ISSN : 2350-0743

www.ijramr.com

International Journal of Recent Advances in Multidisciplinary Research

Vol. 04, Issue 11, pp.3071-3075, November, 2017

RESEARCH ARTICLE

AUTOMATIC QUESTIONS GENERATING SYSTEM

Himani Jain, *Sai Krishna, P.V., Mukul Mittal and Geraldine Bessie Amali D.

VIT University, Vellore, India

ARTICLE INFO

Article History:

Received 09th August, 2017
Received in revised form
28th September, 2017
Accepted 04th October, 2017
Published online 30th November, 2017

Keywords:

Text-blob, Word-net, Wikipedia.

ABSTRACT

All the educational information can be accessible through the Internet sites (e.g., Wikipedia, news administrations). Learning the information by searching through random website is difficult, one of the main difficulties is to frame the questions for famous people who have a Wikipedia page. Here, we address some portion of this test via mechanizing the formation of a particular kind of useful inquiries that can improve the learning background. In particular, we focused consequently producing true questions from the text or keywords provided. We have attempted to make a customised framework that can take as information a content from a site like Wikipedia and deliver as yield inquiries as questions for evaluating a person learning of the data in the content.

INTRODUCTION

Our main focus is to generate automatic questions which creates real questions by scraping the Wikipedia pages (more than one pages at given moment by scrapping them to produce inquiries or questions from numerous themes) and mainly we check for the questions that start with WH ,we try to eliminate the questions that do not make proper sense, utilizing common dialect handling along with some comparable words by utilizing Word-net.Since we have used text-blob and word-net in the implementation of the question generation system . Hence prior knowledge of these libraries should be there. Also this system is taking the content from the Wikipedia pages.

Wikipedia

Wikipedia is a free lance reference website, composed co-operatively by the general population who utilize it. It is an exceptional kind of site intended to make co-operation simple, called a wiki. Many individuals are always enhancing Wikipedia, rolling out a large number of improvements every hour. Even the people who utilize the site can improvise the sites by editing the content in it.

WordNet

WordNet is a large collection of English words. Things, verbs, modifiers and intensifiers are gathered into groups of subjective proportionate words also known as synsets, each conveying a specific thought. Synsets are interconnected by different techniques. The ensuing arrangement of really related words and thoughts can be investigated with the program. WordNet is moreover freely and easily available for downloading. It is generally made important instrument for computational derivation and regular language. WordNet groups words together in perspective of their suggestions. Regardless, wordnet has few fundamental refinements. In any case, WordNet not only interconnect just the word shapes and arrangement of letters but also specific resources of words. And also the words whichare connected to each other in the form of framework are semantically disambiguated. Secondly, it names the inter-relations among the words.

TextBlob

TextBlob is a Python (2 and 3) library for taking care of printed data. It gives an essential Programming interface to diving into general natural language processing (NLP) errands, for example, checking grammer, phrase extraction, notation examination, understanding, and anything is possible from that point. "Natural language processing" is a field at the joining of programming building, phonetics and synthetic mental aptitude which intends to make the basic structure of language available to PC programs for checking and control. It's an immense and dynamic field with a long history! New research and techniques are being created for more improvement. The point of this part is to present a couple of straightforward ideas and strategies from NLP simply the stuff that'll enable you to do inventive things rapidly, and possibly open the entryway for you to see more advanced NLP ideas that you may experience somewhere else. The most regularly known library for doing NLP in Python is NLTK. NLTK is an awesome library, but at the same time it's a difficult: substantial and dangerous and hard to get it. TextBlob is a more straightforward, more comfortable interface to quite a bit of NLTK's usefulness: ideal for NLP tenderfoots or artists that simply need to complete work.

