Advances in Computational Sciences and Technology (ACST) ISSN 0973-6107 Vol.11, No. 1, (2018)
© Research India Publications http://www.ripublication.com

Automatic Viva Question Generation

Yash jain, Ronak Kachalia, Rahul Gupta, Prof. Shubhada Labde
Department of Computer Engineering
K.J Somaiya Institute of Engineering and Information Technology,Sion,Mumbai
yash.gj@somaiya.edu, ronak.kachalia@somaiya.edu, gupta.rr@somaiya.edu, shubhada.l@somaiya.edu

Abstract—TheTextsisconsideredasveryimportantsourcein educational valuewhicharebecomingavailablethroughthe InternetsuchasWikipedia, news services. Hence using these new texts in classrooms brings many challengesandsooneofwhichis thatthereisusuallylackofpracticeexercisesand assessments.In this paper we are addressing a part of this challenge by automatingthecreationofaspecifictypeofassessment.Tobe specific we are focusing on automatically generating WH questions.Themainaimis tocreate an automated system that can take input a text and produce output questionsforassessing areader'sknowledgeoftheinformationinthetext.This question will be given to teacher and they will revise each and every question and will select all he important question. After introducing the problem, we are also describing all the computational and linguistic challengeswhicharepresentedby factualquestiongeneration.Thesystem usestherankingof question based on priority used by students for answer of question.Offeringsuchautomaticsuggestionsreducedthetime andeffort spentbyparticipants(e.g.teachers),thoughitalso affectedthetypesofquestions that were created. The research supports the idea that natural language processingcanbeusedto help teachers and other participants efficiently in creating instructionalcontent.

Keywords—Simplify;Extraction;QuestionGeneration; Ranking;

I. INTRODUCTION

If you are highly talented and dedicated reader, and there no need to assess whether you read and retained basic factual information. However, this is not the same with every readers. For example, an engineering college teacher ask numerous question to students for viva. Generating such questions, and framing the reading assessments more generally, can be a time consuming and effortful process. In this research, we work on automating that process and help teachers in reducing some of the efforts. In particular, we focusing on the problem of automatically generating factual questions from the given paragraph.

We aim to create a system for question generation (QG) that can take as input a subject (e.g., a web page or Wikipedia article that a teacher might select to supplement the materials in a textbook), and create all the possible question from the given subject. This system will also help user to select and revise all the possible questions for practice exercises or part of a quiz to assess whether students read the text and retained knowledge about its topic. We focus on QG about informational texts—that is, that convey factual information rather than nonfactual.

To make our factual QG system very much useful, we make the use of domain-specific knowledge (e.g., about historical events or geographical locations) and also focus on modeling fairly general lexical and syntactic phenomena related to questions and the presentation of factual information.

i. IllustrativeExampleofFactualQuestionGeneration

In this section, we have provided an examples that illustrate that QG about factual information is a very challenging but still achievable task given current advancement of natural language processing (NLP) technologies. We begin with a relatively straightforward example, taken from an Encyclopedia Britannica Elementary Edition article about the city of Monrovia [2].

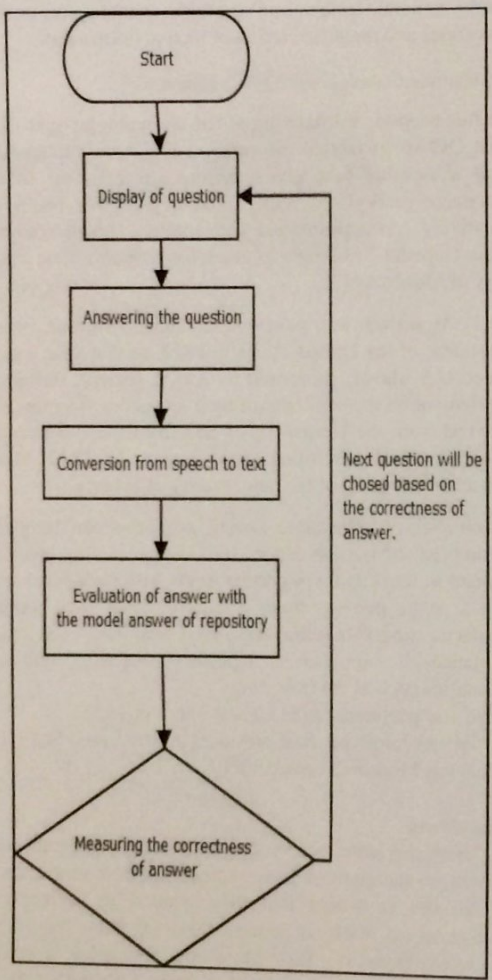
(1.1) Monrovia was named after James Monroe, who was president of the United States in 1822. In that year a group of freed U.S. slaves, sponsored by a U.S. society, started a new settlement on the continent of their ancestors. As more settlers arrived from the United States and from the Caribbean area, the area they controlled grew larger. In 1847 Monrovia became the capital of the new country of Liberia.

