

Experiment - 9

6/4/23

Implementation of NLP programs.

Aim: To implement NLP algorithms.

Algorithm: ① Preprocess the text:

Remove punctuation, stopwords and other irrelevancies from the sample text.

② Tokenize the text into words and sentences and performing any necessary stemming and lemmatization.

③ Calculate importance ~~as~~ with TF-IDF on text Rank.

④ create summary: concatenate the selected sentences to create summary.

⑤ check for coherence.

⑥ Evaluate the ~~system~~ summary: Use evaluation metrics to calculate result and measure accuracy.

(7) Refine the algorithm and analyze the evaluation results and refine the algorithm as necessary to improve the quality of result.

result: Thus, we have successfully implemented and executed as NLP programs.

10
on

```
import nltk
from nltk.tokenize import word_tokenize, sent_tokenize
from nltk.corpus import stopwords
from collections import Counter

# Sample text to process
text = "The quick brown fox jumped over the lazy dog. The dog slept well."

# Tokenize the text into words and sentences
words = word_tokenize(text)
sentences = sent_tokenize(text)

# Remove stop words
stop_words = set(stopwords.words('english'))
filtered_words = [word for word in words if word.casefold() not in stop_words]

# Count word frequency
word_counts = Counter(filtered_words)
most_common_words = word_counts.most_common(5)

# Print results
print("Original text:\n", text)
print("\nTokenized words:\n", words)
print("\nTokenized sentences:\n", sentences)
print("\nFiltered words:\n", filtered_words)
print("\nMost common words:\n", most_common_words)
```

Manual Calculation / Output :

Input text :

input taken from user [code]

Output -

summary :

i/p length : 178

o/p length : 63