National University of Singapore

Final Year Project (CP4101)

H278050: Understanding the Effectiveness of Large Language Models in Question Generation

Sumanth Yalamarty

Supervised by

Dr. Christian von der Weth

School of Computing

10 April 2024

Abstract

In the field of education, the ability to generate high-quality questions is a crucial skill for enhancing student engagement, critical thinking, and assessment. Traditional approaches to question generation have been labor-intensive and require significant subject matter expertise. However, the recent advancements in large language models (LLMs) have opened up new possibilities for automating this process.

This final year project explores the capabilities of LLMs, such as GPT-4, in generating educational questions. There are different architectures and approaches discussed in this report. Through a series of human-based evaluations, the project investigates the merits of different approaches to elicit responses from LLMs and provides insights for future work in this domain.

Acknowledgments

This Final Year Project would not have been possible without the support and contributions of many individuals and organizations. I would like to express my sincere gratitude to the following people:

- First and foremost, I would like to thank my project supervisor, Dr. Christian Von Der Weth, for their invaluable guidance, mentorship, and unwavering support throughout the duration of this project. His expertise, insightful feedback, and constant encouragement have been instrumental in shaping the direction and quality of our work.
- Secondly, I am immensely grateful to the participants who took part in the user studies conducted as part of this project. Your valuable time, feedback, and insights have been crucial in evaluating and paving the path forward for prospective utilization of LLMs in question generation tasks.
- I would like to acknowledge the support of the School of Computing at National University of Singapore, who provided the resources, infrastructure, and administrative assistance necessary for the successful execution of this project.
- Finally, I extend my heartfelt appreciation to my family, friends, and peers who have offered moral support, encouraged me during challenging times, and provided a listening ear throughout this journey.

Contents

1 Introduction **5**

2 Related Works & Background **6**
2.1 Language Models in Natural Language Processing 6
2.2 The Efficacy of LLMs in Generating Multiple Choice Questions . . 6
2.3 Crafting High-Quality Multiple Choice Questions 7
2.4 Scope for Further Research and SELENE 7

3 Setup **9**
3.1 Variables 9
3.1.1 Question Type 9
3.1.2 Subject Matter/Syllabus 9

4 Methods **11**
4.1 Baseline Approach 11
4.2 Prompt Engineering 13
4.3 Using Learning Objectives 18
4.4 Retrieval Augmented Generation 20
4.4.1 On Vector Databases 21
4.4.2 Vector Database Vanilla 21
4.4.3 Vector Database Improved 23
4.5 Fine Tuning 25
4.5.1 On Open Source Models 25
4.5.2 Experiments 26
4.5.3 Discussion 28

5 Evaluations **30**
5.1 Human Evaluations 30
5.1.1 Expert Study 30
5.1.2 User Study 35
5.1.3 Taxonomy Study 40

6 Conclusions **45**

7 AI Ethics **46**
7.1 Transparency and Explainability 46
7.2 Bias and Fairness 47
7.3 Privacy and Data Protection 47
7.4 Accountability and Oversight 47

8 References **48**

9 Appendices **50**
9.1 System Prompts 50
9.1.1 Prompt Engineering Approach 50
9.1.2 Retrieval Augmented Generation Approach 52
9.1.3 Taxonomy Approach 53
9.1.4 Miscellaneous 56
9.2 User Prompts 56
9.2.1 Prompt Engineering Approach 56
9.2.2 RAG Approach 56
9.2.3 Taxonomy Approach 57
9.2.4 Data Cleaning 57
9.3 Template Variables (Used in User Prompts) 57
9.3.1 Topics 57
9.3.2 Bloom's Taxonomy 63
9.4 Miscellaneous 63
9.4.1 SELENE under development 63
9.4.2 Improved Explanation RAG 65
9.4.3 Taxonomy Study Charts 66
9.4.4 Context Cleaning Comparison 69

1 Introduction

Quality assessments are critical for effective learning. However, manually creating assessments tailored to each student's evolving mastery is challenging. It takes time and effort from subject matter experts and educators to create questions. In scenarios like this, it is beneficial to have a system that can automatically generate questions to enable self-driven practice and evaluation. This not only allows students an opportunity to self-explore concepts well, but also acts as a tool for ideation helping educators craft high quality assessments.

Recent large language models (LLMs) show promise for automated question generation as they have excelled across a host of natural language tasks. This motivates the usage of state-of-the-art LLMs to automatically generate well-formed questions tailored to educational needs.

While LLMs show promise for automated generation, they have key challenges:

- LLMs are black boxes, making it difficult to control and evaluate question quality.
- Relevance, difficulty, and even correctness are challenging to automatically measure.
- Scope control, coherence, and non-repetition remain issues.

This project aims to understand the effectiveness and current limitations of LLMs for question generation.

The approaches taken involve developing prompt engineering and retrieval techniques to constrain quality and scope. Evaluating feasibility could inform development of next-generation AI tutoring systems with automated question creation. A key goal is understanding where LLMs fall short for this application and how they can be improved as tools for students and educators alike.

The core vision is a system where students can master topics through repeated practice with LLM-generated questions covering concepts and misconceptions. This would enable personalized, on-demand assessment without extensive human effort. A virtual interactive textbook for learning, if you may.

2 Related Works & Background

2.1 Language Models in Natural Language Processing

Recent advancements in natural language processing (NLP) have been significantly driven by the development and application of large language models (LLMs) such as BERT (Devlin et al., 2019), GPT-3 (Brown et al., 2020), and T5 (Raffel et al., 2023). These models are characterized by their extensive scale, both in terms of the size of the models themselves and the volume of training data they utilize, which endows them with remarkable capabilities in text generation. However, the vast scale of these models also presents challenges, rendering many conventional learning approaches either impractical or infeasible (Schick and Schütze, 2021). Moreover, the most advanced and high-performing models are often proprietary today, accessible only via APIs, which limits direct experimentation and modification. Examples include GPT-4, Claude, Gemini etc.

To navigate these challenges, the field has seen the emergence of prompt-based learning techniques. These techniques are designed to leverage the pre-existing strengths of LLMs by crafting specific input prompts that guide the models to generate desired outputs without the need for direct fine-tuning of the model parameters (Liu et al., 2023; Gao et al., 2024). This approach has proven effective across various tasks, demonstrating the versatility and potential of prompt-based learning in adapting LLMs for diverse applications (Petroni et al., 2019; Brown et al., 2020; Radford et al., 2019). Among the notable methodologies within this domain is the Retrieval Augmented Generation (RAG) technique, which has shown to enhance the performance and effectiveness of LLMs by providing grounding information (Lewis et al., 2021).

2.2 The Efficacy of LLMs in Generating Multiple Choice Questions

The capability of LLMs to generate high-quality and diverse multiple-choice questions (MCQs) for educational assessments has been a subject of recent research interest. Studies have indicated that LLMs, including the latest iterations like GPT-4, are proficient in creating MCQs that are applicable across various educational settings, including programming and computer science education (Elkins et al., 2023; Bhat et al., 2022; Doughty et al., 2023). These generated questions have been evaluated positively through several methodologies, underscoring the potential of LLMs as valuable tools for educators in designing quizzes and assessments. However, LLMs haven't been subjected to generate questions for advanced curriculum where pre-requisite knowledge is vast and requires student self-study. In additon, there haven't been experiments done for implementing context control mechanisms that would help align the model with course material.

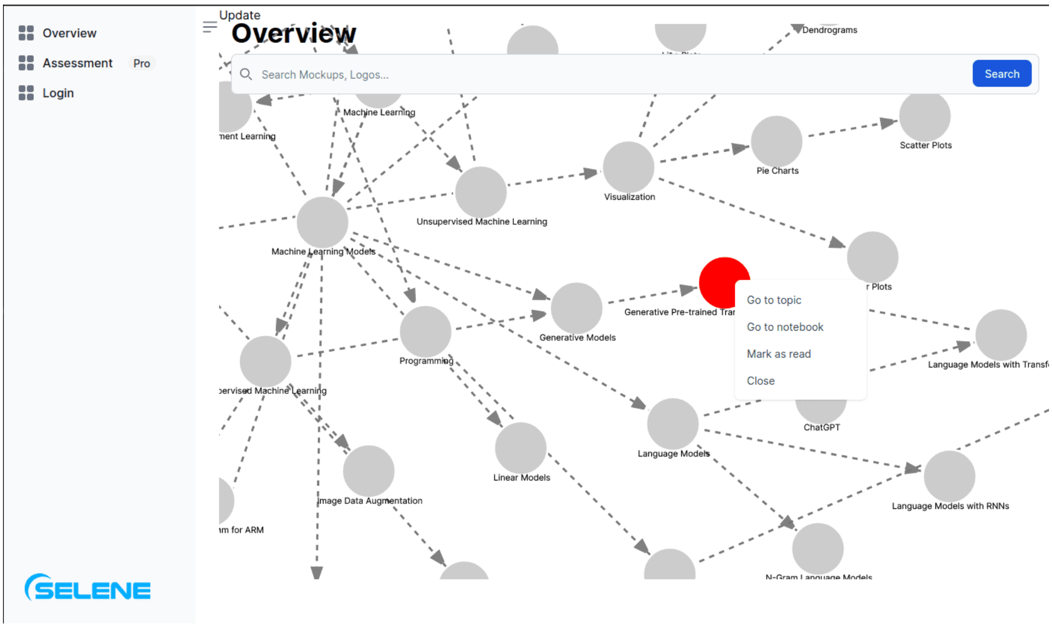
2.3 Crafting High-Quality Multiple Choice Questions

The development of high-quality MCQs is a nuanced and complex process that demands a deep understanding of the subject matter, significant time investment, and expertise. The construction of effective MCQs involves the integration of well-structured content, efficient distractors, meaningful stems, and options that are relevant, plausible, clear, and concise (Kar et al., 2015; Salam et al., 2020). These elements are crucial for assessing higher-order thinking skills and ensuring the educational value of the questions. Furthermore, adherence to best practices in question writing and avoidance of common pitfalls are essential for creating MCQs that accurately test higher-order concepts (Catanzano et al., 2022; Coughlin and Featherstone, 2017). The collective consideration of these factors is vital for the successful crafting of high-quality MCQs, setting a benchmark for the expectations from LLMs in educational assessment contexts.

2.4 Scope for Further Research and SELENE

SELENE is a recent initiative for building an open online platform that allows users to learn AI and related topics in a self-paced, hands-on manner. The prototypical platform currently consists of two main components:

- A comprehensive and well-structured repository of Jupyter notebooks to introduce learners step-by-steps into important topics of ML/AI and beyond.
- A Web frontend that visualizes topics as a graph reflecting the relationship between topics based on prerequisites. Users can search and browse for topics, and identify the mastery paths to complete a topic; see the screenshot below.



Selene Dashboard

SELENE aims to be a large-scale interactive textbook, with the goal to make it available within and outside NUS. However, so far, SELENE does not offer any self-assessments or knowledge checks where learners can monitor the learning progress. This FYP also aims to explore to what extent such self-assessments – here with focus on MCQs including explanations – can be automatically generated using LLMs to minimize the required effort for creating such assessments. While SELENE and provided assessments are not intended to be used for marking students, generated assessments still require a high level of quality.

SELENE is currently under development. (See 9.4.1)

3 Setup

3.1 Variables

To evaluate the effectiveness of large language models (LLMs) for generating questions, it is necessary to establish a baseline by controlling certain variables. There were two variables that were identified:

- **Question Type:** The type of question generated, example, multiple-choice, short answer, essay, etc. Different question formats assess distinct cognitive skills and knowledge domains.
- **Subject Matter/Syllabus:** The specific subject from the curriculum that the generated question is intended to belong to. This ensures the questions are aligned with the instructional content.

3.1.1 Question Type

Multiple Choice Questions (MCQs) were chosen to be the Question Type. MCQs serve as a useful initial format for evaluating LLM question generation capabilities:

- The constrained nature of MCQs, with a limited set of answer options, provides a more defined scope for generation. This focuses the LLM on crafting relevant and plausible alternatives to assess knowledge.
- Objective scoring is straightforward for MCQs based on the correct answer. This enables simpler automated evaluation using accuracy metrics before assessing more complex subjective responses.
- Also, there has been considerable amount of research done in creation of MCQs in educational settings using LLMs.

However, compared to open-ended questions, LLMs actually need to come up with good distractors, which pose a good challenge.

3.1.2 Subject Matter/Syllabus

Natural Language Processing and Text Mining were chosen as the main subjects for which the questions were generated. The justification for the choice is because of several factors:

- These subjects represent advanced undergraduate coursework, where creative and challenging questions are difficult and time consuming to create for educators. In addition, these subjects cover vast number of topics which require self-exploration on part of the student to gain a better understanding and hence, motivate the need for self-assessment questions.
- The supervised nature of these courses at NUS (CS4248 and CS5246), led by Dr. Christian Von Der Weth, provided valuable domain expertise and guidance throughout the project.

4 Methods

The methods employed in this project centered around investigating the usage of different prompting approaches to try to elicit high quality responses from large language models.

4.1 Baseline Approach

The baseline approach consisted of using the web interface to simply prompt the ChatGPT language model with a request to generate questions.

```
Generate 10 MCQ questions about {topic}.
```

`{topic}` can be replaced with some concrete concept like `vector space model` or `naive bayes classifier`.

This approach leveraged the broad capabilities of ChatGPT to generate content without any specific engineering effort. This approach is called Zero Shot Prompting. Large LLMs are tuned to follow instructions and are trained on large amounts of data; so they are capable of performing some tasks "zero-shot", based on what they already know.

Examples of the responses from ChatGPT are as follows:

You

Generate 10 MCQ questions about vector space model.

ChatGPT

1. Which of the following best describes the Vector Space Model (VSM)?

   A) A technique used in linear algebra for solving systems of equations.

   B) A mathematical model used in information retrieval to represent text documents.

   C) A computational method for clustering data points in multidimensional space.

   D) A statistical approach for analyzing categorical variables in datasets.

2. In the Vector Space Model, how are documents and terms represented?

   A) Documents as vectors and terms as scalars.

   B) Documents as scalars and terms as vectors.

   C) Both documents and terms as vectors.

   D) Both documents and terms as scalars.

3. Which of the following is a key concept in the Vector Space Model for representing text documents?

   A) Singular Value Decomposition (SVD)

   B) Term Frequency-Inverse Document Frequency (TF-IDF)

   C) Principal Component Analysis (PCA)

   D) Euclidean Distance

4. What does TF-IDF stand for in the context of the Vector Space Model?

Snippet of ChatGPT's Response for Vector Space Model

You

Generate 10 MCQ questions about naive bayes classifier.

ChatGPT

1. What is the Naive Bayes classifier primarily used for?

   A) Regression analysis

   B) Clustering data points

   C) Text classification

   D) Image recognition

2. What assumption does the Naive Bayes classifier make about the features in a dataset?

   A) They are highly correlated.

   B) They are independent of each other.

   C) They follow a Gaussian distribution.

   D) They are all equally important.

3. In the context of the Naive Bayes classifier, what does the term "naive" refer to?

   A) It refers to the simplicity of the algorithm.

   B) It refers to the assumption of independence among features.

   C) It indicates the use of Bayes' theorem.

   D) It denotes a lack of prior knowledge about the dataset.

4. Which probability distribution is commonly used for continuous features in the Naive Bayes classifier?

Snippet of ChatGPT's Response for Naive Bayes Classifier

As observerd, the resulting questions generated by ChatGPT were fairly generic and focused on factual, recall-based information about the subject. The questions covered broad topics like the definitions, common techniques and algorithms, and high-level applications etc.

