Question Generation Using POS tagging

Project-I (CS57006) report submitted to

Indian Institute of Technology Kharagpur

in partial fulfilment for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Gangaram Sudewad

(20CS30017)

Under the supervision of

Professor Sudeshna Sarkar

Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

Autumn Semester, 2023-24

November 3, 2023

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: November 3, 2023 (Gangaram Sudewad)
Place: Kharagpur (20CS30017)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**Question Generation Using POS tagging**" submitted by **Gangaram Sudewad** (Roll No. 20CS30017) to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering is a record of bona fide work carried out by him under my supervision and guidance during Autumn Semester, 2023-24.

Date: November 3, 2023

Place: Kharagpur

Professor Sudeshna Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur - 721302, India

Abstract

Name of the student: **Gangaram Sudewad** Roll No: **20CS30017**

Degree for which submitted: **Bachelor of Technology**

Department: **Department of Computer Science and Engineering**

Thesis title: **Question Generation Using POS tagging**

Thesis supervisor: **Professor Sudeshna Sarkar**

Month and year of thesis submission: **November 3, 2023**

In education, human-generated questions are commonly used to assess a student's understanding of a text. However, creating a computer program to do this is challenging due to the complexities of human language. Many existing methods rely on expensive tools for accuracy. In this research, we propose a new approach that uses machine learning and pattern matching based on parts of speech to generate knowledge-testing questions from English text without the need for large semantic tools. This thesis leverages the Stanford Question Answering Dataset to enhance question-answering capabilities for textual data in PDF and TXT file formats. Our research demonstrates the effective use of question-answering models, trained on SQuAD, to process and extract valuable information from PDF and TXT documents.

Contents

Abbreviations

CC	**C**oordinating conjunction
CD	**C**ardinal number
DT	**D**eterminer
IN	**P**reposition or subordinating conjunction
JJ	**A**djective
JJR	**A**djective, comparative
JJS	**A**djective, superlative
NN	**N**oun, singular or mass
NNS	**N**oun, plural
NNP	**P**roper noun, singular
NNPS	**P**roper noun, plural
PDT	**P**redeterminer
POS	**P**ossessive ending
PRP	**P**ersonal pronoun
PRP$	**P**ossessive pronoun
RB	**A**dverb
RBR	**A**dverb, comparative
RBS	**A**dverb, superlative
UH	**I**nterjection
VB	**V**erb, base form
VBD	**V**erb, past tense
VBG	**V**erb, gerund or present participle
VBN	**V**erb, past participle
VBP	**V**erb, non-3rd person singular present
VBZ	**V**erb, 3rd person singular present
WDT	**W**h-determiner
WP	**W**h-pronoun
WP$	**P**ossessive wh-pronoun
WRB	**W**h-adverb

Chapter 1

Question Generation

1.1 Introduction

Problem involves transforming sentences into question-answer pairs, which is the issue we are addressing. Question generation using natural language processing is the automated process of creating questions from a given text or context. It's important because it streamlines education, content creation, and information retrieval, making interactions with AI systems more natural and efficient. This technique can be used for generating educational quizzes, enhancing content with questions, improving search engine queries, and advancing the capabilities of conversational AI.The primary challenge in question generation is maintaining both the meaning and grammatical correctness of the original sentence. To craft a question, we may need to alter the sentence structure, add or remove words, change verb tenses or parts of speech, or perform other complex operations. Importantly, through these operations, we must ensure that the question still conveys the same meaning as the original statement and correctly identifies the answer. Nevertheless, a proficient question generator can find applications across various domains, from enhancing automated educational tools to improving AI-driven conversations. Our proposed approach for question generation relies on part-of-speech pattern matching, leveraging

Inversion Transduction Grammars (ITG), and is trained on a dataset of sentence-question pairs. We focus on input sentences containing a single independent clause, with the belief that this method can be applied more broadly if we preprocess more complex inputs.

1.2 Problem Definition

Any collection of English sentences containing one independent clause will be accepted as input. For the best results, sentences should be well-formed and use proper English grammar.The output will be a set of question-answer pairs inquiring about the original text's contextual knowledge. The output questions should be grammatically correct as well. Here are a few examples.

