ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.5

An Automatic Gap-Fill Multiple Choices Question Generator for Quizzing

Pranita Pradip Jadhav[1], Manjushree D.Laddha[2]
M. Tech Student[1], Assistant Professor[2]
Department of Computer Engineering
Dr. Babasaheb Ambedkar Technological University, Lonere, India

Abstract:
An Automatic Gap-fill Multiple Choices Question Generator is one of the research fields which is aim to support increasing demand for specialized educational system and active learning. An Automatic gap-fill Generator can generate gap-fill question (GFQ) with multiple choices (one correct answer and three distractors).The crafting of such type of questions is time-consuming for teachers because making the GFQ from the external source material like textbooks and other electronic texts are a very tedious task. It can be generated in three parts, selecting the informative sentence to ask the question, choose the gap or blank space of the resulting selected sentence and Search the distractors which distract the learner from the correct answer of the question. Natural language Processing (NLP) techniques like Tokenization, Part-of-Speech tagging, Name Entity Recognition are applied on each of these sentences. The advantage of this AGFQG is to provide the services that make it easy for teachers to generate the GFQ and many other competitive exams in which the evaluation can be done through conducting multiple choice questions test (Quiz test).

Keywords: Automatic Gap-fill Question Generator (AGQG), Gap-fill Questions (GFQ), Named Entity Recognition (NER), Natural Language Processing (NLP), Natural Language Toolkit (NLTK), Part-of-Speech (POS).

I. INTRODUCTION

Assessment is an essential facet of education. However, developing an assessment test is a grueling task. It engrosses Teacher's time and efforts which could be spent on teaching performance. There is a necessity for developing educational materials for language learning. An Automatic Gap-fill Question Generator helps to diminish teacher's load and generates questions of consistent CALIBER which provides an objective assessment. Assessment Evaluation plays a deciding role in education and increases its importance in a changing demand teaching environment. In this paper, we propose the system for the Automatic Gap-fill multiple choices questions from the text file and Paragraph using Natural Language Toolkit (NLTK) which is a ruling platform for building Python Programs to work with a human language data processing [10].To initiate the GFQ from the text file, separate the sentences using the symbols like Full-stop, Exclamatory Mark, and Question Mark. After that sentence is divided into tokens known as tokenization and then using Part-of-Speech (POS) tag, we acquire the separate word called as Token and its type like the word is Noun, Pronoun, Verb, Adjective etc. Selection of Informative sentence for GFQ is based on the number of Noun, Pronoun and Superlative degree present in the sentence. For gap selection, put priority to the Noun, Pronoun and Superlative degree present in the sentence and removes the appropriate word from the sentence and makes the Gap. Formerly the question is generated; the annoying task is to find the distractors for the gap which to be selected. For this cause, used Named entity Recognition [4] and used Wikipedia is a help to find the applicable choices for Gap.

II. RELATED WORK

Generating automatic GFQ is relatively new and very emerging research topic which is useful in Education Technology. Here we first discuss the few models or systems for automatic gap-fill question generation. Sheetal Rakangor and Dr. Yogesh Ghodasara (2013) can proposed the System finds fill in the blanks, blanking key generates from the selected statement. Syntactic and lexical features are used in this process. NLP parser is used, part of speech taggers are applied on each of these sentences to encode necessary information [1]. Sheetal Rakangor and Dr. Yogesh Ghodasara (2014) both are proposed the distractor selection on the basis gap selected name, organization and place using Named entity Recognition (NER) [2]. Manish Agarwal and Prashanth Mannem (2011) they used to selects most informative sentences of the chapters and generates gap-fill questions on them using Syntactic features like height of tree i.e Heuristic function[3]. Smith et al. (2010) present a system, TEDDCLOG, from corpus it will generate the test items. Take key as input in TEDDLOG. It finds distractors from a distributional thesaurus. They got 53.33% (40 out of 75) accuracy after post editing (editing either in carrier sentence (GFS) or in distractors) in the generated gap-fill questions [4]. Pino et al. (2009) proposed a baseline technique to generate cloze questions (gap-fill questions) which uses sample sentences from WorldNet. They then refine this technique with linguistically motivated features to generate better questions. They used the Cambridge Advanced Learners Dictionary (CALD) which has several sample sentences for each sense of a word for stem selection (GFS). The new strategy produced high quality cloze questions 66% of the time[5]. Karamanis et al. (2006) report the results of a pilot study on generating Multiple-Choice Test Items (MCTI) from medical text which builds on the work [6]. Mitkov et al. (2006) used NLP techniques like shallow parsing, term extraction, sentence transformation and computation of semantic distance in their works for generating MCQ semi-automatically from an electronic text. They did term extraction from the text using frequency count, generated stems using a

