Impact of Noise on Retrieval-Augmented Generation Models for Question Answering

Adrienn Molnár

Introduction

This semester's project focuses on evaluating the performance of a Retrieval-Augmented Generation (RAG) model in the context of question answering (QA). A unique aspect of this evaluation is the intentional introduction of noise into the database to simulate errors commonly encountered during Optical Character Recognition (OCR) processes. This approach aims to test the model's robustness and accuracy under noisy data conditions.

The primary goal is to assess how well the RAG model can retrieve relevant documents and answer questions when the database is compromised with OCR-like noise. By introducing such noise, we simulate real-world scenarios where data is not perfectly clean, allowing us to understand the practical implications of deploying such models in real-life applications.

Key research questions include examining how OCR-induced noise affects the retrieval of relevant documents, the recall of these documents, and the overall question-answering performance.

Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) enhances large language models by integrating information retrieval systems, improving text generation through external knowledge sources.

Core Components of RAG Models

1. **Neural Language Model**: At the core of a RAG model is a large, pre-trained transformer-based language model, such as the GPT series developed by OpenAI. For our project, we utilize GPT-3.5 Turbo.
2. **Document Retrieval System**: This component involves indexing a large database of text documents - in our case, a Chroma vector database - and retrieving documents that are most relevant to a given query.

How RAG Models Operate

- **Query Processing:** Upon receiving a query, the RAG model uses its retrieval system to fetch relevant documents from an external database, providing additional context.
- **Context Integration:** Retrieved documents from the Chroma vector database are presented alongside the query to the language model. The database uses cosine distance to measure relevance, allowing the model to effectively integrate and cross-reference information.
- **Response Generation:** With enhanced context from the retrieved documents, the language model generates responses that are more informed, accurate, and contextually relevant.

SQuAD Database

For the purpose of this project, we utilized the Stanford Question Answering Dataset (SQuAD) as our primary database. SQuAD is a well-established benchmark dataset widely used in natural language processing for training and evaluating question-answering systems.

Overview of SQuAD

The SQuAD database consists of question-answer pairs based on Wikipedia articles. Each article is divided into paragraphs, and several questions are generated for each paragraph along with corresponding answers, which are typically spans of text within the paragraphs. This structure makes SQuAD suitable for testing retrieval-augmented models.

OCR Simulation

To evaluate the robustness of the RAG model under realistic conditions, we introduced simulated OCR errors into the SQuAD dataset. This approach was guided by the comprehensive analysis of OCR error patterns and characteristics presented in Jatowt et al. [2019].

Implementation of OCR Simulation

An error rate was defined to control the frequency of simulated OCR errors. Using a predefined set of character substitution probabilities, OCR errors were introduced into the text by randomly altering characters based on these probabilities.

The error rate is crucial for controlling the frequency of OCR errors introduced into the dataset. The normalization of probabilities was necessary because the original substitution probabilities for a given character did not add up to one. Specifically, there was an '@' symbol indicating the probability that a character could be replaced by any other character, which was not utilized in our simulation. By applying the error rate and normalizing the probabilities, we effectively managed the occurrence of errors.

The probabilities were adjusted and normalized as follows:

- For each character, the substitution probabilities (excluding the probability of the character remaining the same) were multiplied by the error rate.

  (Examples of the original and adjusted probabilities are provided in the Appendix.)

- After applying the error rate, the adjusted probabilities were then normalized so that their sum equals one.

Recall

In this project, recall is measured to evaluate the effectiveness of the RAG model in retrieving relevant contexts for question answering. For each question, there is exactly one correct context within the database. The recall metric is determined by checking if the correct context is among the top three contexts retrieved by the RAG model. If the correct context is found within these top three results, the recall is recorded as 1, indicating a successful retrieval. If the correct context is not among the top three, the recall is recorded as 0, indicating a failure to retrieve the relevant information.

Experiments

Our experiments were designed to evaluate the performance of the RAG model on a subset of the SQuAD dataset under varying levels of OCR-induced noise.

