An Intelligent Approach to Automatic Query Formation from Plain Text using Artificial Intelligence

Mohd Akbar
Dept. Computer Science & Engg.
Integral University, Lucknow, India
Email: Akbar [AT] iul.ac.in

Mohd Shahid Hussain
Information Technology Department
University of Technology and Applied Sciences-CAS Ibri, Oman
Email: shahid.ibr [AT] cas.edu.om

Mohd Suaib
Dept. Computer Science & Engg.
Integral University, Lucknow, India
Email: Suaib [AT] iul.ac.in

Abstract—**Man have always been, inherently, curious creatures. They ask questions in order to satiate their insatiable curiosity. For example, kids ask questions to learn more from their teachers, teachers ask questions to assist themselves to evaluate student performance, and we all ask questions in our daily lives. Numerous learning exchanges, ranging from one-on-one tutoring sessions to thorough exams, as well as real-life debates, rely heavily on questions. One notable fact is that, due to their inconsistency in particular contexts, humans are often inept at asking appropriate questions. It has been discovered that most people have difficulty identifying their own knowledge gaps. This becomes our primary motivator for automating question generation in the hopes that the benefits of an automated Question Generation (QG) system will help humans achieve their useful inquiry needs. QG and Information Extraction (IE) have become two major issues for language processing communities, and QG has recently become an important component of learning environments, systems, and information seeking systems, among other applications. The Text-to-Question generation job has piqued the interest of the Natural Language Processing (NLP), Natural Language Generation (NLG), Intelligent Tutoring System (ITS), and Information Retrieval (IR) groups as a possible option for the shared task. A text is submitted to a QG system in the Text-to-Question generation task. Its purpose would be to create a series of questions for which the text has answers (such as a word, a set of words, a single sentence, a text, a set of texts, a stretch of conversational dialogue, an inadequate query, and so on).. (Abstract)**

Keywords- Automatic Question Generation, NLP, Intelligent Tutoring System (ITS), IR, Query Processing

I. INTRODUCTION

A question can be a language expression that is used to make a request for information, or it might be the request itself. This information could be sent along with a response. Questioning is a basic cognitive process that underpins higher-order cognitive abilities like comprehension and reasoning. The ability to ask questions is a key cognitive feature that distinguishes human from animal cognition. A excellent question[3] is concise, straightforward, and unambiguous. A The Attributes of an ideal question may be understood by following points[5]:

Eliciting the Truth: Questions must seem to be genuine , realistic and Truth inciting.

1. A question requiring numerous dimensions of response will not give you lots the info you require. "Do you love the texture and flavour of the snack?" a scholar examining a new food snack could ask. If a respondent responds no, the researcher has no idea whether they detest the texture, the flavor, or both.
2. Will it possible to accommodate all probable answers?

   MCO (Multi Choice Options) questions are the most popular type of survey question since they are the easiest for target audience to answer and the easiest to assess & Analyze.
3. Possess options that are mutually exclusive..

   A good question leaves no room for doubt in the mind of the respondent. Only one correct or appropriate option should be available to the respondent.
4. Does not presuppose a certain state of affairs.

   It doesn't seem to mind if it's a presumptive topic referring to a specific condition of affairs..
5. Does not utilise emotionally charged or ambiguous terms..

   In questions, quantifying adjectives (e.g., most, least, majority) are frequently utilised. It's critical to realise that these descriptors have varied meanings for different people.

   Moreover, apart from these attributes as mentioned above, the fine characteristics of a good question may be contemplated as described below-

- Suitable: An appropriate question is one that is well-thought-out. It emphasises memorization of only the content in your lecture and is well-aligned with the broader learning objectives.
- Precise: A excellent query is usually short and to-the-point. This eliminates any extraneous information that pupils must spend time comprehending.

- Comprehensible: A good question is written in plain, easy-to-understand language that leaves no room for ambiguity. Even if they don't know the answer, students should understand what the inquiry needs.
- Objectiveness: A query with no defined goal is useless. The purpose aids in evaluating the question against some predetermined criteria. A good query can elicit both general and specific information.
- Assisting without leading to clue: A smart question encourages the learners to recognize the notion in the image, rather than leading them to it. However, it does not provide them with concrete answers.
- Thought Provoking: A smart question forces students to think about and recall the things being taught. By directly asking, it does not safeguard them.
- Uni-Directional: A excellent inquiry is one that focuses on only one aspect at a time. If you have several ideas to assess, it's best to break them up into separate questions. However, it does not provide them with concrete answers.

