

Assignment No :- 1

Aim :- Write a programme for pre-processing of a text documents such as stop removal stemming

Outcome :-

1) Understand and implement basic text preprocessing technique.

2) Remove stop words from a document

3) Apply stemming to words using NLP libraries.

Hardware Requirements :-

1) Processor - Intel i3/i5 or above

2) RAM :- Minimum 4GB

3) Storage :- 2GB of free space

4) OS :- Window / Linux / os.

Software Requirements :-

1) Python 3x

2) Jupyter Notebook

3) NLTK Library

4) Internet Connection

Theory :-

Text Preprocessing in NLP - Detailed description.

1) Tokenization

It is a process of breaking down a large chunk of text into smaller units called tokens. These tokens can be words, characters or sub words.

Tokenization :-

1) Word Tokenization - split text into words.

2) Sentence Tokenization - split text into sentences.

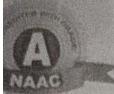
• Importance :-

- Tokenization is a basic step for NLP operation like parsing, POS tagging, named entity recognition.

- Helps in feature extraction in ML models.

2. Stop word Removal :-

Stop words are commonly used



words in a language that are filtered out before passing because they usually don't carry significant meaning.

3. Stemming :-

It is the process of reducing a word to its base root from by removing prefixes or suffixes.

e.g - "running", "runs", "ran" → "run"

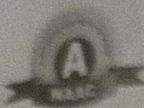
"studies", → "study"

Popular stemming Algorithms :-

- Porter stemmer - commonly used, rule based stemmer.
- snowball stemmer - An improvement over the porter stemmer.
- Lancaster stemmer :- more aggressive than Porter.

~~Advantages :-~~

- Reduces the size of data for processing.
- Helps focus on meaningful words.
- Improves model performance by reducing size.



Dis-Advantage:
stemming may result in non-distinct words.

- stop word in same contexts
- Language-dependent - each language needs its own preprocessing.

Conclusion:-

Hence we conclude, Text pre-processing is a critical step in NLP, like stop words removal, stemming help in cleaning & standardizing data making it ready for machine learning models.

Coding efficiency	Niva	Timely	Total	Sign.
5	3	2	10	
4	2	2	8	Balance

Assignment No:- 2

- Aim :- To implement a program that retrieves documents based on keyword queries using the inverted file indexing technique.
- Outcome :-
 - 1) Understand and implement inverted indexing
 - 2) Store and retrieve document efficiently based on keywords.
 - 3) Apply the concept in search engines and information retrieval system.

Hardware Requirement :-

- 1) Processor - Intel i3 / i5 / i7 or its equivalent
- 2) RAM - minimum 4 GB.
- 3) Disk space - Minimum 1 GB free.
- 4) Keyboard & Monitor.

Software Requirement :-

- 1) Operating System - Windows / macOS / Linux.
- 2) Programming language - Python 3.6.
- 3) IDE :- VS code / pycharm
- 4) Libraries :- No external libraries required.

Theory :- Inverted file indexing is a method of storing mapping from content to their location in a set of documents. It is backbone of modern search engines.

• steps Involved :-

1) Tokenization - split documents into individual words.

2) Stop words removal - Remove common words.

3) Inverted Index Construction - Map word to the document TD's which it occurs.

Conclusion :-

The inverted file index method was successfully implemented. The basic mechanism behind search engine worked using inverted index.

Coding	Efficiency	Nirav	Timely	Total
5	9	3	2	~10
		2	2	8 4