set of linguistic rules, and selected distractors by finding semantically close concepts using WordNet [7]. Liu et al (2008) proposed study proposes generation of high-quality test items by focusing on words with multiple senses. Using the Word Sense Disambiguation (WSD) technique, they generated questions that test student comprehension of the sentence and the vocabularies in the alternatives [8]. Aldabe and Maritxalar (2010) developed systems to generate MCQ in the Basque language. They have divided the task into six phases: selection of text (based on learners and length of texts), marking blanks (manually), generation of distractors, selection of distractors, evaluation with learners and item analysis [9].

III.OUTLOOK

The process of generating and analyzing the GFQ with multiple choices consists of the following steps:
In this Approach,

(i) Gives text file as an input to the Gap-fill question Generator.
(ii) Informative Sentences can be selected from the text file.
(iii) The Process of selecting Gap from the Sentence.
(iv) Selection of Distractors from the dataset of Name entities or from Wikipedia.
(v) Interchange the position of choices.
(vi) Make gap-fill question with four options.

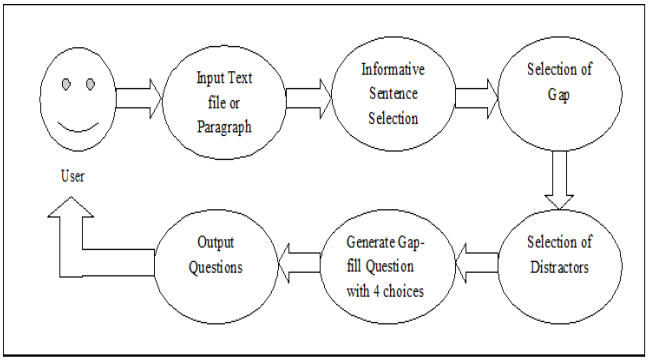

Figure.1. Outlook of an Gap-Fill Question Generator

IV. METHODOLOGY

Gap-fill multiple choice question generators can be creates GFQ in three distinct levels to lighten the load of Teachers for creating Quiz test paper.

(1) **Informative Sentence Selection:** Pick out analytical and meaningful sentences from the text file to inquire the question.
(2) **Gap Selection:** From sentence determine the gap.
(3) **Distractor Selection:** Draft three choices which have the same context of the gap and will trouble the learner to select the precise answer.

Our Generator builds the accurate techniques of in the field of Machine Learning, Natural Language Processing to solve above problem.

A. Informative Sentence Selection

Allow the text file as an input for selecting a consistent and logical sentence from the input text to form the question. Gap fill question can be asked on selected informative sentences and is done by using Natural Language Toolkit (NLTK). Divide each and every sentence using a full stop (.), Question mark (?) and Explanatory mark (!). Get the separate sentences apply the Part-of-Speech (POS) tagging and get the words with its type. In the case of the word is noun, pronoun, adverb, adjective, determiner etc. and superlative degree present in the sentence. Perform the pattern matching for getting the Noun, Pronoun and Superlative Degree. If there is no noun and superlative degree present in the sentence then Generator will discard the sentence.