- *John* drove the car to work. → Who drove the car to work? *John*
- The pump is now operational. → Is the pump operational? Yes
- He waters the garden every day. → What does he do every day? waters the garden

Chapter 2

Used and Related Methods

2.1 Related work for Question Generation

Numerous efforts have been made to automate the interpretation of natural human languages, with most focusing on solving specific sub-problems.

Wolfe[5] brought question generation to the attention of the natural language processing field in 1976. He discussed the objective, applicability, and potential obstacles of a question generator. Since then, several have created question generators with a narrow concentration

Brown[1] focuses solely on vocabulary-testing questions and use a WordNet to boost question complexity without sacrificing semantic accuracy. They also use part of speech (POS) tagging to ensure the question's grammatical accuracy.

Kunichika[4] takes the most general approach of all, examining the original sentence's syntactic and semantic structure before formulating the question. After considering both of these, their method may generate a wide range of questions about the initial declarative text.

However, this strategy is strongly reliant on the correctness of sentence interpretation utilizing tools such as WordNets, which may not be accurate in all circumstances.

These are three depictions of the finest solutions available today, none of which employ machine learning. Our strategy will be highly reliant on a POS tagger.
On a different note, Heilman[3] ranks generated questions that could be added to our question generation process later in the development process.

2.2 Producing POS Pattern Templates from the Corpus

The first significant step in processing our corpus instances is to find the phrases that remain consistent as we move from declarative sentence to question-answer combination. We accomplish this by searching the instance for phrases with the exact same language, beginning with the largest possible phrases and gradually decreasing the size until all of the similar phrases are located. This is referred to as chunking.
Figure 1.1 depicts such an event as well as the words that were recognized once chunking was done.

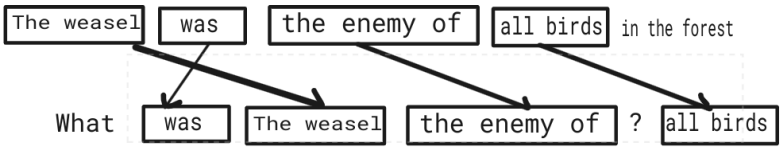

FIGURE 2.1: Sample of chunking the common phrases from an instance in our corpus

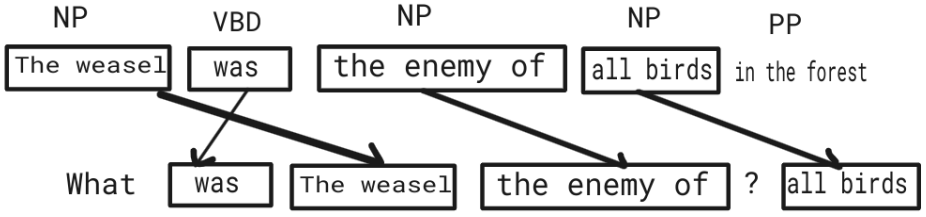

FIGURE 2.2: The example from figure 1 with its sentence chunks labeled with their POS

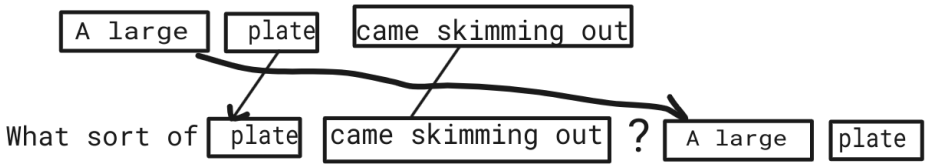

FIGURE 2.3: Sample of chunking an instance where a phrase is repeated in all three parts of an instance. This helps produce templates of questions that quiz on adjectives in the input sentence while still keeping accuracy

After labeling the sentence chunks, we remove all of the phrases that appeared in the instance's sentence part. The phrases that appeared only in the question or answer are retained as part of the template. This is the final step in creating our POS template from a corpus instance. This process is repeated for each instance in the corpus before attempting to apply these templates to our input sentences. Figure 1.2 depicts the final step in our example.

