Hack-O-Week

Week 2

Implement text preprocessing for student questions – lowercasing, tokenization, stopword removal, punctuation handling, and basic spelling normalization

Introduction

- Students ask questions in different formats and styles
- Queries may contain spelling mistakes, extra words, symbols, and mixed cases
- Raw text is difficult for machines to understand
- Text preprocessing is the first step in Natural Language Processing (NLP)

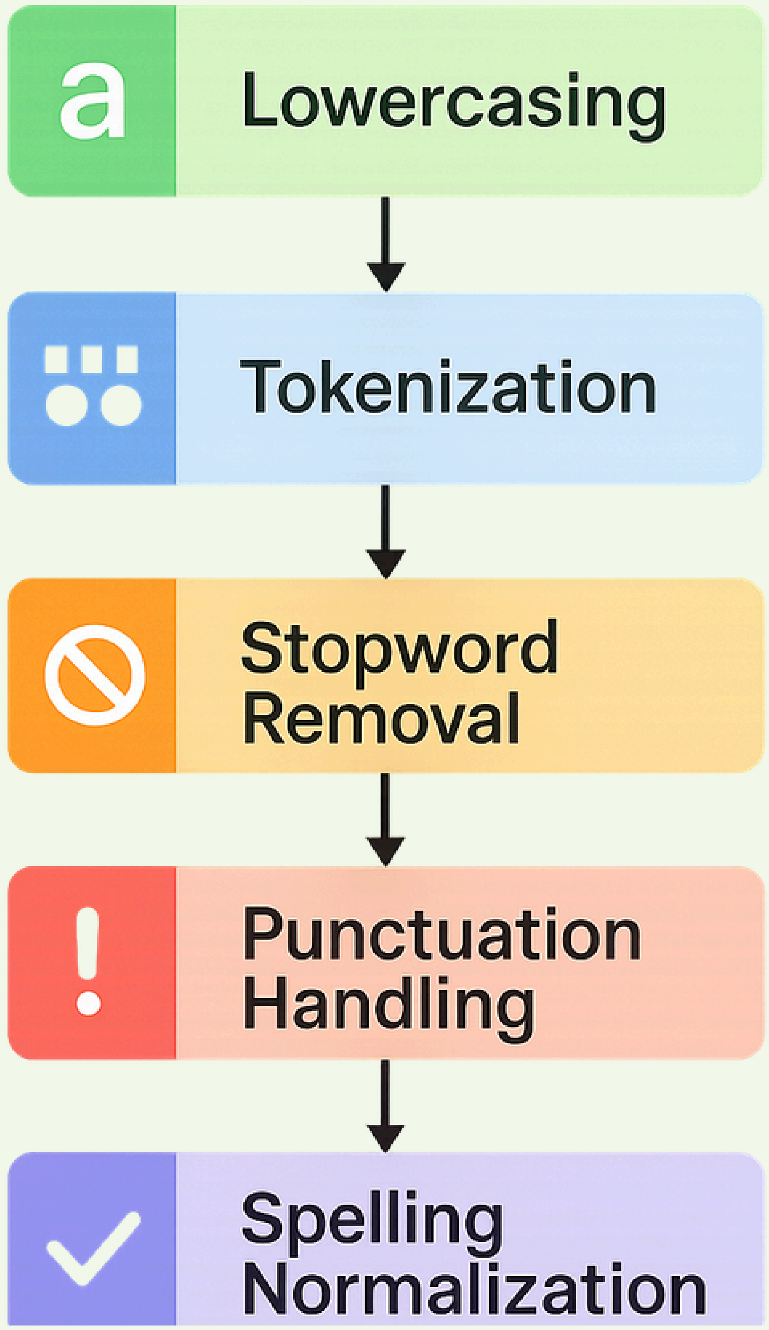
Problem Statement

Student queries are often unstructured and contain noise such as spelling errors, extra words, and inconsistent use of cases and punctuation. These inconsistencies reduce the accuracy of chatbots and search systems. Hence, a text preprocessing system is required to clean and normalize the queries so that they can be efficiently processed by machines.

Objectives

- Convert text to lowercase
- Split sentences into words (Tokenization)
- Remove common unnecessary words (Stopwords)
- Remove punctuation
- Correct basic spelling errors
- Improve quality of input for AI models