1) Extracting Features by POS taggers are:

(i) Sentences calculated in the Text file.
(ii) Shows the nouns (NN), pronoun (NNP), adverbs (RB), adjectives (JJ), determiner (DT) etc. present in the sentence.
(iii) Shows the adjective superlative (JJS) degree of sentence.
-Superlatives are suffix with –est. like biggest.

2) Algorithm for Informative Sentence Selection

i. Extract the Text file.
ii. Read the sentences from the Text file.
iii. Calculate the number of sentences.
iv. For all sentences, calculate the nouns, pronouns, adjectives, adverbs etc.
v. Find the named entity in each statement (Name, Location, city, country etc) Store the named entity into the database where all the previous name entities are stored.
vi. If the sentence which contains a noun and superlative degree then the sentence is selected.
vii. Else if the sentence contains max [noun] then it is selected.
viii. If superlative degree is not found then catch the sentence which having a number of noun and pronoun present.
ix. Else if sentence will be disposed by the generator, if no noun and superlative degree present.
x. End if.
xi. End for.
xii. Display the selected sentence.

B. GAP SELECTION

Once the sentence is selected for the gap-fill question, there will be the task to select a gap which is very important level. Part-of-speech (POS) tag can administer a linguistic image between the words in the sentence. The task of Sentence Selection is done from the text file and then pushes all the nouns, pronouns, superlatives present in sentences into the potential key list. From this key list, the generator will select one best key as a gap after checking the number of occurrences of the gap in selected sentence and Paragraph. If any noun is restated in the list it will be detached and formulate a blank space to the specific place of that gap. To find the occurrences of the key in the Text file is calculated by the Euclidean Distance theorem [11]. In which, it gives the unique id to the name entities present in the key list and finds their occurrences in the text file and calculated by using the equation(1),

$$d^2_{xy} = (x_1 - y_1)^2 + (x_2 - y_2)^2 \quad \ldots\ldots\ldots \quad (1)$$

Where, x denotes the unique id of name entities present in potential keys list and the y denotes their occurrences in the paragraph and whole text file.

$$d_{xy} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad \ldots\ldots\ldots\ldots (2)$$

i.e. the distance itself is the square root value. Select the key which has minimum value is found.

1) Algorithm for Gap Selection

i. For each sentence, extracts the words and its types (noun and pronoun) from the sentence and pushed it into the potential keys list of the sentence.
ii. From this list, selection of a gap on the basis of their occurrences in selected sentence and text file.
iii. End for.

iv. Remove the best key from the sentence and generate gap-fill question.

C. DISTRACTORS SELECTION

Once the best key is selected from the pool of potential keys and making the gap in selected sentence, we get one correct choice which is our answer to that question Then find out the distractors or choices for Gap-fill Questions, and for this, it will invoke the Named Entity Recognition (NER). The common NER task is mapping named entities to concepts in vocabulary and dataset. Check whether the selected key is name, organization, number, time, place, country, disease etc. and then retrieves the appropriate choices from the dataset. In which any text file is given as an input to the model it will find the Name entity in each sentence (Name, place, time organization, city, country etc.) and automatically stored into the dataset and this Dataset is dynamically updated every time and also fetch the distractors from the Wikipedia by giving the best key as an input. To choose the distractors from the dataset, used Randomized Theorem [12] and gets the choices in fraction of second For eg: Suppose the 75 entities are present in the dataset which are relevant to the answer then it will divide the 75 entities into three parts and then fetch the one entity from each section as choices. By doing this, it will not select the same choice frequently. It will always give the different choices.

1) Selects Distractors on the Basis of Following Properties:

i. **Semantic Checking**: The choices that are selected for the questions that should be in same meaning or context of the gap.

ii. **Syntactic checking**: Choices that are Complementary and identical to the gap of the sentence. For ex. T-phase probably as good distractor for G-phase.

iii. **Contextual meaning**: Choices need to Competent to the Question.

