Question Generation

Using POS Tagging

Gangaram Arvind Sudewad
20CS30017

01

INTRODUCTION

02

RELATED WORKS

03

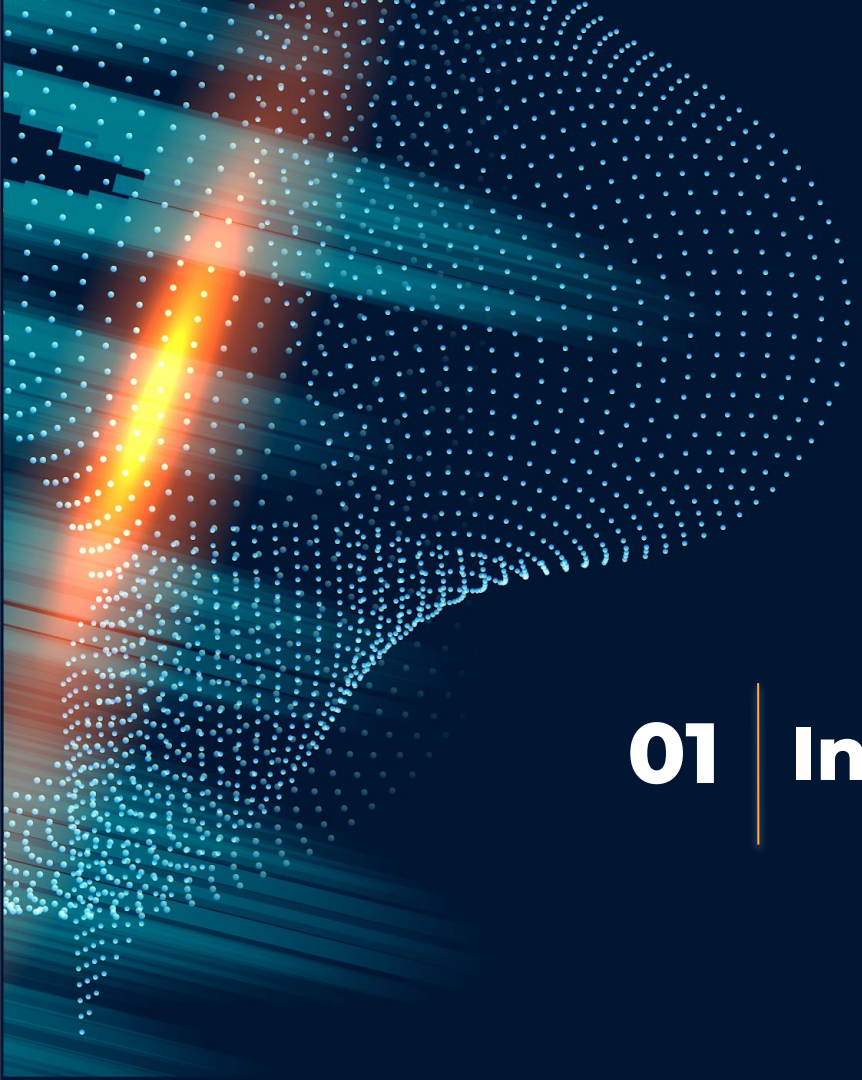
METHODOLOGY

04

RESULTS

05

FUTURE WORK



01 Introduction

What’s the focus?

NLP

Concerns with the interaction of machine and humans in natural languages, leading to applications like chatbots, language translation, sentiment analysis, and text summarization.

QG

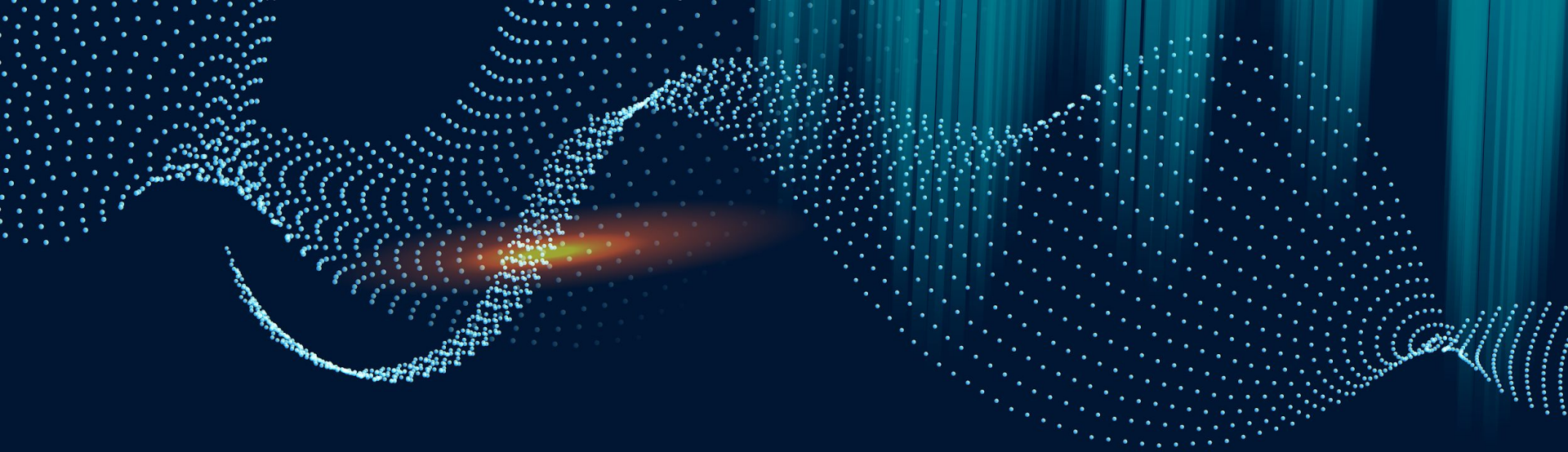
Involves automatically generating questions from a given text or content.
The goal is to create relevant questions that can help in comprehension.
For now only text from .txt and .pdf .

