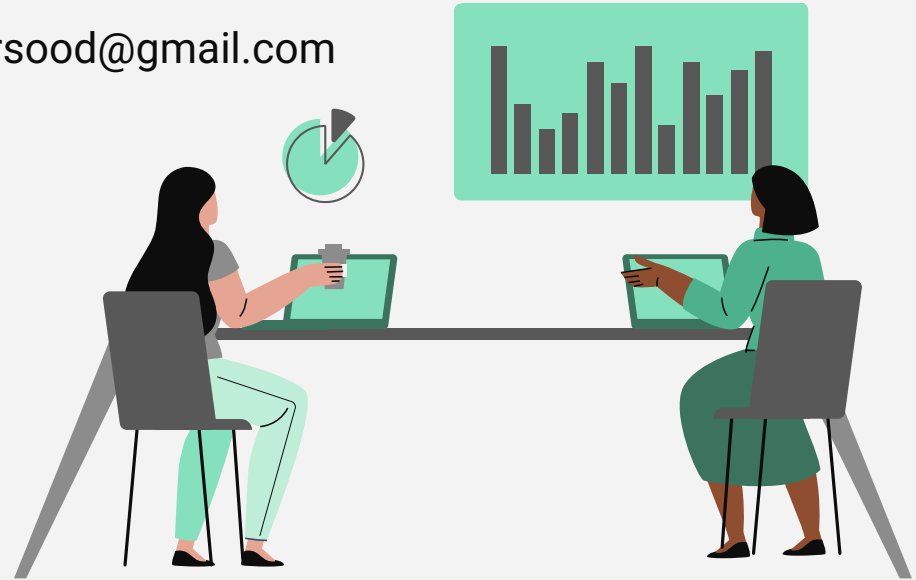
Hack to Hire 2024

Ambar Sood

New Delhi, 110014 | 9555888566 | ambarsood@gmail.com



Question Answer NLP – Quora dataset

This study presents a detailed overview of the processes involved in analyzing, preprocessing, and applying machine learning models to the **Quora Question Answer Dataset**. It covers data integrity checks, text preprocessing techniques, and visualizations to understand the dataset's characteristics. It also includes the **fine-tuning of a GPT-2 model** for generating responses, the creation of an interactive interface using **Gradio**, and the evaluation of model performance using various metrics. Additionally, different models and techniques for **similarity-based query** answering are explored and compared. The methodologies and insights provided serve as a comprehensive guide for implementing effective question-answering systems in NLP.