A. Role and significance of Queries in Cognitive Learning

The most significant use of questions/queries is contemplation, which helps us better understand what we've learned. People frequently spend hours alone studying concepts and pondering challenges brought with what they've read. These concepts and issues are frequently expressed as questions. From the earliest stages of learning to academic research, questions are used. A question is often the starting point of an investigation in the scientific process, and it can be thought of as a bridge in-between the observation and assumption phases. Teachers and students use questions to learn about topics, and one of the most important aspects of inquiry education is having learners create "investigatable" questions. A teacher can utilise the strategy of questioning student responses to lead a student to the truth without giving direct instruction, and it also helps pupils establish logical conclusions. Students' interest in a topic has also been piqued through the use of questions. Another application of questions is to provide students with a road map for self-recognition[6] of reaching understanding milestones while they study a unit by posing some difficult questions at the start of the unit that need understanding and insight into the content to answer. The assessment of students' knowledge through exams is a common [3] and recognised use of questions in an educational context.

B. Generating Questions

Development of Question is the process of automatically producing queries from a variety of sources, such as raw text, databases, or linguistics diagrams. Though there are a variety of approaches to automatic Question Generation, it is generally regarded to be a discourse task including the following [2] four steps:

(1) When to ask the question,

(2) What the question is about, i.e. topic selection, and

(4) How to enquire.

(4) Identification of Question Category, and

(5) Basic Building blocks of Question (eg; Question Constructs)

C. Development of Questions and its Applicability

The deliberate and intentional asking and answering of questions about what's being read, aids reading-comprehension-training not only on its own pretext, but also in conjunction with a variety of other reading comprehension strategies. In addition to being a natural antecedent to Question responder (QA), QG approaches are an excellent complement to another well-proven strategy: understanding observance. Comprehension observance translates into meta-Morph-cognitive thinking and students' abilities to self-select and apply questioning tactics on a situational basis, and is frequently noted as self-regulation. Students learn to employ a variety of strategies to help them understand text material, as well as to actively select and use those approaches. Several of the most significant positive changes in students' ability to assess and enhance the quality of their own questions. It has been discovered that people of various students appear in connection with comprehension observance education. In reciprocal teaching, QG has played a significant role. The National Reading Panel designated self-questioning as the single most useful reading and comprehension strategy to learn eg; teaching children to raise their own concerns about text as they read it, as critics asking questions, except as examples to illustrate the the strategy of questioning to self is the four main methods[1] (e.g. succinct summation, question generation, elaboration, prognostication, prediction and assessment) used in conjunction all through reciprocal-teaching, question generation is the most common. The challenges of QG and QA are peculiar to systems that work on the principle of natural-languages-processing(NLP).

The benefits of using automatic systems to obtain queries can assist reduce the reliance on people for queries and other desires linked to systems functioning with NLP. Finally, QG allows people and, in some situations, AI systems to understand their surroundings and each other. The history of QG analysis can be traced back to AI, psychology, pedagogy, and the language process. One line of inquiry[2] has been theoretical, with the goal of identifying and defining the trigger points (data disparities) and mechanisms (e.g., the relationship between the type of data discrepancy and the question type) that underpin QG. The analysis on the other hand has focused on automatic QG. Automatic QG facilities provide a plethora of uses. The following are a few examples:

i. Taught smart questions that students might think of while reading documents and other media.

ii. Questions that human and computer tutors might ask in order to sell and evaluate deeper learning.

iii. Drug-related questions for patients and caregivers.

iv. Instructed questions, which may be asked by litigants in a legal environment or by interrogators in a security context.

v. Queries derived from some sources like data-repositories for FAQ (Frequently Asked Question) facilities.

D. Classification of Queries

It is vital to classify questions since different[4] types of queries require distinct ways for automatic Question-type production (eg; automatic generation of Questions based on categories).

