Lab 1 NLTK, VSM Exercise

Spring 2022

Task 1

Prerequisites

1. You need to install the NLTK packages:

```
pip3 install --upgrade nltk
```

2. You need to download the following file(s) from lab materials:

   - `lab1_task1_skeleton.py`: the program skeleton.

Q1 Finish codes to perform the following tasks on the corpus named "`austen-sense.txt`" from the project Gutenberg electronic text archive.

Hint: you could refer to Task1/Q1-1 in the .ipynb() for how to retrieve the corpus

1. Print the number of word tokens in the corpus.
2. Print the size of the vocabulary (number of unique word tokens).
3. Print the tokenized words of the first sentence in the corpus.

Q2 Finish codes to perform the following tasks on the `brown` corpus.

Hint: you could refer to Task1/Q2-1 in the .ipynb for how to retrieve the corpus. And you could check nltk.FreqDist function for this question.

1. Print the top 10 most common words in the `romance` category.
2. Print the word frequency of the following words: `[ring,activities,love,sports,church]` in the `romance` and `hobbies` categories respectively.

Q3 Finish codes to perform the following tasks using `WordNet`.

1. Print all synonymous lemma words of the word '`dictionary`'.
2. Print all hyponyms of the word '`dictionary`'.

3. Use one of the predefined similarity measures to score the similarity of the following pairs of synsets and rank the pairs in order of decreasing similarity.
(`right_whale.n.01`, `novel.n.01`)
(`right_whale.n.01`, `minke_whale.n.01`)
(`right_whale.n.01`, `tortoise.n.01`)

Hints: predefined similarity measures can be found in ***http://www.nltk.org/howto/wordnet.html***

Task 2

Prerequisites

1. You need to have some background knowledge about Vector Space Model (VSM). If not, you can check out:
   - https://en.wikipedia.org/wiki/Vector_space_model
   - https://en.wikipedia.org/wiki/Tf–idf
   - https://en.wikipedia.org/wiki/Cosine_similarity
2. You need to install the Numpy packages:

   ```
   pip3 install --upgrade numpy
   ```

   or

   ```
   conda install --upgrade numpy
   ```

3. You need to download the following file(s) from canvas:
   - `lab1_task2_skeleton.py`: the program skeleton.

Q1 Write code to perform the following tasks:

1. Build a vocabulary based on given corpus, and turn raw sentences into Bag-of-words representation.

   Using this corpus as Input:

   ['this is a foo bar', 'foo bar bar black sheep', 'this is a sentence']

2. Implement cosine similarity function. And output pairwise similarity of BOW.

3. Implement TFIDF function to turn Bag-of-words representation into TFIDF representation. And output pairwise cosine similarity of TFIDF(hint: the tfidf equation is as follows).

$$\begin{aligned} \text{tf}(t,d) &= f_{t,d} \Big/ \sum_{t' \in d} f(t',d) \\ \text{idf}(t) &= N/n_t \\ \text{tf-idf}(t,d) &= \text{tf}(t,d) \times \text{idf}(t) \end{aligned}$$

4. Implement TFIDF based 3 using the following equation.

$$\begin{aligned} \text{tf}(t,d) &= (0.5 + 0.5\frac{f_{t,q}}{\max_t f_{t,q}}) \log \frac{N}{n_t} \\ \text{idf}(t) &= \log(\frac{N}{1+n_t}) + 1 \\ \text{tf-idf}(t,d) &= \text{tf}(t,d) \times \text{idf}(t) \end{aligned}$$

Task 3

Prerequisites

1. You need to download the following file(s) from lab materials:
   - `lab1_task3_skeleton.py`: the program skeleton.
2. Similar to assignment 1, you should understand how to use NLTK corpus, such as Brown, Gutenberg etc.

Q1 Zipf's law states frequency of a word is inversely proportional to its rank in the frequency table. verify Zipf's law based on **brown + gutenberg** corpus, and draw a graph of the results via matplotlib.

Hint: the file, ***`lab1_task3_skeleton.py`*** *contains the answer for codeing and drawing based on the gutenberg corpus.*