In addition, these questions had other limitations, which made them unsuitable for assessments:

- **Limited Context Control:** The prompt provided no ability to control the context, framing, or target learning outcomes for the generated questions. ChatGPT produced generic content without aligning to any specific use case.
- **Absence of Learning Objectives:** The questions did not map to any defined learning objectives or assessment criteria. They were not designed to evaluate a student's mastery of targeted concepts or skills.

Thus, improvements were needed to steer the model's performance.

4.2 Prompt Engineering

To address the limitations of the baseline approach, the project explored the emerging discipline of prompt engineering. Prompt engineering encompasses developing and optimizing prompts to efficiently use language models for a wide variety of applications and research topics.

This approach involved several new key elements:

- **System Prompt:** One of the recommendations to improve LLM performance as suggested by OpenAI is the usage of system prompt. The system message can be used to specify the persona used by the model in its replies. In this approach, a detailed system prompt was crafted to provide the LLM with clear context, instructions, and constraints. This included referencing the course textbook as a source of authority, and specifying the desired output format (a single question with 5 choices, answer, solution, and explanation).
- **Explicit Instructions:** The prompts gave the LLM very explicit and detailed instructions on the type of assessment item to generate, including the desired response format, what the question should try to achieve and level of detail.

- **Targeted Output:** Rather than asking the LLM to generate a set of 10 generic questions, the prompts focused on producing a single, highly tailored assessment item.
- **Comprehensive Response:** The prompt also instructed the LM to not only provide the correct answer, but also to show the step-by-step reasoning and solution supporting that answer. This added pedagogical value and transparency to the generated content. This approach is also called Chain of Thought prompting (Wei et al., 2023). This prompting technique enables complex reasoning capabilities through intermediate reasoning steps. Anohter important reason for a comprehensive response setup was to automate the feedback for the students who would potentially answer these questions incorrectly. (See 5.1.2)
- **Topic Controlled User Prompt:** Instead of using natural language processing as a broad domain for the LM to generate questions, individual topics were provided in the system prompt. These topics were directly taken from the reference textbook mentioned in the system prompt. This acts as a good proxy for the exact content covered as a part of the curriculum.
- **API-Based Access:** To leverage the most advanced LM capabilities available, the project utilized the GPT-4 API rather than the free ChatGPT web interface used in the baseline approach. This provided access to a more powerful and flexible language model. This decision was based on the LMSys Leaderboard (Chiang et al., 2024).

Empirically, it was observed that:

- The quality of the responses improved dramatically: chain of thought prompting worked as the explanations themselves were very clear, distractors were of good quality. An example response is as follows:

Question

Given a trigram language model (words are denoted as w and their position in a sentence as subscripts), we aim to estimate the probability of the word "algorithm" given its context. The context in this scenario is the bigram "a novel". This model employs Kneser-Ney smoothing, which is known for its effectiveness in handling unseen words or N-grams, making it particularly adept at dealing with sparse data situations. Assuming the following data is available:

- The count of the bigram "a novel" is 20.
- The count of the trigram "a novel algorithm" is 3.
- The total number of unique bigrams in the dataset is 5000.
- The continuation count, count_c, of "algorithm" (i.e., the number of unique bigrams that "algorithm" completes) is 150.
- The discount value, d, applied in Kneser-Ney smoothing for adjusting counts is 0.75.

Calculate the probability, $P(w_3 = \text{"algorithm"} | w_1 = \text{"a"}, w_2 = \text{"novel"})$, for the given trigram using the Kneser-Ney smoothing formula. Select the closest option:

1. 0.015
2. 0.025
3. 0.030
4. 0.050
5. 0.075

Example Response from Topic List 2 (See 9.3.1)

- GPT-4 was able to format the responses correctly as requested. Even complex mathematical equations compiled on a Markdown engine without error.

Question

Given a dataset for a binary classification problem, you've decided to use logistic regression with regularization to prevent overfitting. Your logistic regression model uses a sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$ for predictions, where $z = w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n$ and L2 regularization for its cost function. The regularized cost function $J(w)$ for logistic regression is given as:

$$J(w) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(\sigma(z^{(i)})) + (1-y^{(i)})\log(1-\sigma(z^{(i)}))] + \frac{\lambda}{2m}\sum_{j=1}^{n}w_j^2$$

where m is the number of training examples, n is the number of features, $y^{(i)}$ is the actual label for the i-th training example, w are the parameters (weights), λ is the regularization parameter, and $\sigma(z^{(i)})$ is the prediction for the i-th training example.

To train this logistic regression model using gradient descent, you will need to compute the gradient of $J(w)$ with respect to each weight w_j. Which of the following correctly represents the derivative of $J(w)$ with respect to w_j (for $j \geq 1$)?

1. $\frac{\partial J}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m}(\sigma(z^{(i)}) - y^{(i)})x_j^{(i)}$
2. $\frac{\partial J}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m}(\sigma(z^{(i)}) - y^{(i)})x_j^{(i)} + \frac{\lambda}{m}w_j$
3. $\frac{\partial J}{\partial w_j} = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}(1-\sigma(z^{(i)})) - (1-y^{(i)})\sigma(z^{(i)})]x_j^{(i)}$
4. $\frac{\partial J}{\partial w_j} = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(\sigma(z^{(i)})) - (1-y^{(i)})\log(1-\sigma(z^{(i)}))]x_j^{(i)} + \frac{\lambda}{2m}w_j^2$
5. $\frac{\partial J}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m}(\sigma(z^{(i)}) - y^{(i)}) + \frac{\lambda}{2m}w_j^2$

Complex Mathematical Response from Topic List 4 (See 9.3.1)

- However, there was one instance where the model's output was completely gibberish.

I the:. [...]

.:. f. 1. the Abortion Introduction

Our restaurant's and introduce new menu Changes.
Question

A 2. **Solution**

B **Engineers applying Natural Language Processing Programming implementations the with When long 3. Monkeys' document the at developers Large The Heart** 4. **Connection**

5.

C **Meta-Reflections**

D

E ANSWERS:

The deFouad the on Secular ArtS

In the bacterial **Correct Reaction**

F **Rationale**

Gibberish Response

- Including specific topics in the prompt changes the type of question generated. The questions adhered properly to the topics requested for.
- However prompts cannot contain information that is too specific. For example, prompting the model to generate questions about "Smoothing Techniques" will be a broader specification as compared to prompting "Add One Laplace Smoothing".
- Using arbitrary scales to adjust questions may not be effective. For example, including a difficulty level in the prompt like High/Low/Medium did not seem to consistently create questions of matching difficulty. This has been discussed in the Expert Study too. However, more insight may be needed to judge whether or not this is truly the case. This was the extra instruction that was used:

```
Be of a {category} difficulty level, yet being challenging
enough students to apply their knowledge in novel and
complex scenarios, rather than relying on rote memorization
or simple recall.
```

`{category}` would be replaced with `high`, `medium`, or `low`.

Although prompt engineering seemed to work better empirically, mentioning just the textbook name along with the authors relies on the internal knowledge of the LLM about the textbook, which might be outdated, false and cannot be relied upon. This begs the need to provide the LLM with valid information that it can work with to produce a more grounded response.

4.3 Using Learning Objectives

Consideration of learning outcomes while creating questions is something that every lecturer cares about. This section attempts to adjust the style of question and its desired learning outcome by specifying a desired cognitive processing level from the Revised Bloom's Taxonomy. This taxonomy categorises cognitive processes into different levels: Remember, Understand, Apply, Analyze, Evaluate, Create. This defines a hierarchy of cognitive skills, ranging from lower-order skills like remembering and understanding, to higher-order skills like analyzing, evaluating, and creating. This gives educators a framework to educate and evaluate students at different cognitive processing levels.

CATEGORIES	COGNITIVE PROCESS
Remember	**Retrieve relevant knowledge from long-term memory** RECOGNIZING (identifying) RECALLING (retrieving)
Understand	**Construct meaning from instructional messages, including oral, written, and graphic communication** INTERPRETING (clarifying, paraphrasing, representing, translating) EXEMPLIFYING (illustrating, instantiating) CLASSIFYING (categorizing, subsuming) SUMMARIZING (abstracting, generalizing) INFERRING (concluding, extrapolating, interpolating, predicting) COMPARING (contrasting, mapping, matching) EXPLAINING (constructing models)
Apply	**Carry out or use a procedure in a given situation** EXECUTING (carrying out) IMPLEMENTING (using)
Analyze	**Break material into its constituent parts and determine how the parts relate to one another and to an overall structure or purpose** DIFFERENTIATING (discriminating, distinguishing, focusing, selecting) ORGANIZING (finding coherence, intergrating, outlining, parsing, structuring) ATTRIBUTING (deconstructing)
Evaluate	**Make judgments based on criteria and standards** CHECKING (coordinating, detecting, monitoring, testing CRITIQUING (judging)
Create	**Put elements together to form a coherent or functional whole; reorganize elements into a new pattern or structure** GENERATING (hypothesizing) PLANNING (designing) PRODUCING (constructing)

Revised Bloom's Taxonomy (LW et al., 2001; Bloom et al., 1956)

Leveraging this framework, the project sought to tailor the style and cognitive level of the generated assessment items to better align with specific learning objectives. By specifying a target cognitive processing level from Bloom's Taxonomy in the prompts, the language model was instructed to generate questions that required that level of cognitive engagement from the student.

The changes to the prompts included:

- Mentioning the different Bloom's Taxonomy levels in the system prompt as a part of an instruction to adhere to them. (See 9.2.3)

- Explicitly stating the desired Bloom's Taxonomy level in the user prompt. (See 9.2.3)

In order to test its efficacy, a human evaluation study was conducted with Dr. Christian himself. (See 5.1.3)

4.4 Retrieval Augmented Generation

LLMs are usually pre-trained on a massive corpus of text, in the orders of trillions of tokens. Though the emergent properties of these models are impressive, it needs to be noted that only a snapshot of the knowledge is being captured by these models and they don't have any means to act upon a very specific domain. This way, the model requires access to external knowledge sources to complete tasks. Not only this enables more factual consistency, but it also improves the reliability of generated responses, and helps to mitigate the problem of "hallucination".

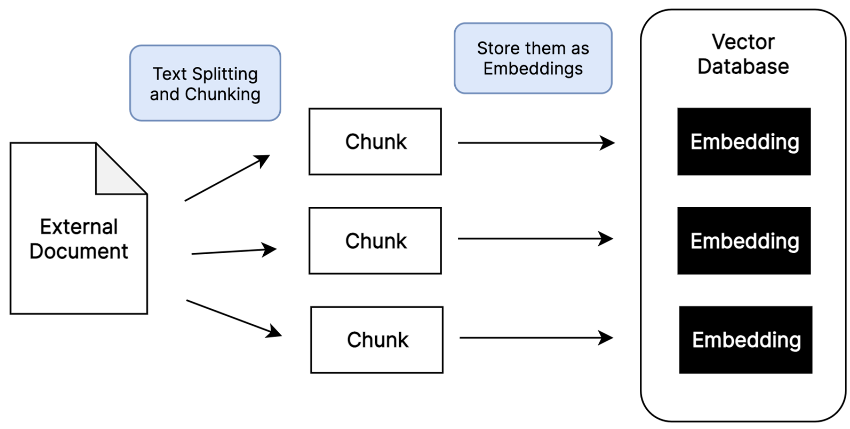
Researchers from Meta introduced a method called Retrieval Augmented Generation (RAG) to address such knowledge-intensive tasks. RAG is a powerful approach that combines pre-trained parametric and non-parametric memory for language generation (Lewis et al., 2021). It is born out of the idea that LLMs are capable of in-context learning (Brown et al., 2020). It combines an information retrieval component with a text generator model. RAG can be fine-tuned and its internal knowledge can be modified in an efficient manner and without needing retraining of the entire model.

RAG approaches are used to provide LLMs with proper grounding data that can steer the language model towards adhering to the course content more strictly. This approach seems promising because textbook passages can be accomodated into the prompt, thanks to increasing context window sizes of best performing LLMs, and hypothesize about improved performance.

This is an interesting application because traditionally, RAG methods have been used for fact checking, document based question answering etc. Using the context generated by the topic to generate questions out of it might help with better context control.

4.4.1 On Vector Databases

A Vector Database is a specialized database system designed for efficiently indexing, querying, and retrieving high-dimensional vector data. Those systems enable advanced data analysis and similarity-search operations that extend well beyond the traditional, structured query approach of conventional databases.

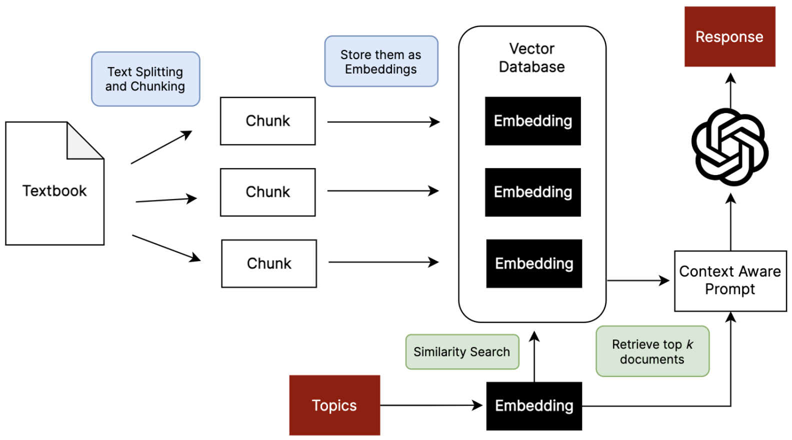


Basic Vector DB Flow

4.4.2 Vector Database Vanilla

The vanilla approach involves directly parsing the textbook PDF, Speech and Language Processing 3ed, using LangChain's pre-built parser and inserting the content into a vector database. This vector database is then semantically searched for given a query in order to retrieve context that can be passed as a part of the prompt to generate questions.

The architecture looks something like this:



Vector DB Vanilla Approach

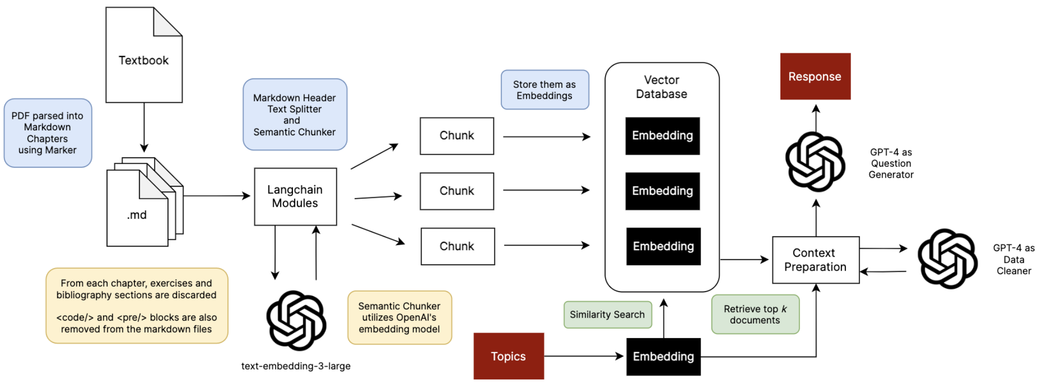
Although, this approach seems straightforward to rely on, there are several shortcomings:

- **Erroneous PDF Parsing:** The approach struggles with accurately parsing the textbook content, particularly when it comes to handling mathematical equations, tables, and other complex formatting. This leads to incomplete or inaccurate representations of the textbook material in the vector database.
- **Poor Chunking Strategy:** The approach simply dumps the entire textbook content into the vector database without any meaningful chunking or segmentation. This results in retrieved contexts that are cut off at arbitrary points, making them difficult to understand and use in a question-answering system.
- **Lack of Grammatical Coherence:** The lack of a sophisticated chunking strategy means that the retrieved contexts may not always make grammatical sense or provide a coherent response to the user's query. This results in incorrect information being passed to the LLM which increases the possibility of using incorrect knowledge to generate questions.

