CHAT-WITH-FILE

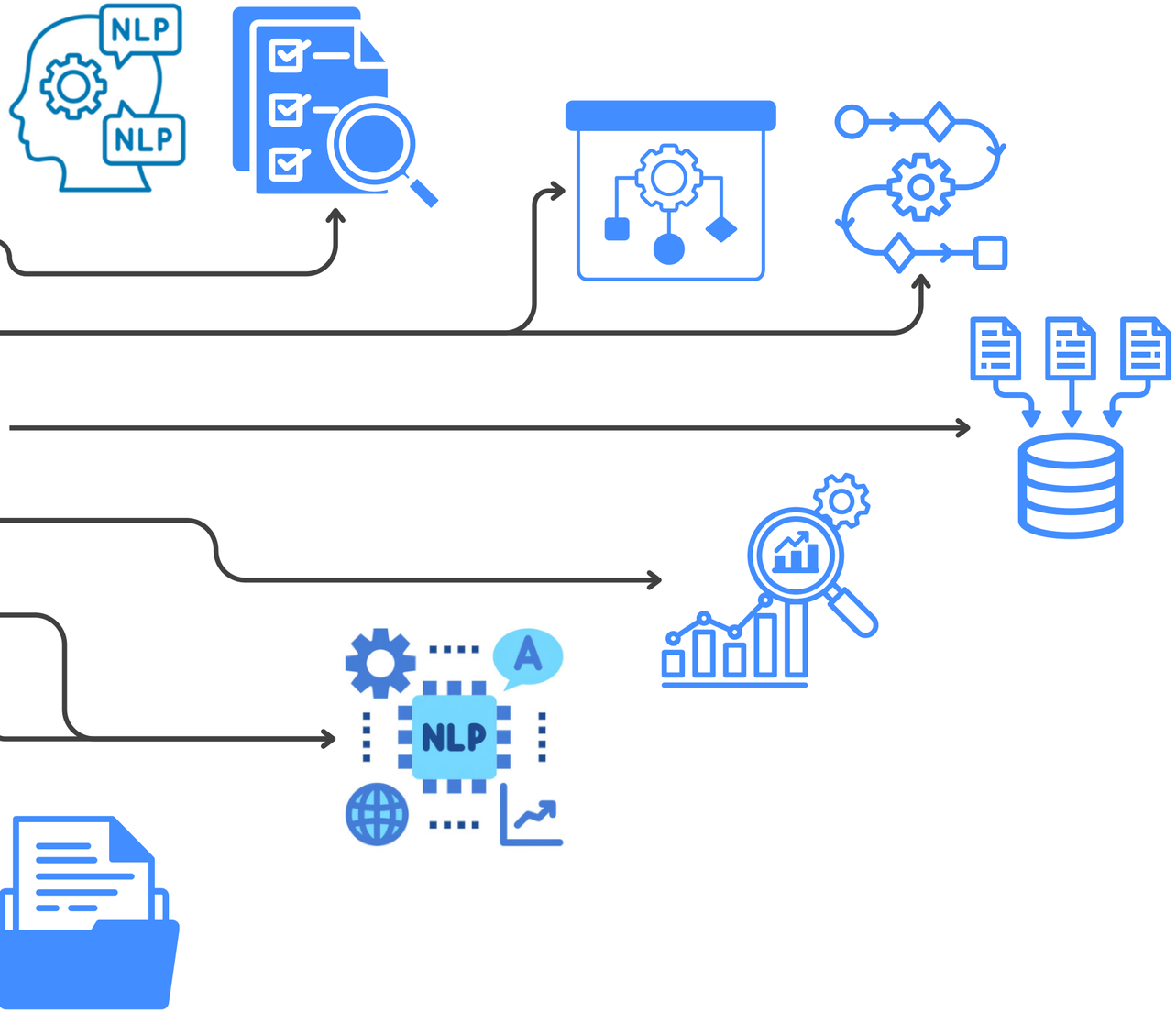
Machine Learning Project

PREPARED BY:
Deepam N. Chhimpi (202203103510092)

public link: https://www.canva.com/design/DAGgTUT_OLc/jMNE7zRBFmzMdQqKLJu9bw/view?utm_content=DAGgTUT_OLc&utm_campaign=designshare&utm_medium=link2&utm_source=uniquelinks&utlId=h6ac3c63377

Outline

1. Introduction
2. Literature Review
3. Algorithms
4. Dataset Description
5. Results Analysis
6. Conclusion
7. Future Work
8. References