**Corresponding author:* Sai Krishna, P.V.
VIT University, Vellore, India

Literature Survey

A lot of works were reviewed to achieve the objectives presented in the paper. Some notable ones have been discussed here like the one by Liu *et al.* (2010) who in their research thesis discussed about the AQG approach for the formation of the questions to support students who learn by writing. They also compared the questions generated by the AQG approach and those developed manually and found out humans have medium difficulty in distinguishing questions generated by their approach. Work by Mannem *et al.* (2010) also relates our work. They used semantic role labelling system to find different relevant points from the text which are suitable to form the questions. Out of all the generated questions, it then finalises six most suitable questions. The approach used by them can be broken down into three stages, which include content selection, question formation and ranking of the questions. Aquino *et al.* (2011) in their study discussed the question formation using declarative and narrative parts of the paragraph. The framework actualizes data reflection strategies, for example, anaphora determination and genuine proclamation extraction for the preparing of its information. It additionally utilizes question age techniques assembled all through the investigation. The framework parses a content into its relating parse tree with the assistance of the Stanford Statistical Parser and digests and scores it appropriately. All sentences from the content with a non-going along score are evacuated. From what stays, however much inquiries as could reasonably be expected are created. Heilman (2011) concentrated on automatically producing authentic WH questions. He likely made a computerized framework that can take as info a content and create as yield inquiries for evaluating a pursuer's learning of the data in the content. The inquiries could then be displayed to an educator, who could choose and reconsider the ones that he or she judges to be helpful. In the wake of presenting the issue, he depicted a portion of the computational and semantic difficulties introduced by authentic inquiry age. He at that point displayed an actualized framework that use existing characteristic dialect preparing procedures to address some of these difficulties.

The framework utilizes a mix of physically encoded change rules and a measurable inquiry ranker prepared on a customized dataset of named framework yield. He introduced tests that assess singular segments of the framework and also the framework all in all and found, in addition to other things, that the inquiry ranker generally multiplied the agreeableness rate of best positioned questions. Rakangor *et al.* (2015) used Natural Language Processing (NLP) approach to generate automatic questions. They gave the review of many different algorithms used by different researchers. Calvo *et al.* (2012) introduced a different approach for semi-automatic question generation to help scholastic composition. Their framework initially extricates key expressions from students' writing the survey papers. Every majorexpression is coordinated with respective to the Wikipedia pages and made into one out of five theoretical idea classes: like Research Areas, latest Technology, etc. Utilizing the substance of the coordinated Wikipedia article, the framework at that point develops a reasonable diagram structure portrayal for every major expression and the inquiries are then produced depending on the structure. To assess the type of the PC created questions, we directed a form of the “Bystander Turing test”.

Algorithm

Algorithm: Questions Generation System

Input: Keywords having Wikipedia page.
Output: Questions related to the article and hint words along with the correct answer.

```
Step 1: START
Step 2: Import the python libraries named TextBlob , WordNet
        and Wikipedia.
Step 3: Get the Wikipedia page of the keywordgiven and
        analyze the article to retrieve the data.
Step 4: Delete the first word/sentence as they arenot useful for
        generating questions.
Step 5: Gather all the synsets.

If no synsets:
Return empty list.

Step 6: Extract all the hypernyms from the gathered synsets.
Step 7: Extract all the hyponyms from the gathered
        hypernyms.
Step 8: Repeat Steps 5-7 till we get the best 8hyponyms.
Step 9: Ignore the sentences that begin with adverbs because
        mostly they won't befit for question generation.
Step 10: Don't consider the proper nouns that are occurring in
        the title.
Step 11: If we get a noun phrase:

Ignore the last two words in the phrase
Else: Continue displaying

Step 12: If nothing is found then, Displayerror.
```

MATERIALS AND METHODS

For the generation of the questions we have used this methodology. Firstly, we have gathered all the synsets from keyword. if there are no synsets , then we'll return an empty list. Then our main aim is to gather all the hyponyms (i.e. a word with broad meaning in which specific meaning words fall).Then we'll gather some more hyponyms for this hyponym. Out of all these we'll take the first 8 hyponyms. For getting the text from which we are generating questions,. Else for each article we will retrieve the generated questions. Finally in the output window we'll get a display the questions, help words, and the answer.

Module 1

Initialise by adding some of the keyword to frame questions. Keyword should be such that it has a particular page in the Wikipedia. Not any keyword is able to form the questions.

Module 2

Import the required libraries of python. To begin with, import the Text Blob library. TextBlob is a library of python so as to collect the phrases from a paragraph, make it grammatically correct and arrange it in the proper format. It uses Natural Language Processing for generating this. Apart from this, import the WordNet which has a vast database of English. It is useful to generate the synonyms for the related words in order to search for the vast region to generate questions. Third is the Wikipedia which is the source for all of our keywords. The keyword should have a page in Wikipedia in the name of query from which the text will be analysed and questions will be formed accordingly. Wikipedia has a vast source of information as a lot of people continuously change the content of it as per the latest information.

Module 3

The page in the name of the keyword is analysed completely and then the useful phrases from the page are extracted.

Module 4

The first sentence or the phrase is deleted for analysing as we have analysed that in almost every case the first sentence is not useful enough to form the questions. Also, if it is relatable, it has the least valuable knowledge regarding the topic. So, it is better to omit this and more valuable questions.

Module 5

When we have all the phrases from the keyword, so it is better to search for the similar type of words that are synonyms for the executed keyword. All such synonyms are stored in a list together. Also if there is no such synonym for that word, then the list is returned as an empty set.

Module 6

After the synonyms for the keyword is found out, then the hypernyms for that keyword and also for the synonyms of that keyword are searched. Hypernyms are wider category of the word given, say the word is a sub-category for that hyponym word. So, we get a lot of topics to collect the questions.

Module 7

Further if we mine deeper, we also find the hyponym of the found out hypernyms. This further gives us a vast region to form question from.

Module 8

Search for the best 8 hyponyms from the previous modules. Only the best hyponyms are collected to improve the quality of the questions generated. If we are unable to find out 8 hyponyms, the previous modules are repeated again and again until we get the 8 hyponyms.

Module 9

Search if any sentence begins with an adverb. We shall remove those sentences from the group of sentences. This is because the sentences with adverb are not much suitable for the formation of questions.

Module 10

Use of the proper nouns that are found out in the keyword or the title of the page shall be omitted from generation of the questions.

Module 11

If a noun phrase is selected to query for the generation of the questions, in that case the last 2 words of the phrase shall be omitted from the phrase. If no such noun phrase is found, then move further to execute the rest of the program.

Module 12

This module is an error case, where if nothing is found out of the whole of keyword then the program should throw an error that the given keyword does not exist or if exist is not capable enough so that the qualitative questions could be formed out of it.