MOTIVATION

Dataset: Generating a large-scale corpus of question-answer pairs of acceptable quality

Education: Generating quality questions with no bias and repetition to evaluate student performance

Customization: Customizing content into questions -answer pairs, for chat boxes and Q/A system



02 | Related Works

Question Generation

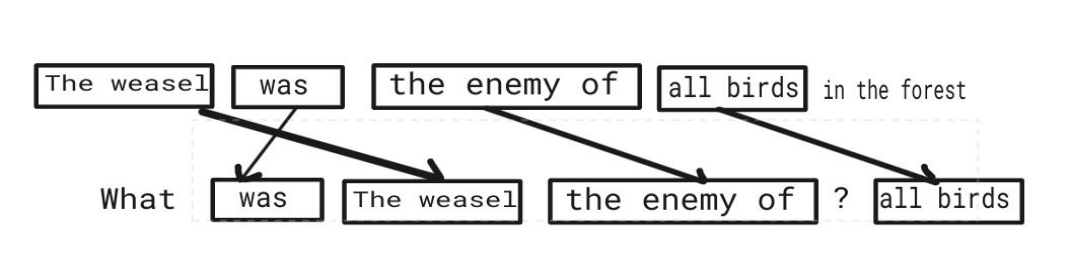
- Wolfe [5] brought question generation to the attention of the natural language processing field in 1976. He discussed the objective, applicability, and potential obstacles of a question generator. Since then, several have created question generators with a narrow concentration
- Brown [1] focuses solely 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 is 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.

03 | Methodology

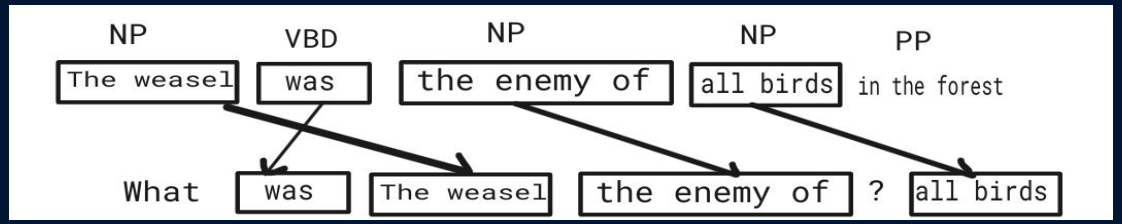
Producing POS Pattern Templates

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.

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



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.



The example from above with its sentence chunks labeled with their POS

Dataset

SQuAD

stanford question answering dataset was used. Extracted sentence-question-answer triples.

Treebank

From nltk for training data for implementing Part of Speech tagger.

Sequential Phases for Question Generation

Input Sentence Preparation

Cleaning involves removing special characters and converting the text to lowercase. It ensures that words are not treated differently due to variations in capitalization.

Part-of-Speech Tagging

The script uses the TextBlob library to tokenize the cleaned sentence and assign grammatical categories or ”tags” to each word in the sentence.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.

POS tagging using Viterbi algorithm

- Initialization: Initialize the first column of the Viterbi matrix with the initial state probabilities.
- Recursion: For each subsequent column in the matrix, calculate the maximum probability for each state based on the previous column's probabilities and transition probabilities.

  Transition[i][j]=Count(transition from POS tag i to POS tag j)/Count(POS tag i)
  Emission[j][k]=Count(word k with POS tag j)/Count(POS tag j)

- Backtracking: Keep track of the best path (sequence of states) by selecting the state with the highest probability at each time step.
- Termination: Once all columns are processed, select the final state with the highest probability as the most likely sequence of hidden states.
- Return: Return the best path, which represents the most likely sequence of hidden states.