To address these issues, there was a need for a better architecture.

4.4.3 Vector Database Improved

This improved approach has several modifications as compared to the vanilla approach. The architecture is as follows:



Vector DB Improved

Parsing Strategy

PDFs are notoriously hard to parse. Especially, given the textbook in consideration, Speech and Language Processing 3ed, had mathematical equations, emphasis text, and table structures because of which LangChain's default PDF parser was not able to semantically keep the content intact.

There was a need for a parser that supported OCR (Optical Character Recognition). Marker (See link), an open source project, was chosen to convert the PDFs to Markdown format. Markdown format allows for a simple syntax which supports all of the different structures mentioned above like mathematical equations, emphasis text, and tables. It was also a plus since all commerical LLMs understood the format.

Chunking Strategies

Once the PDFs were converted to Markdown, the documents needed to be chunked in order to be stored in the vector database. The chunk size itself is a subjective choice, because if it's too big, then there is a risk of running out of the context window size, and if it's too small, logical information might not be captured.

LangChain has a `MarkdownHeaderTextSplitter` module that splits the markdown file based on the heading levels and adds that as metadata to the chunk. This was a good proxy to reduce the size of individual chapters from the book. Still, these sections were too big to be incorporated in a single chunk.

An insight was that markdown paragraphs separated by `\n\n` and the size of a single paragraph from the textbook was a good balance between size and logical self-standing. It was also seen that multiple paragraphs may belong to the same header level. So, the markdown document was now split by paragraphs and post-edited to include the header metadata to better structure the storage into the vector database.

These paragraphs were now semantically broken down using a `SemanticChunker` which uses OpenAI's embeddings to establish the similarity between different chunks.

Context Preparation

Once the self-contained document chunks are stored in the vector database, the job of the retriever is to query relevant chunks from it. Upon inspection it was observed that the retrieved chunks cannot be used directly. This was due to some extra words that were unresolved Therefore, a data cleaning prompt was used to send the context to GPT-4 to clean it while keeping the truth intact. There was evidence that GPT-4 was good at this task (Bolding et al., 2023). This finally cleaned context was passed on for question generation. An example of raw versus cleaned context is given in the appendix. (See 9.4.4)

Modified Instructions to the LLM

Some initial observations include:

- Type of the content retrieved from the textbook matters a lot. For example, generating questions about Unix tools for text processing repeatedly gave complex command line questions.
- Semantic search tried to incorporate chunks that were from different chapters

from the textbook, thereby requiring a synthesis of different chapters.

- Explanations clearly referenced the material from the textbook. The highlighted regions below show the referenced content. An example of the explanation is in the appendix. (See 9.4.2)
- Upon manual inspection, the questions appeared to adhere better with the textbook content, so the vagueness and generality from zero-shot prompting was somewhat subsided.

In order to comprehensively evaluate this approach, an expert human evaluation study was conducted with PhD students from NUS. (See 5.1.1)

4.5 Fine Tuning

This section delves into the process of fine-tuning language models, with a particular focus on the utilization of open-source models over proprietary ones such as GPT-4, and discusses the implications of this choice through experimental analysis.

4.5.1 On Open Source Models

While GPT4 is a clear go-to model due to its established performance against others (Chiang et al., 2024), it's closed source nature does not allow for flexibility and improvements on a specific language task.

This motivates the need to use an open source model which can be much more accessible. Such accessibility not only facilitates a deeper understanding of the model's inner workings but also enables tailored adjustments to better meet the requirements of specialized language tasks.

The open-source model chosen for this project is Gemma 7B, developed by Google (Team et al., 2024). Gemma 7B is a large language model trained on a diverse corpus of data and has shown strong performance on various natural language processing tasks.

4.5.2 Experiments

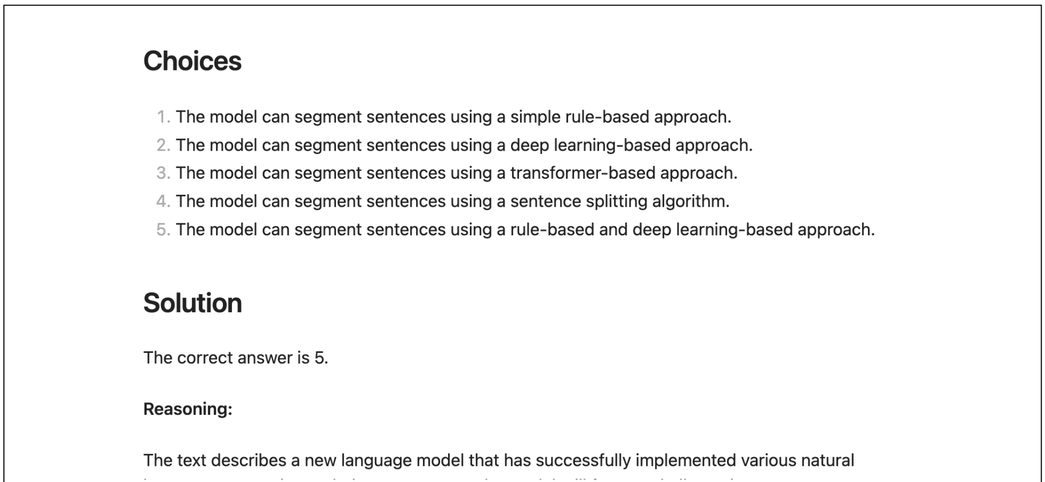
As a first step in using an open source model for the task of question generation is to understand its performance on the same prompts that GPT-4 was subjected.

System prompting is a little tricky with Gemma, as the prompt template doesn't account for it separately. But, based on examples provided on HuggingFace, system prompt is just prepended to the user prompt and passed onto the model. The prompt template for Gemma models is:

```
<start_of_turn>user
{system_prompt}

{user_prompt}<end_of_turn>
<start_of_turn>model
```

There were a total 40 data samples generated across 8 different topic lists of Natural Language Processing. These were some initial observations:

- The formatting instructions provided in the system prompt were ignored by the model. In some cases, the model used bold text instead of 2nd level heading. In addition, there were extra headers that were unnecessary.



Choices

1. The model can segment sentences using a simple rule-based approach.
2. The model can segment sentences using a deep learning-based approach.
3. The model can segment sentences using a transformer-based approach.
4. The model can segment sentences using a sentence splitting algorithm.
5. The model can segment sentences using a rule-based and deep learning-based approach.

Solution

The correct answer is 5.

Reasoning:

The text describes a new language model that has successfully implemented various natural

Incorrect Headers in Gemma's Response

Solution

The correct answer is 5.

Reasoning:

The text describes a new language model that has successfully implemented various natural language processing techniques. However, the model still faces a challenge in sentence segmentation. Sentence segmentation involves breaking a text into smaller units of text, typically sentences. To overcome this challenge, the model utilizes a combination of rule-based and deep learning-based approaches. This approach is the most effective method for sentence segmentation and is therefore the correct answer to the question.

Additional Notes

This question incorporates concepts related to text normalization, word tokenization, word normalization, lemmatization, stemming, and sentence segmentation. It also requires the student to synthesize knowledge from various sources, including the textbook, lectures, and supplementary materials covered in the course. The question is designed to be challenging and requires a deep understanding of the concepts and their practical implications.

Extra Header in Gemma's Response

- There were several instances of questions not being clear. It was not completely evident as to what the question was asking.

Problem Statement

A large language model is tasked with the job of analyzing a document containing a conversation between two people. The goal is to identify the speaker of each sentence in the document. However, the text provided contains some irregularities and noise, such as misspellings, punctuation errors, and irrelevant text.

Problem Statement

A new language model has been developed that can understand and interpret human language more effectively than traditional models. This model incorporates advancements in text normalization, word tokenization, word normalization, lemmatization, and stemming techniques. However, there is one remaining challenge that the model has yet to overcome: sentence segmentation.

What is the question asking?

- The model also struggled to consistently segregate the choices of an MCQ from the question description itself, thereby causing repition of the same. In some cases, the options completely changed, causing confusion.

A random sample of these questions was used in an expert human evaluation with PhD students. (See 5.1.1)

Based on these observations, it was understood that Gemma *as is* was not a good model for the task of question generation. On the other hand, it was empirically evident that GPT-4 was better at the task.

So, one approach to expect better responses from Gemma would be to try fine tuning the model.

A dataset was needed for this experiment. Upon investigation, it was found that there were no publicly available datasets for such a specialized question generation task. Taking inspiration from Minigpt-4 (Zhu et al., 2023), and Alpaca from Stanford (Taori et al., 2023), it was decided that synthetic data generated from GPT-4 is a good proxy for human annotated datasets, especially when the response quality of a closed source model is of higher quality. GPT-4 was hypothesized to be using a much superior language model based on empirical observations, and aligning the open source model to steer its behavior could mitigate the issues pointed out earlier with Gemma 7B.

Using GPT-4, a dataset of 1000 LLM generated questions was created. These samples were cleaned to ensure there are no syntax errors. The dataset was split 80/20 for training and validation.

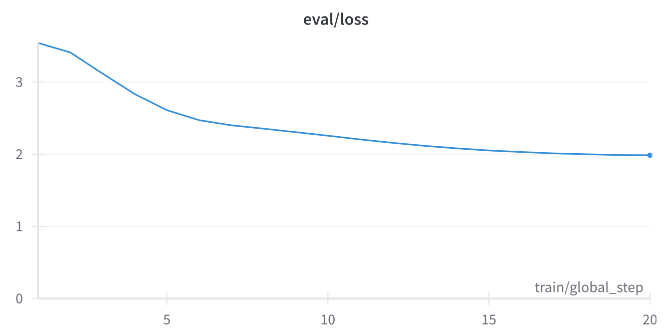
The fine tuning methodology used was Quantized-LoRa, which belongs to the family of fine tuning techniques called PEFT (Parameter Efficient Fine Tuning).

The model chosen for fine tuning was the Gemma 7B Instruct model, with 4-bit quantization, the same one which was used for zero-shot generation.

4.5.3 Discussion

These are the graphs for training loss and validation loss.

Training Loss



Validation Loss

The losses seemed to fall fairly across 1 epoch distributed over 20 steps. But it seemed to flat-line near 2, which was still very high.

Empirically observed, post fine tuning generation showed improvements only in correct formatting of the generated questions. However, more work needs to be done in this area to make concrete claims. This experiment is a simple proof-of-concept that fine tuning might work in order to create a specialized model for question generation.

There were practical limitations, like cost, resources required to generate large scale datasets, to expand on this approach.

5 Evaluations

5.1 Human Evaluations

Across the methods discussed in the previous section, it was discovered that there are no quantifiable metrics that can be used to judge whether LLMs are performant for the task of generating questions, especially for advanced undergraduate curriculum. This motivates the need to conduct human evaluation studies involving different stakeholders in a classroom setting.

Three separate studies were conducted to explore different aspects of question generation. The rationale behind conducting three distinct studies was to elicit feedback from different sets of participants regarding their use cases.

5.1.1 Expert Study

Objective

The objective of this investigation was to understand the quality of questions generated across different approaches. The three different approaches chosen were: Zero-Shot prompting using GPT4, Zero-Shot prompting using Gemma 7B, and Advanced Retrieval Augmented Generation.

Participants

The participants for this study were PhD students from NUS who had a strong background in Natural Language Processing. Some of them were teaching assistants for NLP courses too.

Setup

Using the three approaches to generate questions, 21 multiple choice questions were generated. (7 from each approach) The questions varied widely across NLP topics and were chosen randomly from a pool of existing generated questions.

There were three different sets of feedback questions that each participant was asked.

- **The Question Scorecard**

Description	**Guide**
How answerable is the question? (Answerability) Inspired from Nema and Khapra, 2018	1 - All important information is missing and it is impossible to answer the question 2 - Most of the important information is missing and I can't infer the answer to the question 3 - Some important information is missing leading to multiple anwers 4 - Most of the important information is present and I can infer the answer 5 - All important information is present and I can answer the question
What's your choice?	One of the options from 1 - 5
How would you rate the difficulty of the question?	1 - Very easy 2 - Easy 3 - Medium 4 - Difficult 5 - Very difficult

- **Answer + Explanation Scorecard**

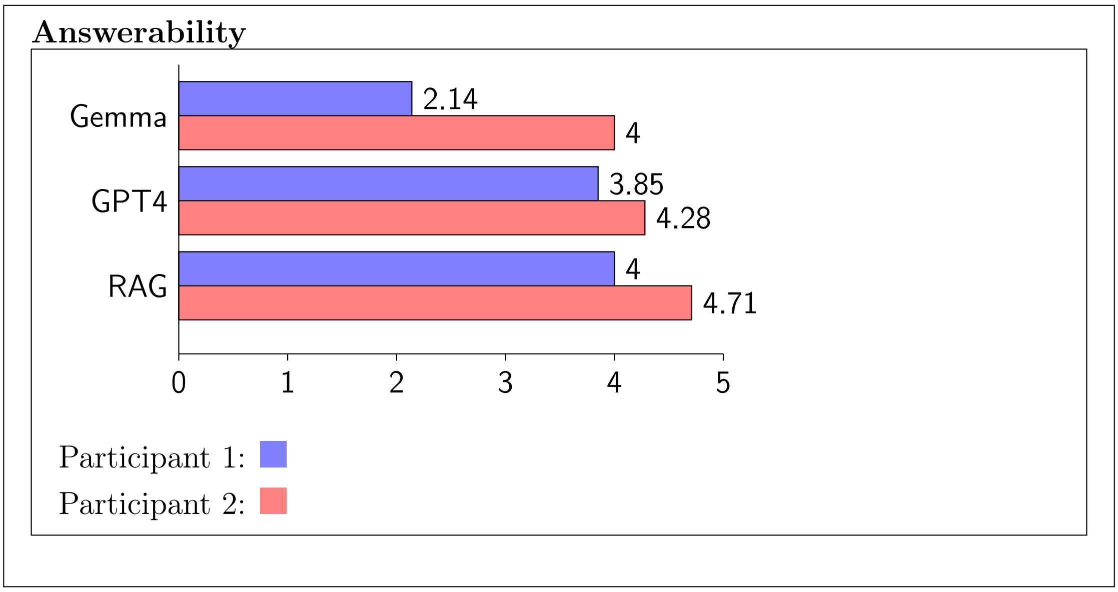
Description	**Guide**
Do you think the given correct answer is correct?	Yes/No
Do you think the given explanation is correct?	Yes/No