- True/False : In semantics, any query, technically known as a polar question, is one that has two possible answers: affirmative or negative. Formally, they provide an exclusive disjuncture, or a set of options from which only one is proper. Such questions can be written in both positive and negative versions in English (for example, can "SHAUN" be here day after tomorrow? and Won't he(SHAUN) be here tomorrow? etc.)..
- 'Wh' Queries: What, why, Whom, when, who, where, which, and other interrogative wordings are considered to be present in "Wh"-questions to request data or ask for answer. They can't just be responded with a yes or no. Non-polar yes-no inquiries are distinct from polar yes-no queries, which do not essentially provide a variety of alternative answers or fundamentally prevent that vary to two options. (What moment did you get across last evening, for instance?).
- FIB / MCQ Questions: A "fill in the blank-question", also known as a diagnostic assessment question, is a statement with one or more blanks and four options for filling those blanks.

Example:

……carried the burden of our nation for 10 years, before Rohit Sharma.!

(a) Rahul (b) Sehwag (c) Sachin (d) Dhoni

II. PROBLEM DESCRIPTION

The main challenge is to produce questions automatically from given text, such as sentences or groups of sentences. An affirmative or declarative statement, or a series of sentences, is the basic input. Based on the type of phrase, the output should be a variety of questions. In general, we can say, for instance-

Input: "Text-Phrase"

Output: "Questions for reading assessment"

For instance, we have input text-phrase *"SHAUN is a good boy"*. The output should be *"Who is a good boy?"* Another Example is *"SHADDY is always murmuring"* The output should be *"Describe SHADDY."*

The main goal of this system is to provide a solution to the problem through breaking down a paragraph and then converting it into questions by mean of NLP and AI techniques using Part of Speech (POS) and Semantic analysis..

A. Model Solution Proposed

Integration and conversion are the most basic approaches to solving the problem. The statement or sentences should be split down into components and categorized before being translated into questions. The following is the basic method for extracting simple statements in order to generate questions:

We've –

Input: Intriguing Sentence infused with Complication (s).

Output: A set of refined & simple declarative-statements encoded with pattern matching rules for various syntactic constructions for decision trees.

In our proposed solution, the Python programming language is used to create the Automatic Question Paper Generator System. There is a repository of curriculum, questions, and question paper patterns in a fully functional system. It accepts a basic text, a document, or a pdf file as input and outputs a list of questions. The steps in the process for automatically creating questions are as follows:

1. Sentence Segmentation: The initial step in the pipeline is to segment the document's text into individual sentences. We can now process our document one sentence at a time now that we've separated it into sentences.

2. Word Tokenization: The second step in our pipeline is tokenization, which is the process of breaking a text into tokens or independent words. When tokenizing, we simply divide words apart anytime there is a space between them. We'll also treat punctuation marks as independent tokens because they have their own significance.

3. Guessing the Parts of Speech for Each Token: Next, we'll look at each token and try to figure out whether it's a noun, verb, or adjective. Learning the function of every single word in the given phrase will assist us in determining what the sentence is all about.

4. Text Lemmatization & Stop Word Detection: Lemmatization is the process of determining the most fundamental form or lemma of each word in a text. Stop words are often identified by checking a list of previously identified stop_words.

5. Dependency Parsing: The final step is to determine how all of the words in a sentence are related to one another. Dependency parsing is the term for this. So far, each item in our sentence has been considered as an independent entity. However, there are situations when grouping words that reflect a single idea or entity makes greater sense. We can use the dependency parse tree's information to automatically group phrases that are all referring to the similar thing.

6. Creating a Question: A usable and meaningful description of sentence have already been compiled and generated wherein each of parts of speech for every single word are understood, as well as how the words connect to one another.

B. Sentence Extraction using POS and Semantic Analysis

Take a Tree "T" as input .

Extract a set of declarative & assertive sentence trees $T_{extracted}$ from constructions in T.

For each t' in $T_{extracted}$:

Simplify "T' by eliminating modifiers.

Extract trees $T_{conjuncts}$ from conjunctions in t'.

For each $t_{conjunct}$ in $T_{conjuncts}$:

$T_{result} = T_{result}$

$\{t_{conjunct}\}$ Return

T_{result}

C. Algorithm Designed for Query Generation

```
BtnGenerateQuestions()
{// initialize the variables
StringBuilder output = new StringBuilder(); string strNounName = "";
string strMainNoun = ""; string[] sentences =
SplitSentences(txtBoxEnteredText.Text); string strQuestion;
int iFlag = 0;
int iMainNounFlag = 0;
foreach (string sentence in sentences)
{ int i = 0; string[] strText =
ParseSentence(sentence).Show().ToString().Split(' '); strText = removeBrackets(strText);
string strAdjective =""; foreach (string s in strText)
{
if ((s == "NNP") || (s == "PRP"))
{
strNounName = strText[i + 1]; if (iMainNounFlag == 0)
{ strMainNoun = strNounName; iMainNounFlag = 1; }
iFlag = 1;
```

Our Method:

1. Uses rules to extract and simplify sentences
2. Is motivated by linguistic knowledge
3. Outperformed a sentence compression baseline

Our idea here is too extract and simplify multiple statements from complex sentences including operations

```
}
if ((s == "NN") && (iFlag != 1))
{
strNounName = strText[i + 1]; if (iMainNounFlag == 0)
{ strMainNoun = strNounName; iMainNounFlag = 1;
}
iFlag = 2;
}
if ((s=="NN") || (s=="JJ"))
{
if (iMainNounFlag == 0)
{
strMainNoun = strNounName; iMainNounFlag = 1;
}
iFlag = 3;
if (s == "JJ")
{
strAdjective = strText[i + 1];
}
}
i = i + 1; }
strQuestion = removeFullStop(sentence); if (iFlag == 1)
{
strQuestion = replaceString(strQuestion, strNounName, "Who"); }
if (iFlag == 2)
{ strQuestion = replaceString(strQuestion, strNounName, "Who"); }
if (iFlag == 3)
{ strQuestion = replaceString(strQuestion, strAdjective, " "); }
if (iMainNounFlag == 1)
{ strQuestion = "Describ " + strMainNoun + "?"; } iFlag = 0;
strQuestion = "";}
}
```

The functions used by the algorithm to generate the questions are

// Replaces the Noun and Adjective's in questions string replace String (string sentence, string to Replace, string replaced With)

{ return (sentence.Replace(toReplace, replacedWith) + "?"); }

string[] removeBrackets(string[] strText)

{ int i = 0;

foreach (string s in strText)

{

strText[i] = strText[i].Trim(new Char[] { '(', ')', '.', ' '}); i = i + 1; }

return strText; }

// Used as preprocessing to remove the parser added text

string removeFullStop(string sentence)

{ return (sentence.Trim(new Char[] { '.' })) ;

}

// Used to split the sentences

string[] SplitSentences(string paragraph)

{ if (mSentenceDetector == null)

{ mSentenceDetector = new OpenNLP.Tools.SentenceDetect.EnglishMaximumEntropy SentenceDetector(mModelPath + "EnglishSD.nbin");

} return mSentenceDetector.SentenceDetect(paragraph); }

// Generates the tokens from the sentences given

string[] TokenizeSentence(string sentence)

{ if (mTokenizer == null)

{ mTokenizer = new OpenNLP.Tools.Tokenize.EnglishMaximumEntropyToke nizer(mModelPath + "EnglishTok.nbin");

} return mTokenizer.Tokenize(sentence); }

// This tags the token generated from sentences

string[] PosTagTokens(string[] tokens)

{ if (mPosTagger == null)

{ mPosTagger = new OpenNLP.Tools.PosTagger.EnglishMaximumEntropyPos Tagger(mModelPath + "EnglishPOS.nbin", mModelPath + @"Parser\tagdict"); }

return mPosTagger.Tag(tokens);}

// Used to parse the sentence to generate a single parse tree

Parser.Parse ParseSentence(string sentence){ if (mParser == null)

{ mParser = new OpenNLP.Tools.Parser.EnglishTreebankParser(mModelPa th, true, false); }

return mParser.DoParse(sentence); }

// Used to find "Names" like name of location, money, organization, percentage, person and time from the postages

string FindNames(string sentence)

{ if (mNameFinder == null)

{ mNameFinder = new OpenNLP.Tools.NameFind.EnglishNameFinder(mModelP ath + "NameFind\\"); }

string[] models = new string[] { "location", "money", "organization", "percentage", "person", "time" };

return mNameFinder.GetNames(models, sentence); }.

