# SUBJECTIVE MODEL ANSWER GENERATION TOOL FOR DIGITAL EVALUATION SYSTEMS


Shubham
Research Scholar
Bhilai Institute of Technology
Durg
+91-9644026902
Shubhamlive1010@gmail.com

Dr. Arpana Rawal
Professor
Bhilai Institute of Technology
Durg
+91-9907180993
arpana.rawal@gmail.com

Dr. Ani Thomas
Professor
Bhilai Institute of Technology
Durg
+91-9893165872
ani.thomas@bitdurg.ac.in



## ABSTRACT
Automated subjective answer assessments in modern digital evaluation environments are promising structural consistency, but they distort the very nature of expressing complex and context-rich information put up for evaluation. In modern teaching-learning environments, with wide variety of biasing observed while fabricating humane scripted Memoranda-Of-Instructions, it becomes difficult to evaluate subjective answers with appropriate justification. Answer evaluation systems have seen an extensive research by academicians since few decades. On the other hand, research on subjective model answer generation is still in its infancy stage. Lately, an algorithm for subjective model answer generation has become necessary for developing a generic framework sufficing all types of subjective questions. In this paper, we described one such algorithm for generating model answers for all types of descriptive (subjective) questions from a given text corpus.



## CCS Concepts
**• CCS → Information systems → Information retrieval → Retrieval tasks and goals → Question answering.**

## Keywords
Answer Generation Systems; Algorithm; Content filtering; Restricted Domain; Vocabulary; Seed; Co-occurring; Domain specific; Answer retrieval


## 1. INTRODUCTION
As the modern education system is augmented with digital environments round the globe, the evaluation systems are also getting digitized progressively.

With the advent of automated subjective answering evaluation tools like Electrnic Essay Rater (E-rater) by Burstein, Kukich, Wolff, Chi and Chodorow (1998), Conceptual Rater (C-rater) (Valenti et al., 2003), Intelligent Essay Assessor (IEA) (Valenti et al., 2003), Educational Testing Service (ETS-I) by Whittington and Hunt (1999), BETSY (Valenti et al., 2003), Schema Extract Analyze and Report (SEAR) (Christie, 1999), a drastic drift is seen in preparation of question paper manuscripts from MCQ questionnaire to a blend of both objective and subjective questions [1,2]. The Academicians are observed spending more of their time in setting question papers and evaluating answer, rather than analyzing the scores and counselling the students. According to recent statistics it takes one month on an average to evaluate 700 candidate answer script for six subjects in total. Thus, taking almost two to three months to declare answer for the same lot of students. Apart from this, even with expert evaluators it is not possible for anyone to justify which answer is better and why? Envisioning such a series of hurdles, an attempt is being made to ease the task of manual answer generation by obtaining machine generated answers to some question categories.

The rest of this paper is organized as follows: Section 2 addresses preprocessing issues that have been addressed by other systems for model answer generation. Section 3 outlines the subjective answer generation algorithm. Section 4 suggests further applications and developments possible in near future.

## 2. PRE-PROCESSING ISSUES
The considerable issues that are needed to be investigated in depth before building the prototype tool of generating model answers are enumerated below:

Language Support: One of the design issues in the algorithm demands the concurrent modification of passive data objects in already existing dictionary, while checking for the terms in that domain-specific vocabulary in an attempt to expand the vocabulary at runtime. Not all languages support the above mentioned feature. Hence, there arises a need to choose the appropriate language for tool development. This issue can also be resolved by using the method as described by D. Clarke et al. in their article [3].

Natural Language Processing (NLP) Tool selection: The selected NLP tool must support the following Annotator properties viz. Tokenization, Sentence Splitting, Lemmatization, Parts of Speech Tagging, Constituency Parsing, Dependency Parsing and Co-reference Resolution

Supporting Domain-specific Vocabularies: Using 'WordNet' as the source of open-domain vocabulary may seem optimal at first, but it usually hinders the generation of most accurate answers, demanding information retrieval for a narrowly specified subject domain. The Information Retrieval model built for Question-Answering (QA) systems by IBM's statistic system finds greatest hindrance observed in the last step of trimming set of optimal sentences from the ranked set of passages obtained in the previous step. The best alternative for reducing such system errors is to use restricted domains as back-ground knowledge rather than open-domains [4]. In another exhaustive survey put up by L. Hirschman and R. Gaizauskas, they emphasized on the crucial role of passages in extraction of answers to the subjective questions through IR techniques [5].

# 3. SUBJECTIVE ANSWER GENERATION ALGORITHM

The syntax used in pseudo code borrows some of the elements from java syntax. All the input and output objects are specified in bold. All the language constructs like conditionals and loops use italics style. Square brackets denote index for storing and accessing array elements. Assignment operation is denoted by Symbolic notation ←. The description of various variables and procedures used in this pseudo code are as follows:

a. **Q** is the raw question string to be used for finding answers

b. **K** is the List of keyword strings present in question **Q** which can be generated using open source NLP tools.

c. **C** is the List of Corpus sections as string which can be sections or chapters defined in a standard text book

d. **V** is the source vocabulary which can be generated for all the keywords of Corpus **C** using either wordnet for enhanced domain vocab or manual human intervention for restricting domain.

e. **Get-Entry-Points** : Function for initial filtering based on count of Question keywords **K** found in different sections of Text Corpus.

f. **Get-Seed-Sentences** : Function for getting seed sentences present in a particular section based on keyword **K** and keyword vocab for given question **Q**.

g. **Get-Section-Sentences** : Function to return list of all sentences present in a text paragraph from a section.

h. **Get-Vocab** : This procedure returns a list of string for all terms supplied separately. Each list in returned composite list is synonym for respective words provided in input list.

i. **Get-Keywords** : This procedure returns a list of related keywords based on NLP dependencies provided by NLP parsers.

j. **Get-Co-Occurring-NP** : This procedure returns co-occurring NP after performing anaphora resolution of supplied text.