A number of acceptable factual questions can be generated from the above sentence by analyzing its grammatical structure, labeling its important items with high-level semantic types (e.g., person, time, location), and later performing syntactic transformations and WH-framing. From the first sentence, we can extract a perfect structured and specific questions such as the following:
Who was president of the United States in 1822?
When was James Monroe president of the United States?
Who was Monrovia named after?

RelatedWorks:
[1]There are some works that are related to segmentation of sentences and analyze those sentences. There was a repository of all the keywords that they wanted to segment. After analyzing of each sentences they segment the important keyword from it. This keywords resembles a particular meaning such as scheduling an alarm, disabling an alarm, storing a reminder message and active an alarm.

II. WORKING

During viva, teacher will open the application which will be having all the subjects. After that teacher will select the subject that he/she wants to generate the question. The system will find a moderate level question and it will convert that question from text to speech. Initially the question level will be moderate. The application will frame the question in terms of voice. After that the application will be ready to hear the answer from the student. The system will chec if there input from for certain period of time, it will consider that question as an unanswered question. Once students completes the answer the audio will be converted to text format and so will be given for evaluation. Based on correctness the system will decide what should be the next question. This process continues until teacher wishes to end.



Steps involved in automatic question generation and self evaluation.

a.Converting complex sentence to simple factual.

b. Generation of question from the repository.

c. Based on question the student will answer the question and the system will record the voice of it.

d. This voice will be converted text form and will send for evaluation.

e. The evaluation process will analyze the answer with the model answer and will assign the correctness value to it.

f. Based on correctness value we will decide weather to decrease or increase the toughness of the answer.

g. In order to provide the levels of each question we have classified the question in the following category.

1. Simple
2. Moderate
3. Difficult

III, CONVERTING COMPLEX SENTENCE TO SIMPLE FACTUAL.

Sentences often reflects not just one but many pieces of factual information which makes the use of nested syntactic constructions. Our method for extracting correct meaning-preserving, simplified sentences totally depends on two linguistic phenomena: semantic entailment and presupposition. We provide brief discussion of these phenomena to motivate our approach.

1. ExtractandSimplificationbySemanticEntailment.

A semantically need B if and only if for each and every situation in which A is true, B is also true. It make simple the complex sentences by removing adjunct modifiers and discourse connectives and by splitting conjunctions of clauses and verb phrases. These change in sentences preserve the truth conditions of the original input sentence, while giving more incisive sentences from which questions can be generated.

2. RemoveDiscourseMarkersandAdjunctModifiers.

Many sentences can be made simple by removing some of the adjunct modifiers from clauses, verb phrases, and noun phrases. We can extract by aking off the discourse marker however and the relative clause which restricted trade with Europe.

(2.1) However, Jefferson did not believe the Embargo Act, which restricted trade withEurope,wouldhurttheAmerican economy.

(2.2) JeffersondidnotbelievetheEmbargoActwouldhurtthe American economy.

Example 2.2 is true in every situations where example 2.1 is true, and hence it is semantically entailed. Discourse markers such as 'however' does not affect the truth conditions in and of themselves, but actually it serve to inform a reader about how the current sent nce relative to the prior discourse. Adjunct modifiers do have some meaning in sentence, but also it does not affect semantic entailment. Of course, many adjuncts provide useful information that we should preserve for later QG steps. Such example, prepositional phrases that identifies the locations and times at which events occurred gives clear indication of where and when questions, respectively.

3. Splitting Conjunctions.

We also separate conjunctions of clauses and verb phrases. In most cases, the conjuncts in these conjunctions are needed by the original sentence. For example, it is given John studied on Monday but went to the park on Tuesday, both John studied on Monday and John went to the park on Tuesday are entailed. Exceptions where conjuncts are not entailed include the following conjunctions: with or and nor, which we do not split; and conjunctions within low monotone contexts, which we have to split. We do not split conjunctions except those conjoining clauses and verb phrases. To be specific, we keep noun phrases as it is. Avoiding conjoined noun phrases is generally advisable for factual QG, because of difficult semantic and pragmatic issues which involves nouns.

4. ExtractionbyPresupposition.

In addition with the strict notion of semantic entailment, the pragmatic phenomenon of presupposition has an crucial significance in giving information. The semantically entailed information in highly complex sentences generally covers only some of what readers understand. Extracting semantically entailed statement have a chances of losing some of the useful question of the facts (example 4.1).