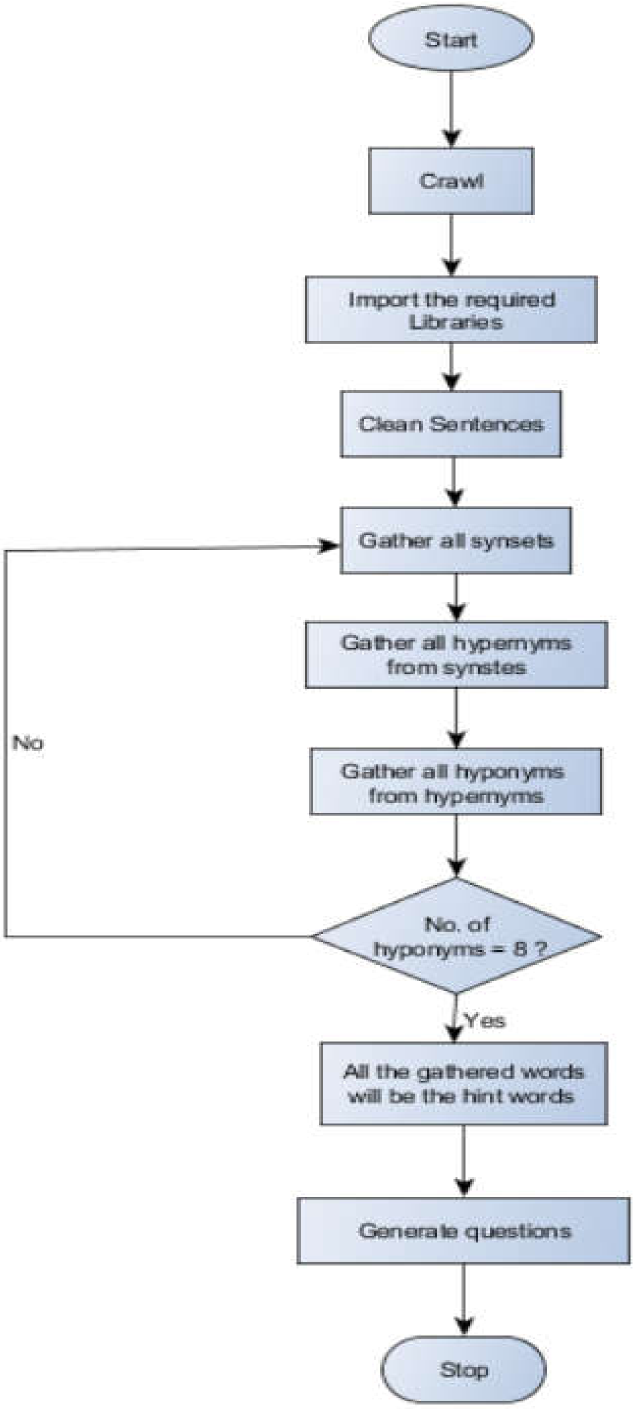