III. RESULTS AND YANALYSIS

The phrased lines were put to the test on the system, and the results were compared to questions made by a human who is fluent in English and can ask efficient questions. The human subject is regarded to be the best subject for comparing outcomes. The findings are shown in table 1.1, followed by inferences.-

Table-1.1: POS Analysis Stats

S. No	No of Sentences	No of Correct Questions	No of Incorrect Questions	No of Questions By Human
1	1	2	0	2
2	1	2	0	2
3	2	3	1	2
4	1	2	0	2
5	1	2	0	2
6	1	2	0	2
7	2	1	1	3
8	1	2	0	2
9	1	2	0	2
10	1	3	0	3

Total no of sentences = 13

Total no of correct questions = 21

Total no of incorrect questions = 1

Total no of questions by the human = 22

The above result shows that the system is working fairly correct with an accuracy of over 90% which may further be improved.

IV. CONCLUSION

The algorithm, proposed and implemented in our research, performs admirably well when dealing with fairly large amounts of text, such as sentences. With a few simple tweaks, rather accurate questions are created. Not only does the system generate factual questions, but it also generates some descriptive questions. That is the system's additional flavour. In rare circumstances, the system fails when faced with extremely complicated sentences. I buried the roots of a tree that will grow tall in the future in this work. This work exemplifies what NLP is capable of. The proposed system of ours can be utilised in a variety of settings, including educational institutions. Because it is a web-based application, the user's location is irrelevant. This system can be accessed by the administrator from any location at any time. This technique also efficiently assesses the candidate's capability and talents. It's a totally automated method that generates quick results. There is no need to carry paper to all colleges using police/security vehicles in this approach. This system generates findings that are completely unbiased. The use of this method significantly minimizes human effort and reduces time and resources.

V. FUTURE WORK

The future appears to be promising. Following the successful implementation of this task, it will lead to the generation of questions from the complete document in the future. In that instance, it would be necessary to first identify which sentences from a document can be used to generate questions. The invented system along with developed algorithm can be deployed with optimal improvements. The eventual system will be able to handle any pdf, word, or other sort of text file independently, evaluate it, and find key sentences for QG. With minimal error & inaccuracy, a variety of questions might be derived from those sentences. Our proposed system can be expanded to produce True-False questions as well as FIB questions(fill-in-the-blanks). In addition to this, the feature of "automatic checking of answer-copies" can also be incorporated as a future work extension to check the answer sheets of examinee producing assessment result with fine accuracy.

REFERENCES

[1] Karen Mazidi, Rodney D. Nielsen, "Linguistic Considerations in Automatic Question Generation", Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 321–326, Baltimore, Maryland, USA, June 23-25 2014 " Automatic Question Genaration Asyntactical Approach" Husam Deeb Abdullah Deeb Ali, 2012

[2] Manish Agarwal and Prashanth Mannem, "Automatic Gap-fill Question Generation from Text Books", Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications, pages 56–64, Portland, Oregon, 24 June 2011.

[3] "Using Questions to Teach Better"Rick Garlikov-http://www.garlikov.com/teaching/usingquestions.html, 2010

[4] D. Metzler and W. B. Croft, "Analysis of statistical question classification for fact-based questions," Inf. Retr., vol. 8, no. 3, pp. 481–504, May2005.

[5] Jonathan C. Brown, Gwen A. Frishkoff, Maxine Eskenazi "Automatic Question Generation for Vocabulary Assessment", Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 819–826, Vancouver, October 2005

[6] M. Agarwal, "Cloze and open cloze question generation systems and their evaluation guidelines," Journal of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 819–826, Vancouver, October 2005

[7] Goto, T. Kojiri, T. Watanabe, T. Iwata, and T. Yamada, "An automatic generation of multiple-choice cloze questions based on statistical learning," in proceedings of the 17th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education, 2009, pp.415–222.

[8] J. C. Brown, G. A. Frishkoff, and M. Eskenazi, "Automatic ques- tion generation for vocabulary assessment," inroceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, ser.

[9] HLT '05. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp.819–826.

[10] C.-Y. Chen, H.-C. Liou, and J. S. Chang, "Fast: An automatic generation system for grammar tests," in Proceedings of the COL- ING/ACL on Interactive Presentation Sessions, ser. COLING-ACL '06. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp.1–4.