Correctness of Context Retrieval

Answer Evaluation

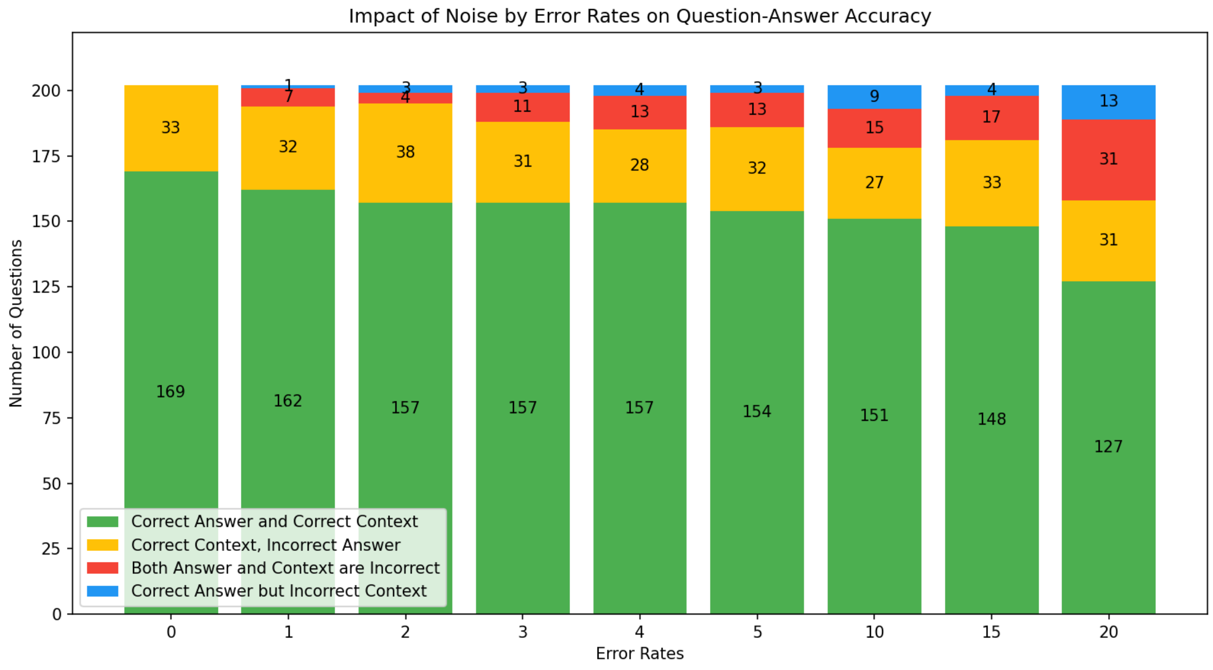
To evaluate whether an answer provided by the model is correct, we follow a two-step process. First, we check if all the exact words from the correct answer are present in the model's response. If the exact words are not found, it may be due to different phrasing. To account for this, we apply a fuzzy match as a secondary check. Fuzzy matching involves comparing the entities within the correct answer to those in the model's response, allowing for minor differences in wording. If the highest similarity score for any entity pair is below a certain threshold (e.g., 90%), the answer is considered incorrect.

Initial Experiment

First, we ran the RAG model on a set of 500 questions, including those marked as impossible in the SQuAD database. These impossible questions are labeled as such in the database because they cannot be answered based on the provided context, and we excluded these from our analysis (263 questions). From these initial runs, we identified the questions for which the model successfully retrieved the correct context. This initial filtering resulted in a subset of 202 questions where the model found the correct context.

Noise Experiments

Next, we ran the model on the identified 202 questions using databases with different levels of simulated OCR noise. The error rates applied were 1, 2, 3, 4, 5, 10, 15, and 20. The results are summarized using a stacked bar chart, shown below.



The chart displays the impact of different error rates on the performance of the RAG model in question-answering tasks. The x-axis represents various error rates (0, 1, 2, 3, 4, 5, 10, 15, 20), while the y-axis shows the number of questions (out of 202) for which the model retrieved specific outcomes. The different outcomes are represented by the colors on the chart, with corresponding labels also provided within the image.

Conclusions:

- **Difficulty in Finding Correct Context**: Introducing more noise makes it more difficult for the model to find the correct context, which in turn affects the accuracy of the answers.
- **Decreasing Performance**: As the error rate increases, the model's ability to find the correct context and provide the correct answer diminishes significantly. In some cases when the model provides an incorrect answer, it often indicates that the provided context does not contain the appropriate information.
- **Stable Incorrect Answer Rate**: Even without noise, the model fails to answer some questions correctly despite finding the correct context. This number of failures does not change drastically with the introduction of noise, suggesting that the inherent difficulty of the questions plays a significant role.
- **Correct Answers**: Interestingly, in some cases, the model can provide a correct answer even without retrieving the correct context, and this occurrence tends to increase with the error rate.

