|  |
| --- |
| **Python**  **Production of programs that analyze Korean PDF and extract keywords**  progress report **#2** |

|  |  |
| --- | --- |
|  | Date : 2023.12.10  Name : Haegeon Lee  ID : 183014 |

**1. Introduction**

**1) Background**

With the advent of the information age, the era of reading and interpreting a large amount of documents has arrived. If the amount of documents is vast when a particular document is first encountered, fatigue accumulates before reading. Therefore, if the Python program provides keywords that readers should focus on first, the speed increases rapidly and more efficient reading is possible. Therefore, the existence of these programs is necessary.

**2) Project goal**

It aims to develop a program that analyzes pdf to analyze what key keywords are based on the frequency of words and provides them to users.

**3) Differences from existing programs**

Existing programs are produced based on English, so there is a big difference in reading Korean pdf and it is impossible to use. Therefore, there is a difference from existing programs because we focus on extracting keywords from Korean pdf file for Koreans.

**2. Functional Requirement**

**1) Function 1** – Extract text from file

- Extract the text file from the pdf file and store it in the list.

**2) Function 2 -** Remove non-critical words

- Remove investigations such as '은','는','이','가' from list tocken. And import Morpheme analyzer to make up a list with only nouns. We will also add the ability to remove words that are frequently used but are not important.

**3) Function 3 -** ranking based on frequency to output the top 15

- Saves the frequency of nouns in dictionary form and outputs the most written value.

**4) Function 4**

- We plan to save the result value as a txt file so that it can be used in the future.

**3. progress**

**1)** Implementation of a feature

**(1)** Extract text from file

- In

Pdf file

-out

Text(Words / variable)

- Explanation

After receiving pdf, import the txt file on a page-by-page basis from pdf and save it in text(variable).

- applied learning

For loop, package, file inout

- screen shot

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**(2)** Remove non-critical words

- Explanation

Other are excluded and only alphabetic or numeric words are added to the word list.

Gets the list of words that are not important.  
Put the rest in the list called yes\_words, except for the list of words mixed with numbers and non-important words.

- applied learning

for loop, list comprehension

- screen shot

텍스트, 소프트웨어, 멀티미디어 소프트웨어, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**(3)** ranking based on frequency to output the top 15

- In

Yes\_words

-out

1~15 등 : (단어) n회

- Explanation

Find the frequency of words from the yes\_words list and add them to the dictionary.

After sorting by frequency, it outputs up to the top 15 according to the format.

- applied learning

Dictionary, sorted, for loop

- screen shot

**텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명2) Test Result**

**(1) Feature Name Tested**

- Explanation

Read the letter as txt from the input pdf, remove insignificant words, determine the frequency, and output to the top 15 depending on the format.

- Test Result screen shot

**텍스트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명**

**#** Example of pdf used for testing

**텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명**

**4. Changes to plan**

**1) Change History Subject**

- before

Remove investigations such as '은','는','이','가' from list tocken. And import Morpheme analyzer to make up a list with only nouns.

- after

Write a code that removes words and investigations that you don't use directly without importing morpheme analyzers.

- reason

I was thinking of importing morpheme analyzers. However, since the analyzer to be imported works using the java environment, we found that it is possible only if a specific environment is established. There was a problem that it was not easily and conveniently available on any computer. Therefore, we didn't import the content, but we directly configured it to be simple and maintained.

In that case, the removal of frequently used but insignificant words can also be maintained from time to time.

**5.** **What I'm considering to add**

1) The ability to print and save a list according to frequency as a file so that you can check it at any time

2) Ability to load non-critical word lists from external files and modify and add them from external files (still undecided whether to implement)

**6. Schedule**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **11/3** | **11/10** | **11/26** | **12/4** | **12/10** | **12/14** | **12/23** |
| **Write Proposal** | **Finished** |  |  |  |  |  |  |
| **Function 1** |  | **finished** | | |  |  |  |
| **Function 2** |  |  | **finished** | |  |  |  |
| **Function 3** |  |  |  | **finished** | |  |  |
| **Function 4** |  |  |  |  |  | **------>** | |