2) Algorithm for Distractor Selection

i. For each sentences, diff (distractor, key) with comparable importance to the key.

ii. Retrieve the arbitrary three name entities with equal importance perhaps close in their semantic meanings.

iii. Interchange their positions.

iv. Display the gap-fill question into text file with the four options.

v. End for.

When all the three levels are achieved by the An Automatic Gap-fill Question generator it will deliver the sorted gap-fill questions with suitable choices.

D. PROCESS FLOW OF AN AUTOMATIC GAP-FILL MULTIPLE CHOICES GENERATOR

Gives Text file or pdf document as an input to the Gap-fill multiple choice Generator It will separate all the sentences present in the file and calculate the number of sentences. In Example, There are two sentences extracted from paragraph which is C++ is Object Oriented Programming Language and OOP follows Bottom-up Approach and by applying the POS tagging and get number of Noun, Pronoun and Adjective present in sentence and from this Selection of informative sentence can be done. From the selected sentence, key list is generated in which all the nouns, pronouns, adjectives present in the sentence are pushed, and from this list select one key as a gap on the basis of their occurrences in Paragraph and Sentence and Generate:

(1) ____ is Object Oriented Programming Language.
(2) OOP follows _____ approach..

After that next level is to find the distractors for the gap. For C++ and Bottom-up select the appropriate choices from the pool of name-entity dataset and Wikipedia Interchange the Position of choices and display the Gap-fill Question with Choices.

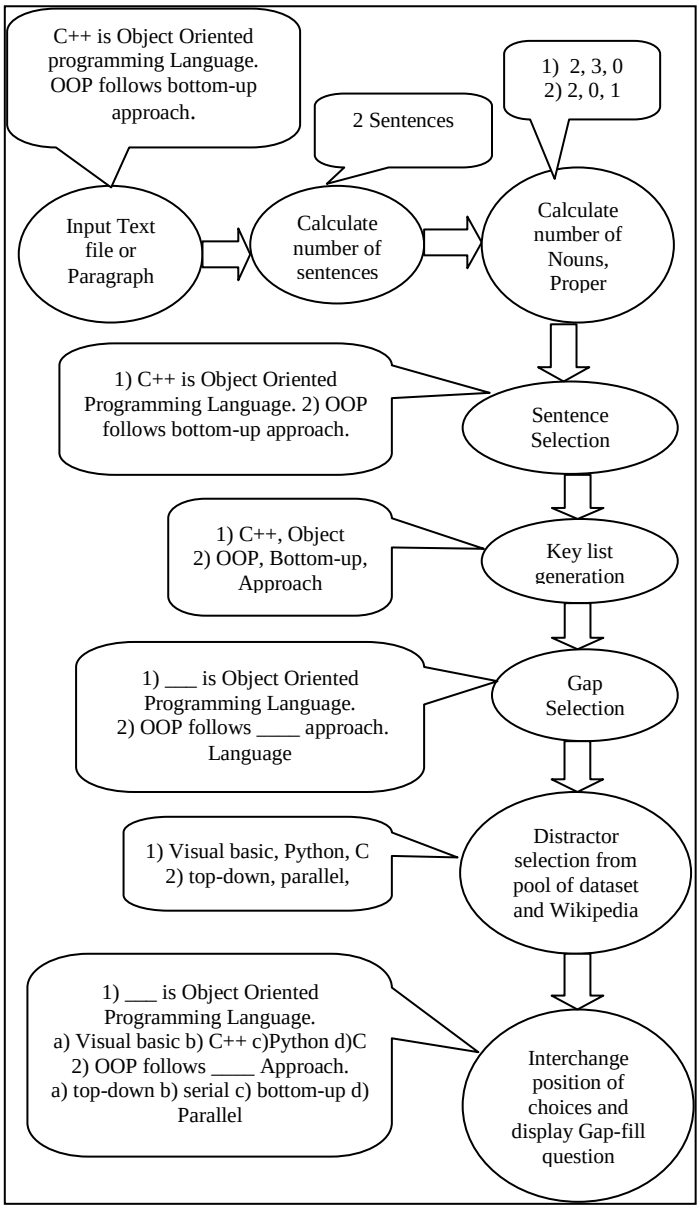

Figure.2. Process Flow Of An Automatic Gap-Fill Multiple Choice Question Generator

