

1402-02-29



Persian Gulf University

Faculty of Intelligent Systems Engineering and Data Science

Natural Language Processing

Dr. Mohammad Bidoki

Homework #3

By: Mohammad Barzegar

Student ID: 4010724001

تمرین a (توکنایز و حذف علامت های اضافی):

ابتدا با استفاده از `WhiteSpaceTokenizer`، اسپیس ها و تب ها و لاین بریک ها را پاک کرده تا فایل یکدست شود:

```
1 import nltk
2 from nltk.tokenize import WhitespaceTokenizer

3 # Deleting spaces and tabs
4
5 input_filenames = ['sport.txt', 'tech.txt']
6 output_filenames = ['sport1.txt', 'tech1.txt']
7
8 for i in range(len(input_filenames)):
9     with open(input_filenames[i], "r") as f:
10         input_text = f.read()
11
12     # Tokenize the text using WhitespaceTokenizer
13     tokenizer = WhitespaceTokenizer()
14     tokens = tokenizer.tokenize(input_text)
15     output_text = ' '.join(tokens)
16
17     # Save the tokens to an output text file
18     with open(output_filenames[i], "w") as f:
19         f.write(output_text)
```

تمرین b (حذف کردن **stopword** ها و حروف تک کاراکتری):

خروجی مرحله قبل، ورودی این مرحله است. با استفاده از `RegexpTokenizer` همه ی کاراکتر ها بجز

حروف الفبای انگلیسی را حذف می کنیم:

```
1 # Delete all characters except alphabet letters
2
3 from nltk.tokenize import RegexpTokenizer
4
5 input_filenames = ['sport1.txt', 'tech1.txt']
6 output_filenames = ['sport2.txt', 'tech2.txt']
7
8 tokenizer = RegexpTokenizer(r'[a-zA-Z]+')
9
10 for i in range(len(input_filenames)):
11     with open(input_filenames[i], "r") as f:
12         input_text = f.read()
13         words = tokenizer.tokenize(input_text.lower())
14
15         # Save the tokens to an output text file
16         with open(output_filenames[i], "w") as f_out:
17             f_out.write(" ".join(words))
```

خروجی مرحله قبل را برداشته و سپس حروف تک کاراکتری را حذف می کنیم:

```
1 input_filenames = ['sport2.txt', 'tech2.txt']
2 output_filenames = ['sport3.txt', 'tech3.txt']
3 for i in range(len(input_filenames)):
4     with open(input_filenames[i], 'r') as file:
5         text = file.read()
6
7         # Split the text into words
8         words = text.split()
9
10        # Remove one-character words
11        words = [word for word in words if len(word) > 1]
12
13        # Join the remaining words back into a string
14        updated_text = ' '.join(words)
15
16        # Write the updated text back into the file
17        with open(output_filenames[i], 'w') as file:
18            file.write(updated_text)
```

با استفاده از خروجی مرحله قبل، حروف stopword را نیز حذف میکنیم:

```

1 # Removing stopwords
2
3 nltk.download('stopwords')
4
5 from nltk.corpus import stopwords
6 from nltk.tokenize import word_tokenize
7
8 input_filenames = ['sport3.txt', 'tech3.txt']
9 output_filenames = ['sport4.txt', 'tech4.txt']
10
11 stop_words = set(stopwords.words('english'))
12
13 for i in range(len(input_filenames)):
14     with open(input_filenames[i], 'r') as f:
15         text = f.read()
16         words = word_tokenize(text)
17         cleaned_words = [word for word in words if word not in stop_words]
18         cleaned_text = ' '.join(cleaned_words)
19
20         with open(output_filenames[i], 'w') as f_out:
21             f_out.write(cleaned_text)
22
23     print(input_filenames[i], 'Before:', len(text), 'After:', len(cleaned_text))

```

```

sport3.txt Before: 908175 After: 638400
tech3.txt Before: 1141164 After: 817591

```

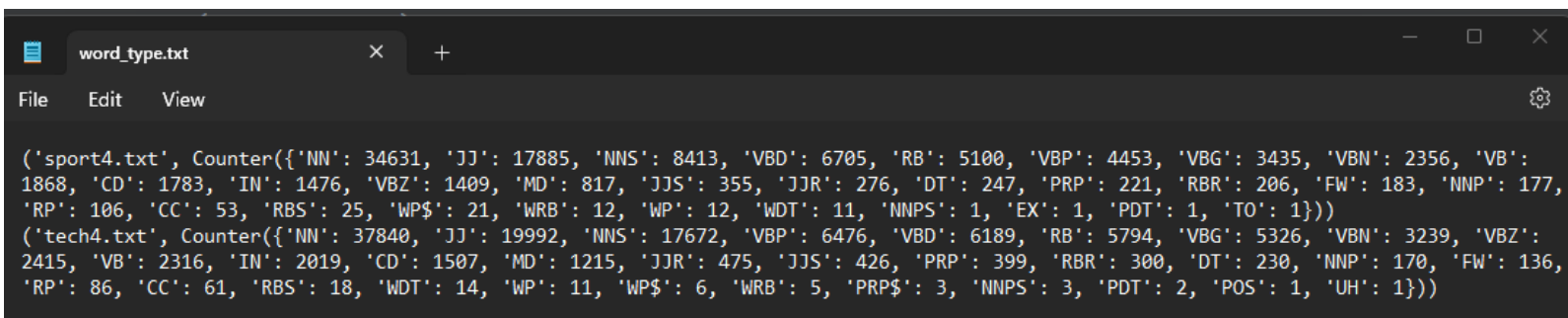
تمرین c (type های موجود در هر متن):