Chapter 3

Sequential Phases for Question Generation

3.1 Input Sentence Preparation

- **Initial Step:** Input sentence preparation is the initial step in the question generation process.
- **Cleaning the Input Sentence:** The script takes an input sentence and begins by cleaning it. This cleaning involves removing special characters and converting the text to lowercase.
- **Removing Special Characters:** Special characters such as quotation marks, curly quotes (e.g., “” or ‘’), and double quotes are removed to ensure uniformity in text analysis.
- **Converting to Lowercase:** Converting the text to lowercase helps standardize the text for consistent analysis. It ensures that words are not treated differently due to variations in capitalization.

3.2 Part-of-Speech Tagging

- **Understanding Grammatical Roles:** Part-of-speech tagging is a crucial NLP technique used to understand the grammatical roles of words in a sentence.
- **Tokenization and Tag Assignment:** The script uses the TextBlob library to tokenize the cleaned sentence and assign grammatical categories or "tags" to each word in the sentence.
- **Importance for Question Construction:** These tags help identify whether a word is a noun, verb, adjective, pronoun, adverb, etc. This information is essential for determining the roles of words in constructing questions.

3.3 Identifying Questionable Elements

- **Candidate Identification:** After part-of-speech tagging, the script identifies specific types of words and grammatical elements that are potential candidates for forming questions.
- **Types of Elements:** These elements include nouns, pronouns (such as "he," "she," "it"), possessive pronouns (e.g., "his," "her"), and more.
- **Central Role in Question Formation:** The script focuses on these elements because they often play a central role in question formation. For example, "Who is [someone]?" or "What is [something]?"

3.4 Question Generation

- **Initiating Question Generation:** Once the script has identified the potentially relevant elements, it begins the process of generating questions.

- **Searching for Question Candidates:** It iterates through the tagged words in the sentence and searches for specific combinations of words and tags that indicate the potential formation of a question.
- **Constructing Questions:** When a suitable combination is identified, the script constructs a question based on the context. The specific question format may vary, such as "Who is [someone]?" or "What is [something]?"
- **Predefined Rules and Conditions:** The script uses a set of predefined rules and conditions to determine the question format, including recognizing verbs and personal pronouns.

Here default question is "Who." However, it may be modified based on the following conditions:

- If the part-of-speech tag of the following word (`item[index + 1]`) is 'VBG' (present participle), the question is changed to "Who is."
- If the part-of-speech tag of the current word (`item[index]`) is 'PRP$' (possessive pronoun), the question is changed to "Whose."
- If the part-of-speech tag of the current word (`item[index]`) is 'NN' (singular noun) and it's not the word 'i' or 'ive,' and the part-of-speech tag of the following word is not 'is,' the question is changed to "What."
- If the current word is 'it,' the question is changed to "What."

3.5 Question Formatting

- **Formatting Process:** The generated questions are then formatted into a required generalized form.

- **Modifications for Structure and Grammar:** Formatting involves making modifications to the questions to ensure they follow a desired structure and adhere to grammatical rules.
- **Tense Modifications:** In the script, formatting includes making tense modifications to questions (e.g., changing past tense to future tense) and removing specific names to make the question more general and applicable to various contexts.
- **Generic Term Usage:** For instance, it may replace a specific name with a generic term like "someone."

3.6 Saving Questions and Metadata

- **Knowledge Base Maintenance:** The script maintains a knowledge base, which is essentially a dictionary that stores the generated questions, original text, and formatted questions.
- **Easy Access to Questions and Metadata:** This knowledge base allows for easy access to the generated questions and their associated metadata.
- **Metadata Storage:** Metadata, including the original text, generated questions, and formatted questions, is saved in two formats: JSON and CSV files for future reference and analysis.

3.7 POS tagging using Viterbi algorithm

The Viterbi algorithm is a dynamic programming algorithm used in part-of-speech (POS) tagging, which is the process of assigning a grammatical category (such as noun, verb, adjective, etc.) to each word in a natural language text. It helps in determining the most likely sequence of POS tags for a given sentence, based on the

words in that sentence.