The algorithm for generating answers is as follows:

```
ALGORITHM Generate-Answer is
   INPUT: Question Q with Keywords K,
         Text Corpus C as List of section fragments,
         Vocab Source V
   OUTPUT: Answer A comprising concatenated fragments

   E ← Get-Entry-Points(Q,C)
   CREATE an empty list Answer_Fragments of type String
   FOR i = 0 to E.size do
      Cur_Segment ← E[i]
      Seed_Sentences ← Get-Seed-Sentences(E[i],K,C)
      Section_Sentences ← Get-Section-Sentences(E[i],C)
      CREATE an empty list SectionWise_Fragments of type
string
      FOR j = 0 to Seed_Sentences.size do
         Seed_Index ← Get-Seed-
Index(Section_Sentences,Seed_Sentences[j])
         CREATE List Seed_Vocab of type String
         CREATE List Co_Occurring_NP of type String
         Seed_Vocab ← Get-Vocab(Get-
Keywords(Seed_Sentences[Seed_Index]),V)
         Left_Marker ← Seed_Index
         Right_Marker ← Seed_Index
         WHILE there exists a String from Seed_Vocab or
Co_Occurring_NP
                  in Seed_Sentences[Left_Marker]
            Cur_Co_Occurring_NP ← Get-Co-Occurring-
NP(Seed_Sentences[Left_Marker])
            add Cur_Co_Occurring_NP to Co_Occurring_NP
            Left_Marker ← Left_Marker - 1
            IF Left_Marker = 0
               break from while loop
            END IF
         END WHILE
         WHILE there exists a String from Seed_Vocab or
Co_Occurring_NP
                  in Seed_Sentences[Right_Marker]
            Cur_Co_Occurring_NP ← Get-Co-Occurring-
NP(Seed_Sentences[Right_Marker])
            add Cur_Co_Occurring_NP to Co_Occurring_NP
            Right_Marker ← Right_Marker + 1
            IF Right_Marker = Seed_Sentences.size
               break from while loop
            END IF
         END WHILE
         INITIALIZE Cur_Frag to empty String
         FOR k = Left_Marker to Right_Marker
            Concatenate Seed_Sentences[k] to Cur_Frag
         END FOR
         Add Cur_Frag to SectionWise_Fragments
      END FOR
      Remove Duplicate sentences from SectionWise_Fragments
      INITIALIZE Cur_Section_Answer to empty String
      FOR j = 0 to SectionWise_Fragments.size do
         Concatenate SectionWise_Fragments[j] to
Cur_Section_Answer
      END FOR
      Add Cur_Section_Answer to Answer_Fragments
   END FOR
   INITIALIZE A to empty String
   FOR i = 0 to Answer_Fragments.size
      Concatenate Answer_Fragments[i] to A
   END FOR
   RETURN A
```

# 4. FURTHER APPLICATIONS AND DEVELOPMENT

This tool is observed to provide answer fragments that go fairly congenial, when compared with model answers fabricated by human assessors. The algorithm presented here is capable of generating answers with highest precision depending on the vocabulary source but some other parameters like context continuity and context span must be included in order to limit the locality of context for more accurate results with high recall. The software testing of the tool seems to provide promising results in performing fair and unbiased evaluation of students' answer scripts. Combining this algorithm with a good answer evaluation approach can provide robust answer evaluation feature for automating the digital evaluation systems.

Another field of this tool application is evaluation of online assignments at the institute level for analyzing students' appraisals on continuous scale. The up gradation scopes of such a tool development follow with real-time answer generation for different types of subjective questions presented in wide variety of grammatical styles and for versatile subject domains.

## 5. ACKNOWLEDGMENTS

This work was supported by Research and Development Laboratory, Department of Computer Science and Engineering at Bhilai Institute of Technology, Durg, Chhattisgarh, India, awaiting sponsorship from suitable funding agencies.

## 6. REFERENCES

[1] Valenti, S., Neri, F. and Cucchiarelli, A. 2003. An Overview of Current Research on Automated Essay Grading. J. of Information Technology Education (JITE), pp. 319-330.

[2] Christie, J. 1999. Assessment of Essay Marking - focus on Style and Content. In 3rd International Computer Assisted Assessment Conference (CAA) , pp. 39-45.

[3] R.Diekema, Ozgur Yilmazel, and E.D.Liddy, 2004. Minimal Ownership of Active Objects. In Proceedings of the 6th Asian Symposium on Programming Languages and Systems, APLAS 2008, Bangalore, pp. 139-154A.

[4] Parag A. Guruji, Mrunal M. Pagnis, Sayali M. Pawar and Prakash J. Kulkarni, 'Evaluation Of Subjective Answers Using Glsa Enhanced With Contextual Synonymy', International Journal on Natural Language Computing (IJNLC) Vol. 4, No.1, February 2015, pp. 51-60.

[5] Jorg Tiedemann, "Integrating linguistic knowledge in passage retrieval for question answering." Proceedings of Conference on Human Language Technology and Empirical Methods in Natural LanguageProcessing, Vancouver, British Columbia, Canada, pp. 939 - 946, 2005.