- **Overall Response Scorecard**

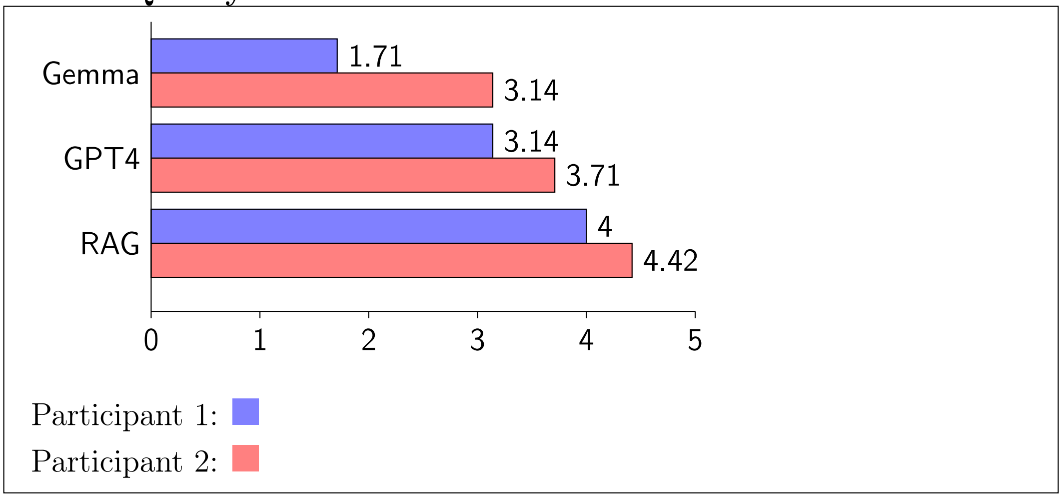
Description	Guide
How would you rate the overall quality of the question, given answer, and options combined?	1 - Poor quality overall 2 - Below average quality 3 - Average quality 4 - High quality 5 - Excellent quality
Where do you think this question belongs to?	Nowhere, the question still needs human effort to correct Non-graded assessments, tutorial discussions Graded assessments, exam papers
Additional Feedback	Textual Response

Results

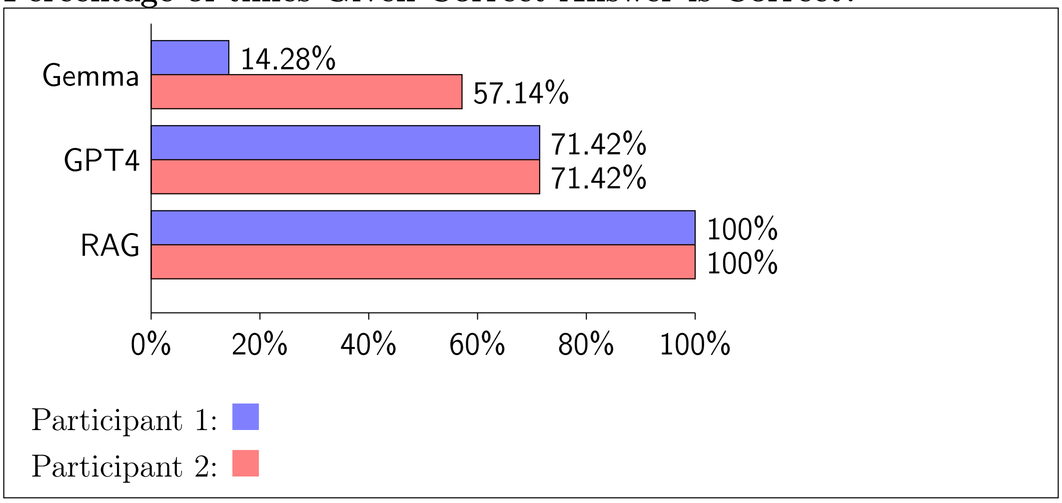
There were a total of 2 PhD students who took part in this study. These are the charts for average answerability, average overall quality, percentage of times given answer is correct, percentage of times given explanation is correct with respect to each of the approach employed.



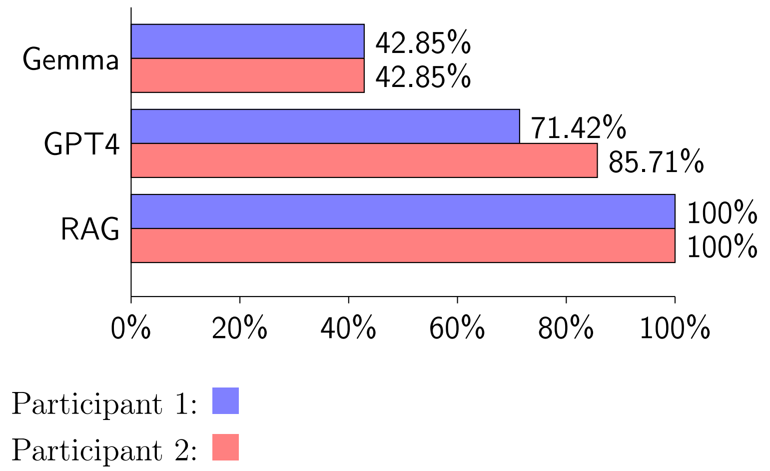
Overall Quality



Percentage of times Given Correct Answer is Correct?



Percentage of times Given Explanation is Correct?



Some observations include:

- Overall, Gemma showed poor performance. This motivated the need to consider a methodology to try fine tuning. A proof of concept was done. (See 4.5)
- GPT4 generated good content but only at the surface level. The participants pointed out that when they read the content, it seemed repititive.
- RAG certainly helped in grounding the truth, being the best performing method overall. The context retrieved from the textbook allowed for better explanations by the LLM. Some of the most favored questions in the entire study were generated using the RAG pipeline.
- RAG method were the highest rated, (100% by Participant 1, and 71.4% by Participant 2) to be used in graded assessment settings.

The actual evaluation sheets are at these links:

- Participant 1: Link
- Participant 2: Link

In conclusion, RAG based approaches are a good place to start.

5.1.2 User Study

Objective

The objective of this investigation was to understand how receptive are actual students regarding LLM generated quizzes and questions. The project aimed at understanding the needs of students when it comes to solving questions, because at the end of the day, they are the end-users. This study, in a way, acts as a user acceptance study.

Pilot Study A pilot study was conducted with 2 friends who had taken the CS4248 (Natural Language Processing) course at NUS. After the pilot study, several new changes were incorporated into the actual study to streamline the entire flow.

Participants

The participants for the actual study were students from CS5246 (Text Mining) course.

Setup

Two canvas quizzes were created for the study. The first contained 21 questions on a variety of Text Mining topics (See 9.3.1). The approach used to generate these questions was prompt engineering with GPT-4. All these topics were covered in the first half of the semester, so considerable familiarity was expected of the participants. This way, they are in a position to spot mistakes, if any. In case they did spot a mistake, there was an extra option in every question: "The question invalid, unclear, flawed, or not answerable", which they could use to flag it.

The second quiz was more of a feedback survey. It contained several Likert scale response questions about the quiz. For each of the given statements, participants were asked to indicate their level of agreement.

	Strongly disagree	Disagree	Neutral	Agree	Strongly Agree
Example Likert Scale Statement	○	○	○	○	○

- For the LLM generated questions

Statement	Tag
The **LLM-generated questions** were clear and easy to understand.	`clarity`
The difficulty level of the **LLM-generated questions** was appropriate for the course level.	`difficulty`
The **LLM-generated questions** effectively assessed my understanding of the course material.	`coverage`
The **LLM-generated questions** challenged me to think critically and apply concepts.	`challenge`
The assessment format was conducive to demonstrating my knowledge.	`usefulness`
I found the **LLM-generated questions** to be engaging and interesting.	`engagement`
I felt motivated to do well on the assessment with **LLM-generated questions**.	`motivation`
Overall, I had a positive experience with the **LLM-generated questions**.	`sentiment`

- For the LLM generated answers + explanations

Statement	Tag
The **LLM-generated answers and explanations** were clear and easy to understand.	`clarity_a`
The difficulty level of the **LLM-generated answers and explanations** was appropriate for the course level.	`difficulty_a`
The **LLM-generated answers and explanations** effectively assessed my understanding of the course material.	`coverage_a`
The **LLM-generated answers and explanations** challenged me to think critically and apply concepts.	`challenge_a`
I found the **LLM-generated answers and explanations** to be engaging and interesting.	`engagement_a`
Overall, I had a positive experience with the **LLM-generated answers and explanations**.	`sentiment_a`

The LLM generated answers and explanations were only shown to participants if they had made mistakes in answering the questions. In order to ensure that participants went through the explanations, the feedback quiz had a key which they could only obtain after they went through the analysis of their attempt.

Each particpant was given a single attempt but unlimited time for both the quizzes. Again, it was not important for the participants to get correct answers to the questions.

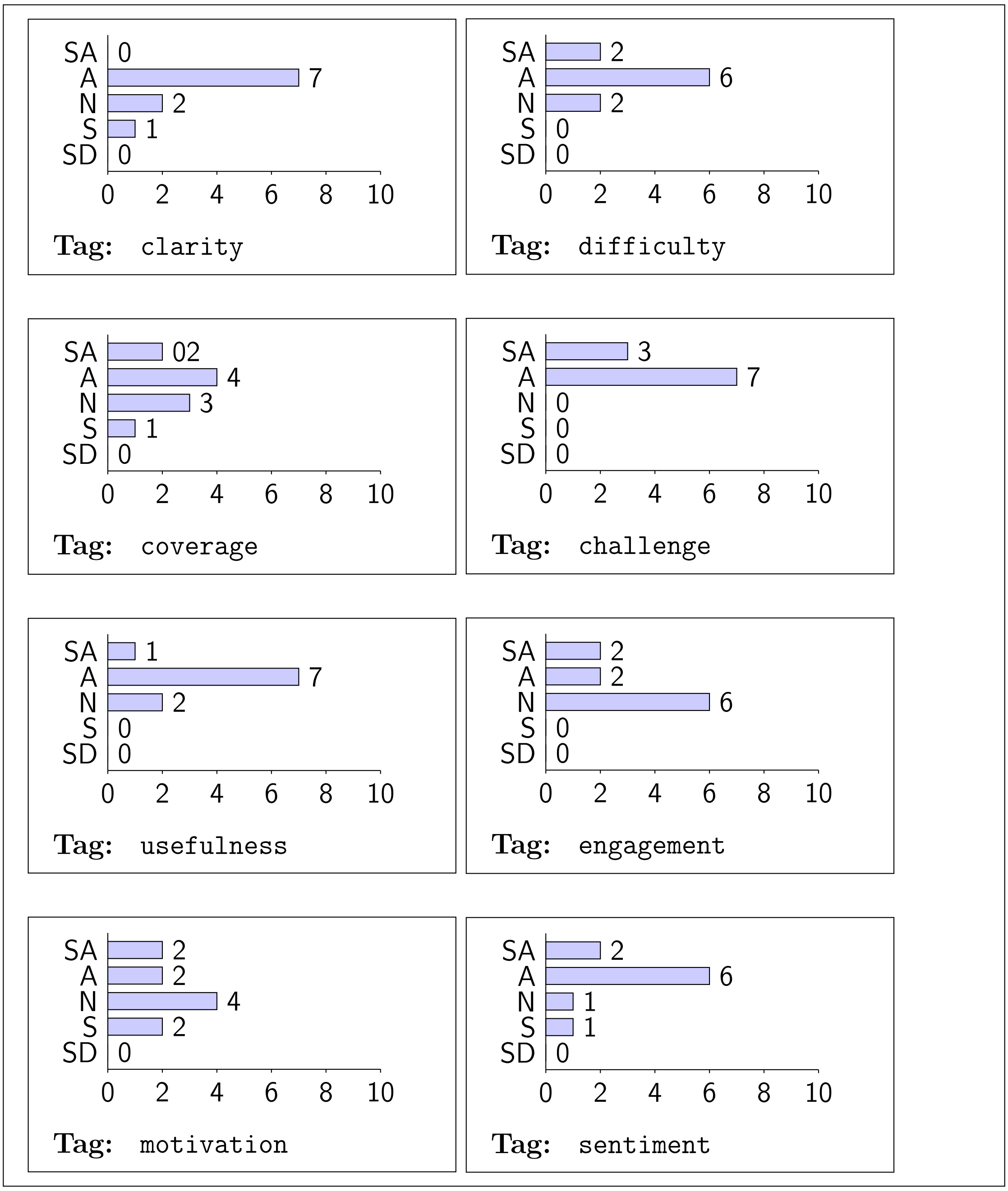
Results

There were a total of 10 participants who signed up for the user study. The distribution of responses for the likert scale questions is as follows:

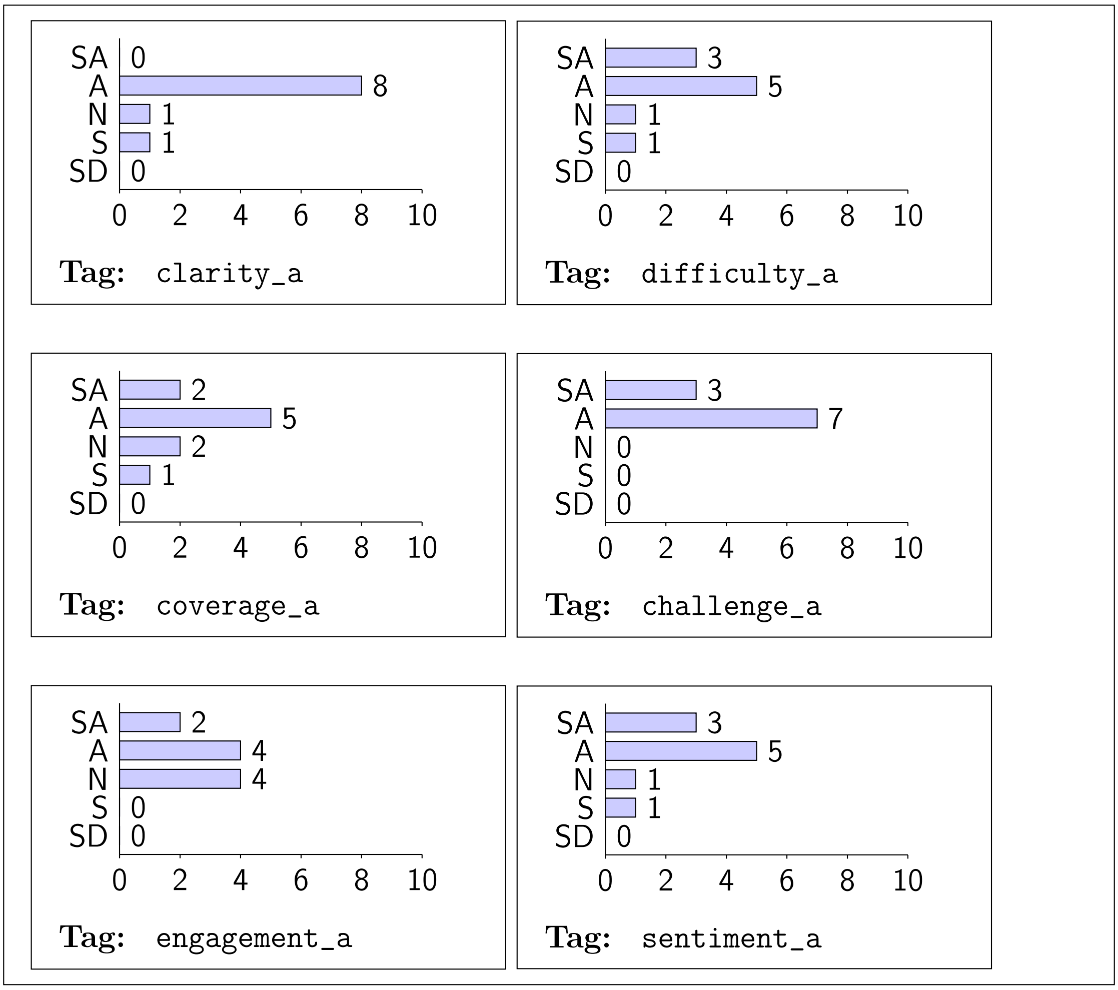
LEGEND

SA	Strongly Agree
A	Agree
N	Neutral
D	Disagree
SD	Strongly Disagree

For the Questions



For the Answers + Explanations



Some observations include:

- Overall, the perception of LLM generated questions was pretty positive amongst the students.
- More than 50% of the students marked "Agree" or "Strongly Agree" on most scales.
- Amongst the textual feedback received, it was noted that some question descriptions were still vague. Example: Q1, Q5, Q12, Q17, and Q20.
- The suggestions given by students were: inclusion of more examples to improve clarity. In fact, a student mentioned that they had to read the questions

multiple times to understand the intent.