Here's a step-by-step explanation of how the Viterbi algorithm works for POS tagging:

- **Input Data:** The input to the algorithm is a sequence of words (a sentence), and the goal is to find the most likely sequence of POS tags for each word in the sentence.
- **Initialization:** Start by defining a set of possible POS tags for the first word in the sentence. For example, if the first word is "The," the possible POS tags might include "article" or "determiner."
- **Transition Probabilities:** Define a set of transition probabilities that represent the likelihood of transitioning from one POS tag to another. These probabilities are typically derived from a training corpus, which contains labeled examples of word-POS tag pairs.

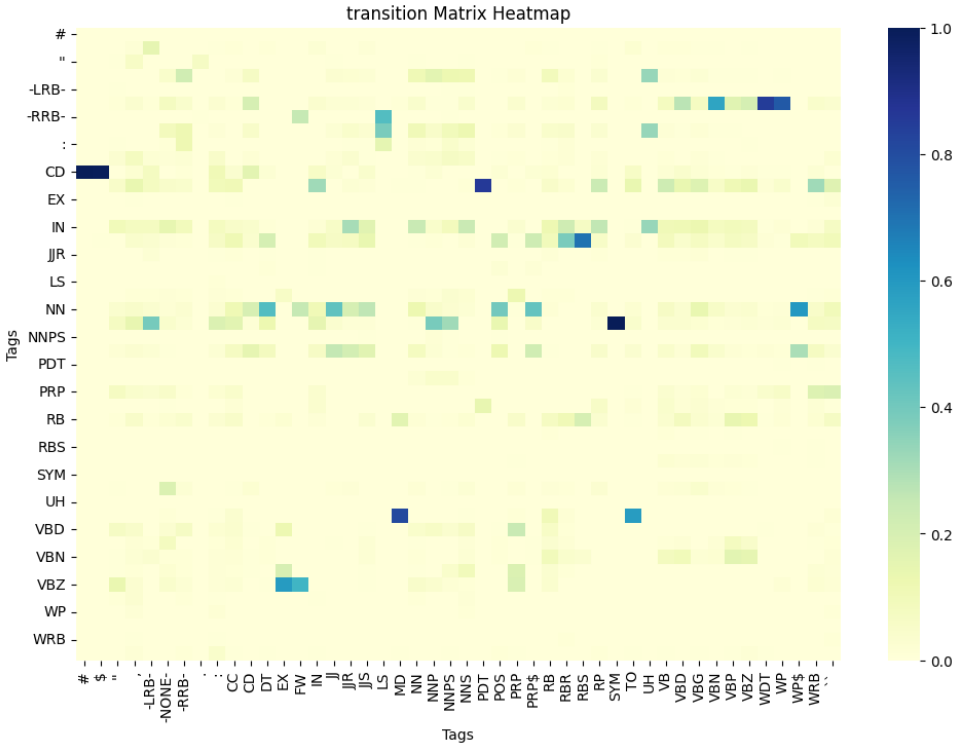

FIGURE 3.1: Transition matrix heatmap

$$\text{Transition[i][j]} = \frac{\text{Count(transition from POS tag } i \text{ to POS tag } j)}{\text{Count(POS tag } i)}$$

- **Emission Probabilities:** Define a set of emission probabilities that represent the likelihood of a particular word being associated with a specific POS tag. These probabilities are also derived from the training corpus.
$$\text{Emissionmatrix[j][k]} = \frac{\text{Count}(word\ k\ with \text{ POS tag } j)}{\text{Count(POS tag } j)}$$

- **Dynamic Programming:** The Viterbi algorithm uses dynamic programming to find the most likely sequence of POS tags for each word in the sentence. It does this by considering all possible POS tags for the current word and then determining the most likely sequence of POS tags for the previous word.