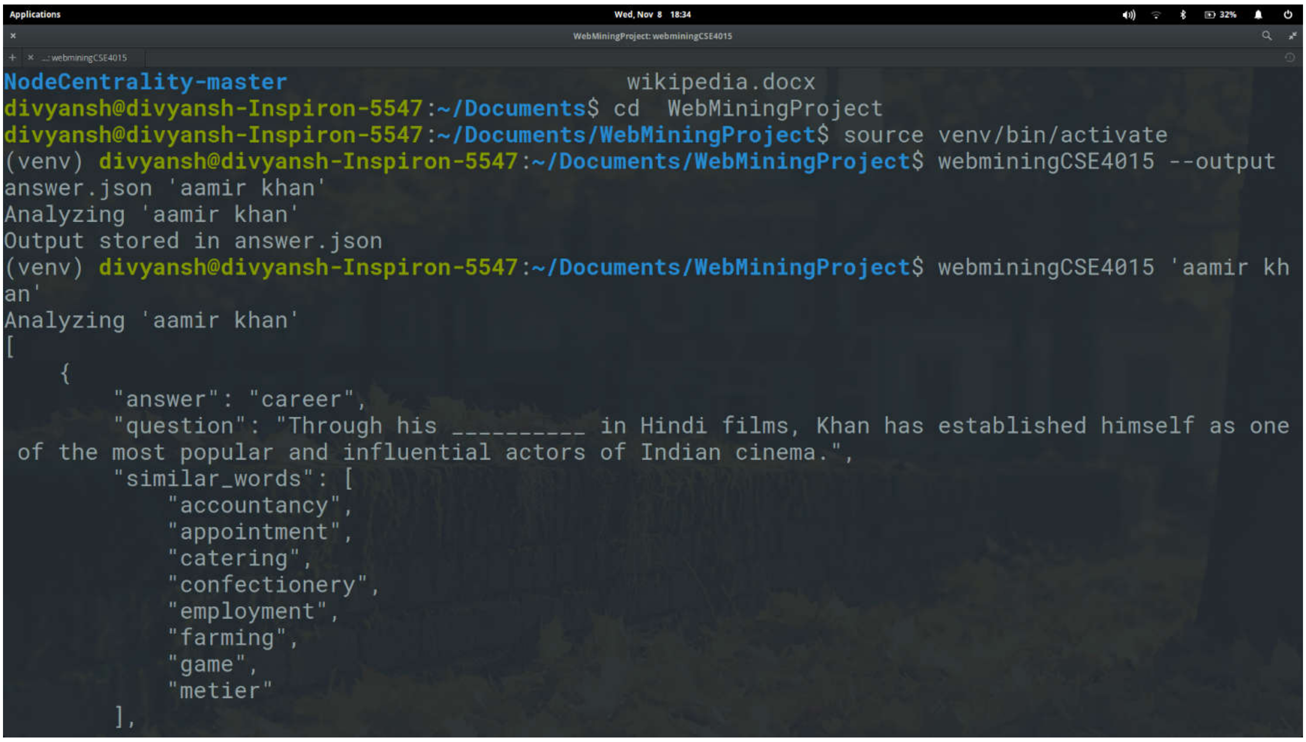