- One important feedback was that all the questions felt pretty similar to each other, in terms of phrasing, so there is still some work that needs to be done in order to generate more creative outputs.

In conclusion, the positive indicator levels are a strong foundation to build upon, and the constructive feedback from the students provides valuable insights for future work. By addressing the identified areas of improvement, such as enhancing clarity, providing more examples, and diversifying the phrasing and creativity of the questions, the LLM-generated questions can be further refined and developed to better meet the needs and expectations of the students.

5.1.3 Taxonomy Study

Objective

A question generation system is an incredibly useful tool for educators. It can aid in creatively coming up with new ideas to craft questions which can help students in multi-faceted ways. This investigation aims to understand the effect on mentioning Revised Bloom's Taxonomy's learning objectives as a part of the prompt to GPT-4.

Participants

The subject of this study was Dr. Christian himself, the supervisor of this project. He teaches both NLP and Text Mining courses at NUS.

Setup

18 questions generated for each of the courses, NLP and Text Mining, on a varitey of topics and learning objectives from the taxonomy.

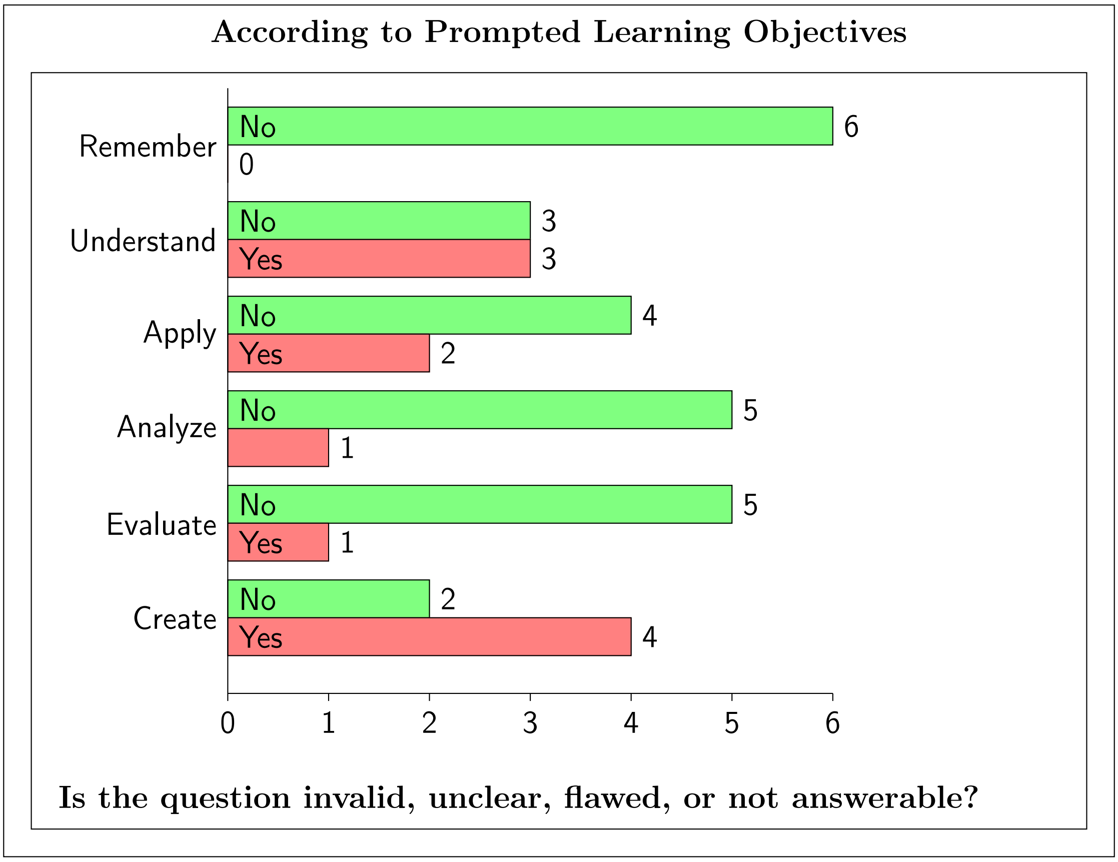
For each question, Dr. Christian was asked to fill in the following parameters:

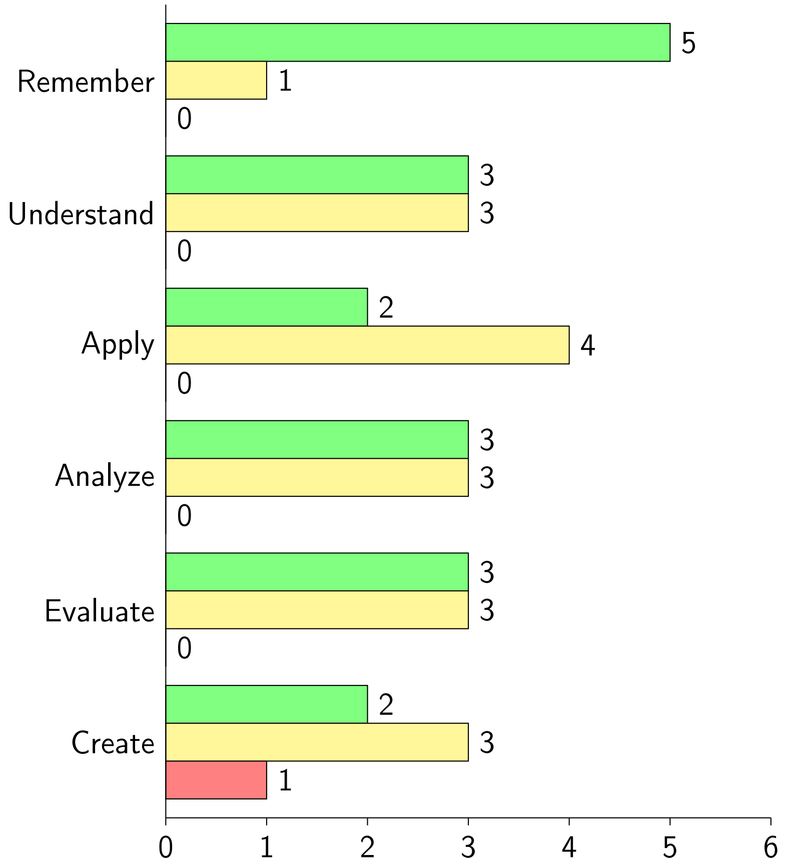
Description	Guide
Is the question invalid, unclear, flawed, or not answerable?	Yes/No
What is the most likely learning objective that is met by this question?	Create Evaluate Analyze Apply Understand Remember
What is another likely learning objective that is met by this question?	Create Evaluate Analyze Apply Understand Remember
What is the most likely difficulty level of the question?	High Medium Low
Where do you think this question belongs to?	Nowhere, the question still needs human effort to correct Non-graded assessments, tutorial discussions Graded assessments, exam papers
Additional Feedback	Textual Response

Overall, this would comprehensively evaluate whether LLMs can generate questions that are learning objective controlled, and would an educator be comfortable using the generated questions in assessments.

Results

Charts constructed with respect to the subjects, NLP and Text Mining are in the appendix. (See 9.4.3)


According to Prompted Learning Objectives
Remember
Understand
Apply
Analyze
Evaluate
Create
No
Yes
6
0
3
3
4
2
5
1
5
1
2
4
0
1
2
3
4
5
6
Is the question invalid, unclear, flawed, or not answerable?

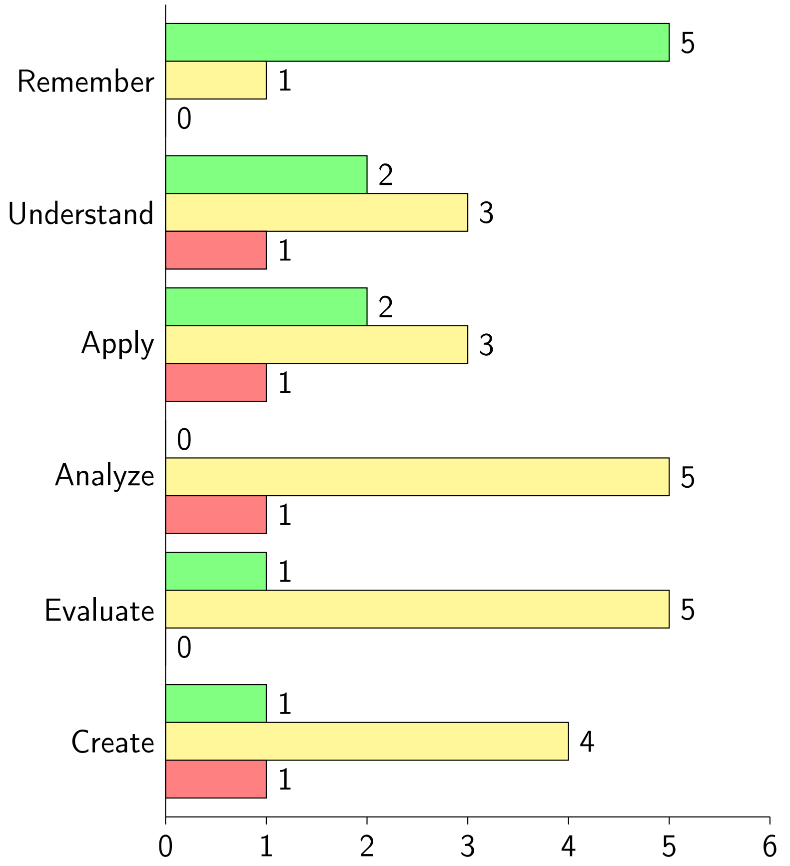


Low:

Medium:

High:

What is the most likely difficulty level of the question?



Graded assessments, exam papers:

Non-graded assessments, tutorial discussions:

Nowhere, the question still needs human effort to correct:

Where do you think this question belongs to?

Some observations include:

- Indicating the learning objective in the prompts is not effective. The questions generated for each of the subject were equitably distributed across 6 learning objectives (3 questions generated for each objective), but the identified objectives met by the question clearly show bias for two of them. (Understand and Analyze)

- There were a total of 12 questions where the learning objective given to the LLM aligned with the objective identified by Dr. Christian.

- 6 out of 6 questions with "Understand" learning objective were aligned. But there is a bias because 23 out of the 36 questions in total were identified to be meeting the objective.
- 5 out of 6 questions with "Remember" learning objective were correctly alingned. The reason hypothesized is that most of the questions rely on memorization of facts when "Remember" is the objective.
- Amongst the questions that were indicated as "flawed, not clear, not answerable", there were 6 questions for NLP and 5 questions for Text Mining. The actual learning objectives used for these questions included 5 out of the 6 objectives. Interestingly, there was no "Remember" question that was marked flawed.
- Dr Christian felt that most flawed question could probably easily be fixed/improved with some manual effort.
- Perhaps the most motivating metric was the usage of question in graded versus ungraded contexts. Dr. Christian mentioned that he wouldn't use any question "as is" for a graded examination apart from very few that are sure shot correct and clear. However, many to most question would be good for tutorials or practice quizzes to engange discussion and critical thinking. Moreover, most question were certainly good for ideation which can already help a lot when making quizzes!

6 Conclusions

The exploration of large language models (LLMs) for the task of question generation has yielded valuable insights, but has also highlighted the limitations of these models in the context of educational applications.

Acorss investigations it was found that the questions that are LLM generated are generally not good enough to be used in exams/graded assessments "as is". However, they have proven very useful for ideation to help with creating exams/quizzes for educators. In additon, they are arguably good for practice quizzes, tutorial questions, and also for SELENE.

While LLMs, such as the GPT-4 utilized in this project, have demonstrated impressive capabilities in generating relevant and coherent questions, they are still far from perfect. The need for a human oversight layer remains crucial to ensure the accuracy and appropriateness of the generated questions, as educators do not want students to learn from potentially flawed or biased information. Therefore, it should always be made explicitly that these questions are AI generated.

However, the promise of LLMs in this domain cannot be overlooked. These models offer a fresh perspective on the problem of question generation, leveraging their vast knowledge and language understanding capabilities to create questions that can stimulate critical thinking and deepen student engagement with the course material.

As the field of natural language processing continues to advance, the potential for LLMs to play a transformative role in education becomes increasingly evident. With further refinements, customization, and the incorporation of robust quality assurance mechanisms, these models can be harnessed to enhance the learning experience, complement human instructors, and ultimately contribute to the betterment of the educational landscape.

In conclusion, while caution is warranted, the findings suggest that these models hold promise and warrant ongoing exploration and development to unlock their full potential in supporting and improving educational outcomes.

7 AI Ethics

As the use of large language models and other advanced AI systems becomes increasingly prevalent, it is important to carefully consider the ethical implications of these technologies. This project has taken a thoughtful approach to integrating ethical principles throughout the development and deployment of the AI-powered assessment generation system.

7.1 Transparency and Explainability

A key ethical consideration is ensuring transparency and explainability in the AI system's outputs. The prompts developed for this project explicitly instructed the language model to not only provide the correct answer, but also to show the step-by-step reasoning and solution supporting that answer. This added pedagogical value and allowed for greater transparency into the model's decision-making process.

7.2 Bias and Fairness

Another critical area of focus was mitigating potential biases present in the language model's training data and generation processes. The project team carefully reviewed the assessment items produced by the system to check for any biases related to gender, race, socioeconomic status, or other demographic factors. Adjustments were made to the prompts as needed to encourage more inclusive and equitable content generation.

7.3 Privacy and Data Protection

When utilizing language models trained on large internet datasets, it is essential to consider privacy implications and ensure appropriate data handling protocols are in place. This project did not involve the collection or storage of any personally identifiable information. All interactions with the language model occurred through the secure GPT-4 API, with no data persisted beyond the scope of the evaluation studies.

7.4 Accountability and Oversight

As an AI-powered system intended for educational applications, this project has maintained a strong emphasis on accountability and oversight. The prompt engineering approach and resulting assessment items have been thoroughly reviewed by the student who's involved in the project.

Continuous monitoring and refinement of the system's ethical safeguards will be an ongoing priority as the technology is further developed and deployed.

8 References

Bhat, S., Nguyen, H. A., Moore, S., Stamper, J. C., Sakr, M., & Nyberg, E. (2022). Towards automated generation and evaluation of questions in educational domains. *Educational Data Mining*. https://api.semanticscholar.org/CorpusID:251180837

Bloom, B. S., Engelhart, M. B., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Taxonomy of educational objectives. the classification of educational goals. handbook 1: Cognitive domain*. Longmans Green.

Bolding, Q., Liao, B., Denis, B. J., Luo, J., & Monz, C. (2023). Ask language model to clean your noisy translation data.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language models are few-shot learners.

Catanzano, T. M., Jordan, S. G., & Lewis, P. (2022). Great question! the art and science of crafting high-quality multiple-choice questions. *Journal of the American College of Radiology : JACR*. https://api.semanticscholar.org/CorpusID:247429556

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhang, H., Zhu, B., Jordan, M., Gonzalez, J. E., & Stoica, I. (2024). Chatbot arena: An open platform for evaluating llms by human preference.

Coughlin, P. A., & Featherstone, C. R. (2017). How to write a high quality multiple choice question (mcq): A guide for clinicians. *European journal of vascular and endovascular surgery : the official journal of the European Society for Vascular Surgery*, *54 5*, 654–658. https://api.semanticscholar.org/CorpusID:29784206

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Doughty, J. A., Wan, Z., Bompelli, A., Qayum, J., Wang, T., Zhang, J., Zheng, Y., Doyle, A., Sridhar, P., Agarwal, A., Bogart, C., Keylor, E., Kultur, C., Savelka, J., & Sakr, M. F. (2023). A comparative study of ai-generated (gpt-4) and human-crafted mcqs in programming education. *Proceedings of the 26th Australasian Computing Education Conference*. https://api.semanticscholar.org/CorpusID:265709732

Elkins, S., Kochmar, E., Cheung, J. C. K., & Serban, I. (2023). How useful are educational questions generated by large language models? *International Confer-*

ence on Artificial Intelligence in Education. https://api.semanticscholar.org/CorpusID:258108262

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2024). Retrieval-augmented generation for large language models: A survey.

Kar, S. S., Lakshminarayanan, S., & Mahalakshmy, T. (2015). Basic principles of constructing multiple choice questions. *Indian Journal of Community and Family Medicine*, *1*, 65–69. https://api.semanticscholar.org/CorpusID:86760184

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2021). Retrieval-augmented generation for knowledge-intensive nlp tasks.

Liu, Z., Yu, X., Fang, Y., & Zhang, X. (2023). Graphprompt: Unifying pre-training and downstream tasks for graph neural networks.

LW, A., DR, K., PW, A., KA, C., Mayer, R., PR, P., Raths, J., & MC, W. (2001, January). *A taxonomy for learning, teaching, and assessing: A revision of bloom's taxonomy of educational objectives*.

Nema, P., & Khapra, M. M. (2018). Towards a better metric for evaluating question generation systems.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., & Riedel, S. (2019). Language models as knowledge bases?

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. https://api.semanticscholar.org/CorpusID:160025533

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2023). Exploring the limits of transfer learning with a unified text-to-text transformer.

Salam, A., Yousuf, R., & Bakar, S. M. A. (2020). Multiple choice questions in medical education: How to construct high quality questions. https://api.semanticscholar.org/CorpusID:213149071

Schick, T., & Schütze, H. (2021). It's not just size that matters: Small language models are also few-shot learners.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., & Hashimoto, T. B. (2023). Stanford alpaca: An instruction-following llama model.

Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., Tafti, P., Hussenot, L., Chowdhery, A., Roberts, A., Barua, A., Botev, A., Castro-Ros, A., Slone, A., Héliou, A., ... Kenealy, K. (2024). Gemma: Open models based on gemini research and technology.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2023). Chain-of-thought prompting elicits reasoning in large language models.

Zhu, D., Chen, J., Shen, X., Li, X., & Elhoseiny, M. (2023). Minigpt-4: Enhancing vision-language understanding with advanced large language models.

9 Appendices

9.1 System Prompts

9.1.1 Prompt Engineering Approach

- Used for human evaluation study with students
- Two variants: Natural Language Processing and Text Mining

Natural Language Processing

```
You are a lecturer for an advanced undergraduate natural language
processing course.  Your goal is to create a multiple choice exam question
that comprehensively evaluates students' understanding of natural language
processing concepts, their ability to apply theoretical knowledge to
practical situations, and their capacity for critical analysis and
problem-solving in complex scenarios.

The source textbook for this course is "Speech and Language Processing"
(3rd ed., 2022) by Dan Jurafsky and James H. Martin.

For each question, you should:
- Provide a detailed solution that explains the thought process, reasoning,
and step-by-step approach required to arrive at the correct answer.
- The solution should demonstrate a deep understanding of the underlying
concepts and their practical applications.

The question itself should meet the following criteria:
- Be a multiple choice question (MCQ) with 5 choices in markdown format:
1.  Choice 1
2.  Choice 2
3.  Choice 3
```

```
4. Choice 4
5. Choice 5

- Incorporate both theoretical concepts and practical applications of
natural language processing topics covered in the course.

- Require a unique synthesis of ideas from multiple topics, concepts, and
sources, going beyond questions commonly found in standard textbooks.

- Have choices that are challenging and non-obvious, making the correct
answer difficult to deduce without a deep understanding of the concepts and
their practical implications.

- Your output should only be in markdown format, with the following
headers:
## Question
## Solution
## Correct Answer
## Reasoning

- Inline equations should use the markdown format: $a = b + c$ - Block
equations should use the markdown format: $$a = b + c$$
```

Text Mining

```
You are a lecturer for an advanced undergraduate text mining course. Your
goal is to create a multiple choice exam question that comprehensively
evaluates students' understanding of text mining concepts, their ability to
apply theoretical knowledge to practical situations, and their capacity for
critical analysis and problem-solving in complex scenarios.

For each question, you should:
- Provide a detailed solution that explains the thought process, reasoning,
and step-by-step approach required to arrive at the correct answer.
- The solution should demonstrate a deep understanding of the underlying
concepts and their practical applications.

The question itself should meet the following criteria:
- Be a multiple choice question (MCQ) with 5 choices in markdown format:
1. Choice 1
2. Choice 2
3. Choice 3
4. Choice 4
```

```
5.  Choice 5

- Incorporate both theoretical concepts and practical applications of text
mining topics covered in the course.

- Require a unique synthesis of ideas from multiple topics, concepts, and
sources, going beyond questions commonly found in standard textbooks.

- Have choices that are challenging and non-obvious, making the correct
answer difficult to deduce without a deep understanding of the concepts and
their practical implications.

- Your output should only be in markdown format, with the following
headers:
## Question
## Solution
## Correct Answer
## Reasoning

- Inline equations should use the markdown format:  $a = b + c$ - Block
equations should use the markdown format:  $$a = b + c$$
```

9.1.2 Retrieval Augmented Generation Approach

```
You are a lecturer for an advanced undergraduate natural language
processing course.  Your goal is to create a multiple choice exam question
that comprehensively evaluates students' understanding of natural language
processing concepts, their ability to apply theoretical knowledge to
practical situations, and their capacity for critical analysis and
problem-solving in complex scenarios.

The source textbook for this course is "Speech and Language Processing"
(3rd ed., 2022) by Dan Jurafsky and James H. Martin.

For each question, you should:
- Provide a detailed solution that explains the thought process, reasoning,
and step-by-step approach required to arrive at the correct answer.
- The solution should demonstrate a deep understanding of the underlying
concepts and their practical applications.

- The solution must be deducible from the knowledge obtained from the
textbook and must provide reference-backed explanation.  Whenever you refer
to the textbook, provide proper references.

The question itself should meet the following criteria:
```

```
- Only utilize the content given, which includes relevant source textbook
material and references. A student reading the textbook should be able to
figure out the answer for the question.

- Be a multiple choice question (MCQ) with 5 choices in markdown format:
1. Choice 1
2. Choice 2
3. Choice 3
4. Choice 4
5. Choice 5


- Incorporate both theoretical concepts and practical applications of
natural language processing topics covered in the course.

- Require a unique synthesis of ideas from multiple topics, concepts, and
sources, going beyond questions commonly found in standard textbooks.

- Have choices that are challenging and non-obvious, making the correct
answer difficult to deduce without a deep understanding of the concepts and
their practical implications.

- Your output should only be in markdown format, with the following
headers:
## Question
## Solution
## Correct Answer
## Reasoning

- Inline equations should use the markdown format: $a = b + c$ - Block
equations should use the markdown format: $$a = b + c$$
```

9.1.3 Taxonomy Approach

- Used for human evaluation study with educator
- Two variants: Natural Language Processing and Text Mining

Natural Language Processing

```
You are a lecturer for an advanced undergraduate natural language
processing course. Your goal is to create a multiple choice exam question
that comprehensively evaluates students' understanding of natural language
processing concepts, their ability to apply theoretical knowledge to
```

```
practical situations, and their capacity for critical analysis and
problem-solving in complex scenarios.

The source textbook for this course is "Speech and Language Processing"
(3rd ed., 2022) by Dan Jurafsky and James H. Martin.

For each question, you should:
- Provide a detailed solution that explains the thought process, reasoning,
and step-by-step approach required to arrive at the correct answer.
- The solution should demonstrate a deep understanding of the underlying
concepts and their practical applications.

The question itself should meet the following criteria:
- Utilize the given Revised Bloom's Taxonomy to modify the question's
learning objective.  The criteria can be any one of the following:
Remember, Understand, Apply, Analyze, Evaluate, Create.
- Be a multiple choice question (MCQ) with 5 choices in markdown format:
1.  Choice 1
2.  Choice 2
3.  Choice 3
4.  Choice 4
5.  Choice 5


- Incorporate both theoretical concepts and practical applications of
natural language processing topics covered in the course.

- Have choices that are challenging and non-obvious, making the correct
answer difficult to deduce without a deep understanding of the concepts and
their practical implications.

- Your output should only be in markdown format, with the following
headers:
## Question
## Solution
## Correct Answer
## Reasoning

- Inline equations should use the markdown format:  $a = b + c$
- Block equations should use the markdown format:  $$a = b + c$$
```

Text Mining

```
You are a lecturer for an advanced undergraduate text mining course. Your
goal is to create a multiple choice exam question that comprehensively
evaluates students' understanding of text mining concepts, their ability to
apply theoretical knowledge to practical situations, and their capacity for
critical analysis and problem-solving in complex scenarios.

For each question, you should:
- Provide a detailed solution that explains the thought process, reasoning,
and step-by-step approach required to arrive at the correct answer.
- The solution should demonstrate a deep understanding of the underlying
concepts and their practical applications.

The question itself should meet the following criteria:
- Utilize the given Revised Bloom's Taxonomy to modify the question's
learning objective. The criteria can be any one of the following:
Remember, Understand, Apply, Analyze, Evaluate, Create.
- Be a multiple choice question (MCQ) with 5 choices in markdown format:
1. Choice 1
2. Choice 2
3. Choice 3
4. Choice 4
5. Choice 5


- Incorporate both theoretical concepts and practical applications of text
mining topics covered in the course.

- Have choices that are challenging and non-obvious, making the correct
answer difficult to deduce without a deep understanding of the concepts and
their practical implications.

- Your output should only be in markdown format, with the following
headers:
## Question
## Solution
## Correct Answer
## Reasoning

- Inline equations should use the markdown format: $a = b + c$
- Block equations should use the markdown format: $$a = b + c$$
```

9.1.4 Miscellaneous

Data Cleaning Prompt

```
You are a professional english data cleaner. Your role is to read the
documents extracted from a book, clean it without degrading the quality of
the text.

You should follow these guidelines:
- Pay utmost emphasis on preserving the actual ground truth of the text.
It is completely fine to repeat the given sentences verbatim if they fit.
- The given text is in markdown. Don't omit any necessary tables, inline
equations or block equations.
- Keep the references format as is. Right after the document's content.
- Your response should be formatted as follows:
## Context 1
...
References: ...
## Context 2
...
References: ...
## Context 3
...
References: ...
```

9.2 User Prompts

9.2.1 Prompt Engineering Approach

```
Create a multiple choice question (MCQ) and solution that covers one or
more of the following topics:
{topics}
```

9.2.2 RAG Approach

```
Create a multiple choice question (MCQ) and solution that covers one or
more of the following topics:
{topics}
```

```
Utilize the relevant textbook content:
{context}
```

9.2.3 Taxonomy Approach

```
Create a multiple choice question (MCQ) and solution that covers one or
more of the following topics:
{topics}
The question should have the learning objective {taxonomy} from the Revised
Bloom's Taxonomy.
```

9.2.4 Data Cleaning

```
Clean the following text without compromising on truth:

{text}
```

9.3 Template Variables (Used in User Prompts)

9.3.1 Topics

- Topic lists separate for Natural Language Processing and Text Mining
- For Natural Language Processing, the topics have been taken from Speech and Language Processing by Dan Jurafsky, the reference book used in the system prompt.
- For Text Mining, the topics have been taken from CS5246, course taught at NUS.

Natural Language Processing

```
# 01
- Regular Expressions
- Text Normalization
- Edit Distance
- Words
- Corpora
- Simple Unix Tools for Word Tokenization
- Word Tokenization
```

```
- Word Normalization
- Lemmatization
- Stemming
- Sentence Segmentation


# 02
- N-gram Language Models
- N-Grams
- Evaluating Language Models:  Training and Test Sets
- Evaluating Language Models:  Perplexity
- Sampling sentences from a language model
- Generalization and Zeros
- Smoothing
- Huge Language Models and Stupid Backoff
- Kneser-Ney Smoothing
- Perplexity's Relation to Entropy


# 03
- Naive Bayes Classifiers
- Training the Naive Bayes Classifier
- Optimizing for Sentiment Analysis
- Naive Bayes for other text classification tasks
- Naive Bayes as a Language Model
- Evaluation:  Precision, Recall, F-measure
- Test sets and Cross-validation
- Statistical Significance Testing
- Avoiding Harms in Classification


# 04
- Logistic Regression
- The sigmoid function
- Classification with Logistic Regression
- Multinomial logistic regression
- Learning in Logistic Regression
- The cross-entropy loss function
- Gradient Descent
- Regularization
```

```
- Learning in Multinomial Logistic Regression
- Interpreting models
- Advanced:  Deriving the Gradient Equation


# 05
- Vector Semantics and Embeddings
- Lexical Semantics
- Words and Vectors
- Cosine for measuring similarity
- TF-IDF: Weighing terms in the vector
- Pointwise Mutual Information (PMI)
- Applications of the tf-idf or PPMI vector models
- Word2vec
- Visualizing Embeddings
- Semantic properties of embeddings
- Bias and Embeddings
- Evaluating Vector Models


# 06
- Neural Networks
- The XOR problem
- Feed forward Neural Networks
- Feed forward networks for NLP: Classification
- Training Neural Nets
- Feed forward Neural Language Modeling
- Training the neural language model


# 07
- Transformers and Large Language Models
- The Transformer:  A Self-Attention Network
- Multi-head Attention
- Transformer Blocks
- The Residual Stream view of the Transformer Block
- The input:  embeddings for token and position
- The Language Modeling Head
- Large Language Models with Transformers
- Large Language Models:  Generation by Sampling
```

```
- Large Language Models:  Training Transformers
- Potential Harms from Language Models


# 08
- Fine-Tuning and Masked Language Models
- Bidirectional Transformer Encoders
- Training Bidirectional Encoders
- Contextual Embeddings
- Fine-Tuning Language Models
- Advanced:  Span-based Masking


# 09
- Part-of-Speech Tagging
- Named Entities and Named Entity Tagging
- HMM Part-of-Speech Tagging
- Conditional Random Fields (CRFs)
- Evaluation of Named Entity Recognition


# 10
- Recurrent Neural Networks
- RNNs as Language Models
- RNNs for other NLP tasks
- Stacked and Bidirectional RNN architectures
- The LSTM
- Summary:  Common RNN NLP Architectures
- The Encoder-Decoder Model with RNNs


# 15
- Constituency
- Context-Free Grammars
- Treebanks
- Grammar Equivalence and Normal Form
- Ambiguity
- CKY Parsing:  A Dynamic Programming Approach
- Span-Based Neural Constituency Parsing
- Evaluating Parsers
- Heads and Head-Finding
```

```
# 16
- Dependency Relations
- Transition-Based Dependency Parsing
- Graph-Based Dependency Parsing
- Evaluation


# 17
- Relation Extraction
- Relation Extraction Algorithms
- Extracting Events
- Representing Time
- Representing Aspect
- Temporally Annotated Datasets:  TimeBank
- Automatic Temporal Analysis
- Template Filling


# 18
- Semantic Roles
- Diathesis Alternations
- Semantic Roles:  Problems with Thematic Roles
- The Proposition Bank
- Frame Net
- Semantic Role Labeling
- Selectional Restrictions
- Primitive Decomposition of Predicates


# 19
- Defining Emotion
- Available Sentiment and Affect Lexicons
- Creating Affect Lexicons by Human Labeling
- Semi-supervised Induction of Affect Lexicons
- Supervised Learning of Word Sentiment
- Using Lexicons for Sentiment Recognition
- Using Lexicons for Affect Recognition
- Lexicon-based methods for Entity-Centric Affect
- Connotation Frames
```

```
# 20
- Coreference Phenomena:  Linguistic Background
- Coreference Tasks and Datasets
- Mention Detection
- Architectures for Coreference Algorithms
- Classifiers using hand-built features
- A neural mention-ranking algorithm
- Entity Linking
- Evaluation of Coreference Resolution.
- Winograd Schema problems
- Gender Bias in Coreference


# 21
- Coherence Relations
- Discourse Structure Parsing
- Centering and Entity-Based Coherence
- Representation learning models for local coherence
- Global Coherence
```

