Assignment 1 – Text Preprocessing and Similarity

CIS8045 Unstructured Data Management · MM2 · AY2023/2024 Spring

In this assignment, you focus on text preprocessing and the calculation of text similarity through TF-IDF.

The dataset **kickstarter_desc_sample.csv** is given. The data was collected from a crowdfunding website, Kickstarter (www.kickstarter.com). The dataset is about a sample of entrepreneurial fundraising campaigns on Kickstarter. The sample contains around five thousand project descriptions. Each row in the data includes the unique identifier of a project (**project_id**) and a column of the textual project description (**description_str**).

You will load the data using the code provided in **Assignment 1.ipynb**. Each student will select a random sample for the assignment tasks, which means every student will work on a different random sample and submit a Jupiter Notebook file with different codes/outputs. The random sample includes 1,000 projects.

When you work on the assignment, please put the data file kickstarter_desc_sample.csv under the same directory as the notebook Assignment 1.ipynb.

You will complete two tasks in this assignment.

Task 1. Text Preprocessing

The text processing includes three steps:

1.1 Using a tokenizer to split the text (i.e., textural project description) into tokens. Print the results in the notebook.
- The code for this step is given as an example.

1.2 Write your code to remove the stop words in the text. Print the results in the notebook.
- You will complete the missing code in the function remove_stopwords().

1.3 Write your code to implement the WordNet lemmatizer with POS tags on the text. Print the results in the notebook.
- You will complete the missing code in the function postag_lemmentization().

Task 2. Use TF-IDF to represent text and calculate similarity

In the second task, you will calculate cosine similarity from TF-IDF representation of the text.

The 1,000 projects will be randomly split into a main sample (i.e., 997 projects) and one small test sample (i.e., 3 projects).

Your tasks are twofold:

2.1 Write your code to build TF-IDF matrices for all texts in the main sample (i.e., 997 projects)

2.2 For each of the 3 projects in the test sample, write your code to find the most similar projects from the main sample using cosine similarity measure based on their TF-IDF vectors. Print out their corresponding project_id and the similarity score.

Hint: to ensure that the TF-IDF matrices have a correct shape, you may use CountVectorizer and TF-IDF transformer to build the matrices. A similar approach can be found in in-class practice 9.Chatbot.ipynb.

DELIVERABLE

1. A Jupiter Notebook file containing the completed code and the printed results.
    - Note that the notebook should print out information about the current computer. (The code is given) Please work on this assignment on your own computer. The submission will not receive a grade if this information is not printed in your file.
    - Given the two random sampling processes in this assignment, the probability is extremely low that two students will get the same set of projects to work on. If two notebooks produce the same results, we assume that both students did not do independent work.
    - If helpful, you can add necessary comments to explain your code and/or results.

Submit your notebook in iCollege before **March 10, 11:59 pm**. Name your notebook as **{LastName}_{FirstName}.ipynb**.

Please refer to the course syllabus for the late submission policy, plagiarism policy, and AI-aided tool policy.