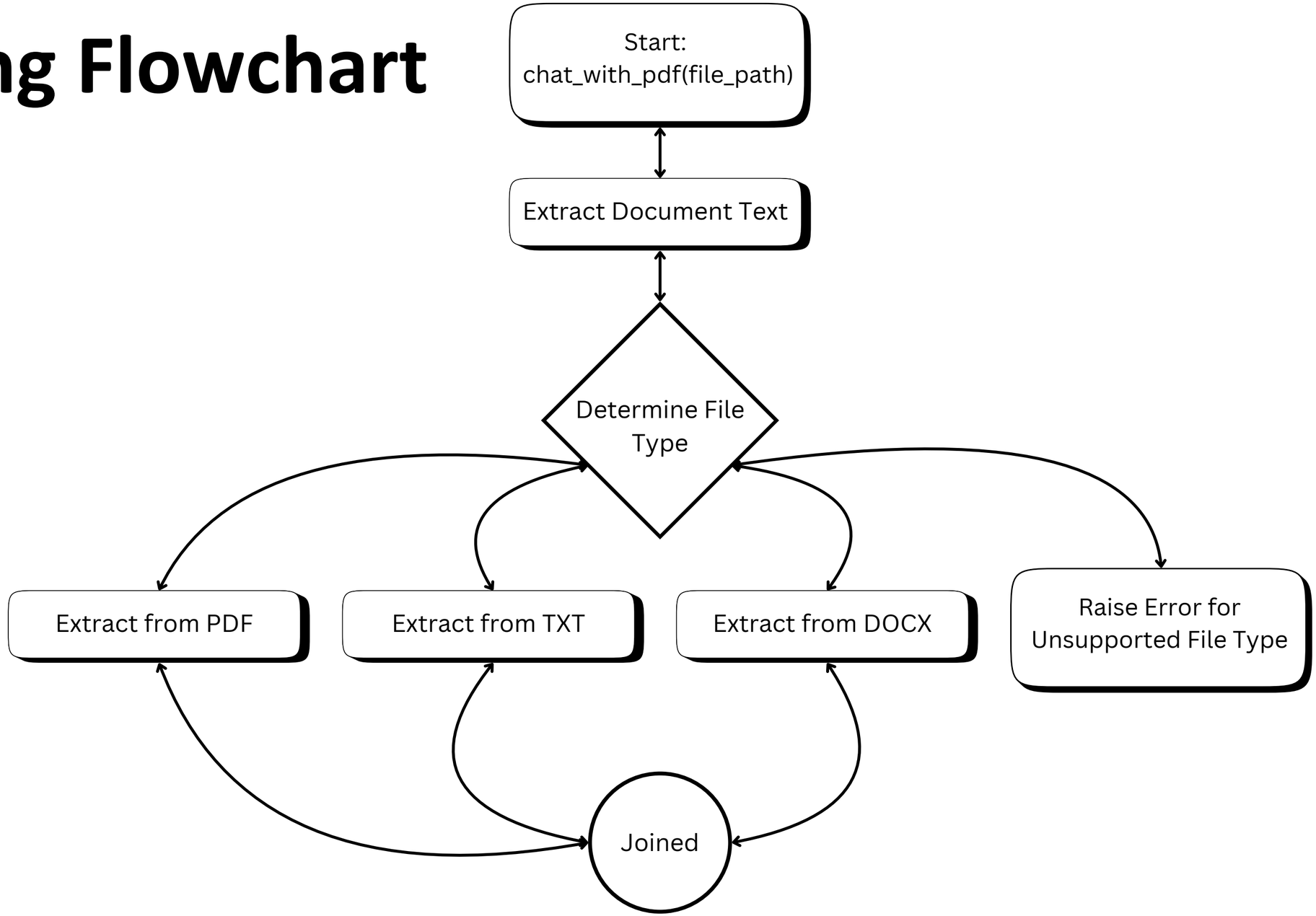
Introduction

- **Topic Importance**
  This Machine Learning Model transforms information retrieval by enabling Natural Language Queries(NLQ) across documents, addressing information overload and saving valuable time searching through content.

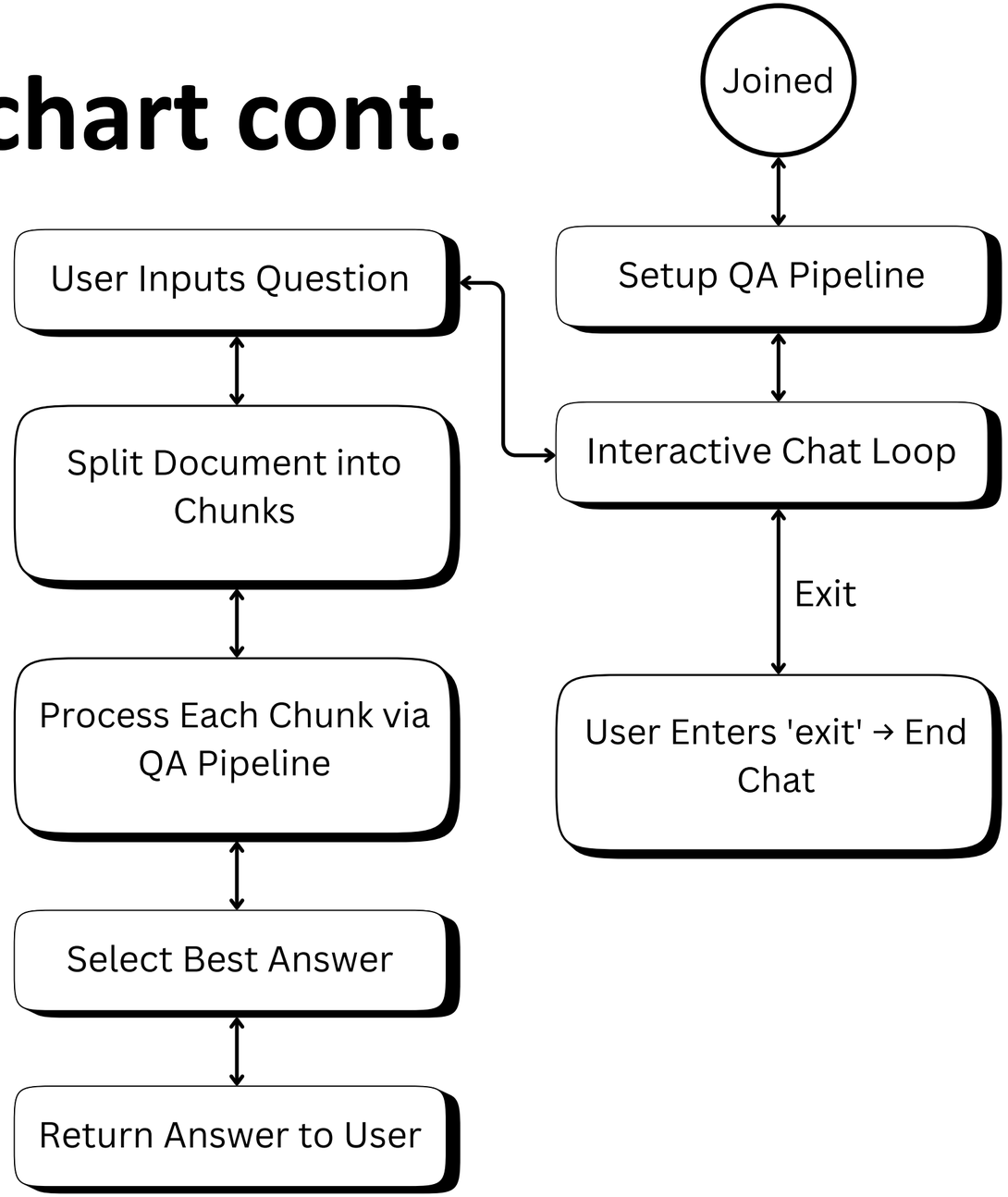
- **Research Problem/Objective**
  This project creates a document chatbot that processes multiple file formats (PDF, DOCX, TXT) through Natural Language Processing(NLP), extracting relevant answers to natural language questions regardless of document type(i.e. from PDF, DOCX, TXT).

- **Key Contributions**
  1) multi-format document processing,
  2) optimized text chunking for contextual integrity(CI), and
  3) RoBERTa language model integration with confidence scoring for reliable answer extraction.

Working Flowchart



Working Flowchart cont.



Literature Review

1. **Research Paper 1: “Progress in Neural NLP: Modeling, Learning, and Reasoning (March, 2020)”**
   - The literature review highlights advances in neural NLP through modeling (embedding, RNNs, attention),and learning (supervised, unsupervised, transfer learning) for improved QA and dialog systems.

2. **Research Paper 2: “PyTorch and TensorFlow Performance Evaluation in Big data Recommendation System (August, 2024)”**
   - Recommender systems(RS) use deep learning to improve accuracy, addressing challenges like scattered data and dynamic preferences. Integrating NLP enhances predictions by analyzing user reviews. Studies show PyTorch is flexible for research, while TensorFlow excels in deployment. Despite advancements, scalability and adaptability remain key challenges in RS development.

Literature Review cont.

3. Text-Book: “RoBERTa: A Machine Reading Comprehension for Climate Change QA (July, 2023)”

- QAS(Question-Answering System) has evolved from rule-based to deep learning models, improving accuracy.
- NLP(Natural Language Processing) advancements enhance text understanding, crucial for QAS.
- MRC(Machine Reading Comprehension), using models like BERT and RoBERTa, extracts precise answers.
- RoBERTa excels in small datasets like Climate Change MRC. These advancements make QAS more effective for specialized domains.

Research paper 1(Progress in Neural NLP: Modeling, Learning, and Reasoning)

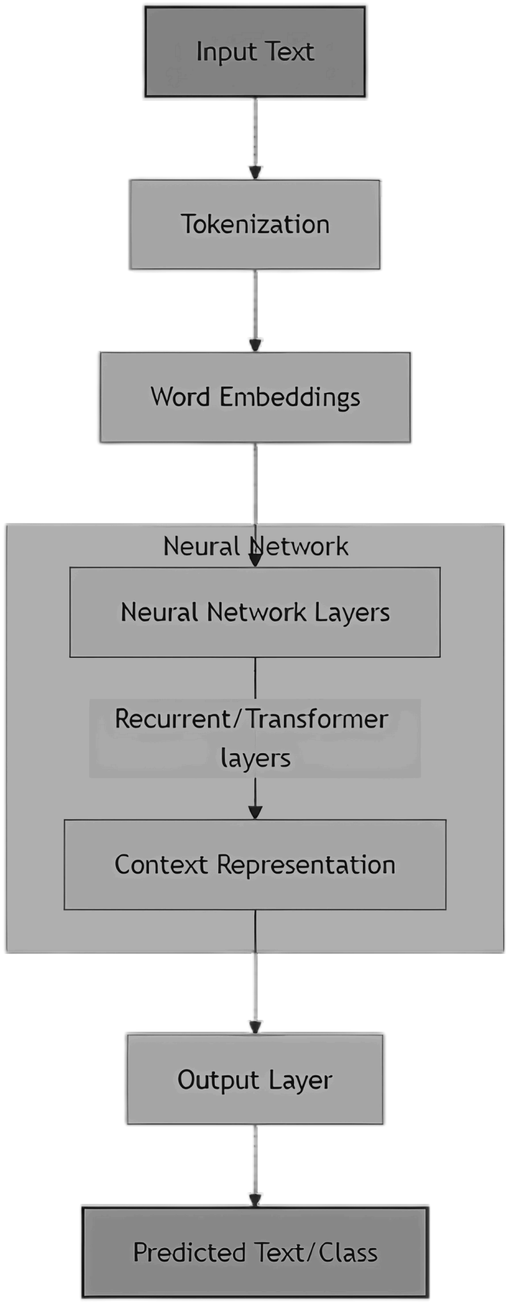
- Abstract and Introduction: These research paper explaining the rapid development of neural like NLP, its importance, and the historical context behind the methods discussed.

- Modeling: The paper reviews various neural network-based modeling approaches. This includes methods for creating word and sentence embeddings like self-attention mechanisms (as seen in models like GPT, BERT,etc). It also covers sequence-to-sequence modeling frameworks (such as encoder–decoder and attention-based models) for tasks like translation and question-answering.

Research paper 1(Progress in Neural NLP: Modeling, Learning, and Reasoning)

- Learning: This part focuses on how NLP models are trained. It discusses supervised learning, semi-supervised and unsupervised learning (including techniques like back-translation), multitask learning, transfer learning with pre-trained models, and active learning strategies.

- Conclusions: Finally, the paper sums up the progress and outlines promising directions for further research in neural NLP.



