Assignment 3 Set A1

March 23, 2024

1 Consider any text paragraph. Preprocess the text to remove any special characters and digits. Generate the summary using extractive summarization process.

[3]:

```
import nltk
import re #for Regular Expression

text ="""Hello all, Welcome to Python Programming Academy. Python Programming␣
 ↪Academy is a nice platform to learn
new programming skills. It is difficult to get enrolled in this Academy 1."""

text = re.sub(r'[[0-9]*]', ' ', text)
text = re.sub(r'\s+', ' ', text)
formatted_text = re.sub('[^a-zA-Z]', ' ', text)
print(formatted_text)
```

```
Hello all  Welcome to Python Programming Academy  Python Programming Academy is
a nice platform to learn new programming skills  It is difficult to get enrolled
in this Academy
```

[4]:

```
from nltk.tokenize import word_tokenize
# Passing the string text into word tokenize for breaking the sentences and␣
 ↪generating tokens
token = word_tokenize(text)
token
```

[4]:

```
['Hello',
 'all',
 ',',
 'Welcome',
 'to',
 'Python',
 'Programming',
 'Academy',
 '.',
 'Python',
 'Programming',
 'Academy',
 'is',
 'a',
 'nice',
 'platform',
 'to',
 'learn',
 'new',
 'programming',
 'skills',
 '.',
 'It',
 'is',
 'difficult',
 'to',
 'get',
 'enrolled',
 'in',
 'this',
 'Academy',
 '1',
 '.']
```

[5]:

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
stopWords = set(stopwords.words("english"))
words = word_tokenize(formatted_text)
```

[6]:

```
# Creating a frequency table of words
wordfreq = {}
for word in words:
    if word in stopWords:
        continue
    if word in wordfreq:
        wordfreq[word] += 1
    else:
        wordfreq[word] = 1

#Compute the weighted frequencies
maximum_frequency = max(wordfreq.values())
for word in wordfreq.keys():
    wordfreq[word] = (wordfreq[word]/maximum_frequency)

# Creating a dictionary to keep the score # of each sentence
sentences = sent_tokenize(text)
sentenceValue = {}
for sentence in sentences:
    for word, freq in wordfreq.items():
        if word in sentence.lower():
            if sentence in sentenceValue:
                sentenceValue[sentence] += freq
            else:
                sentenceValue[sentence] = freq

import heapq
summary = ''
summary_sentences = heapq.nlargest(5, sentenceValue, key=sentenceValue.get)
summary = ' '.join(summary_sentences)
print(summary_sentences)
print(summary)
```

```
['Python Programming Academy is a nice platform to learn new programming
skills.', 'It is difficult to get enrolled in this Academy 1.', 'Hello all,
Welcome to Python Programming Academy.']
Python Programming Academy is a nice platform to learn new programming skills.
It is difficult to get enrolled in this Academy 1. Hello all, Welcome to Python
Programming Academy.
```

[ ]: