FlashCard program Documentation

Zahra Hosseini Pozveh

Feb 2021

Table of Contents

1-Summary:

This program receives a text and a number from user to create a flashcard quiz for practicing and maintaining the concepts.

This program is written in Python 3.8 using anaconda and Spyder.

It has three main parts qg_NER, text_sum and flashcard. The qg_NER, receives the input and perform some preprocessing and passes the preprocessed text to txt_sum which summarize the document and extracts most important concepts. The summarized text gets back to qg_NER, for named entity extraction (Person, Organization, Time, Date, ..) and generates as set of sentence completion questions and their answers (Q&A). These repositories of Q&A are sent to flashcard to return random set of questions. Figure 1 depicts the block diagram of this program. Datils of each program and their functions will be discussed in the following sections.

The *main.py* connects the three parts and *api.py* uses flask microservice to create an endpoint to receive an input and pass it to *main.py* to handle it.

2-Requierments

This program requires the following packages:

- ✓ flask=1.1.2
- ✓ nlp=0.4.0
- ✓ nltk=3.5
- ✓ numpy=1.18.5
- ✓ scikit-learn=0.23.1
- ✓ scipy=1.4.1
- ✓ selenium=3.141.0
- ✓ spacy=2.3.5

The requierments.txt file is prepared to ease the process of package installation.

3- Flashcard codes

3-1- api.py

This program creates an endpoint using *Flask*, gets an input (text) and the number of preferred flashcards (qty) from the user and passes it to main.py

The default values for the input are read from data/data.txt and 10 is assigned to the number of flashcards.

After debugging, open your browser and launch http://127.0.0.1:5000/

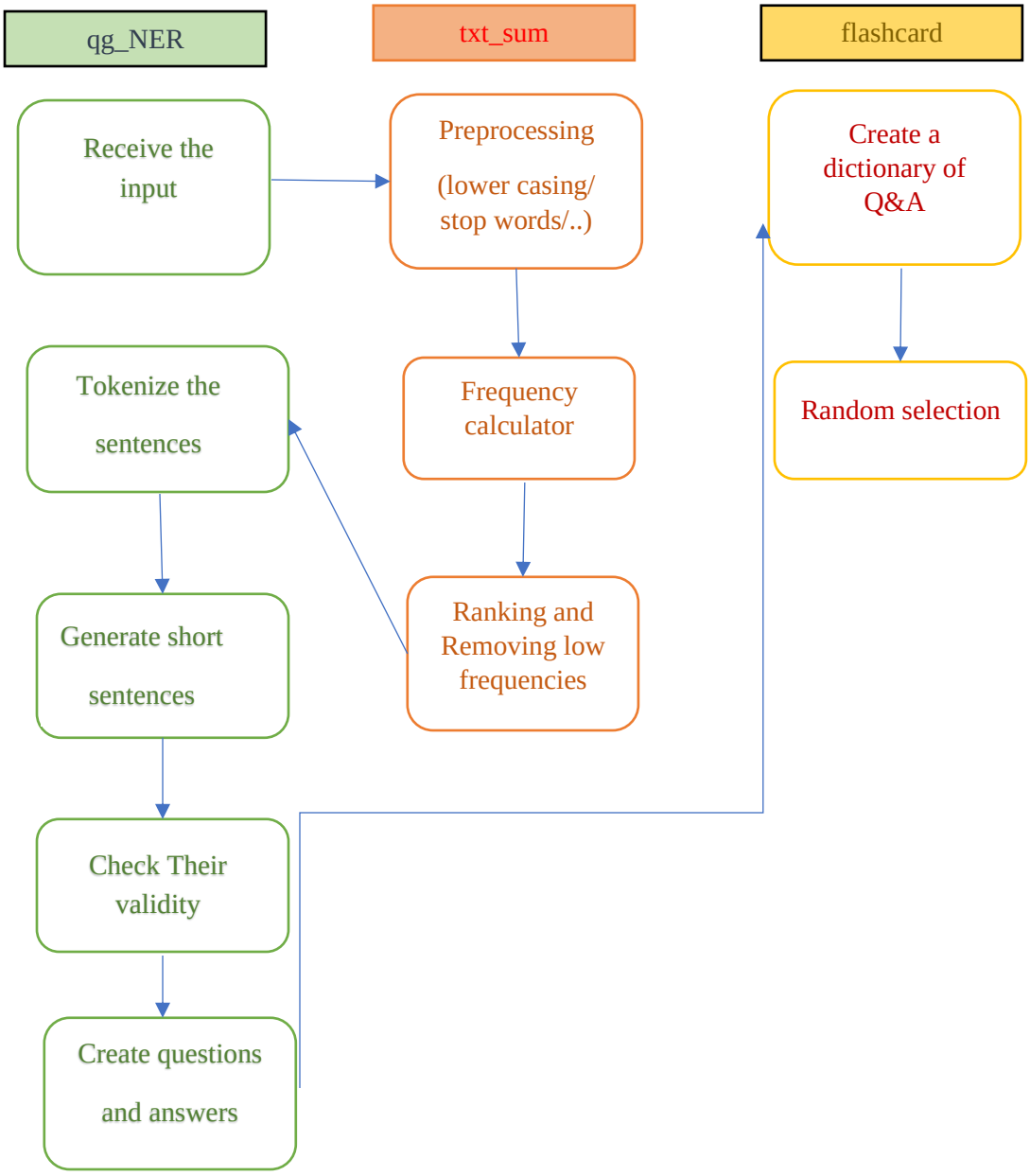

Figure1. Flashcard block diagram

3-2- main.py

Receives the arguments from the *api.py* and calls the other programs to create the flash cards.

3-3- qg_NER.py

Receives the input text, pass it to summarizer (txt_sum.py) to retrieve the most important entities. It breaks the sentences to smaller meaningful sub-sentences and adjust the size of questions (sent_Ref). Then, it performs NER using *NER_d* to detect the meaningful concepts which will be the answers. Finally, it generates the Q&A respiratory.

3-4-txt_sum.py

This program has a list as the input, then converts it to a string. It has a function *freq_Calc* that computes the higher frequency words and passes them to the *text_summ* which creates summarized sentences around these words.

3-5-flashcard.py

This file gets the questions and answers repositories and uses the *qty* variable to create a random number of questions with their answers. If the value of *qty* is bigger than the repository size, it will be adjusted according to the repository size. It generates a dictionary which is suitable for json and passes it to *main.py* to be displayed via *api.py*.