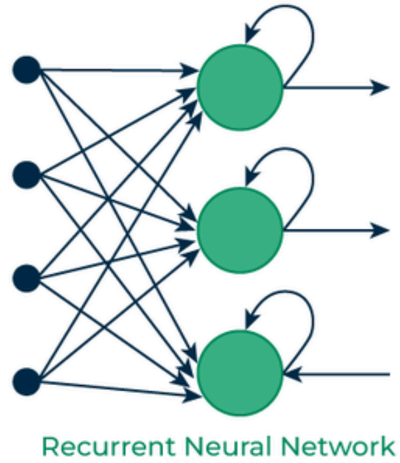
Natural Language Processing Working

Input text:

"The quick brown fox jumps over the lazy dog."

Tokenized output (subwords in BERT-style model):

["The", "quick", "brown", "fox", "jump", "##s", "over", "the", "lazy", "dog", "."]



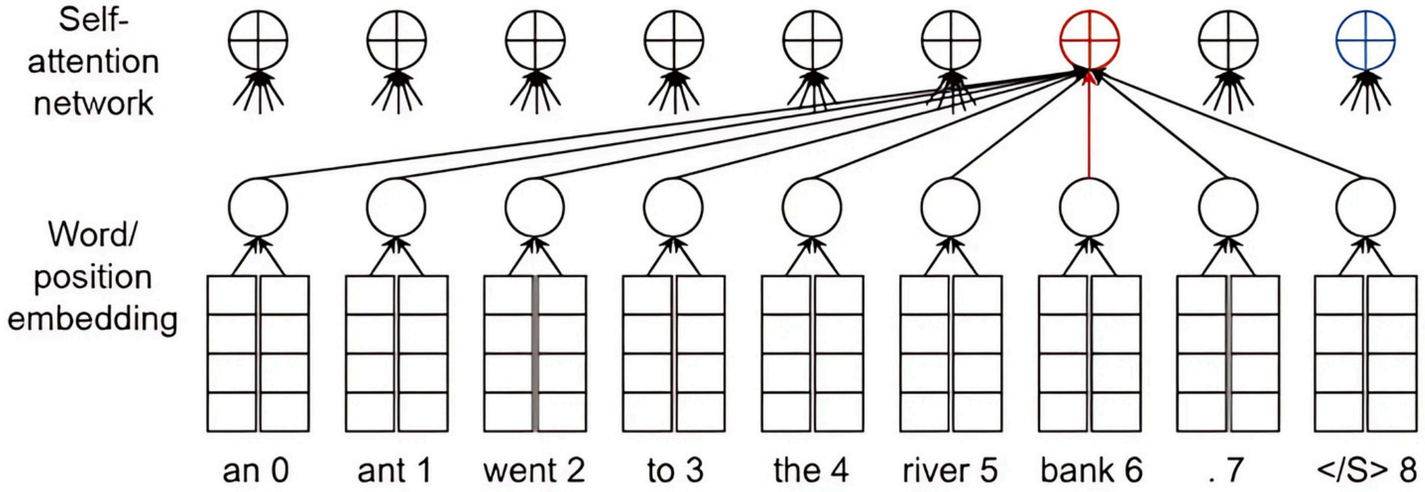


Fig. 3. Self-attention-based context-aware word embedding. : sentence-ending symbol.

Research paper 2(PyTorch and TensorFlow Performance Evaluation in Big data)

- Text Extraction from Multiple File Types:
  - The research paper emphasizes/extracting big data from various sources (e.g., user reviews and ratings), our model includes functions to extract text from PDF, TXT, and DOCX files. This modular approach ensures that the system can work with different formats, much like the diverse data types considered in the research.

- Natural Language Processing (NLP) for Unstructured Data:
  - This research paper discusses on enhancing recommendation accuracy on an incorporating text (e.g., user review comments) and using NLP techniques to process this information. Similarly, the code sets up a question-answering (QA) pipeline using a transformer model (RoBERTa based on SQuAD2) to interpret and extract relevant answers from the extracted document text.

Research paper 2(PyTorch and TensorFlow Performance Evaluation in Big data)

- Handling Long Texts Through Token Chunking:
  - To manage extensive text data—a challenge mentioned in the paper when dealing with big data—the code tokenizes the document and creates overlapping chunks. This strategy ensures that context is preserved across chunks, analogous to how the paper discusses techniques to mitigate issues like data scattering and dynamic preferences in recommendation systems.

- Leveraging Deep Learning Frameworks:
  - Although the paper's focus is on comparing PyTorch and TensorFlow for building recommendation models, our code demonstrates a related concept by using a transformer-based model (a deep learning approach) for QA. This highlights the broader idea of using state-of-the-art deep learning frameworks to process and derive insights from large textual datasets.

Text-Book (RoBERTa: A Machine Reading Comprehension for Climate Change QA)

- Intelligent Question-Answering Systems
  - QAS has evolved from early rule-based systems to modern deep learning models, significantly improving answer accuracy and efficiency.

- Natural Language Processing (NLP)
  - NLP advancements, from rule-based approaches to neural networks, have enhanced text understanding, making it essential for QAS.

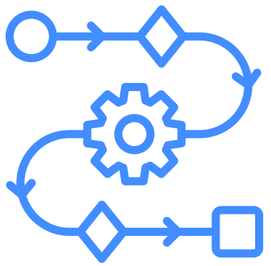
- Machine Reading Comprehension (MRC)
  - MRC enables systems to extract answers from text using deep learning models like BERT, ALBERT, and RoBERTa, improving QAS performance.

Text-Book (RoBERTa: A Machine Reading Comprehension for Climate Change QA)

- <u>RoBERTa for MRC in Small Datasets</u>
  - RoBERTa refines BERT's capabilities, excelling in small datasets like Climate Change MRC by leveraging pre-training and fine-tuning.

- <u>Conclusion</u>
  - Advancements in NLP and MRC, particularly with RoBERTa, enhance QAS accuracy, making them effective for specialized domains like Question-Answer Model.