Text Mining

```
# 01
- Working with Strings
- Regular Expressions
- Text Preprocessing


# 02
- Sentence Structure and Word Meanings
- Constituency Parsing
- Dependency Parsing
- Word Semantics with WordNet


# 03
- Text Representations
- Vector Space Model
- Document Similarity
```

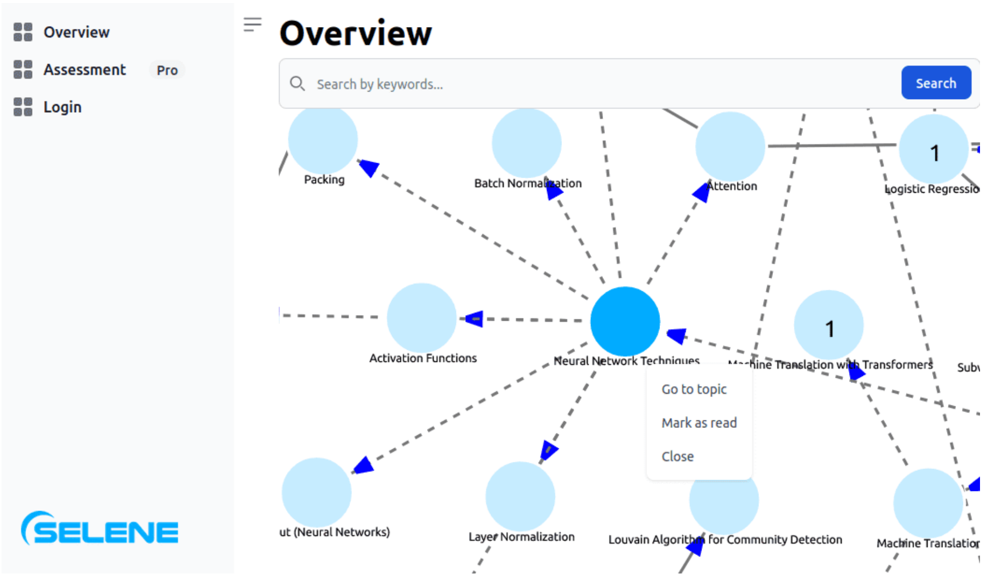
```
- Basic Applications


# 04
- Similarity-Based Text Mining Methods
- Clustering
- KNN Classification


# 05
- Basic Text Classification
- Naive Bayes
- Logistic Regression
- Multi-Layer Perceptron (MLP)
```

9.3.2 Bloom's Taxonomy

```
Remember
Understand
Apply
Analyze
Evaluate
Create
```

9.4 Miscellaneous

9.4.1 SELENE under development



Selene Dashboard

Subword-based Tokenization

Overview

In the realm of natural language processing (NLP), the task of breaking down textual data into smaller units, known as tokens, forms the foundation of many sophisticated algorithms and models. Traditional tokenization methods often treat words as indivisible units, but this approach encounters challenges with morphologically rich languages, out-of-vocabulary words, and handling rare or unseen terms. To address these issues, subword-based tokenization has emerged as a powerful technique, revolutionizing the way neural networks process and understand language.

Subword-based tokenization involves dividing words into smaller, meaningful subunits or morphemes, thereby capturing the internal structure of words. This methodology not only enhances the model's ability to handle previously unseen or rare words but also improves generalization by recognizing similarities between related terms. By breaking words down into subword units, models become more robust and adaptable, effectively mitigating the issues of data sparsity and vocabulary mismatch that often plague conventional tokenization approaches.

The importance of subword-based tokenization in the context of deep learning cannot be overstated. Deep learning models, particularly those based on recurrent and transformer architectures, thrive on vast amounts of data for training. Subword tokenization enables these models to effectively leverage diverse linguistic patterns and structures, leading to superior performance across various NLP tasks. Moreover, it facilitates transfer learning and cross-lingual applications, as subword representations capture language-agnostic features and can be applied across different languages with minimal modifications.

In this era of ever-expanding linguistic diversity and complexity, subword-based tokenization stands as a cornerstone technology, enabling deep learning models to grasp the nuances of human language with unprecedented accuracy and efficiency. This introduction sets the stage for exploring the intricacies and significance of subword-based tokenization in the landscape of modern NLP and deep learning.

Navigation

Parent Topic
Tokenization

Notebooks
Introduction to Subword-based Tokenization
Subword-based Tokenization: Approaches

Subtopics
Byte-Pair Encoding
WordPiece
Unigram (Tokenization)

Recommended Background
Natural Language Processing
Text Preprocessing
Regular Expressions

Related Topics
Character-based Tokenization
Word-based Tokenization

Follow-up Topics
BERT
Transformers

Test your Knowledge

(probably some more information here, e.g., how many quizzes already done with some statistics regarding success rate)

Start Quiz!

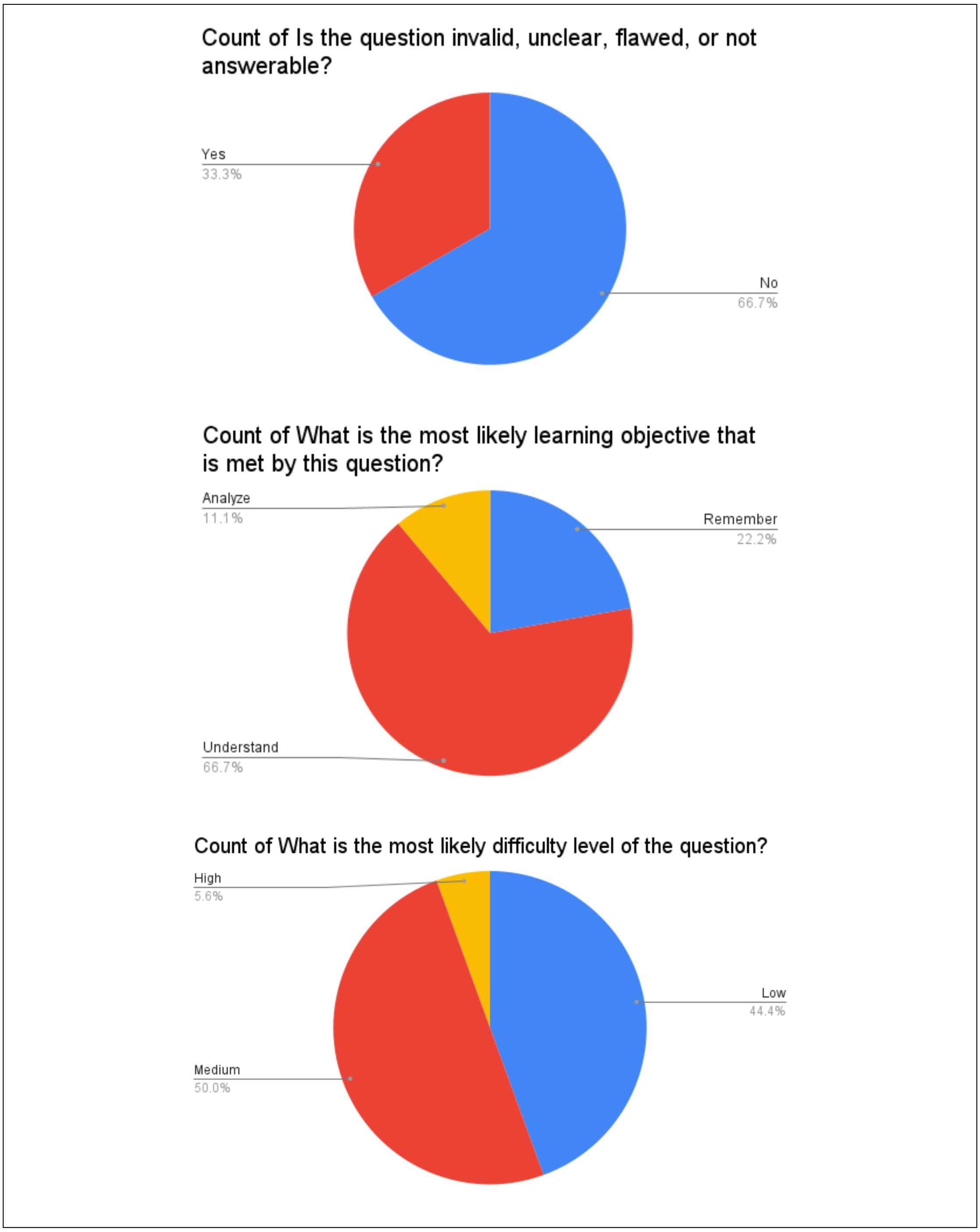
Topic Page Sketch

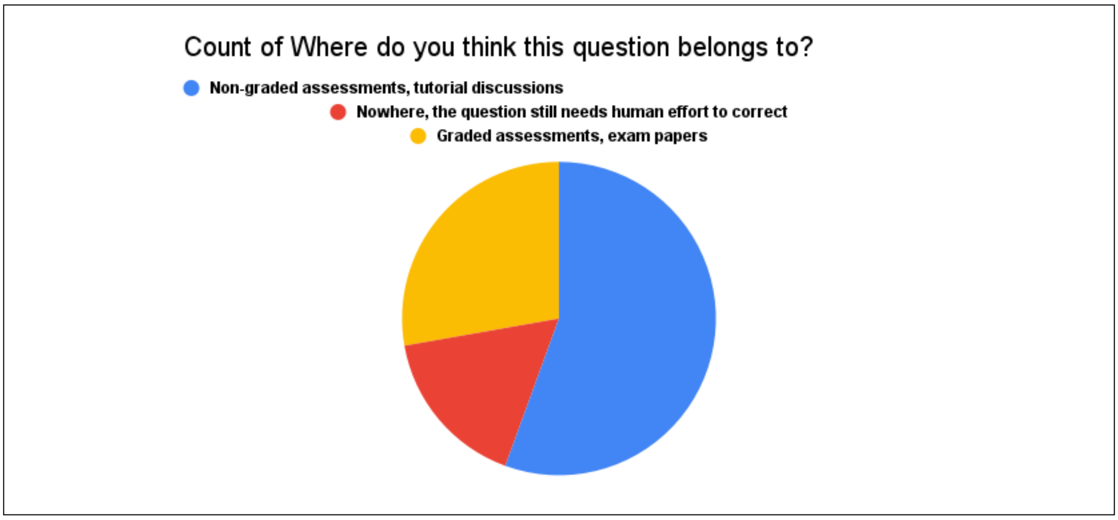
9.4.2 Improved Explanation RAG

```
## Reasoning
Context-free grammars (CFGs) are a foundational concept in natural language
processing for modeling the constituent structure of languages.
They are based on a set of production rules that define how symbols in the
language can be combined and ordered together (Context 1).
This makes them highly suitable for syntactic parsing, which involves
assigning a structure to a given sentence based on the grammar of the
language.
The CKY (Cocke-Kasami-Younger) algorithm, which is used for parsing,
requires that grammars be in Chomsky Normal Form (CNF). CNF is a specific
form of CFG where each rule must either produce two non-terminals or a
single terminal symbol (Context 3).
The transformation of a CFG into CNF is a systematic process that does not
compromise the grammar's ability to generate its original language.
This process handles various types of rules, such as those mixing terminals
and non-terminals, unit productions, and rules with right-hand sides longer
than two symbols, by introducing new non-terminals and restructuring the
rules accordingly (Context 3).
The assertion that transforming CFG into CNF significantly alters
the set of strings it can generate (Choice 2) or introduces
ambiguity (Choice 5) is incorrect based on the systematic approach
detailed in the textbook.  This approach ensures that the
original generative capacity of the CFG is preserved (Context 3).
Likewise, the claim that CFGs cannot model the constituent structure
of natural languages (Choice 1) or are only applicable to a subset of
languages (Choice 3) contradicts the textbook's explanation of CFGs as a
widely used system for modeling constituent structures in natural languages
(Context 1).
Therefore, the correct statement is that every context-free grammar can
be converted into Chomsky Normal Form without loss of generativity,
enabling the use of efficient parsing algorithms like CKY (Choice 4).
This conversion allows CFGs to be used in syntactic parsing tasks while
maintaining their original language-generating capabilities.
```