V. CONCLUSION AND FUTURE SCOPE

System will select the informative sentence from the paragraph and generate fill in the blanks with distractor from the paragraph and with the help of Wikipedia. For that used Natural Language Toolkit for selection of informative sentence and Selection of gap from the paragraph. To obtain the distractors we look for the synonyms, antonyms, and similar words for the distractors that are find from dataset and Wikipedia or the given paragraph. For testing different paragraph's downloaded and tested through system as well as manual question were also generated from paragraph. It is very difficult for questions that are automatically generated to be as good as questions generated by human experts. Currently, our methodology focuses on improving the

correctness of the answer. In the future, the quality improvement of distractors will be the next step of this work.

VI.ACKNOWLEDGEMENT

This work is just not an individual contribution still its completion. I take this Opportunity to thank all for bringing it close to the conclusion. Special thanks go to my guide, Head of Department and other faculty members for leading me to many new insights, valuable and timely suggestions at crucial stages, encouraging and teaching me how to get down to the root of the problem. I am also thankful to all friends and well wishers for support during the demanding period of this work. I express my deep sense of gratitude towards my parents whose strength and compassion are my constant inspiration. I would also like to thank my wonderful colleagues and friends for listens my ideas, asking questions and providing feedback and suggestions for improving my ideas.

VII. REFRENCES

[1]. Sheetal Rakangor and Dr. Yogesh Ghodasara (2013) Computer aided environment for drawing (to set) fill in the blank from given paragraph.e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 15, Issue 6 (Nov. - Dec. 2013), PP 54-58.

[2]. Sheetal Rakangor and Dr. Yogesh Ghodasara (2014) Automatic Fill in the blanks with Distractor Generation from given Corpus, International Journal of Computer Applications (0975 – 8887) Volume 105 – No. 9, November 2014.

[3]. Manish Agarwal and Prashanth Mannem(2011) Automatic Gap-fill Question generation from text books.

[4]. Simon Smith, P.V.S Avinesh and Adam Kilgarriff. 2010. Gap-fill Tests for Language Learners: Corpus-Driven Item Generation.

[5]. Juan Pino, Michael Heilman and Maxine Eskenazi. 2009. A Selection Strategy to Improve Cloze Question Quality, Wkshop on Intelligent Tutoring Systems for Ill-Defined Domains. 9th Int. Conf. on ITS.

[6] .Nikiforos Karamanis, Le An Ha and Ruslan Mitkov. 2006 Generating Multiple-Choice Test Iterms from Medical Text: A Pilot Study, In Proceedings of INLG 2006, Sydney, Australia.

[7]. Ruslan Mitkov, Le An Ha and Nikiforos Karamanis. 2006 A computer-aided environment for generating multiple-choice test items, Natural Language Engineering 12(2): 177-194

[8]. Liu et al (2008). Applications of lexica information for algorithmically composing multiple-choice cloze items. In Proceedings of the Second Workshop on Building Educational Applications Using Natural Language Processing, pages 1–8, Ann Arbor, Michigan, U.S., June 2005. Association for Computational Linguistics.

[9]. Aldabe, I., Maritxalar, M., (2010). Automatic Distractor Generation for Domain Specific Texts. Proceedings of IceTAL, LNAI 6233. pp. 27-38.

[10]. Natural Language Processing with Python by Steven Bird, Ewan klein and Edward Loper, by O'Reilly Publication.

[11]. https://en.wikipedia.org/wiki/Euclidean_distance.

[12]. http://www.cs.cornell.edu/courses/cs684/2005fa/10-26