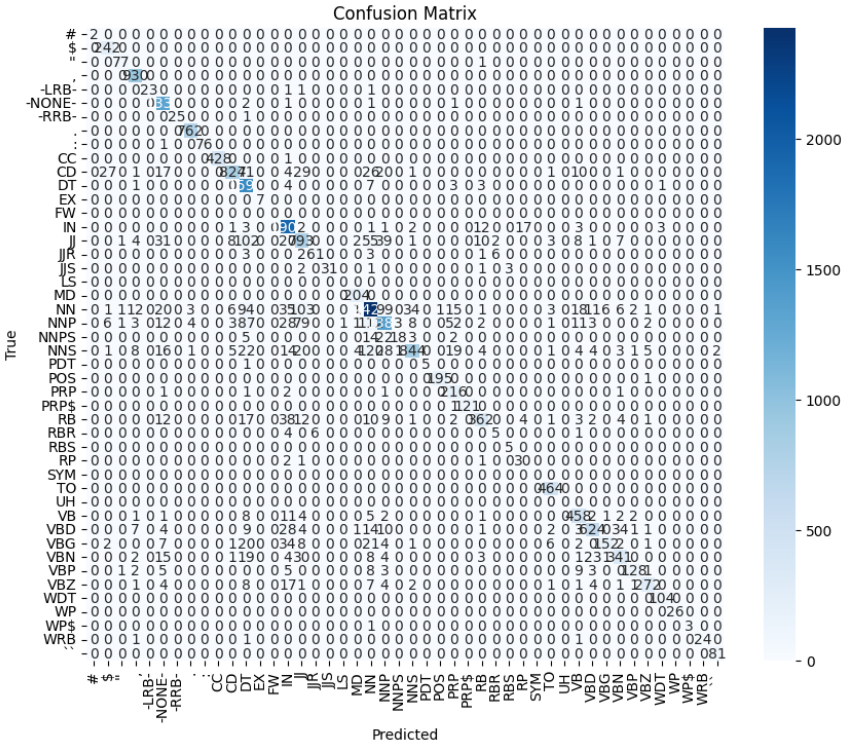

FIGURE 3.2: Confusion Matrix

- **Calculate the Best Path:** For each word in the sentence, the algorithm calculates the probability of the current word having each possible POS tag, taking into account the transition probabilities from the previous word and the

emission probability for the current word. It then selects the POS tag with the highest probability as the most likely tag for the current word.

- **Backtracking:** To find the overall most likely sequence of POS tags for the entire sentence, the algorithm backtracks from the last word to the first word, selecting the best POS tags at each step.

- **Output:** The result of the Viterbi algorithm is the most likely sequence of POS tags for each word in the sentence. These POS tags are used to analyze the grammatical structure of the text.

$$\text{Precision} = \frac{TruePositives}{TruePositives + FalsePositives} F1Score = \frac{2 \cdot (Precision \cdot Recall)}{Precision + Recall}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

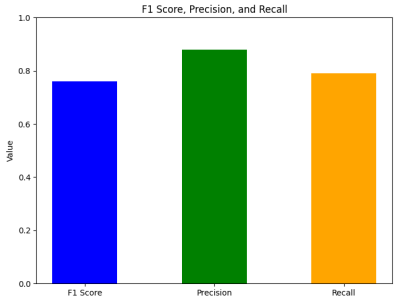

FIGURE 3.3: Result

Metric	Value
F1 Score	0.7600
Precision	0.8800
Recall	0.7900

FIGURE 3.4: Result table

3.8 Result

In the results, we successfully generated questions from the part-of-speech tagged data. The part-of-speech tagging process revealed valuable insights into the structure of Anne Frank's diary text. Examples of the generated questions highlight the effectiveness of our approach. Screenshots of these questions are included for visual reference, demonstrating the practical application of part-of-speech tagging in question generation

Some of the sample questions generated from Anne Frank diary are provided below:

---

1 **Question:** Who began her diary and fifteen when she was forced to stop

Tags: [('who', 'WP'), ('was', 'VBD'), ('t', 'RB'), ('hirteen', 'JJ'), ('when', 'WRB'), ('she', 'PRP'), ('began', 'VBD'), ('her', 'PRP'), ('diary', 'JJ'), ('and', 'CC'), ('fifteen', 'JJ'), ('when', 'WRB'), ('she', 'PRP'), ('was', 'VBD'), ('forced', 'VBN'), ('to', 'TO'), ('stop', 'VB')]

Original Text: who was thirteen when she began her diary and fifteen when she was forced to stop