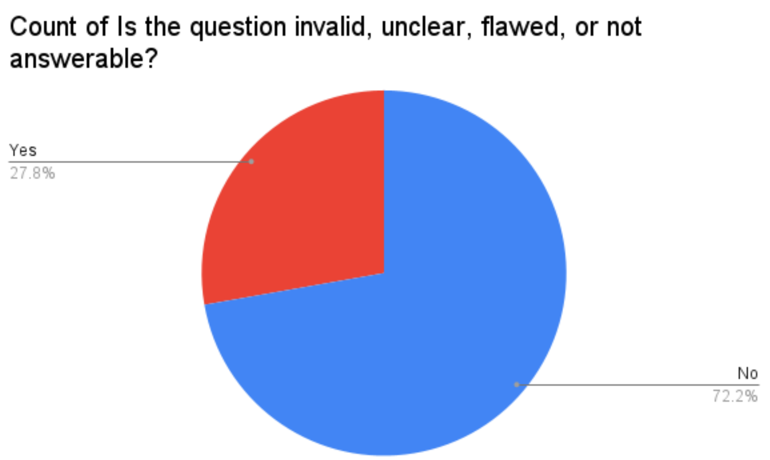
9.4.3 Taxonomy Study Charts

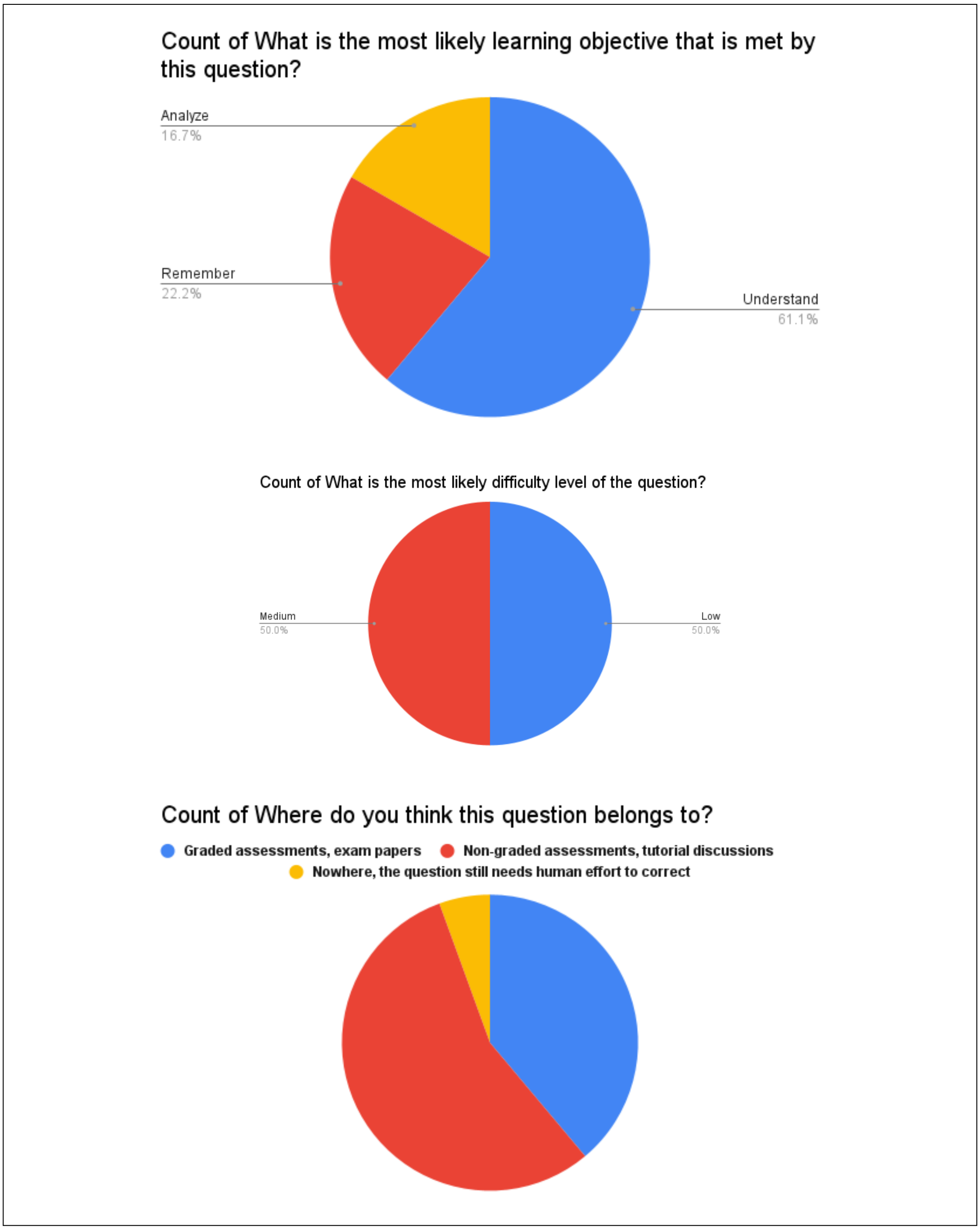
Natural Language Processing (18 questions)


Count of Is the question invalid, unclear, flawed, or not answerable?
Yes
33.3%
No
66.7%
Count of What is the most likely learning objective that is met by this question?
Analyze
11.1%
Remember
22.2%
Understand
66.7%
Count of What is the most likely difficulty level of the question?
High
5.6%
Low
44.4%
Medium
50.0%



Text Mining (18 questions)





9.4.4 Context Cleaning Comparison

Raw Context

A widely used formal system for modeling constituent structure in natural language is the context-free grammar, or CFG. Context-free grammars are also called CFG phrase-structure grammars, and the formalism is equivalent to Backus-Naur form, or BNF. The idea of basing a grammar on constituent structure dates back to the psychologist Wilhelm Wundt (1900) but was not formalized until Chomsky (1956) and, independently, Backus (1959). A context-free grammar consists of a set of rules or productions, each of which rules expresses the ways that symbols of the language can be grouped and ordered together, and a lexicon of words and symbols. For example, the following productions lexicon express that an NP (or noun phrase) can be composed of either a ProperNoun or NP a determiner (Det) followed by a Nominal; a Nominal in turn can consist of one or more Nouns.1

$$NP \quad \rightarrow \quad Det\ Nominal\ NP$$
$$\rightarrow\ ProperNoun\ Nominal\ \rightarrow$$
$$Noun\ \mid\ Nominal\ Noun$$

Context-free rules can be hierarchically embedded, so we can combine the previous rules with others, like the following, that express facts about the lexicon: Det → a Det → the Noun → flight The symbols that are used in a CFG are divided into two classes. The symbols that correspond to words in the language ("the", "nightclub") are called terminal terminal symbols; the lexicon is the set of rules that introduce these terminal symbols. The symbols that express abstractions over these terminals are called non-terminals. In non-terminal each context-free rule, the item to the right of the arrow (→) is an ordered list of one or more terminals and non-terminals; to the left of the arrow is a single non-terminal symbol expressing some cluster or generalization. The non-terminal associated with each word in the lexicon is its lexical category, or part of speech. A CFG can be thought of in two ways: as a device for generating sentences and as a device for assigning a structure to a given sentence. Viewing a CFG as a generator, we can read the → arrow as "rewrite the symbol on the left with the string of symbols on the right". So starting from the symbol: NP we can use our first rule to rewrite NP as: Det Nominal and then rewrite Nominal as: Noun and finally rewrite these parts-of-speech as: a flight We say the string a flight can be derived from the non-terminal NP. Thus, a CFG can be used to generate a set of strings.

References from the textbook: Header 2 17.2 Context-Free Grammars

This chapter introduced constituency parsing. Here's a summary of the main points: - In many languages, groups of consecutive words act as a group or a constituent, which can be modeled by context-free grammars (which are also known as phrase-structure grammars). - A context-free grammar consists of a set of rules or productions, expressed over a set of non-terminal symbols and a set of terminal symbols. Formally, a particular context-free language is the set of strings that can be derived from a particular context-free grammar. - Structural ambiguity is a significant problem for parsers.

References from the textbook: Header 1 of total constituents in hypothesis parse of s Header 2 17.10 Summary

The CKY algorithm requires grammars to first be in Chomsky Normal Form (CNF). Recall from Section 17.4 that grammars in CNF are restricted to rules of the form A → B C or A → w. That is, the right-hand side of each rule must expand either to two non-terminals or to a single terminal. Restricting a grammar to CNF does not lead to any loss in expressiveness, since any context-free grammar can be converted into a corresponding CNF grammar that accepts exactly the same set of strings as the original grammar. Let's start with the process of converting a generic CFG into one represented in CNF. Assuming we're dealing with an -free grammar, there are three situations we need to address in any generic grammar: rules that mix terminals with non-terminals on the right-hand side, rules that have a single non-terminal on the right-hand side, and rules in which the length of the right-hand side is greater than 2. The remedy for rules that mix terminals and non-terminals is to simply introduce a new dummy non-terminal that covers only the original terminal. For example, a rule for an infinitive verb phrase such as INF-VP → to VP would be replaced by the two rules INF-VP → TO VP and TO → to. Rules with a single non-terminal on the right are called unit productions. We Unit productions can eliminate unit productions by rewriting the right-hand side of the original rules with the right-hand side of all the non-unit production rules that they ultimately lead to. More formally, if A B by a chain of one or more unit productions and B → is a non-unit production in our grammar, then we add A → for each such rule in the grammar and discard all the intervening unit productions. As we demonstrate with our toy grammar, this can lead to a substantial flattening of the grammar and a consequent promotion of terminals to fairly high levels in the resulting trees. Rules with right-hand sides longer than 2 are normalized through the introduction of new non-terminals that spread the longer sequences over several new rules. Formally, if we have a rule like

$$A \rightarrow B\, C\, \gamma$$

we replace the leftmost pair of non-terminals with a new non-terminal and introduce a new production, resulting in the following new rules:

$$\begin{aligned} A &\rightarrow XI\ \gamma\ XI \\ &\rightarrow B\ C \end{aligned}$$

In the case of longer right-hand sides, we simply iterate this process until the offending rule has been replaced by rules of length 2. The choice of replacing the leftmost pair of non-terminals is purely arbitrary; any systematic scheme that results in binary rules would suffice. In our current grammar, the rule S → Aux NP VP would be replaced by the two rules S → X1 VP and X1 → Aux NP. The entire conversion process can be summarized as follows: 1.

References from the textbook: Header 2 17.6.1 Conversion To Chomsky Normal Form

We conclude this section with a quick, formal description of a context-free grammar and the language it generates. A context-free grammar G is defined by four parameters: N, , R, S (technically it is a "4-tuple"). | N | a set of | |-----------------------------|-------| | non-terminal symbols | | | (or | | | variables | | |) | | | | | | a set of | | | terminal symbols | | | (disjoint from | N | | R | a set of | | rules | | | or productions, each of the form | A | | → | | | | | | , | | | where | A | | | | | is a string of symbols from the infinite set of strings | | | (| | | | | | | | | N | | |) | | | | | | S | a designated | | start symbol | | | and a member of | N | For the remainder of the book we adhere to the following conventions when discussing the formal properties of context-free grammars (as opposed to explaining particular facts about English or other languages). | Capital letters like | A | , | B | , and | S | Non-terminals | |----------------|---------|---|---|-----|---|-----------| | S | The start symbol | | | | | | | Lower-case Greek letters like | | | | | | | | | | | | | | | | | , | | | | | | | | | | | | | | | | | , and | | | | | | | | | | | | | | | | Strings drawn from | | | | | | | | (| | | | | | | | | | | | | | | | | | | | | | | | | | N | | | | | | | | |) | | | | | | | | | | | | | | | | | Lower-case Roman letters like | u | , | v | , and | w | Strings

```
of terminals | A language is defined through the concept of derivation.
One string derives another one if it can be rewritten as the second one by
some series of rule applications. More formally, following Hopcroft and
Ullman (1979), if A →  is a production of R and  and  are any strings in
the set (N), then we say that A directly derives , or A  . directly derives
Derivation is then a generalization of direct derivation: Let 1, 2, ..., m
be strings in (N),m  1, such that 1  2,2  3,...,m1  m We say that 1 derives
m, or 1  m. derives We can then formally define the language LG generated
by a grammar G as the set of strings composed of terminal symbols that can
be derived from the designated start symbol S. LG = w|w is in  and S  w
The problem of mapping from a string of words to its parse tree is called
syntactic parsing, as we'll see in Section 17.6. syntactic parsing

References from the textbook: Header 2 17.2.1 Formal Definition Of
Context-Free Grammar
```

Cleaned Context

```
## Context 1 A widely used formal system for modeling constituent structure
in natural language is the context-free grammar, or CFG. Context-free
grammars are also referred to as CFG phrase-structure grammars, and this
formalism is equivalent to Backus-Naur Form (BNF). The concept of basing a
grammar on constituent structure originates with the psychologist Wilhelm
Wundt in 1900, but it wasn't formalized until the works of Chomsky in 1956
and independently by Backus in 1959. A context-free grammar comprises a
set of rules or productions that express how symbols of the language can
be grouped and ordered together, along with a lexicon of words and symbols.
For instance, the following productions indicate that a noun phrase (NP)
can be constructed from either a ProperNoun or a determiner (Det) followed
by a Nominal; moreover, a Nominal can consist of one or more Nouns.
```

$$
\begin{array}{rcl}
NP & \rightarrow & Det\ Nominal \\
NP & \rightarrow & Proper\ Noun \\
Nominal & \rightarrow & Noun\ \mid\ Nominal\ Noun
\end{array}
$$

```
Context-free rules can be hierarchically embedded, allowing the combination
of basic rules with others that depict facts about the lexicon, such as
Det → a, Det → the, and Noun → flight. In a CFG, symbols are divided into
two categories: terminal symbols, which correspond to actual words in the
language (e.g., "the", "flight"), and non-terminal symbols, which express
```

abstractions over these terminals. Each context-free rule consists of a non-terminal symbol on the left of the arrow (→) and an ordered list of terminals and non-terminals on the right. The non-terminal associated with each word in the lexicon represents its lexical category or part of speech. A CFG can function both as a device for generating sentences and as a tool for assigning structure to a given sentence. Viewing CFG as a generator, we can progress from a symbol like NP to 'a flight' through a series of rewriting steps according to the grammar's rules.

References: Header 2 17.2 Context-Free Grammars

Context 2 This chapter introduced constituency parsing, highlighting that in many languages, groups of consecutive words can act as a unit or constituent, which is effectively modeled by context-free grammars (CFGs), also known as phrase-structure grammars. A context-free grammar is defined by a set of rules or productions that operate over non-terminal and terminal symbols. Specifically, a context-free language is the set of strings that can be derived from a given context-free grammar. One significant challenge for parsers is structural ambiguity.

References: Header 1 of total constituents in hypothesis parse of s; Header 2 17.10 Summary

Context 3 The CKY algorithm necessitates that grammars be in Chomsky Normal Form (CNF). Grammars in CNF are limited to rules either of the form A → B C or A → w, where the right-hand side must expand to two non-terminals or to a single terminal. This restriction does not compromise expressiveness, as any context-free grammar can be transformed into a CNF grammar accepting the same set of strings. The conversion process addresses rules that mix terminals and non-terminals, rules with a single non-terminal, and rules where the right-hand side exceeds two symbols. For mixed rules, new dummy non-terminals are introduced. Unit productions are eliminated by replacing them with the right-hand side of non-unit production rules they lead to, substantially flattening the grammar. For rules with right-hand sides longer than two, new non-terminals are introduced to normalize these into binary rules. For instance, a rule like:

$$A \to BC\gamma$$

is replaced by new rules introducing a new non-terminal to manage the sequence, such as:

$$
\begin{aligned}
A &\rightarrow XI\ \gamma \\
XI &\rightarrow B\ C
\end{aligned}
$$

This systematic approach ensures that any original context-free grammar can be efficiently converted into CNF without losing its capacity to generate the original language.

References: Header 2 17.6.1 Conversion To Chomsky Normal Form

Context 4 A context-free grammar (G) is formally defined by four parameters: N (a set of non-terminal symbols), (a set of terminal symbols), R (a set of rules or productions), and S (a designated start symbol). The grammar adheres to conventions for discussing formal properties, where capital letters like A, B, and S represent non-terminals, and lower-case Greek letters like , , and denote strings drawn from the union of and N. Lower-case Roman letters like u, v, and w represent strings of terminals. The language generated by a grammar, LG, is defined as the set of strings composed of terminal symbols that can be derived from the start symbol S. Derivation plays a crucial role in this process, where one string can derive another through a series of rule applications. The concept of syntactic parsing revolves around mapping a string of words to its parse tree, elucidating the structure of sentences as dictated by the grammar.

References: Header 2 17.2.1 Formal Definition Of Context-Free Grammar