Vector Embeddings Analysis

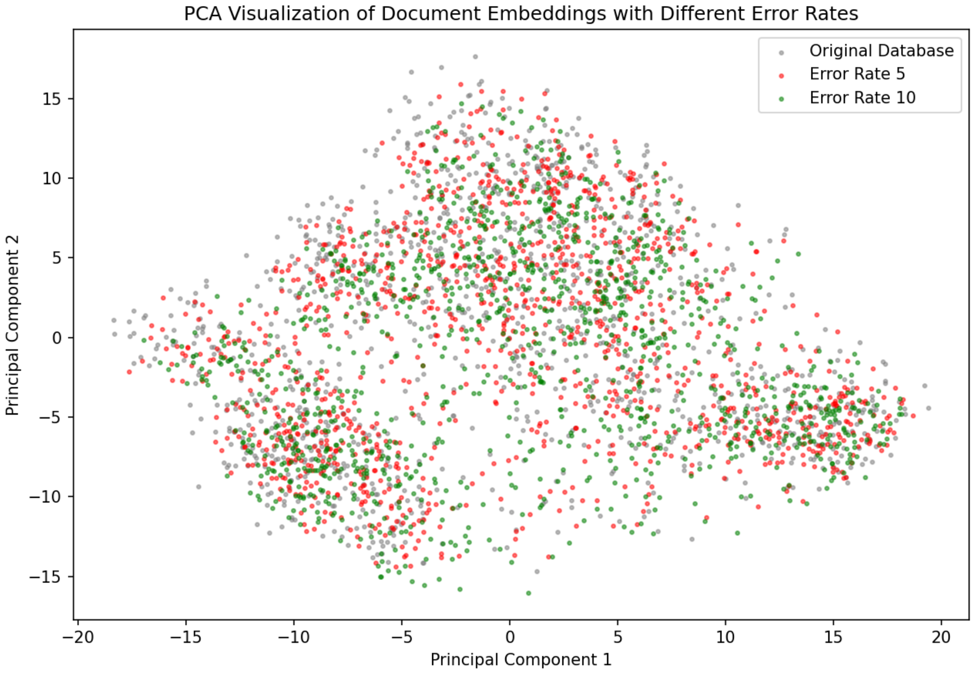
The second experiment was conducted to examine the impact of noise on the vector embeddings in the vector space. We compared the vectors from two databases: one without noise and the other with noise at a specified error rate. The aim was to understand how noise affects the embeddings and the structure of the vector space.

Principal Component Analysis (PCA) and Mean Deviation

To investigate the impact of noise on vector embeddings, we employed Principal Component Analysis (PCA), a dimensionality reduction technique that captures the most significant variance in data. This approach allows us to visualize and analyze structural changes in the vector space induced by noise.

We calculated the mean changes between the original and noisy embeddings in the PCA space to assess the impact of noise, determining the average deviation introduced across the principal components and revealing how noise alters the vector space structure. This analysis aims to identify methods for reversing specific types of noise by altering the vector embeddings. By understanding how different components respond to varying levels of noise, we aim to develop strategies to counteract these changes and restore the original vector representations.

The calculations and visualization of the embedding vectors provide significant insights into the impact of varying levels of noise, introduced as error rates, on document vectors. By analyzing the first two principal components, we observe a distinct pattern. From error rates 1 to 11, the vectors exhibit a consistent directional shift. The magnitude of this shift increases proportionally with the error rate.



The explained variance ratios provided by PCA indicate the proportion of the dataset's total variance captured by each principal component. In our analysis, the first component captures approximately 4.60% of the total variance, while the second component captures about 3.17%. Together, these two components account for around 7.77% of the overall variance.

Focusing on these two components helps us understand the most critical patterns within the data. This dimensionality reduction simplifies the dataset while preserving its essential structure, allowing for effective visualization and analysis of the primary sources of variance.

For the first principal component, the mean deviation gradually increased with noise up to the 11th error rate. However, at the 12th error rate, there was a significant shift: the mean deviation turned negative and continued to increase in the negative direction as the error rate increased. This suggests that beyond a certain threshold, the nature of the modifications drastically affects the vector representation.

In contrast, the second principal component exhibited a different pattern. As the error rate increased, the mean deviation initially moved further into the negative direction until the 13th error rate. After this point, there were sudden jumps from a negative to positive deviation, while the magnitude of the deviation continued to increase.

For the other components, we did not observe any consistent patterns. In some cases, the vectors tended to shift in a specific direction, but there was no clear trend related to the error rate. This variability suggests that the impact of noise on these components is less predictable and may depend on the specific nature of the modifications introduced.

Appendix

Example Probabilities:

- `'a'`: {'a': 97.5, 'u': 0.4, 'n': 0.2, 'e': 0.2, 'i': 0.2}
- `'b'`: {'b': 96.7, 'h': 1.6}

Adjusted Probabilities with Error Rate 2:

- `'a'`: {'a': 97.5, 'u': 0.8, 'n': 0.4, 'e': 0.4, 'i': 0.4}
- `'b'`: {'b': 96.7, 'h': 3.2}

References

Adam Jatowt, Mickael Coustaty, Nhu-Van Nguyen, Antoine Doucet, et al. Deep statistical analysis of ocr errors for effective post-ocr processing. In *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 29–38. IEEE, 2019.