INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
NAGPUR

Department of Computer Science and Engineering

Topics in Artificial Intelligence
CSL 421

Duplicate Question Detection

Supervisor:
Dr. Nishat A. Ansari

Enrollment No. and Name of Student:-
BT21CSE194 Gyanbardhan

DUPLICATE QUESTION DETECTION

PROBLEM STATEMENT

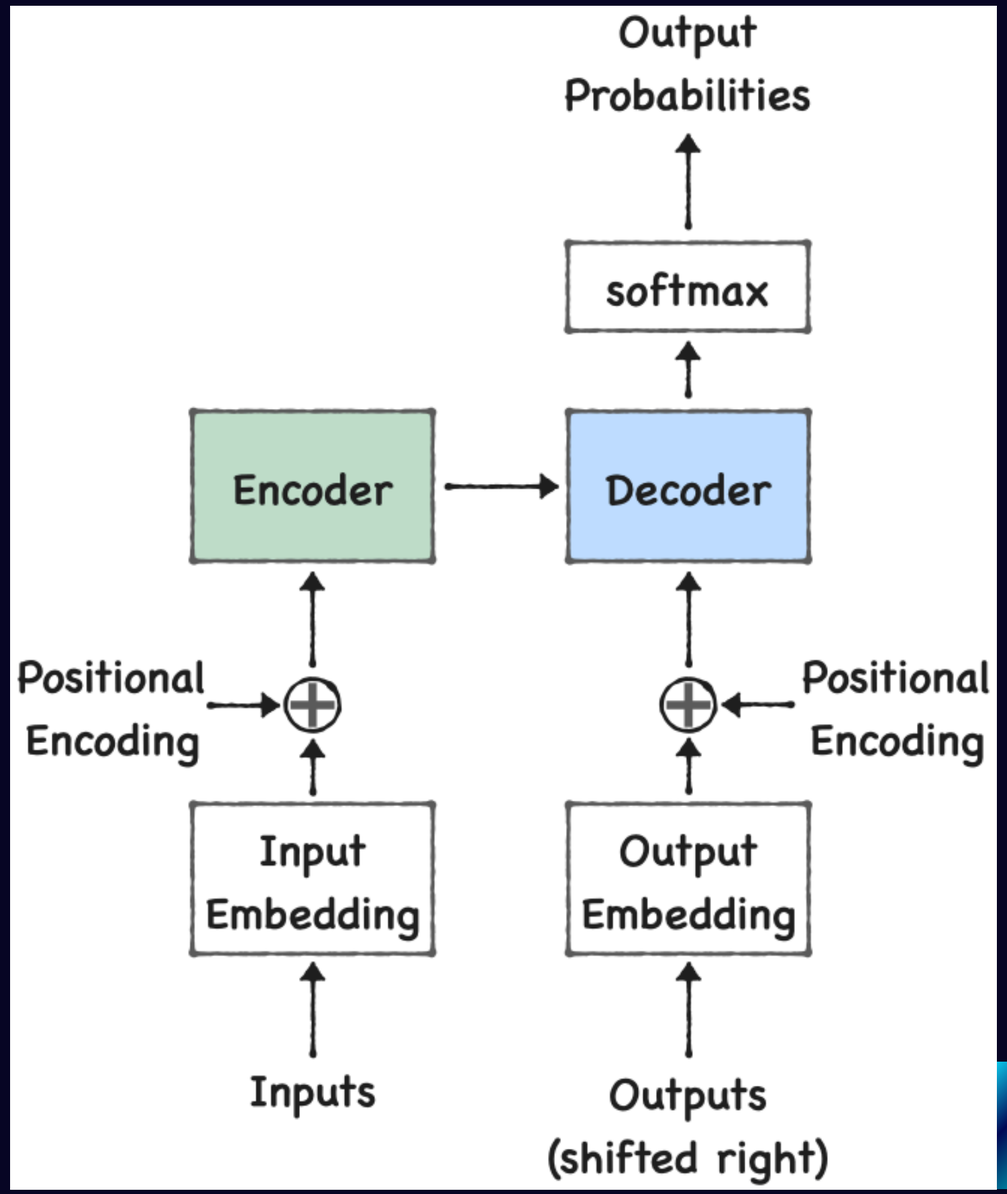
Duplicate questions are a common problem on online question-and-answer platforms, where people often ask similar or even the same questions. This repetition fills up the platform with too much of the same information, making it harder for users to quickly find useful answers. To solve this, our project uses natural language processing (NLP) and machine learning to spot and mark question pairs as either duplicates or not. This makes it easier for users to find answers without getting lost in repeated content.

TRANSFORMERS [1]

- Transformers are a type of deep learning model architecture introduced in 2017 in the paper "Attention is All You Need."
- Unlike traditional sequential models (RNNs, LSTMs), transformers process all input data at once, allowing for parallelization and faster training.

Core Idea: Attention Mechanism

- Attention helps the model focus on relevant parts of the input for each word, improving how it understands context and relationships within data.
- Transformers use self-attention to assess relationships between each word and every other word in a sentence.