Start
Crawl
Import the required Libraries
Clean Sentences
Gather all synsets
Gather all hypernyms from synstes
Gather all hyponyms from hypernyms
No. of hyponyms = 8 ?
No
Yes
All the gathered words will be the hint words
Generate questions
Stop

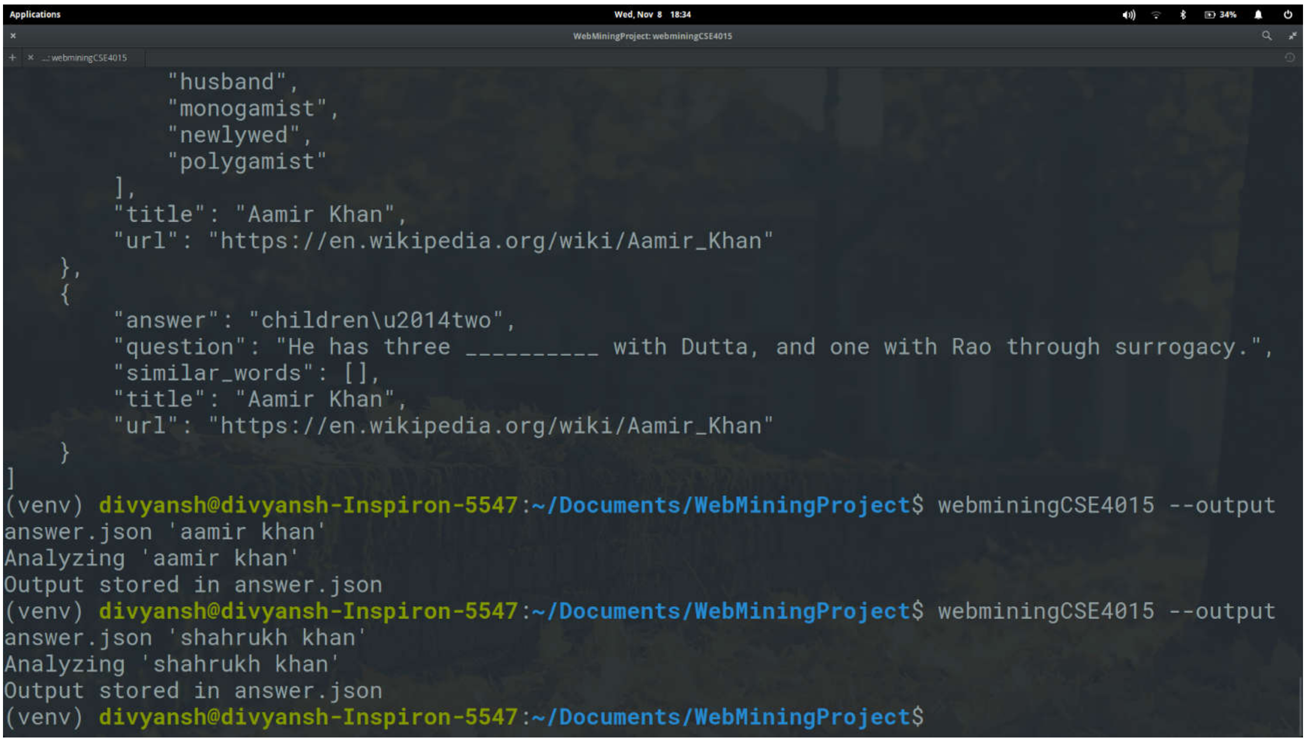
Figure 1. Flowchart

Output



```
NodeCentrality-master                          wikipedia.docx
divyansh@divyansh-Inspiron-5547:~/Documents$ cd  WebMiningProject
divyansh@divyansh-Inspiron-5547:~/Documents/WebMiningProject$ source venv/bin/activate
(venv) divyansh@divyansh-Inspiron-5547:~/Documents/WebMiningProject$ webminingCSE4015 --output
answer.json 'aamir khan'
Analyzing 'aamir khan'
Output stored in answer.json
(venv) divyansh@divyansh-Inspiron-5547:~/Documents/WebMiningProject$ webminingCSE4015 'aamir kh
an'
Analyzing 'aamir khan'
[
    {
        "answer": "career",
        "question": "Through his __________ in Hindi films, Khan has established himself as one
 of the most popular and influential actors of Indian cinema.",
        "similar_words": [
            "accountancy",
            "appointment",
            "catering",
            "confectionery",
            "employment",
            "farming",
            "game",
            "metier"
        ],
```

Figure 2. Output (b)



```
            "husband",
            "monogamist",
            "newlywed",
            "polygamist"
        ],
        "title": "Aamir Khan",
        "url": "https://en.wikipedia.org/wiki/Aamir_Khan"
    },
    {
        "answer": "children\u2014two",
        "question": "He has three __________ with Dutta, and one with Rao through surrogacy.",
        "similar_words": [],
        "title": "Aamir Khan",
        "url": "https://en.wikipedia.org/wiki/Aamir_Khan"
    }
]
(venv) divyansh@divyansh-Inspiron-5547:~/Documents/WebMiningProject$ webminingCSE4015 --output
answer.json 'aamir khan'
Analyzing 'aamir khan'
Output stored in answer.json
(venv) divyansh@divyansh-Inspiron-5547:~/Documents/WebMiningProject$ webminingCSE4015 --output
answer.json 'shahrukh khan'
Analyzing 'shahrukh khan'
Output stored in answer.json
(venv) divyansh@divyansh-Inspiron-5547:~/Documents/WebMiningProject$
```

Figure 3. Output (a)

Enhancements

We can enhance the question generation system by adding some additional features like:

a) Multiword

We can search for many keywords at a single instance. The present system generates questions on both keywords and display the questions of on each keyword separately in the same query but the questions displayed are not of mixed type i.e. not containing the things which are common in both the keywords.

b) Saving questions in a file

We can save all the questions generated in a separate text file so that we can refer to the questionslater and use them in sharing with others.

