IRTM Homework1 Report

B10705054 劉知微

1 Execution Environment

Google Colab

2 Programming Language

Python 3

3 Execution Method

To run the code on Google Colab, no additional environment setup is required, but remember to mount Google Drive. However, if you are using VS Code or another compiler, you may need to install the necessary packages such as pandas, numpy, and nltk.

```
from google.colab import drive
drive.mount('/content/drive')
import pandas as pd
import numpy as np
from nltk.stem import PorterStemmer
```

Please note that 1.txt and stopwords.txt(included in the folder) should be uploaded to Google Drive in advance. If you are running the code on a local desktop, ensure to update the file paths accordingly.

```
file_path = '/content/drive/My Drive/1.txt'
stopword_file = '/content/drive/My Drive/stopwords.txt'
# Load stopwords from file
with open(stopword_file, 'r') as f:
    stop_words = set(line.strip().lower() for line in f if
        line.strip())
```

```
with open(file_path, 'r') as f:
    text = f.read()
```

Lastly, you can run the code by clicking the "Run" button (play icon) next to each cell or by pressing Shift + Enter.

4 Workflow

I use the built-in function `split()` to tokenize the text. `split()` separates words based on spaces and saves the split text into tokens. Next, the `lower()` function is used to convert every letter in tokens to lowercase. I use `PorterStemmer()` from `nltk.stem` to implement the Porter Stemmer algorithm. For each word in tokens, if the word is not in the stopwords set, I stem the word and save it into `filtered_tokens`.

```
# Tokenization
tokens = text.split()
# Lowercasing everything
tokens = [word.lower() for word in tokens]
# Create a Porter Stemmer instance
porter_stemmer = PorterStemmer()
# Filtering stopwords and stemming
filtered_tokens = [porter_stemmer.stem(word) for word in tokens if
    word.isalnum() and word not in stop_words]
```

Lastly, I save the `filtered_tokens` to Google Drive using the `join()` method. The `join()` method combines the `filtered_tokens` into a single string by spaces, which is then written to the output file.

```
# Save processed terms to output file
output_file = '/content/drive/My Drive/result.txt'
with open(output_file, 'w') as f:
    f.write(' '.join(filtered_tokens))
```