Algorithms

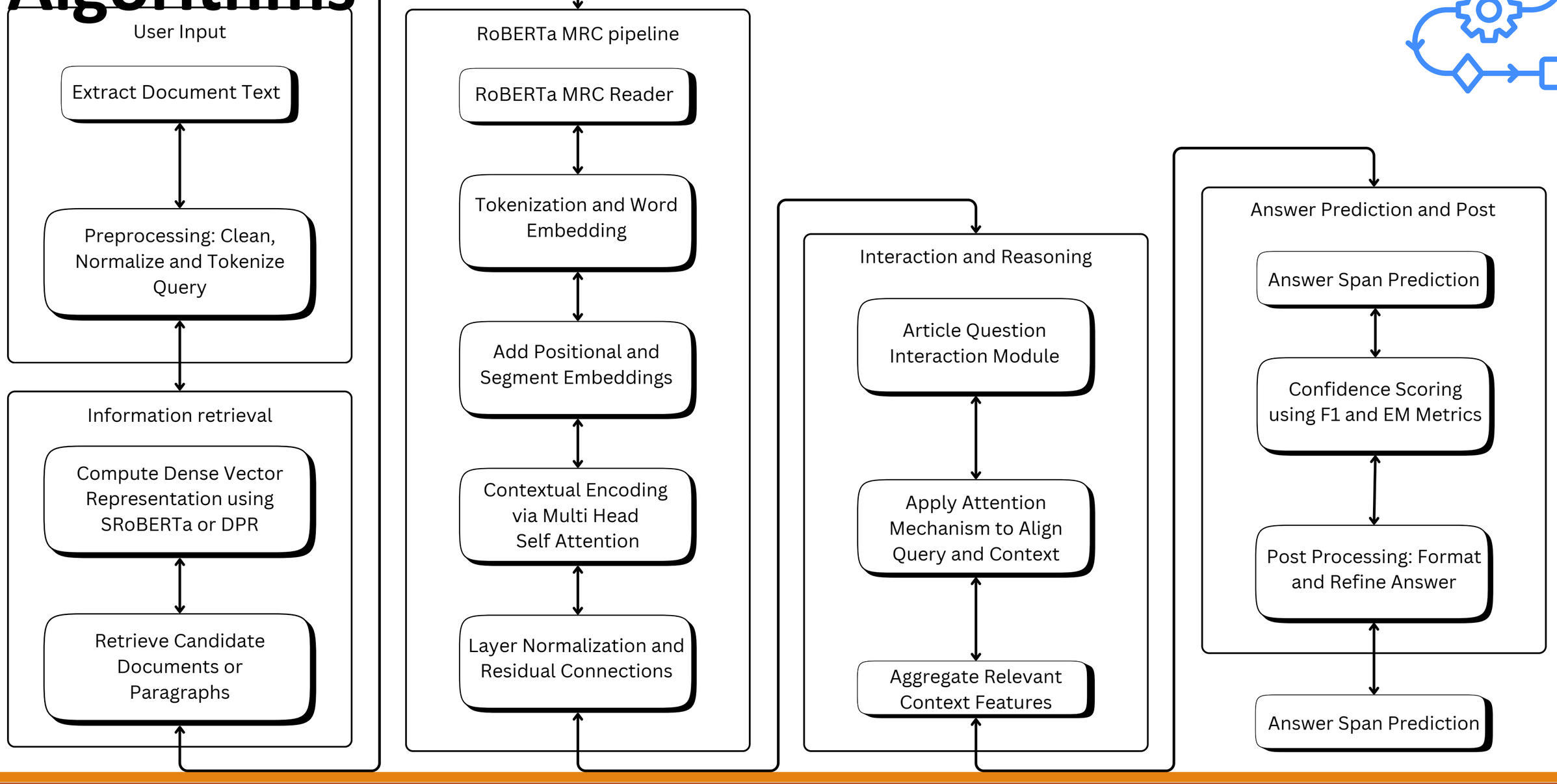


- Code implements on an algorithm of Transformer-based question answering system,The core algorithm is the RoBERTa model (specifically "deepset/roberta-base-squad2"), which is a transformer architecture fine-tuned on the SQuAD2 dataset.

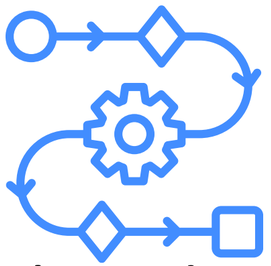
1. **User Input and Preprocessing:**
   - User Input Question: The user submits a natural language question.
   - Preprocessing: The system cleans, normalizes, and tokenizes the query to prepare it for further processing.

2. **Information Retrieval:**
   - Dense Vector Representation: The query is transformed into a dense vector using methods such as Semantic RoBERTa(SRoBERTa) or DPR (Dense Passage Retrieval) to capture semantic meaning.
   - Candidate Retrieval: The system retrieves relevant documents or paragraphs from a large corpus based on this representation.

Algorithms

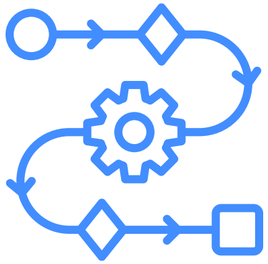

User Input
Extract Document Text
Preprocessing: Clean, Normalize and Tokenize Query
Information retrieval
Compute Dense Vector Representation using SRoBERTa or DPR
Retrieve Candidate Documents or Paragraphs
RoBERTa MRC pipeline
RoBERTa MRC Reader
Tokenization and Word Embedding
Add Positional and Segment Embeddings
Contextual Encoding via Multi Head Self Attention
Layer Normalization and Residual Connections
Interaction and Reasoning
Article Question Interaction Module
Apply Attention Mechanism to Align Query and Context
Aggregate Relevant Context Features
Answer Prediction and Post
Answer Span Prediction
Confidence Scoring using F1 and EM Metrics
Post Processing: Format and Refine Answer
Answer Span Prediction

Algorithms



3. **RoBERTa MRC Pipeline:**
   - RoBERTa MRC Reader: The retrieved text is fed into the RoBERTa-based machine reading comprehension reader.
   - Tokenization and Word Embedding: The text is tokenized and each token is converted into an embedding vector.
   - Positional and Segment Embeddings: Additional embeddings are added to capture token order and context boundaries.
   - Contextual Encoding: Multiple layers of multi head self attention are used to generate deep contextual representations.
   - Normalization and Residual Connections: These techniques stabilize training and help capture more robust features.
4. **Interaction and Reasoning:**
   - Article Question Interaction Module: Aligns the encoded context with the query.