Formatted Text: Who will begin their diary and fifteen when she will be forced to stop

Question Tags: [('Who', 'WP'), ('began', 'VBD'), ('her', 'PRP$'), ('diary', 'JJ'), ('and', 'CC'), ('fifteen', 'JJ'), ('when', 'WRB'), ('she', 'PRP'), ('was', 'VBD'), ('forced', 'VBN'), ('to', 'TO'), ('stop', 'VB')]

---

2 **Question:** Who willed his daughter 's manuscripts to the netherlands state institute for war documentation in amsterdam

Tags: [('he', 'PRP'), ('willed', 'VBD'), ('his', 'PRP$'), ('daughter', 'NN'), ("'s", 'POS'), ('manusc', 'NN'), ('ripts', 'NNS'), ('to', 'TO'), ('the', 'DT'), ('netherlands', 'NNS'), ('state', 'NN'), ('institute', 'NN'), ('for', 'IN'), ('war', 'NN'), ('documentation', 'NN'), ('in', 'IN'), ('amsterdam', 'NN')]

Original Text: he willed his daughter's manuscripts to the netherlands state institute for war documentation in amsterdam

Formatted Text: Who will will their daughter's manuscripts to the netherlands state institute for war documentation in amsterdam

Question Tags: [('Who', 'WP'), ('willed', 'VBD'), ('his', 'PRP$'), ('daughter', 'NN'), ("'s", 'POS'), ('manusc', 'NN'), ('ripts', 'NNS'), ('to', 'TO'), ('the', 'DT'), ('netherlands', 'NNS'), ('state', 'NN'), ('institute', 'NN'), ('for', 'IN'), ('war', 'NN'), ('documentation', 'NN'), ('in', 'IN'), ('amsterdam', 'NN')]

Question generated from other datasets:

Who will talk in hushed:

Who talked in hushed

Ans: While the boys talked in hushed

Who will pick their way carefully through the drifts:

Who picked their way carefully through the drifts

Ans: The riders picked their way carefully through the drifts

What will be laughing and joking as someone rode:

What was laughing and joking as he rode

Ans: Greyjoy was laughing and joking as he rode

What will hear the breath go out of someone:

What heard the breath go out of him

Ans: Bran heard the breath go out of him

What will grin and look up from the bundle in their arms:

What grinned and looked up from the bundle in his arms

Ans: Robb grinned and looked up from the bundle in his arms

What will be afire with curiosity by then:

What was afire with curiosity by then

Ans: Bran was afire with curiosity by then

What will jump off and run:

What jumped off and ran

Ans: Bran jumped off and ran

3.9 Future Works and Upgrades for Question Generation Tool

- **Enhance Question Variety:** Expand rules for more diverse question types.
- **Input Format Support:** Add support for more file formats.
- **Improved Verb Handling:** Consider verb tenses and modal verbs for accuracy.
- **Customizable Output:** Allow different output formats.
- **NLP Integration:** Use advanced NLP libraries for better analysis.
- **User Interface:** Create a user-friendly web app for accessibility.
- **Performance Optimization:** Optimize for large documents and datasets.
- **Multilingual Support:** Extend to multiple languages and grammar rules.

Bibliography

[1] Jonathan Brown, Gwen Frishkoff, and Maxine Eskenazi. Automatic question generation for vocabulary assessment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 819–826, 2005.

[2] Shay Cohen. Part-of-speech tagging. 2015.

[3] M. Heilman and N. Smith. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617, Los Angeles, USA, 2010. Association for Computational Linguistics.

[4] H. Kunichika, T. Katayama, T. Hirashima, and A. Takeuchi. Automated question generation methods for intelligent english learning systems and its evaluation. In *Proceedings of International Conference of Computers in Education 2004*, pages 2–5, Hong Kong, China, 2003.

[5] John H Wolfe. Automatic question generation from text-an aid to independent study. In *Proceedings of the ACM SIGCSE-SIGCUE technical symposium on Computer science and education*, pages 104–112, 1976.

[6] Jacob Zerr. Question generation using part of speech information. *Final Report for REU Program at UCCS*, pages 19–23, 2014.