Lab 6

Q Develop a Python program that can:

1. Read a long article, report, or paragraph as input.

2. Generate summary:

o Extractive Summary: Using an algorithmic or statistical method.

Code:-

```
import nltk
# Ensure required resources are available
for pkg in ['punkt', 'punkt_tab', 'stopwords']:
    nltk.download(pkg, quiet=True)

nltk.download('punkt')
nltk.download('stopwords')
"""
Extractive Text Summarizer
-------------------------
Approach: TextRank (graph-based algorithm)
Libraries: NLTK, NetworkX
"""
import numpy as np
import networkx as nx
from nltk.corpus import stopwords
from nltk.tokenize import sent_tokenize, word_tokenize

# Download required resources once
nltk.download('punkt')
nltk.download('stopwords')
```

```
# --------------------------
# Preprocessing functions
# --------------------------
def preprocess_text(text):
    sentences = sent_tokenize(text)
    return [s.strip() for s in sentences if len(s.strip()) > 0]

def sentence_similarity(sent1, sent2):
    stop_words = set(stopwords.words('english'))
    words1 = [w.lower() for w in word_tokenize(sent1) if w.isalpha() and w.lower() not in stop_words]
    words2 = [w.lower() for w in word_tokenize(sent2) if w.isalpha() and w.lower() not in stop_words]
    all_words = list(set(words1 + words2))
    v1 = [0] * len(all_words)
    v2 = [0] * len(all_words)
    for w in words1:
        v1[all_words.index(w)] += 1
    for w in words2:
        v2[all_words.index(w)] += 1
    # Cosine similarity
    num = np.dot(v1, v2)
    den = np.sqrt(sum(np.square(v1))) * np.sqrt(sum(np.square(v2)))
    return 0 if den == 0 else num / den

# --------------------------
# TextRank Algorithm
# --------------------------
def textrank_summarizer(text, top_n=3):
    sentences = preprocess_text(text)
    n = len(sentences)
    if n == 0:
        return ""
```

```
    if n <= top_n:
        return text

    # Build similarity matrix
    sim_matrix = np.zeros((n, n))
    for i in range(n):
        for j in range(n):
            if i != j:
                sim_matrix[i][j] = sentence_similarity(sentences[i], sentences[j])

    # Apply PageRank on the similarity graph
    nx_graph = nx.from_numpy_array(sim_matrix)
    scores = nx.pagerank(nx_graph)

    # Rank sentences and pick top N
    ranked_sentences = sorted(((scores[i], s) for i, s in enumerate(sentences)), reverse=True)
    selected = [s for _, s in ranked_sentences[:top_n]]
    summary = " ".join(selected)
    return summary

# --------------------------
# Main
# --------------------------
if __name__ == "__main__":
    print("\n=== Extractive Text Summarizer (TextRank) ===\n")
    text = input("Enter a long paragraph or article:\n\n")
    summary = textrank_summarizer(text, top_n=3)
    print("\n--- Original Text ---")
    print(text)
    print("\n--- Extractive Summary ---")
    print(summary)
```

```
--- Original Text ---
Artificial intelligence (AI) is transforming industries by enabling machines to perform tasks that typica
lly require human intelligence. AI is used in applications like image recognition, speech processing, aut
onomous vehicles, and healthcare. The rapid progress in deep learning has made AI more capable than ever
before.

--- Extractive Summary ---
Artificial intelligence (AI) is transforming industries by enabling machines to perform tasks that typica
lly require human intelligence. AI is used in applications like image recognition, speech processing, aut
onomous vehicles, and healthcare. The rapid progress in deep learning has made AI more capable than ever
before.
```