Algorithms

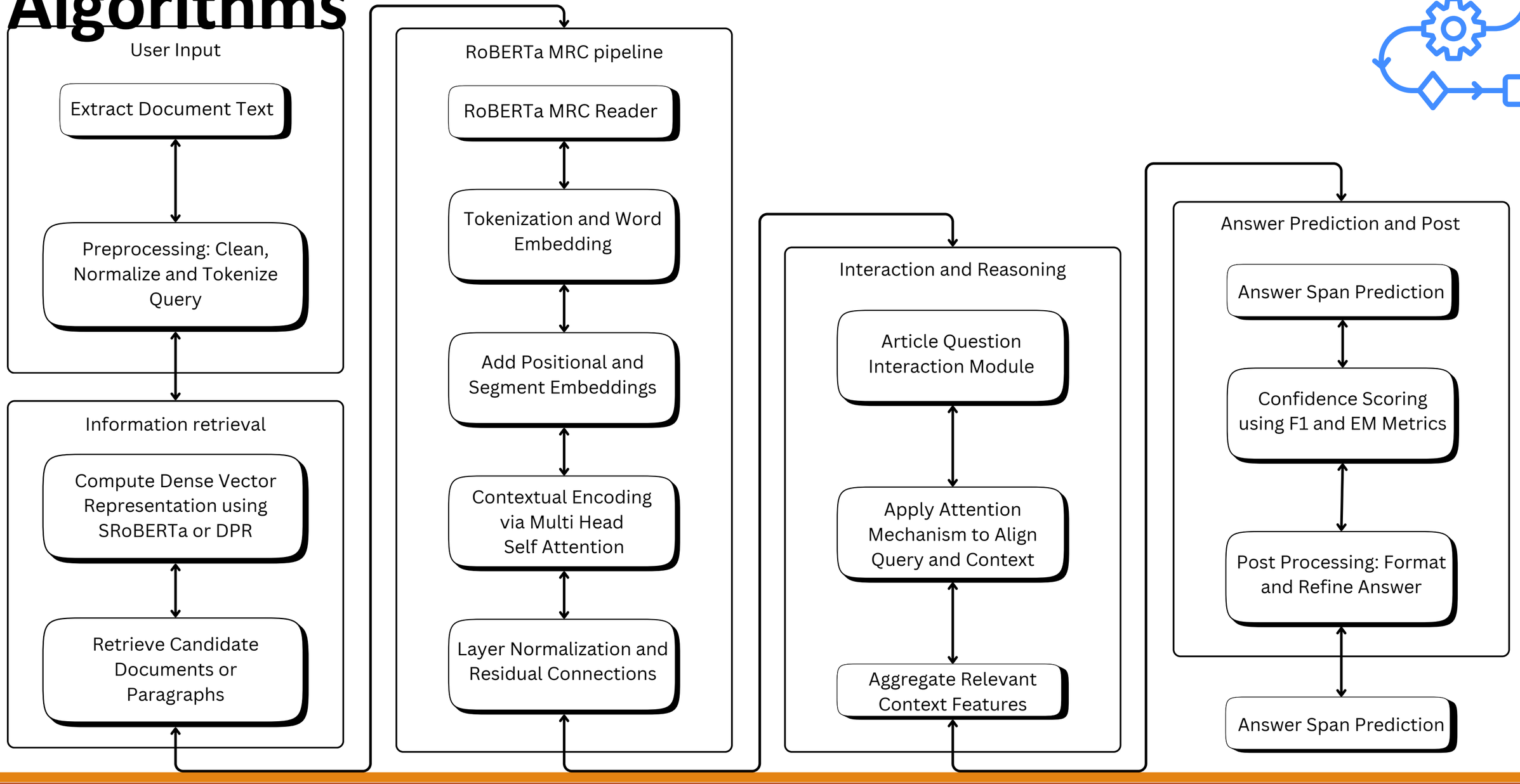


- Attention Mechanism: Focuses on the parts of the document that are most relevant to the question.
- Context Aggregation: Aggregates these relevant features to form a comprehensive context for answer prediction.

5. **Answer Prediction and Post Processing:**
   - Answer Span Prediction: The system predicts the most likely span in the text that contains the answer.
   - Confidence Scoring: Metrics such as F1 and Exact Match (EM) are used to evaluate the prediction's confidence.
   - Post Processing: The raw prediction is refined and formatted into a coherent final answer.

6. **Final Output:**
   - The final answer is output to the user.