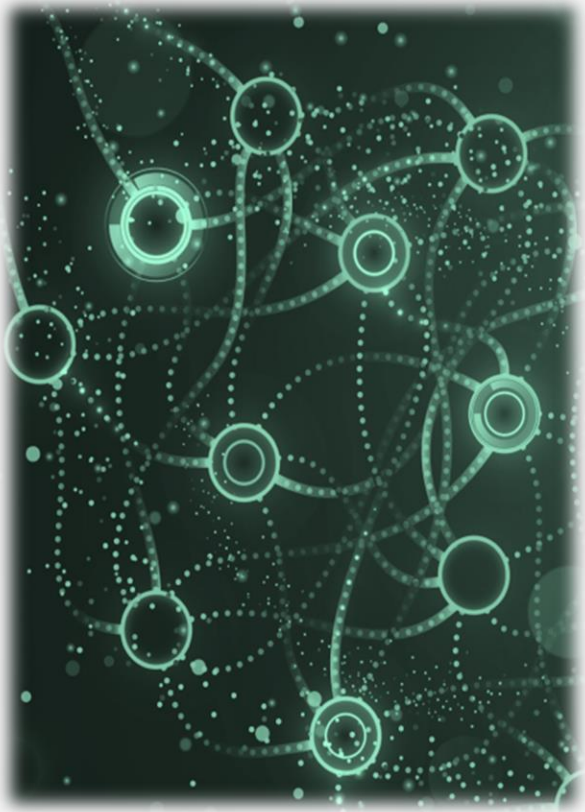


Table of contents

Approach

Finetuning GPT-2

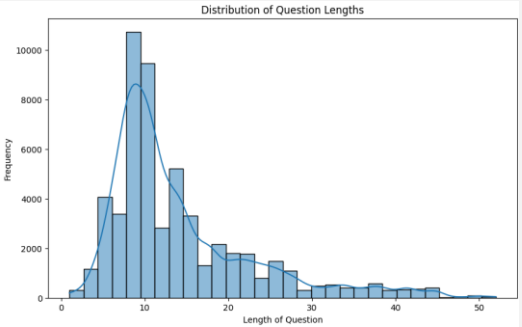
This approach to adapting a pre-trained GPT-2 model to our specific dataset. The process involves preparing the dataset, configuring the tokenizer, and fine-tuning the model using the Hugging Face library. We also describe the creation of an interactive interface with Gradio, allowing users to input questions and receive responses from the fine-tuned model. The model's performance is evaluated using metrics like ROUGE, BLEU, and F1-score, highlighting areas for improvement.

Similarity-based : Word embeddings

In this approach we explore various models and techniques to match user queries with the most relevant predefined questions and answers. These models include Bag of Words (BOW), Word2Vec, GloVe, and BERT. Each model is described in detail, covering their training, embedding generation, and similarity measurement processes. We compare the strengths and limitations of each approach, concluding that BERT offers the best performance for understanding and handling complex queries.

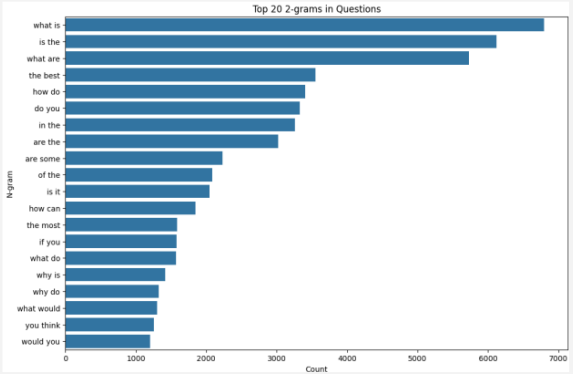
Initial Analysis

Question Length Distribution:



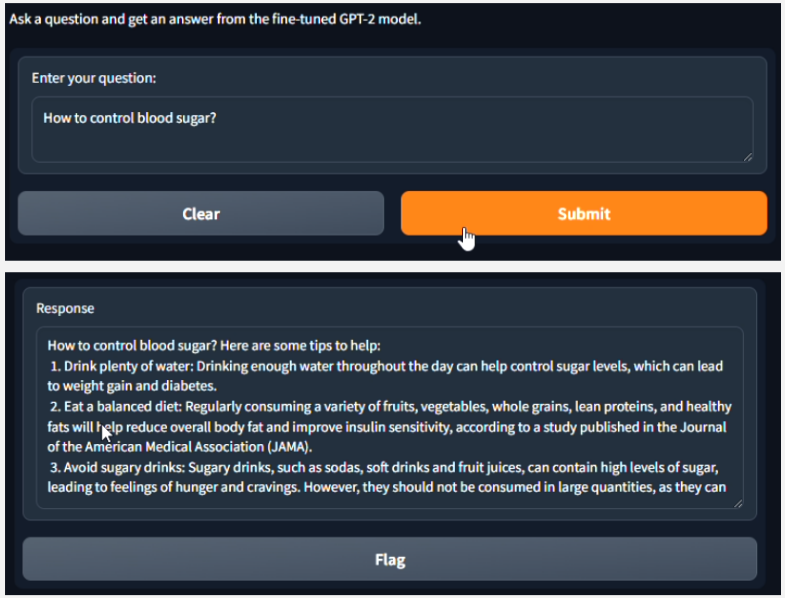
Word Cloud Visualization

N-gram Analysis:



*For in-depth analysis result please refer to the ipynb file in the repository

Finetuning GPT-2



Brief explaination and results

1. **Dataset Preparation (Load & Preprocess)**: Use CSV with Q&A pairs, format as "Question: {question} Answer: {answer}", convert to Hugging Face Dataset.
2. **Model Fine-Tuning Setup**: Load GPT-2 tokenizer/model, modify tokenizer. Training: Tokenize data, set training parameters, use Hugging Face Trainer, and save model.
3. **Model Interaction Inference:** Generate responses with beam search, temperature, nucleus sampling, decode to text.
4. **Interface Creation Gradio:** Build a web interface for user Q&A interaction.
5. **Practical Analysis:** Working for demonstration
6. **Evaluation Metrics:**

- ROUGE: Low precision/good recall.
- BLEU: Low(more epochs needed)
- F1-Score: Low.

**For in-depth results and model please refer to the ipynb file in the repository*

Similarity-based : Word embeddings

Brief explaination

Models & Techniques

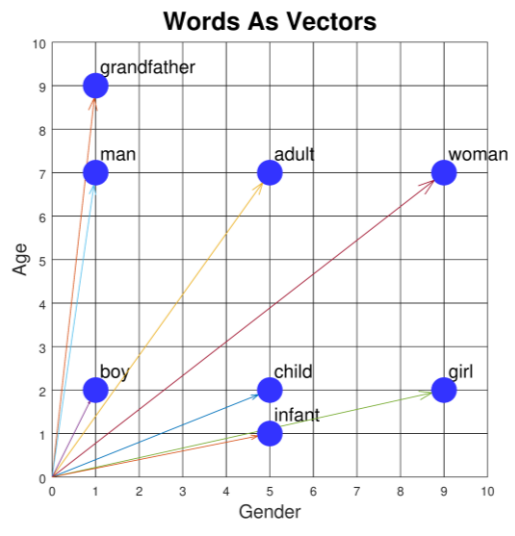
Bag of Words (BOW):
Description: Represents text as vectors of word frequencies.
Process: Preprocess text, create vectors, compute cosine similarity.
Limitations: Lacks semantic context.

Word2Vec:
Description: Creates dense word embeddings capturing semantic meaning.
Process: Train on large text corpus, average embeddings for phrases, compute cosine similarity.
Advantages: Better semantic understanding.
Limitations: May miss subtle nuances and out-of-vocabulary words.

GloVe:
Description: Uses global word co-occurrence statistics to create embeddings.
Process: Train on co-occurrence matrix, aggregate word embeddings, compute cosine similarity.
Advantages: Good for global word relationships.
Limitations: May struggle with nuanced meanings.

BERT:
Description: Context-aware model using deep learning to capture nuanced meanings.
Process: Pre-train on large corpus, generate context-based embeddings, compute cosine similarity.
Advantages: Excellent for understanding complex queries.
Limitations: Computationally intensive.3.

Comparison BOW: Simple but lacks semantic depth.
Word2Vec & GloVe: Capture semantic relationships but miss some nuances.
BERT: Best for complex queries but requires more resources.



Word embedding is a technique where words are represented as dense vectors in a continuous vector space, capturing semantic meaning and relationships based on their context in a text corpus.

*For in-depth analysis result please refer to the ipynb file in the repository

Novel Improvements

Model Architecture Adjustments:

Multi-Task Learning: Fine-tune the model on multiple related tasks (e.g., summarization or text classification) to leverage shared learning.
Hybrid Models: Combine GPT-2 with other models (e.g., BERT) in a multi-model architecture to capture different aspects of language understanding.

Advanced Similarity Measures:

Hybrid Similarity Metrics: Combine multiple similarity measures (e.g., cosine similarity, Euclidean distance) to improve retrieval accuracy.

Interactive Interface Enhancements:

Feedback Loop: Implement a feedback mechanism in the Gradio interface that allows users to rate responses or provide corrections, helping to iteratively improve the model.
Contextual Prompts: Allow users to provide additional context or examples to help the model generate more accurate responses.

Optimization:

Model Compression: Apply model compression techniques to reduce the computational load of embedding models while maintaining performance.
Batch Processing: Use batch processing for similarity calculations to improve efficiency, especially for large-scale datasets.