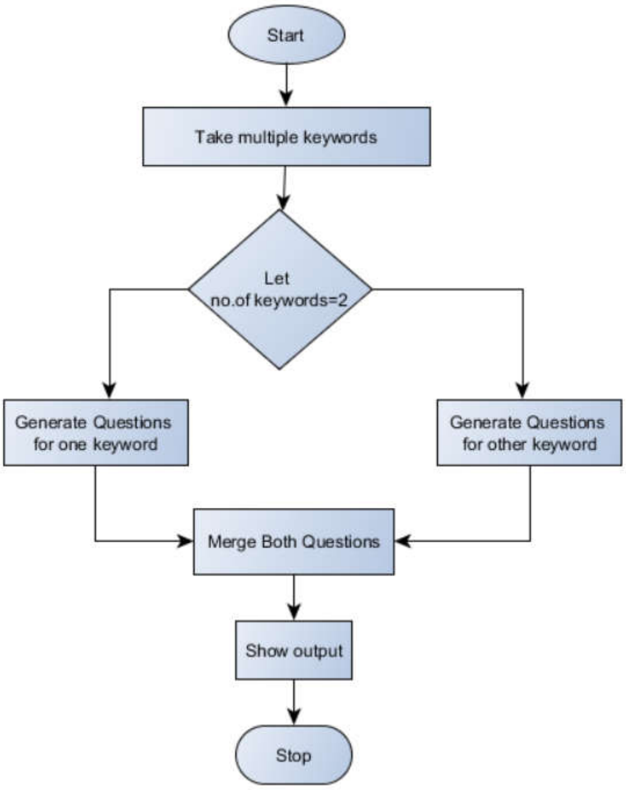


Figure 4. Flowchart

Conclusion

When a new system is created we are bound to face some or the other challenges. This was also implied in the creation of the questions generator system. Some of these challenges are:

a) The major challenge is to create meaningful questions from the text given because even after so much cleansing of data sometimes a few irrelevant questions might pop up.
b) Another challenge is faced during the enhancement of adding multiword feature in the system. The questions needed for a multiword system is that questions which have answers which are common for all the keywords should also come.
c) Also the current system works only if the keyword has a Wikipedia page.

In the current system we are considering only Wikipedia , in future we can try getting the input text from other websites also, which will be helpful up to a very vast extent. Also instead of giving hint words we create MCQ type questions. By implementing this system we can generate meaningful questions from a block of text. This will be a revolution in the field of education. This can save a lot of time of both the teachers and the students as teachers now don't have to invest their time in preparing questions from a text block also the students can practice the questions that can come by giving important keywords which are coming in their syllabus. Hence we can say that this system will be a very influential in the field of education, despite it's flaws we can improve it and work on the future work.

REFERENCES

Aquino, Jessica Franz, *et al*. "Text2Test: Question Generator Utilizing Information Abstraction Techniques and Question Generation Methods for Narrative and Declarative Text."

Calvo, Rafael, A., *et al*. 2011. "Collaborative writing support tools on the cloud." *IEEE Transactions on Learning Technologies,* 4.1, 88-97

Heilman, Michael. 2011. *Automatic factual question generation from text*. Diss. Carnegie Mellon University.

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

Liu, Ming, *et al*. 2012. "Using wikipedia and conceptual graph structures to generate questions for academic writing support." *IEEE Transactions on Learning Technologies,* 5.3, 251-263.

Mannem, Prashanth, Rashmi Prasad, and Aravind Joshi, 2010. "Question generation from paragraphs at UPenn: QGSTEC system description." *Proceedings of QG2010: The Third Workshop on Question Generation*.

Neural Question Generation from Text: A Preliminary Study Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, Ming Zhou

Rakangor, Sheetal, and Dr Y.R. Ghodasara, 2015. "Literature review of automatic question generation systems." *ACM International Journal of Scientific and Research Publications"* 1-5.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and fast but is it good?: evaluating non-expert annotations for natural language tasks. In Proceedings of the conference on empirical methods in natural language processing, pages 254– 263. Association for Computational Linguistics. Lucy Vanderwende. 2008.

The importance of being important: Question generation. In Proceedings of the 1st Workshop on the Question Generation Shared Task Evaluation Challenge, Arlington, VA. John H Wolfe. 1976. Automatic question generation from text-an aid to independent study. In ACM SIGCUE Outlook, volume 10, pages 104–112. ACM. John H Wolfe. 1977. Reading retention as a function of method for generating interspersed questions. Technical report, DTIC Document.

****