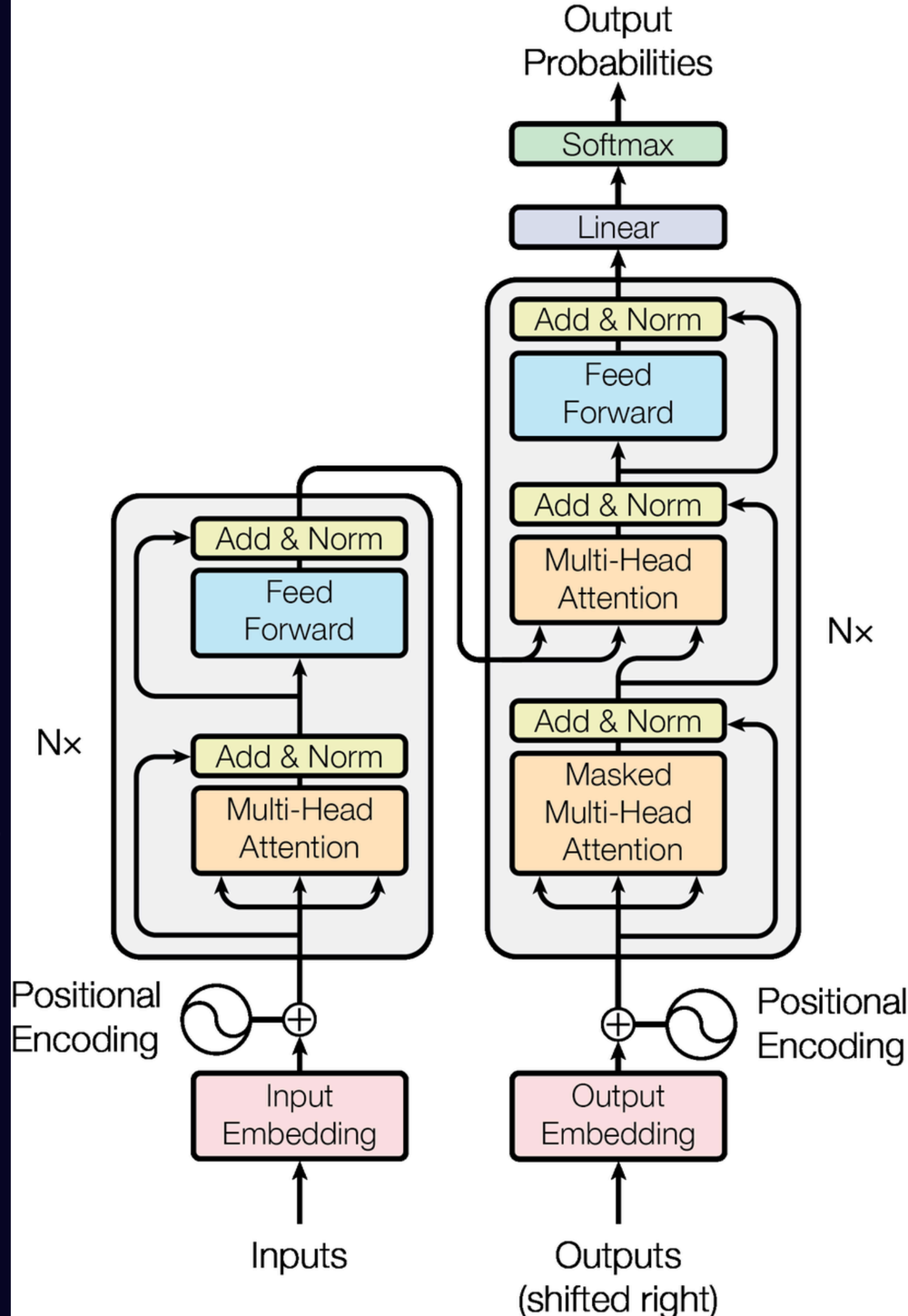


Key Components :

- Encoder-Decoder Structure: Standard transformer consists of an encoder to process the input and a decoder to generate the output (used in tasks like translation).
- Self-Attention Layers: Computes relevance of each word with respect to others, helping the model understand dependencies, even across long distances in text.
- Feed-Forward Neural Network: Adds non-linearity and complexity after the self-attention mechanism for each layer.
- Positional Encoding: Adds information about the order of words (since transformers process input as a whole rather than sequentially).

Benefits :

- Parallel Processing: Handles entire sentences at once, making it efficient and scalable for large datasets.
- Handling Long Dependencies: Better at capturing relationships in long sentences than RNNs or LSTMs.