[11] Manish Agarwal and Prashanth Mannem, "Automatic Gap-fill Question Generation from Text Books", Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications, pages 56–64, Portland, Oregon, 24 June 2011.

[12] I. S. Jacobs and C. P. Bean, "Generating question from text using context analysis," in International Journal of Computer Science & Information Reasearch Advancement, vol. III, M. Akbar,. A.K Shaunl, Eds. Boasten: Academic, 2018, pp. 312–323.

[13] Alessi, S. M., Anderson, R. C., Anderson T. H., Biddle, W. B., Dalgaard, B. R., Paden, D. W., Smock, H. R., Surber, J. R., & Wietecha, E. J. Development and implementation of the computer assisted instruction study management system (CAISMS). San Diego, California: Navy Personnel Research and Development Center, Technical Report TR 74-29, February 1974.

[14] Anderson, T. H., Anderson, R. C., Alessi, S. M., Dalgaard, B. R., Paden, D. W., Biddle, W. B., Surber, J. R., & Smock, H. R. A multifaceted computer-based course management system. San Diego, California: Navy Personnel Research and Development Center, Technical Report TR 75-30, April 1975.

[15] Anderson, T. H., Anderson, R. C., Dalgaard, B. R., Paden, D. W., Biddle, W. R., Surber, J. R., & Alessi, S. R. An experimental evaluation of a computer-assisted instruction study management system (CAISMS). San Diego, California: Navy Personnel Research and Development Center, Technical Report TR 75-31, April.

[16] 197"Automatic Question Generation using Discourse Cues and Distractor Selection for Cloze Questions" Rakshit Shah Language Technology and Research Center (LTRC)

[17] Isha Sood, 7 Characteristics Of A Good Question. Online available at:https://elearningindustry.com/characteristics-of-a-good-question-7

[18] "Automatic Factual Question Generation from Text" Michael Heilman CMU-LTI-11-004

[19] "Towards Automatic Topical Question Generation" Y llias ,Chali Sadid , A.Hasan University of Lethbridge, Lethbridge, AB, Canada, 2011

[20] https://www.statpac.com/surveys/question- qualities.htm

[21] Ming Liu, R. A. Calvo, A. Aditomo, L.A. Pizzato, "Using Wikipedia and Conceptual Graph Structures to Generate Questions for Academic Writing Support" Third Quarter 2012, pp. 251-263, vol. 5 DOI Bookmark: 10.1109/TLT.2012.5

[22] T. Tokunaga, H. Nishikawa, and H. Obari, "Evalu- ation of automatically generated english vocabulary questions," Research and PracticeinTechnologyEnhancedLearning,vol.12,no.1,p.11,2017.

[23] R. Mitkov, L. An Ha, and N. Karamanis, "A computer-aided en- vironment for generating multiple-choice test items," Nat. Lang. Eng., vol. 12, no. 2, pp. 177–194, Jun.2006.

[24] M. Liu, V. Rus, and L. Liu, "Automatic chinese factual question generation," IEEE Transactions on Learning Technologies, vol. 10, no. 2, pp. 194–204,2017.

[25] D. Metzler and W. B. Croft, "Analysis of statistical question classification for fact-based questions," Inf. Retr., vol. 8, no. 3, pp. 481–504, May2005.

[26] M. Agarwal, "Cloze and open cloze question generation systems and their evaluation guidelines," International Institute of Informa- tion Technology, Hyderabad,2012.

[27] T. Goto, T. Kojiri, T. Watanabe, T. Iwata, and T. Yamada, "An automatic generation of multiple-choice cloze questions based on statistical leang," in roceedings of the 17th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education, 2009, pp.415–222.

[28] J. C. Brown, G. A. Frishkoff, and M. Eskenazi, "Automatic ques- tion generation for vocabulary assessment," in Proceedings of the ConferenceonHumanLanguageTechnologyandEmpiricalMethod s in Natural Language Processing, ser. HLT '05. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp.819–826.

[29] C.-Y. Chen, H.-C. Liou, and J. S. Chang, "Fast: An automatic generation system for grammar tests," in Proceedings of the

[30] COL- ING/ACL on Interactive Presentation Sessions, ser. COLING-ACL '06. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp.1–4.