Algorithms


User Input
Extract Document Text
Preprocessing: Clean, Normalize and Tokenize Query
Information retrieval
Compute Dense Vector Representation using SRoBERTa or DPR
Retrieve Candidate Documents or Paragraphs
RoBERTa MRC pipeline
RoBERTa MRC Reader
Tokenization and Word Embedding
Add Positional and Segment Embeddings
Contextual Encoding via Multi Head Self Attention
Layer Normalization and Residual Connections
Interaction and Reasoning
Article Question Interaction Module
Apply Attention Mechanism to Align Query and Context
Aggregate Relevant Context Features
Answer Prediction and Post
Answer Span Prediction
Confidence Scoring using F1 and EM Metrics
Post Processing: Format and Refine Answer
Answer Span Prediction

Dataset Description

1. **User-Input are Dataset ? :**
    - The files(pdf, docx, txt) that are the input in this model, which are not a dataset.
    - The input files are just there for the context of the question
2. **Dataset in Present ? :**
    - The thing is that dataset is not present in this Machine learning model, Because it uses RoBERTa model
    - Which is pre-Trained on SQuAD Dataset , in which three main feature are :
        - Context
        - Question
        - Answer

Implementation

ML project

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text

RAM Disk

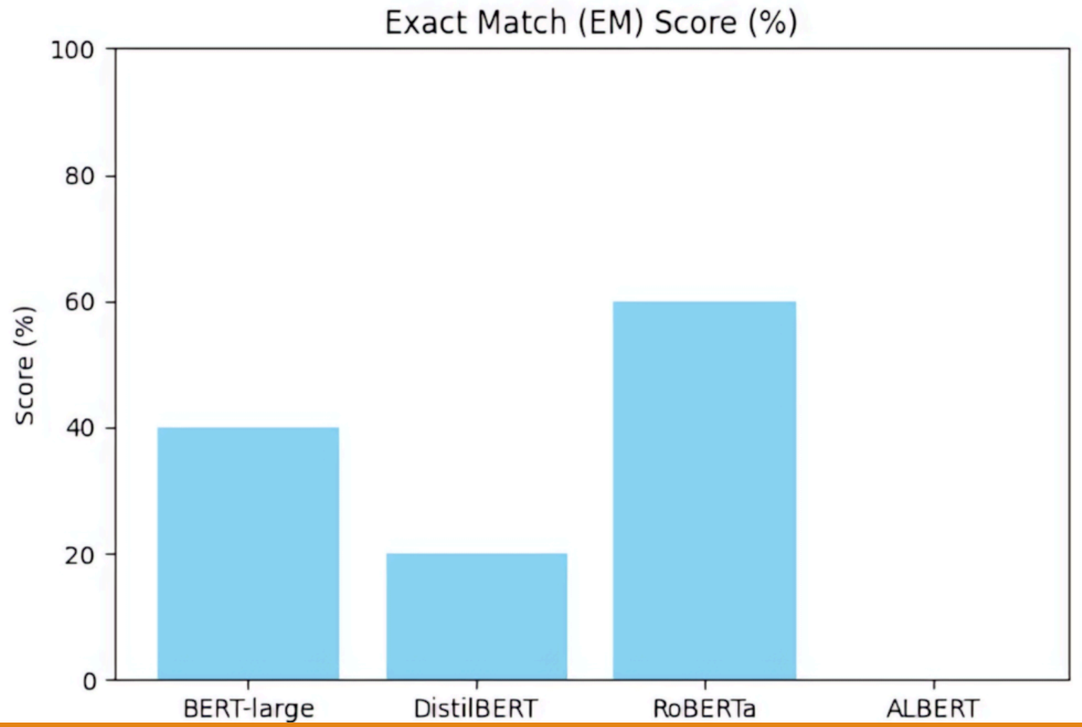
```
!pip install python-docx
!pip install PyPDF2
import PyPDF2
from transformers import pipeline, AutoModelForQuestionAnswering, AutoTokenizer
import torch
import docx
import os

def extract_text(file_path):
    file_extension = os.path.splitext(file_path)[1].lower()

    if file_extension == '.pdf':
        return extract_text_from_pdf(file_path)
    elif file_extension == '.txt':
        return extract_text_from_txt(file_path)
    elif file_extension == '.docx':
        return extract_text_from_docx(file_path)
    else:
        raise ValueError(f"Unsupported file type: {file_extension}")

def extract_text_from_pdf(file_path):
    text = ''
    try:
        with open(file_path, 'rb') as file:
            reader = PyPDF2.PdfReader(file)
            for page in reader.pages:
                page_text = page.extract_text()
                if page_text:
                    text += page_text + " "
    except Exception as e:
        print(f"Error reading PDF: {e}")
    return text.strip()

def extract_text_from_txt(file_path):
    try:
        with open(file_path, 'r', encoding='utf-8') as file:
            return file.read().strip()
    except Exception as e:
        print(f"Error reading TXT: {e}")
        return ""
```

Executing (2m 32s) <ce

```
Device set to use cpu

Bot is ready! Ask any question about the document content (type 'exit' to end).

You: name ?
Chunk 0 score: 0.0045
Bot: Low confidence answer (score: 0.0045): Deepam N.C


You: what is my name ?
Chunk 0 score: 0.3471
Bot: Deepam N.C


You: what is my education?
Chunk 0 score: 0.3943
Bot: B.Tech-IT

You: exit
Bot: Goodbye!
```