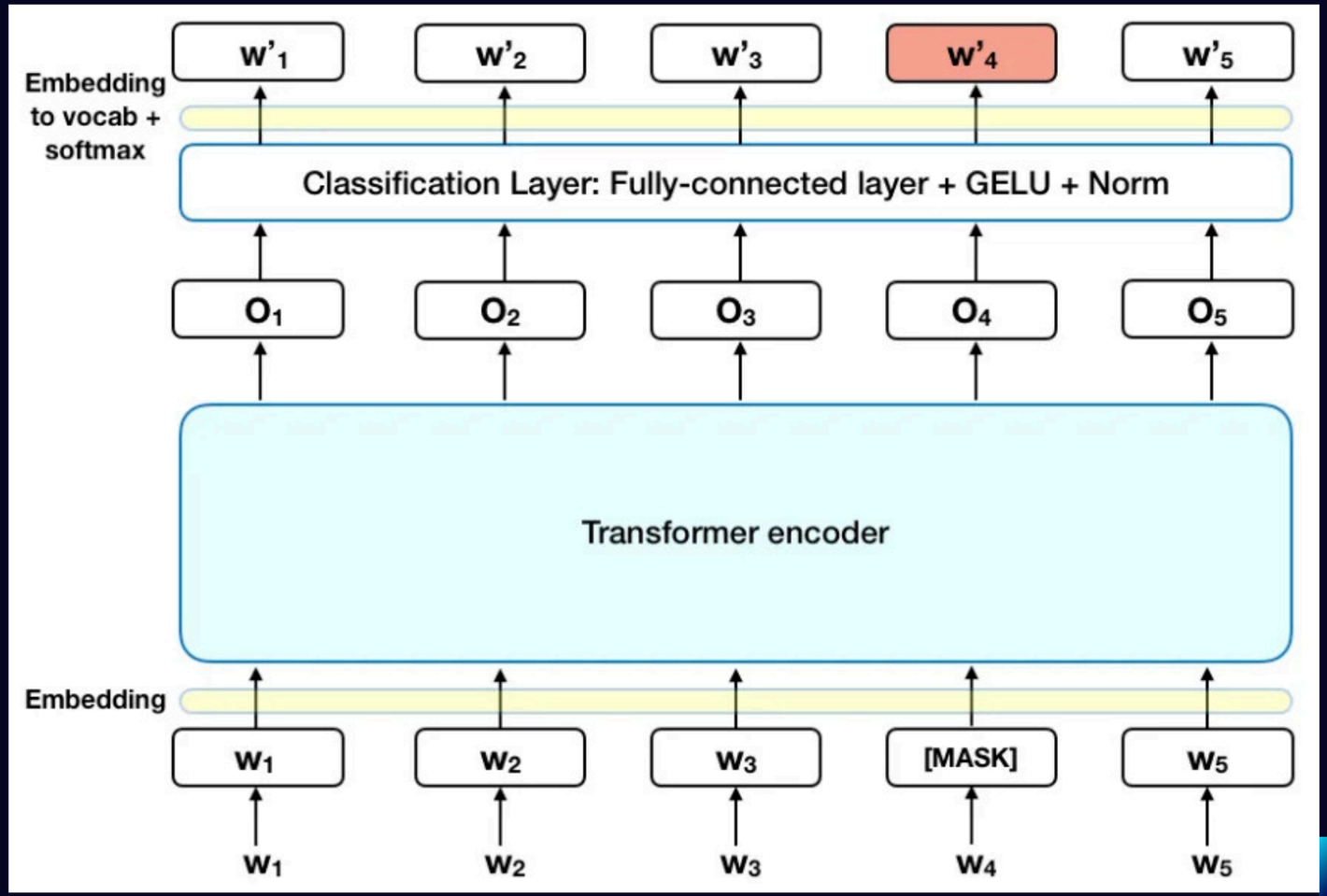


BERT [2]

- BERT (Bidirectional Encoder Representations from Transformers) is a model by Google designed to understand language context from both directions.

Key Features:

- Bidirectional Context: Reads text in both directions for a deeper understanding.
- Pre-training Tasks:
- Masked Language Model (MLM): Predicts masked words in sentences.
- Next Sentence Prediction (NSP): Learns relationships between sentence pairs.



DISTILBERT [3]

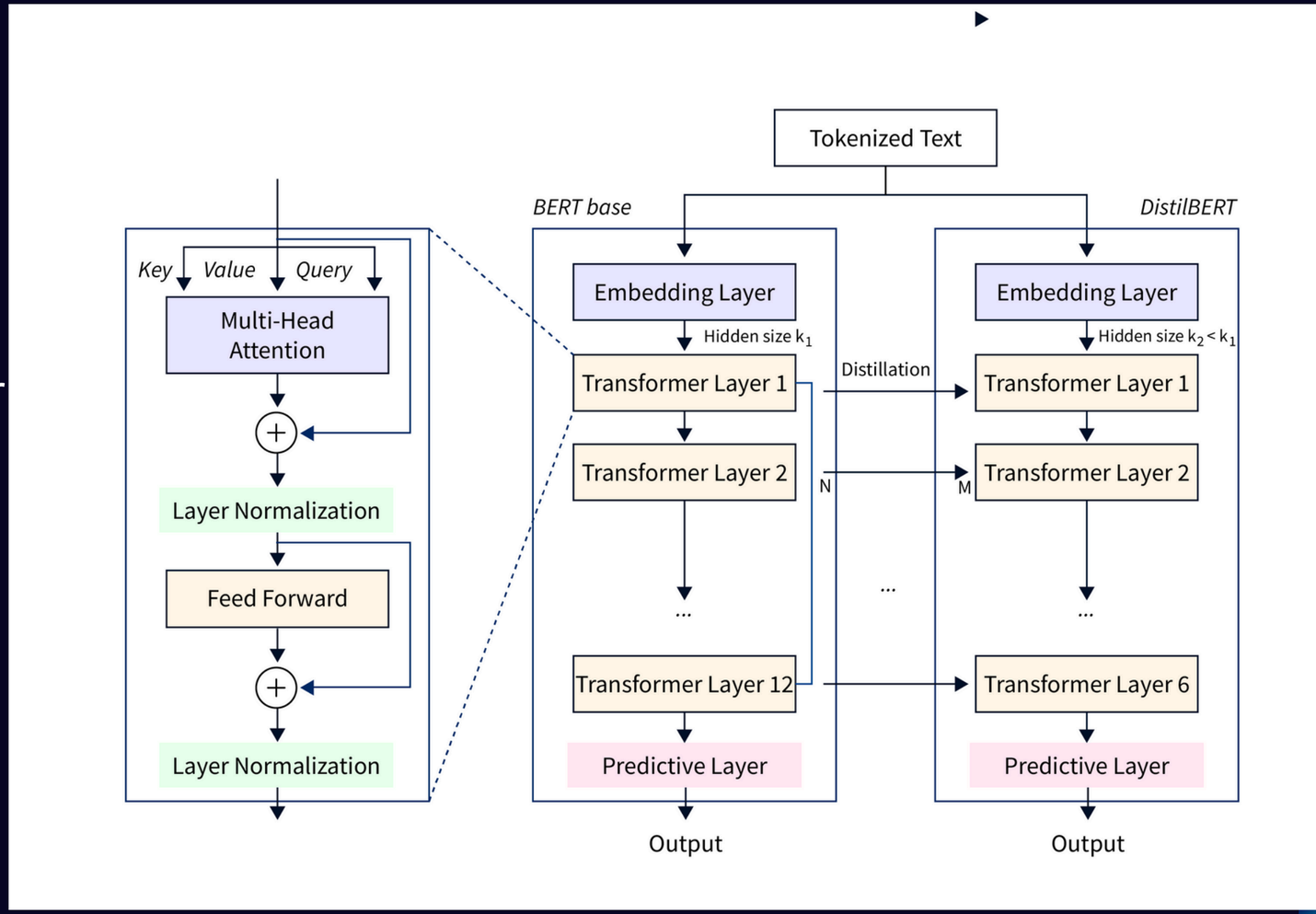
- Created by Hugging Face, DistilBERT is a compact version of BERT that is faster and requires fewer resources.