Identifying Questionable Elements

Focuses on these elements because they often play a central role in question formation. For example, ”Who is [someone]?” or ”What is [something]?” These elements include nouns, pronouns (such as ”he,””she,” ”it”), possessive pronouns (e.g., ”his,” ”her”), and more.

Rules And Conditions To Determine The Question

- If the POS tag of the following word is ’VBG’ (present participle), the question is changed to ”Who is.”
- If the POS tag of the current word is ’PRP$’ (possessive pronoun), the question is changed to ”Whose.”
- If the POS tag of the current word 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.”

Saving Questions and Metadata

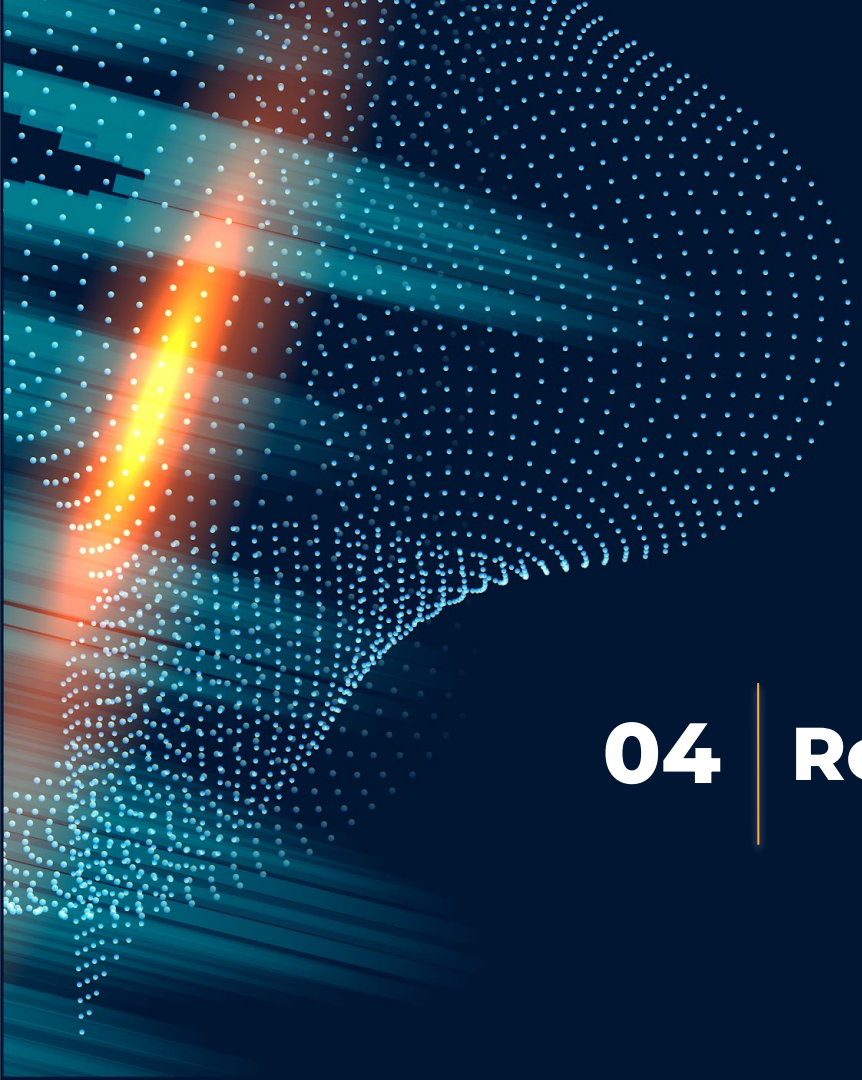
Storing key information.
Benefit: Enables quick access to questions and metadata.
Importance: Aids in future reference and analysis.
Explanation: "Used to centralize and organize generated questions and their related information for efficient retrieval and analysis."

Metadata

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



04 | Results

Results

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

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

Using spaCy model ”en core web sm-3.0.0” and cosine similarity to assess the similarity between generated questions and Stanford Question Answering Dataset (SQuAD 2.0) ground truth questions. It calculates key similarity metrics, such as average, maximum, minimum, standard deviation, and range, for quality evaluation.

Generated Qns: What was fought between the colonies of British America and New France?
Most Similar Ground Truth Qns: Who fought in the French and Indian war?
Similarity Score: 0.648 (64.8%)

Metric	value	percentage
Avg similarity	0.478	47.8%
Max similarity	0.648	64.8%
Max similarity	0.144	14.4%
Std deviation	0.155	-
Range	0.504	-

05 Future work

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.

References

[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.

THANKS!