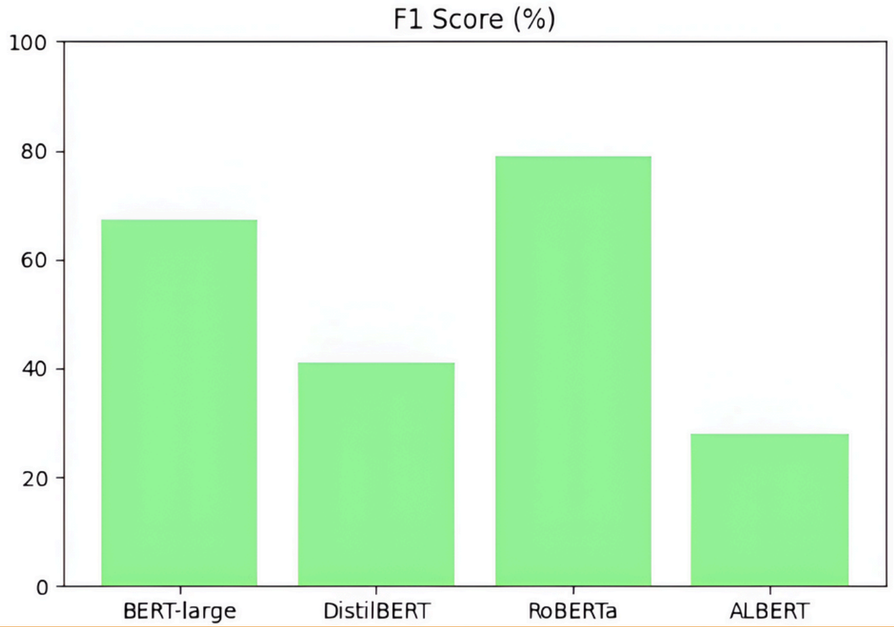
Result Analysis

1. **Exact Match(EM) Score (%):**
   - Measures how often the model's answer exactly matches the ground truth answer.
   - Higher EM means better precision in directly retrieving correct answers without any deviation.
   - RoBERTa has the highest EM score (~60%), followed by BERT-large (~40%).
   - DistilBERT (~20%) and ALBERT (0%) perform worse in this metric.



Result Analysis

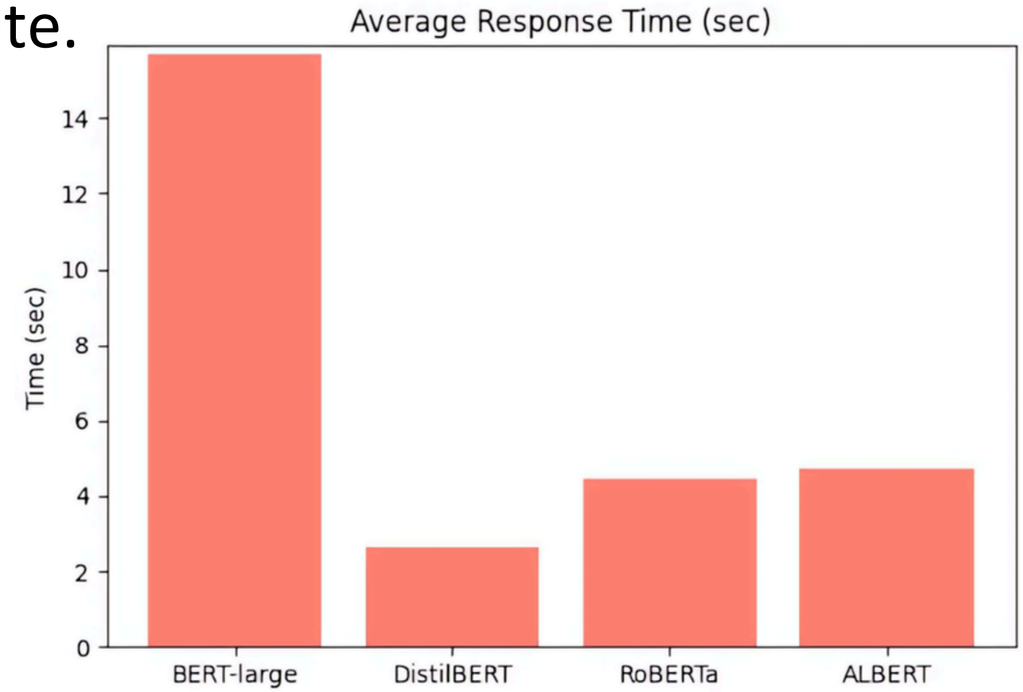
2. **F1 Score (%):**
   - The F1 score measures how much overlap exists between the model's predicted answer and the ground truth, even if they aren’t exact matches.
   - Unlike EM, this metric rewards partial correctness (e.g., missing or adding a minor word won't significantly reduce the score).
   - RoBERTa has the best F1 score (~80%), followed by BERT-large (~65%).
   - DistilBERT (~40%) and ALBERT (~30%) have lower performance.



Result Analysis

3. **Average Response Time (sec):**

- Measures how long each model takes to generate an answer.
- BERT-large is the slowest (~15 sec).
- ALBERT and RoBERTa (~5 sec) are moderate.
- DistilBERT is the fastest (~2 sec).



Conclusion

- The Conclusion we conclude for this presentation is as follows:
  - We understand the NLP(Natural Language Processing),
  - And understanding the Key concept :
    - Information retrieval
    - MRC(Machine Reading Comprehension)
    - Answer Prediction and Post

- **Key Takeaways:**
  a. Multi-format Text Extraction
  b. Transformer-based Question Answering(question-answering pipeline)
  c. Token-based Chunking Strategy
  d. Answer Scoring and Confidence
  e. Interactive Chatbot Interface (loop util exit)

Future Work

- Enhanced Parsing: Support more file types and implement advanced text cleaning.
- Model Improvements: Experiment with different models, fine-tuning, and adaptive chunking.
- Performance Optimization: Add caching, parallel processing, and better memory management.
- User Experience: Integrate conversational memory, feedback mechanisms, and a graphical interface.
- Robustness: Implement comprehensive testing, logging, and improved confidence handling.

References

Book References:

1. Dr. Kalpdrum Passi, Dr. Ratvinder Grewal, Dr. Luckny Zephyr, "RoBERTa: A Machine Reading Comprehension for Climate Change Question Answering in Natural Language Processing", Mohasina Shaikh, 2023

Web References:

1. SQuAD2.0 Dataset

   Link : https://www.kaggle.com/datasets/thedevastator/squad2-0-a-challenge-for-question-answering-syst

References

Paper References:

1. Ming Zhou, Nan Duan, Shujie Liu, Heung-Yeung Shum, “**Progress in Neural NLP: Modeling, Learning, and Reasoning**”, ELSEVIER Engineering 6, pp. 275-290, March 2020
2. Hoger K. Omar, Alaa Khalil Jumaa, “**PyTorch and TensorFlow Performance Evaluation in Big data Recommendation System**”, Information Systems Engineering Vol. 29 No. 4, pp. 1357-1364, August, 2024

THANKS OF LISTENING

“Never ending fight,
Its WITHIN, And END”