SHIV NADAR UNIVERSITY
CHENNAI

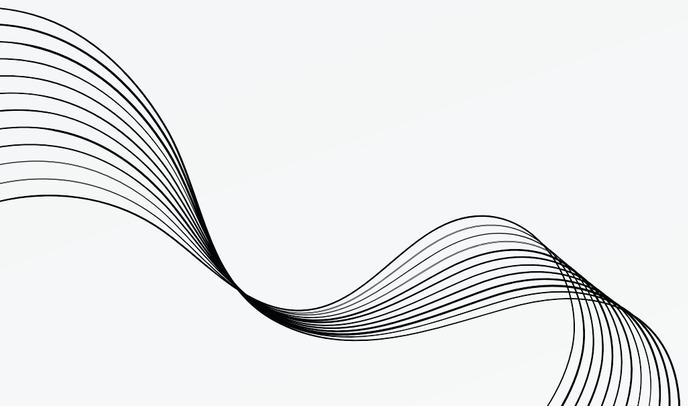
Natural Language Processing

TEXT GENERATION

Nakshatiraa K N
Srikiruthika S

PROBLEM STATEMENT

The objective is to develop a Natural Language project centered around text generation and prediction. The project seeks to showcase a variety of techniques, including Markov chain analysis, n-gram models, and the generation of text based on recurrent patterns observed within a provided text dataset.



PROJECT PIPELINE

DATA PREPROCESSING

Sourcing from repositories like Project Gutenbergas our dataset, we obtain a diverse text corpus. Tokenization breaks the text into words or tokens. Irrelevant elements such as stopwords and punctuation are removed, and text is normalized to lowercase.

These steps ensure a clean dataset for subsequent analysis and model training.

MARKOV CHAIN ANALYSIS

We implement a Markov chain model to analyze the text dataset, focusing on identifying patterns of word sequences and their probabilities of occurrence. By comprehending these patterns, the model gains insights into the structure and style of the original dataset, enabling the generation of new text that closely resembles its characteristics.

This involves constructing a transition matrix, which captures the likelihood of transitioning between words based on observed sequences in the dataset. Subsequently, the model leverages this matrix to probabilistically generate coherent and contextually relevant text that mirrors the style and structure of the input dataset.

N-GRAM MODEL CONSTRUCTION

We construct n-gram models, including unigrams, bigrams, trigrams, etc., to predict subsequent words in a sentence based on preceding words. These models are trained on the preprocessed text dataset, wherein probabilities of word sequences are calculated and stored.

By analyzing the frequency of word co-occurrences, the models learn contextual relationships and generate text with coherence and relevance.

COMPARING THE MODELS

- For our project using the "A Girl of High Adventure" dataset, both Markov chain and n-gram models have their strengths and limitations.
- Markov chains may offer simplicity and efficiency in modeling basic linguistic patterns, but they may struggle with capturing complex narrative structures and character interactions.
- N-gram models, while more computationally intensive, have the potential to provide more accurate and contextually relevant text generation by considering longer sequences of words.
- Ultimately, the choice between Markov chain and n-gram models depends on the specific requirements of our project, including the desired level of accuracy, computational resources available, and the complexity of linguistic patterns present in the dataset.

OUTPUT

MARKOV CHAIN ANALYSIS

In [18]:
```
if __name__ == '__main__':
    url = 'http://www.gutenberg.org/cache/epub/63632/pg63632.txt'
    words = gatherBook(url)

    cleansedWords = [cleanse(word) for word in words]

    print(predict_sentence(cleansedWords, 'he said', 10))
    # this will predict an 11-word sentence (first input word, plus 10 suffixes)
```

```
he said slowly this is the result it was now nothing
```

OUTPUT

N-GRAM MODEL

In [57]:

```
# Step 4: Use the n-gram model to generate next words
if __name__ == "__main__":
    # Fetch data from URL
    url = 'http://www.gutenberg.org/cache/epub/61995/pg61995.txt'
    with urllib.request.urlopen(url) as response:
        text = response.read().decode('utf-8')

    # Preprocess text
    tokens = preprocess_text(text)

    # Build n-gram model
    ngram_model = build_ngram_model(tokens, n=2)  # Adjust n-gram size as needed

    # Generate next 5 words for a given word
    input_word = 'the'
    next_words = generate_next_words(ngram_model, input_word, num_words=5)
    print(f"Next 5 words for '{input_word}': {next_words}")
```

```
Next 5 words for 'the': ['use', 'of', 'a', 'girl', 'of']
```

RESULT

After implementing both the Markov chain and n-gram models for text generation, the n-gram model emerged as the superior approach in our project.

- Markov Model Result: The Markov model generated a predicted sentence, based on the prefix 'he said'. While the sentence exhibits some coherence, it lacks fluency and context.
- N-gram Model Result: In contrast, the n-gram model produced a more coherent and contextually relevant text snippet, based on the seed text 'The'. This generated text demonstrates a higher degree of fluency and relevance compared to the Markov model output.

The project's outcome highlights the effectiveness of the **n-gram model** in generating coherent and contextually relevant text. Its ability to consider sequences of words and capture contextual information leads to superior text generation compared to the Markov chain approach. Therefore, for our project's objectives, the n-gram model stands out as the preferred algorithm for text generation tasks.

.