Text Preprocessing Steps



Lowercasing

Tokenization

Stopword Removal

Punctuation Handling

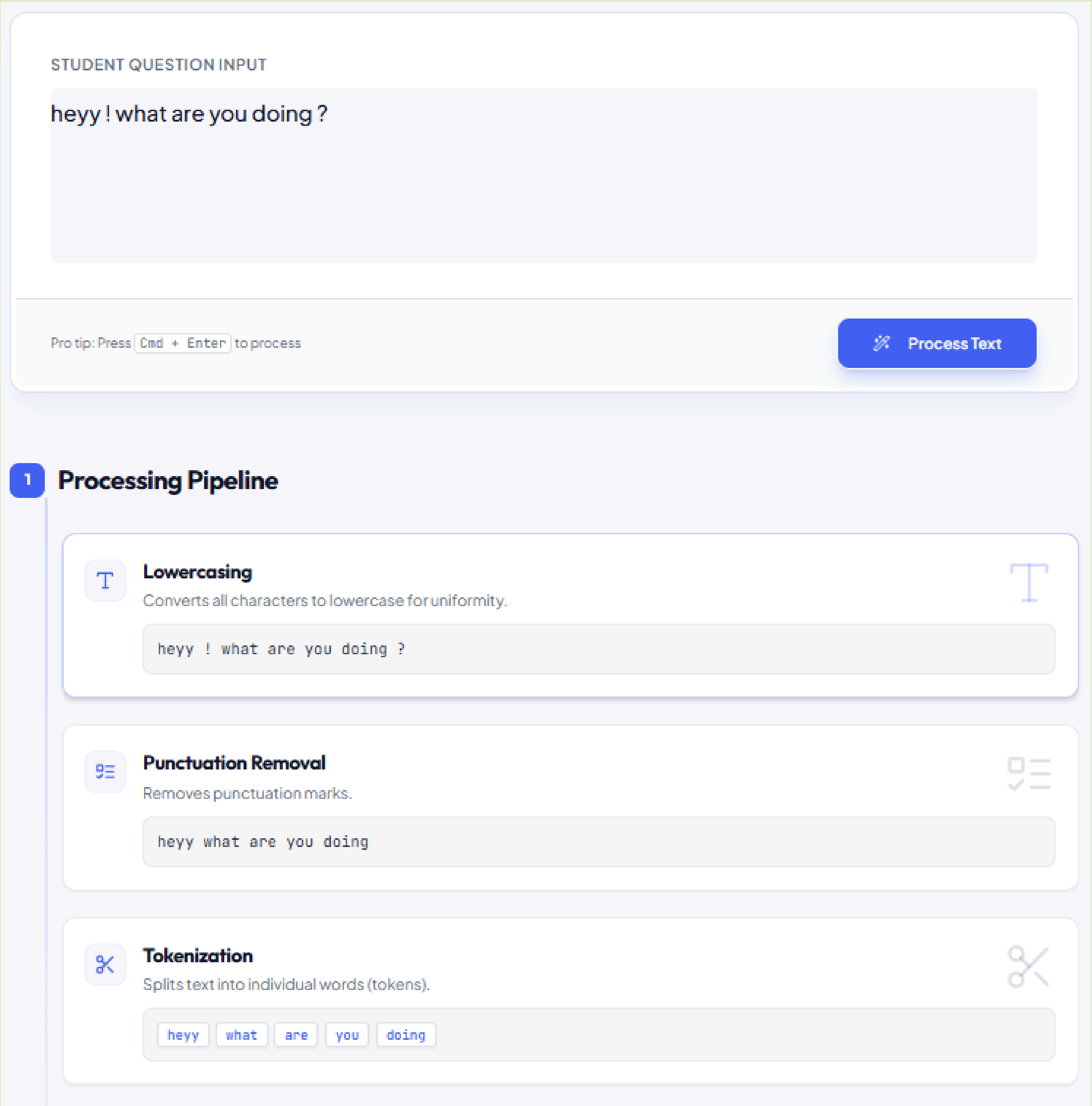
Spelling Normalization

Implementation Details

- Programming Language: Python
- Libraries Used: NLTK / spaCy for NLP processing
- Input: Student question in text form
- Steps Implemented:
- Convert text to lowercase
- Remove punctuation and special characters
- Tokenize sentence into words
- Remove stopwords
- Apply basic spelling normalization
- Output: Cleaned and normalized list of words ready for further processing

Output and Results

- The system successfully converts raw student queries into clean and normalized text.
- Unnecessary words, punctuation, and spelling variations are removed.
- The processed output is a clear set of meaningful tokens.
- This improves the understanding of queries by chatbots and search systems.
- Overall, the accuracy and efficiency of further NLP tasks are enhanced.



STUDENT QUESTION INPUT

heyy ! what are you doing ?

Pro tip: Press `Cmd + Enter` to process

1 Processing Pipeline

Lowercasing
Converts all characters to lowercase for uniformity.

```
heyy ! what are you doing ?
```

Punctuation Removal
Removes punctuation marks.

```
heyy what are you doing
```

Tokenization
Splits text into individual words (tokens).

```
heyy  what  are  you  doing
```

Your paragraph text

Conclusion and Future scope

Conclusion:

The preprocessing system cleans and normalizes student queries, improving their clarity and helping NLP systems understand them more accurately.

Future Scope:

It can be extended with advanced spelling correction, stemming, lemmatization, and multi-language support for better performance.

Thank you