The Embargo Act restricted trade with Europe.

The Embargo Act did not restrict trade with Europe.

As the examples conveys, some of the information in some of the syntactic constructions of sentence is not semantically entailed but rather it has been presupposed, or imagined to be true and not stated directly, by the author. The meaning conveyed by these constructions do not get change by the non-factive verbs that affect the meaning of the main clause of the sentence. This phenomenon of presupposition is generally subsumed by the term "conventional implicature". Many presuppositions have perfect syntactic or lexical associations, or "triggers." These triggers helps in the extraction of simple statements which can lead to many useful, concise questions. A list of presupposition triggers-

- non-restrictive appositives
- non-restrictive relative clauses
- participial modifiers
- temporal subordinate clauses

IV. EXTRACTION ALGORITHM.

This section presents our algorithm for seperating simplified factual statements from complex sentences. The primary method, extractSimplifiedSentences, which is shown below is high-level pseudocode in Algorithm 1, takes tree t as an input and returns all the trees in Tresult as output. There is an helper function, which is shown in Algorithm 2, recursively seperates conjunctions and also checks to confirm that outputs have subjects and finite main verbs. After parsing it with the Stanford Parser and identifying key nodes of Tregex, we manipulate trees using the Stanford Parser API, which gives complete independence for inserting and deleting children, changing labels on tree nodes, etc.

We have involve an additional filtering step in the extractor so to restrict processing parse trees which is not grammatically correct sentence, or sentences with improper constructions. If suppose the parse tree generated for an input sentence has any of the non-terminal symbols, the extractor returns a set containing an unmodified tree as its output.

```
Algorithm 1 extractSimplifiedSentences(t)
Tresult ← Ø
Textracted ← {t} U extract new sentence parse trees from t for the following: non-restrictive
appositives; non-restrictive relative clauses; subordinate clauses with a subject and finite verb; and
participial phrases that modify noun phrases, verb phrases, or clauses.
for all t' ∈ Textracted do
    Tresult ← Tresult U extractHelper(t')
end for
return Tresult

Algorithm 2 extractHelper(t)
Tresult ← Ø
move any leading prepositional phrases and quotations in t to be the last children of the main verb
phrase.
remove the following from t: noun modifiers offset by commas (non-restrictive appositives,
nonrestrictive relative clauses, parenthetical phrases, participial phrases), verb modifiers offset by
commas (subordinate clauses, participial phrases, prepositional phrases), leading modifiers of the main
clause (nodes that precede the subject).
If t has S, SBAR, or VP nodes conjoined with a conjunction c /∈ {or, nor} then
    Tconjuncts ← extract new sentence trees for each conjunct in the leftmost, topmost
set of conjoined S, SBAR, or VP nodes in t.
    for all tconjunct ∈ Tconjuncts do
        Tresult ← Tresult U extractHelper(tconjunct)
    end for
else if t has a subject and finite main verb then
    Tresult ← Tresult U {t}
end if
return Tresult
```

Figure 2: Convert from complex to simple

V. GENERATION OF QUESTION FROM THE REPOSITORY.

This section gives the very important strategy to generate the question. In order to do this, it is compulsory that we remove complex sentences from the passages. We are taking review of the all the important keywords from the teacher. All the question generated will be completely based on these keywords. We will analyse all the linkage of that particular keyword in the passage. After that we will analyse the meaning of that sentences. This will done by using stanford parser which will be doing tagging process of grammar.

Eg. My dog also likes eating sausage.

We have classified sentences based on the above tagging. For what question there are fourteen types of question. Using parse tree the system will how the words on sentence is grammatically connected. Using this tree we will generate simple factual sentence and then we will concatenate the particular WH keyword. Similar process will carried for where, why, when, who and which. Once we analyse the sentence we will append the particular tag.

<u>Tagging</u>:
My/PRP$ dog/NN also/RB likes/VBZ eating/VBG sausage/NN

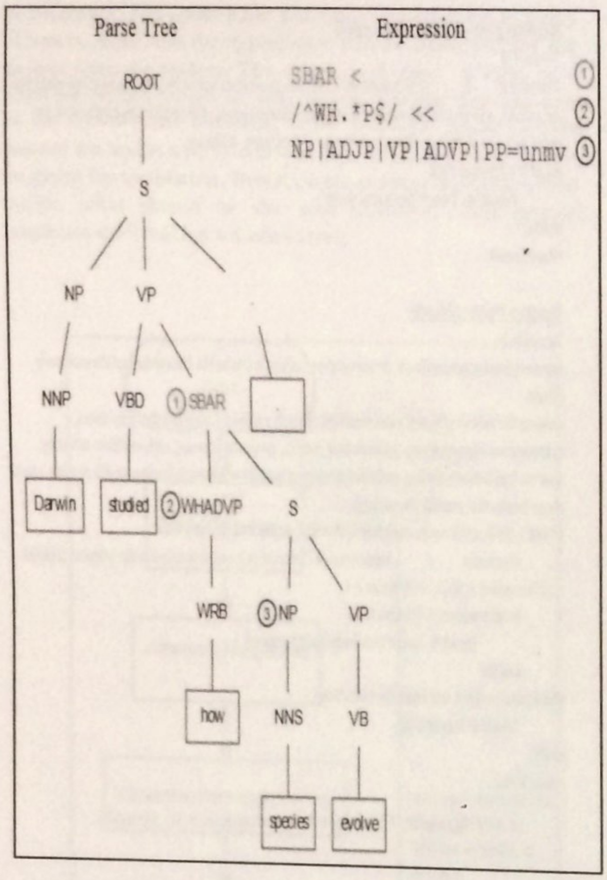


FIGURE 3: PARSE TREE

VI. RANKING OF QUESTION.

We are implementing a system which will classify the question into low, moderate and difficult. Most users of QG tools will only consider a relatively short list of approved questions, and so we want to be able to identify the questions that are most likely to be acceptable for viva examination(i.e., at the top of a ranked list)

In order to do these we will generate all the possible from the passages. This will be having all types of question such as

1. True/False

   Eg. Dracula was a real-life figure in history.
2. Brief Answer Question.

   Eg. Explain what is agent?
3. Short Answer Question.

   Eg. Tell me about youself.

This question will be given to students. Based on their answering of question we will we classify them into their respected difficulty level. A pattern will be seen in solving of their question. These question will be given to brilliant as well as to average and below average student.

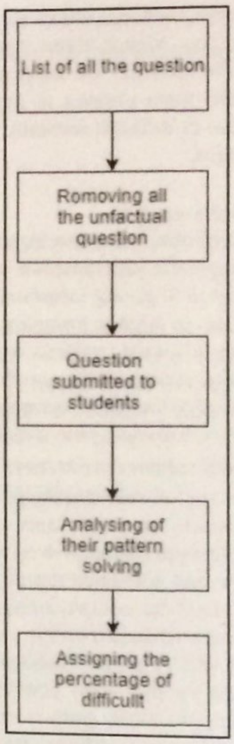


Figure 4: Classification of question.

During viva session the initial difficulty will be 50%. If he/she gives the correct answer then the percentage of difficulty will be increased or else it will be reduced by some amount. So this process of increasing the difficulty or decreasing the difficulty will be continued till the viva session of that particular students ends.

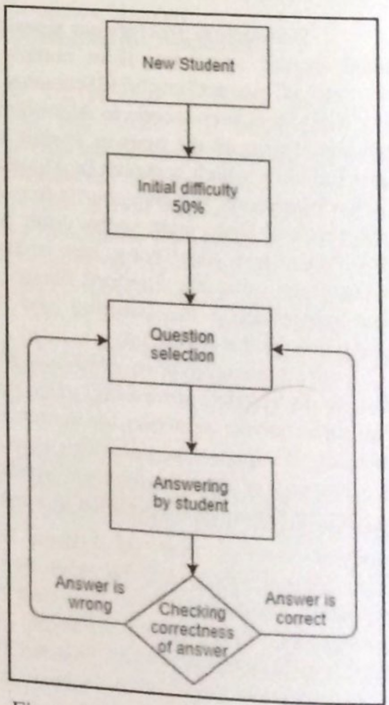


Figure 5: Complete Working

VII. CONCLUSIONS

We have also analyzed many linguistic and computational challenges involved in automatic QG. Our analysis can be beneficial for future QG researchers, also for those who all are working on similar problems, with a proficient roadmap. We have described the complete process for generation and-ranking approach to QG that allows us to use all the existing NLP tools for syntactic analysis. As a first step in this QG technique, we have described a methodology for extracting simplified statements, which can then be use in conversion of questions, from complex input sentences.

The significant feature of our approach is question ranking. We have used live question testing technique to rank candidate questions for viva so that the best questions are selected based on student knowledge.

REFERENCES

[1] http://www.cs.cmu.edu/~ark/mheilman/questions/papers/heilman-question-generation-dissertation.pdf

[2] https://pdfs.semanticscholar.org/903b/72a12a92bdbaad8f978dfd287e1132de556e.pdf

[3] https://pdfs.semanticscholar.org/113e/5ff6fd0465fd8ef9c8580556c09163056581.pdf