Key Features:

- Knowledge Distillation: Trained to replicate BERT's knowledge in a smaller model.
- Efficient Architecture: Fewer layers (6 vs. 12 in BERT-base) and smaller size (~60% of BERT).

Advantages:

- Faster processing, ideal for mobile apps and low-resource environments with minimal performance loss.



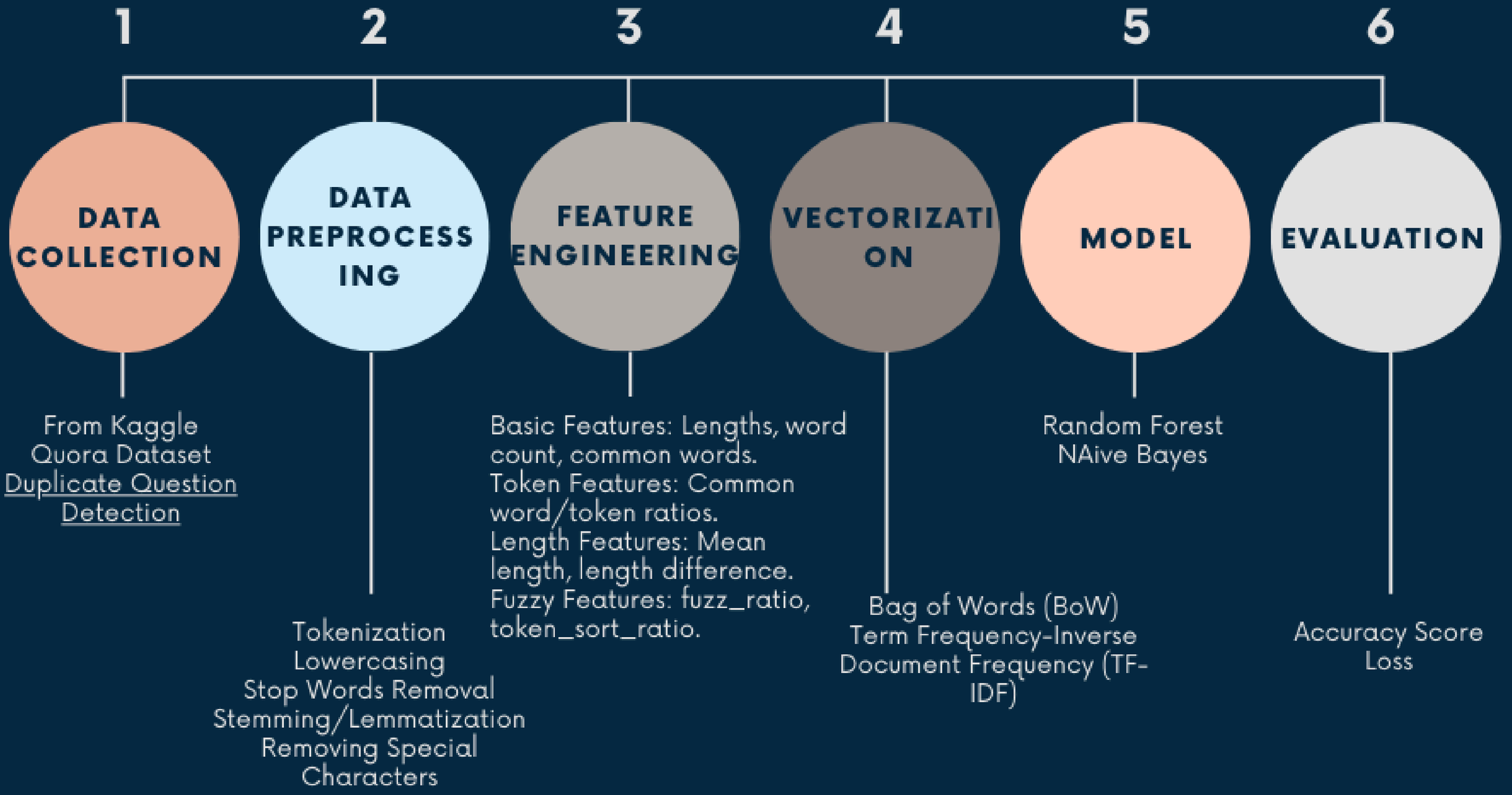
DISTILBERT TOKENIZER

- DistilBERT tokenizers convert text into tokenized inputs, allowing models to process raw language data.
- Uses *WordPiece* Tokenization, breaking words into smaller subwords for better handling of rare or unknown terms.

Key Features:

- [CLS] and [SEP] Tokens: Adds [CLS] at the start (used for classification tasks) and [SEP] to separate sentences or mark sentence boundaries.
- Padding and Truncation: Standardizes input length by adding padding or truncating longer texts, ensuring consistent batch sizes.
- Attention Masks: Adds masks to distinguish real tokens from padding, helping the model focus on relevant text.
- Optimized for faster processing with fewer parameters, ideal for lightweight applications.

APPROACH 1


1
2
3
4
5
6
DATA COLLECTION
DATA PREPROCESSING
FEATURE ENGINEERING
VECTORIZATION
MODEL
EVALUATION
From Kaggle Quora Dataset Duplicate Question Detection
Tokenization
Lowercasing
Stop Words Removal
Stemming/Lemmatization
Removing Special Characters
Basic Features: Lengths, word count, common words.
Token Features: Common word/token ratios.
Length Features: Mean length, length difference.
Fuzzy Features: fuzz_ratio, token_sort_ratio.
Bag of Words (BoW)
Term Frequency-Inverse Document Frequency (TF-IDF)
Random Forest
NAive Bayes
Accuracy Score
Loss

FEATURE ENGINEERING

Basic Features:

- q1_len, q2_len: Character length of Question 1 and Question 2.
- q1_words, q2_words: Number of words in Question 1 and Question 2.
- words_common: Number of unique common words between the two questions.
- words_total: Total number of words in both questions.
- word_share: Ratio of common words to total words.

Token Features:

- cwc_min/cwc_max: Ratio of common words to the length of the smaller or larger question.
- csc_min/csc_max: Ratio of common stop words to the smaller or larger stop word count.
- ctc_min/ctc_max: Ratio of common tokens to the smaller or larger token count.
- last_word_eq: 1 if last words are the same, 0 otherwise.
- first_word_eq: 1 if first words are the same, 0 otherwise.

Length-Based Features:

- mean_len: Average length of the two questions (in words).
- abs_len_diff: Absolute difference in the length (word count) of the two questions.
- longest_substr_ratio: Ratio of the longest common substring length to the length of the smaller question.

Fuzzy Features:

- fuzz_ratio: Overall similarity score between the two questions (from fuzzywuzzy).
- fuzz_partial_ratio: Partial matching score from fuzzywuzzy.
- token_sort_ratio: Similarity score after sorting the tokens (from fuzzywuzzy).
- token_set_ratio: Similarity score based on the token set (from fuzzywuzzy).

MODELS

Bag of Words (BoW):

- A simple method where text is represented as a matrix of token counts. Each row represents a question pair, and each column corresponds to a unique token in the vocabulary.

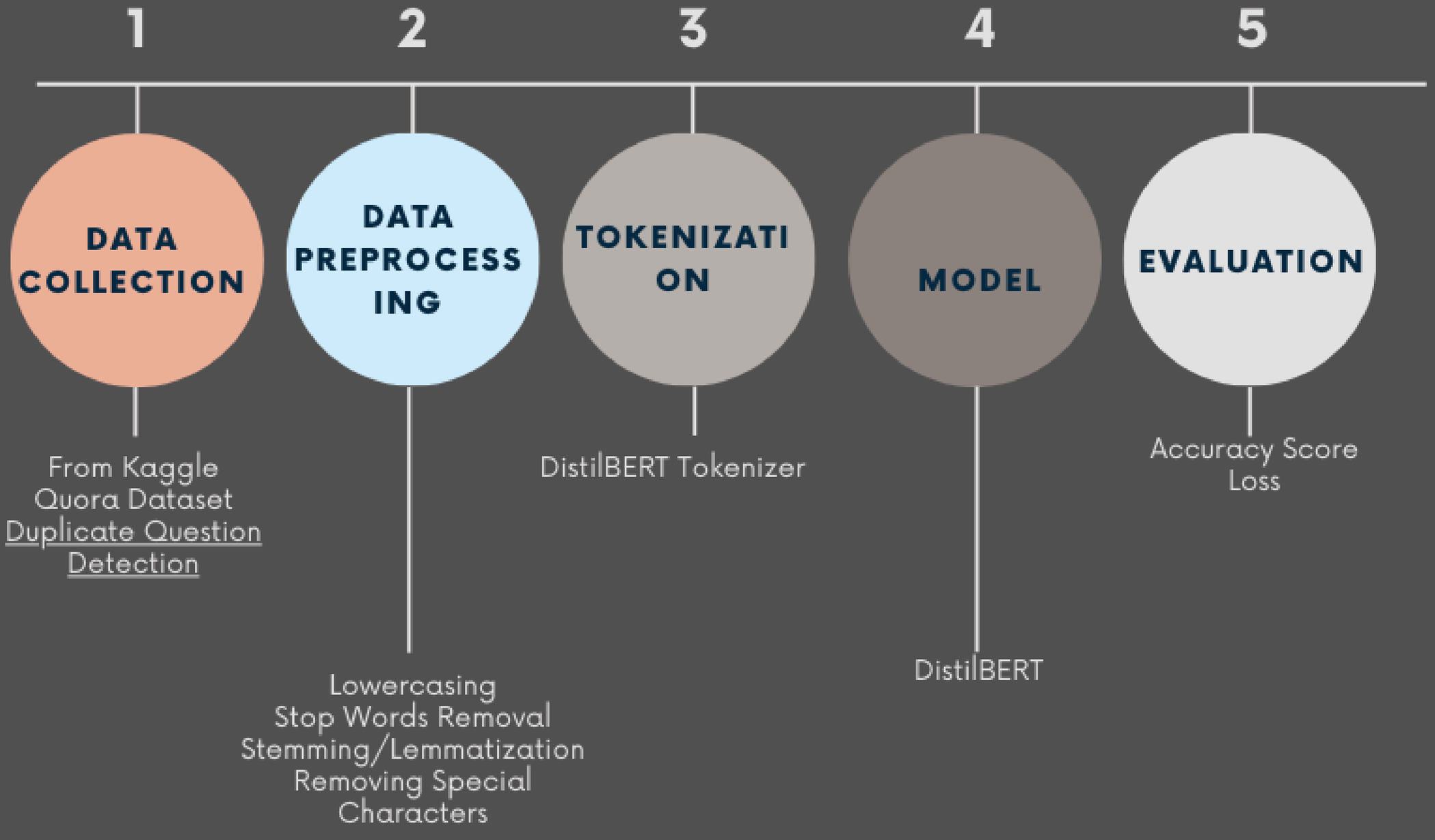
Term Frequency-Inverse Document Frequency (TF-IDF):

- A more advanced technique that adjusts the importance of words by considering their frequency in the entire corpus. It gives higher weight to rare but important words and reduces the weight of common terms.

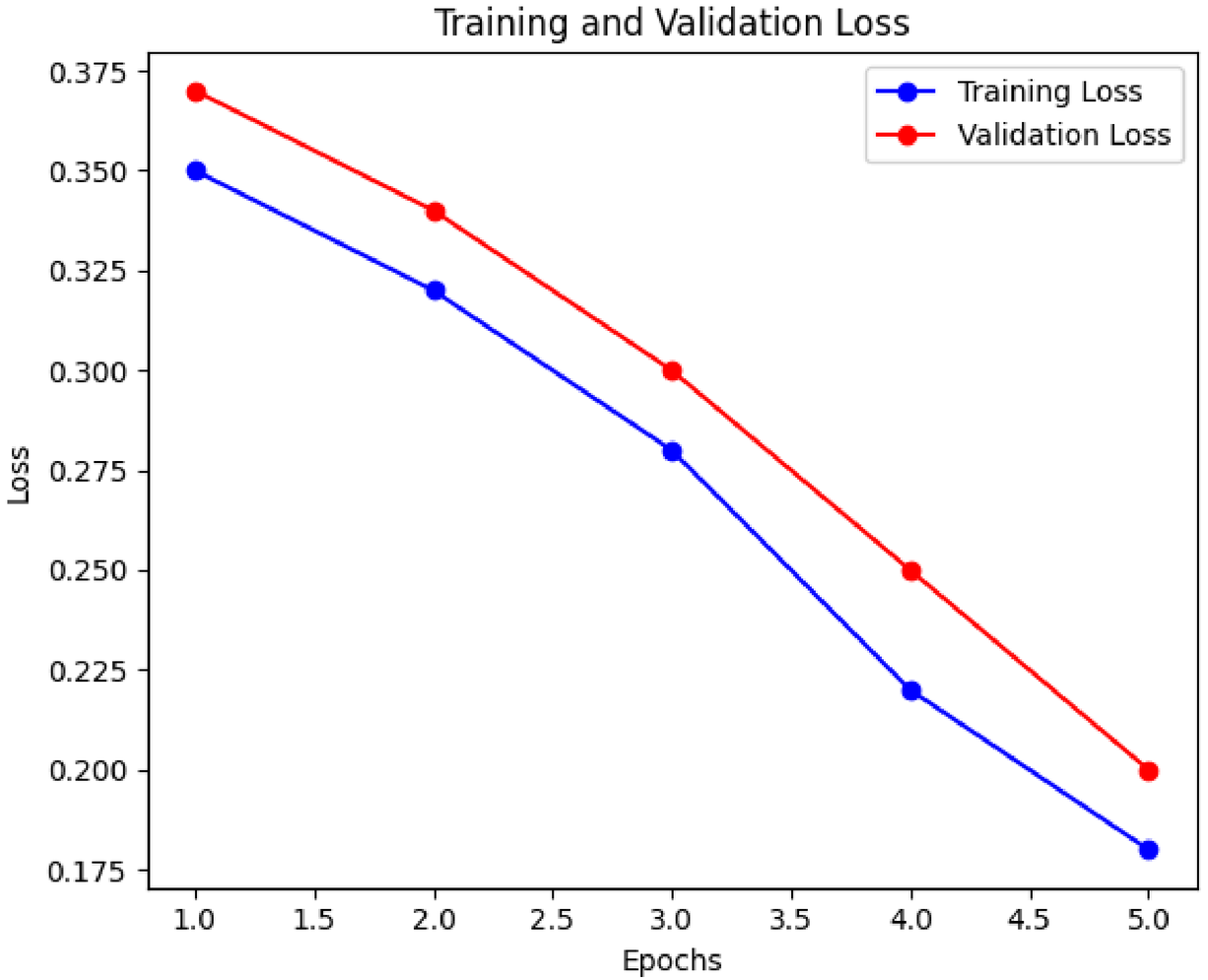
Random Forest Classifier:

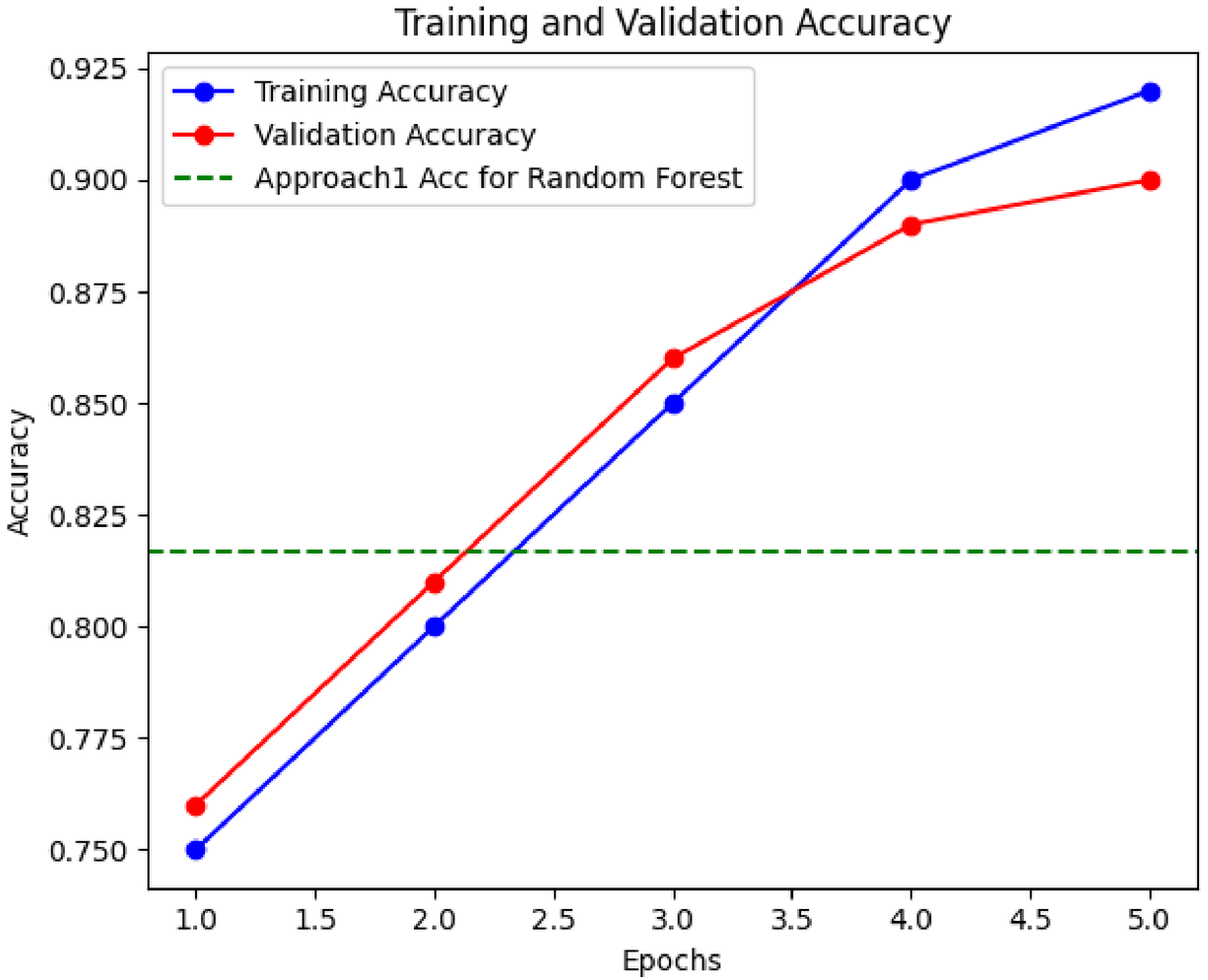
- Used to classify whether a pair of questions is a duplicate or not, based on the features generated through the above techniques.

APPROACH 2

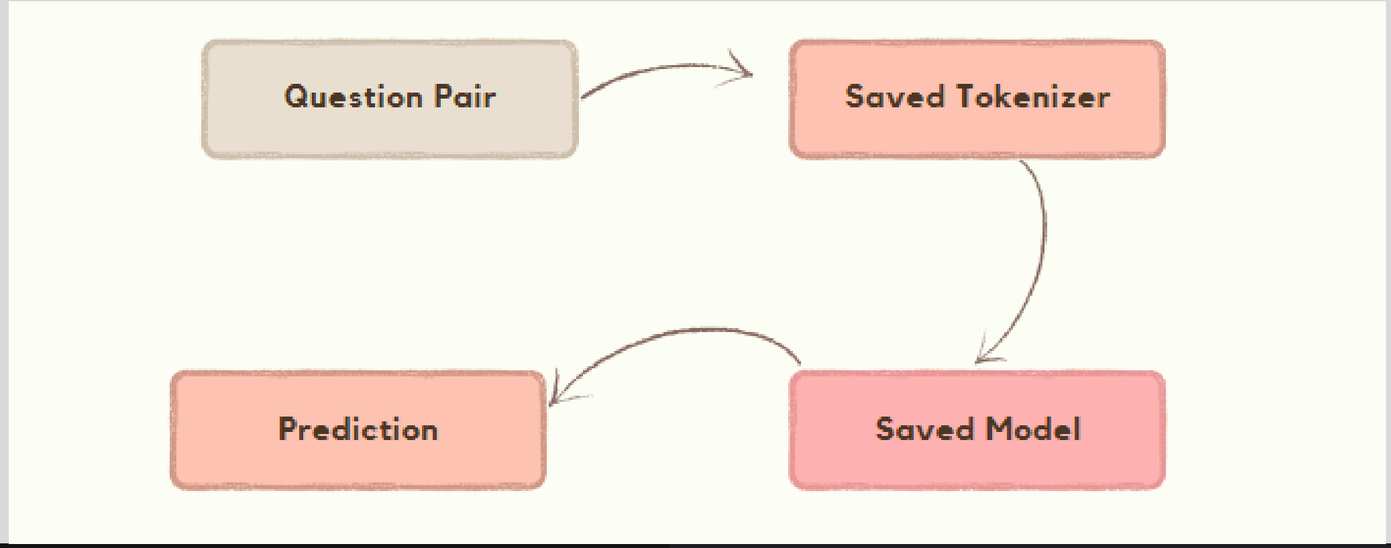

1
2
3
4
5
DATA COLLECTION
DATA PREPROCESSING
TOKENIZATION
MODEL
EVALUATION
From Kaggle
Quora Dataset
Duplicate Question
Detection
Lowercasing
Stop Words Removal
Stemming/Lemmatization
Removing Special
Characters
DistilBERT Tokenizer
DistilBERT
Accuracy Score
Loss

RESULTS/EVALUATION





OUTPUT




```python
question_pairs = [
    ("How do I reset my password?", "What is the process to reset my password?"),
    ("Where can I find my order history?", "How do I view my past orders?"),
    ("What is the return policy for online purchases?", "How can I return an item bought online?"),
    ("Can I change my delivery address after ordering?", "Is it possible to update the shipping address once the order is
    ("How do I contact customer support?", "What's the best way to reach customer service?"),
    ("How do I reset my password?", "How do I delete my account?"),
    ("Where can I find my order history?", "What are the delivery options for my area?"),
    ("What is the return policy for online purchases?", "How long does it take to get a refund after returning an item?"),
    ("Can I change my delivery address after ordering?", "How do I apply a discount code to my order?"),
    ("How do I contact customer support?", "What payment methods are accepted?")
]
```

```python
inputs = tokenizer(
    [q[0] for q in question_pairs], [q[1] for q in question_pairs],
    return_tensors='tf',
    truncation=True,
    padding=True,
    max_length=50
)
```

```python
outputs = model(inputs)
logits = outputs.logits

# Convert logits to probabilities
probabilities = tf.nn.softmax(logits, axis=-1)
predictions = tf.argmax(probabilities, axis=1).numpy()  # 0 or 1 for binary classification
```

```
Question 1: How do I reset my password?
Question 2: What is the process to reset my password?
Prediction: Duplicate
Probability: [0.12865898 0.871341  ]
------
Question 1: Where can I find my order history?
Question 2: How do I view my past orders?
Prediction: Not Duplicate
Probability: [9.9971086e-01 2.8908864e-04]
------
Question 1: What is the return policy for online purchases?
Question 2: How can I return an item bought online?
Prediction: Not Duplicate
Probability: [9.9984324e-01 1.5668685e-04]
------
Question 1: Can I change my delivery address after ordering?
Question 2: Is it possible to update the shipping address once the order is placed
Prediction: Not Duplicate
Probability: [0.9857459  0.01425405]
------
```

WEB LINK

APPROACH 1

APPROACH 2

Duplicate Question

Enter Question1

How do I reset my password?

Enter Question2

What is the process to reset my password?

Find

Duplicate

Clear

Duplicate Question Detection

Enter the first question:

How do I reset my password?

Enter the second question:

What is the process to reset my password?

Predict

Prediction: Duplicate

Probability: Not Duplicate 0.1286584734916687 Duplicate 0.8713415265083313

CONCLUSION

This project addresses the issue of duplicate questions on online platforms by using natural language processing and machine learning to quickly identify and organize similar questions. This improves platform efficiency, reduces repetitive content, and helps users find answers more easily.

We compared two models for detecting duplicate questions:

- BOW/TF-IDF with Random Forest Classifier: Achieved 81.67% accuracy but lacks a deep understanding of context.
- DistilBERT Transformer: Reached a higher 89.89% accuracy, capturing better context and meaning in questions.

DistilBERT outperformed the traditional model by over 8%, making it the preferred choice due to its ability to understand question context more effectively, resulting in a more accurate and reliable solution.

REFERENCES/RESOURCES

- [1]. Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, Lukasz, Polosukhin, Illia. "Attention Is All You Need." arXiv, vol. 1706.03762, 12 Jun. 2017, revised 2 Aug. 2023, https://doi.org/10.48550/arXiv.1706.03762.
- [2]. Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, Toutanova, Kristina. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv, vol. 1810.04805, 11 Oct. 2018, revised 24 May 2019, https://doi.org/10.48550/arXiv.1810.04805.
- [3]. Sanh, Victor, Debut, Lysandre, Chaumond, Julien, Wolf, Thomas. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." arXiv, vol. 1910.01108, 2 Oct. 2019, revised 1 Mar. 2020, https://doi.org/10.48550/arXiv.1910.01108.
- Approach 1 Notebook - BOW - TF/IDF - web link
- Approach 2 Notebook - DistilBERT - web link
- Duplicate Question Detection Quora Dataset

THANK YOU