با استفاده از متد `pos_tag`، تایپ هر کلمه را برای هر فایل مشخص میکنیم. خروجی را در فایل `word_type.txt` می توانید مشاهده بفرمایید.

```

1 from collections import Counter
2 nltk.download('averaged_perceptron_tagger')
3
4 def count_word_types(filename):
5     with open(filename, 'r') as file:
6         text = file.read()
7         # Tokenize the text
8         tokens = nltk.word_tokenize(text)
9         # Perform POS tagging
10        tagged_words = nltk.pos_tag(tokens)
11        # Count the occurrences of each word type
12        word_types = [tag[1] for tag in tagged_words]
13        word_type_counts = Counter(word_types)
14        print("----- FILE NAME: ", filename, "-----")
15        for word_type, count in word_type_counts.items():
16            print(f'{word_type}: {count}')
17        print("-----")
18        return str(filename), word_type_counts
19 # Count the word types in the text file
20 word_type_counts1 = count_word_types('sport4.txt')
21 word_type_counts2 = count_word_types('tech4.txt')
22
23 with open('word_type.txt', 'w') as file:
24     file.write(str(word_type_counts1))
25     file.write('\n')
26     file.write(str(word_type_counts2))

```



```

('sport4.txt', Counter({'NN': 34631, 'JJ': 17885, 'NNS': 8413, 'VBD': 6705, 'RB': 5100, 'VBP': 4453, 'VBG': 3435, 'VBN': 2356, 'VB': 1868, 'CD': 1783, 'IN': 1476, 'VBZ': 1409, 'MD': 817, 'JJ$': 355, 'JJR': 276, 'DT': 247, 'PRP': 221, 'RBR': 206, 'FW': 183, 'NNP': 177, 'RP': 106, 'CC': 53, 'RBS': 25, 'WP$': 21, 'WRB': 12, 'WP': 12, 'WDT': 11, 'NNPS': 1, 'EX': 1, 'PDT': 1, 'TO': 1}))
('tech4.txt', Counter({'NN': 37840, 'JJ': 19992, 'NNS': 17672, 'VBP': 6476, 'VBD': 6189, 'RB': 5794, 'VBG': 5326, 'VBN': 3239, 'VBZ': 2415, 'VB': 2316, 'IN': 2019, 'CD': 1507, 'MD': 1215, 'JJR': 475, 'JJ$': 426, 'PRP': 399, 'RBR': 300, 'DT': 230, 'NNP': 170, 'FW': 136, 'RP': 86, 'CC': 61, 'RBS': 18, 'WDT': 14, 'WP': 11, 'WP$': 6, 'WRB': 5, 'PRP$': 3, 'NNPS': 3, 'PDT': 2, 'POS': 1, 'UH': 1}))

```

تمرین d (تعداد کلمه football و computer):

تعداد تکرار هر کلمه در هر دو فایل را به دست می آوریم:

```

4 1 input_filenames = ['sport4.txt', 'tech4.txt']
2  football_count = {'sport': 0, 'tech': 0}
3  computer_count = {'sport': 0, 'tech': 0}
4
5  for i in range(len(input_filenames)):
6      with open(input_filenames[i], 'r') as file:
7          text = file.read()
8
9      # Convert the text to lowercase for case-insensitive matching
10     text = text.lower()
11
12     # Count the occurrences of the words
13     football_count['sport'] = text.count('football')
14     computer_count['tech'] = text.count('computer')
15
16
17 print("Occurrences of 'football':", football_count)
18 print("Occurrences of 'computer':", computer_count)
19
~  Occurrences of 'football': {'sport': 15, 'tech': 0}
   Occurrences of 'computer': {'sport': 0, 'tech': 444}

```

تعداد تکرار football در فایل sport 15 و در فایل tech 0 بار است.

تعداد تکرار computer در فایل sport 0 و در فایل tech 444 بار است.

تمرین e (ماتریس term-document):

ابتدا تابعی مینویسیم که با گرفتن نام 2 فایل، ماتریس term-document را بسازد. از متد countvectorizer استفاده خواهیم کرد:

```

1 from sklearn.feature_extraction.text import CountVectorizer
2 import numpy as np
3
4 def create_word_term_matrix(file1_path, file2_path):
5     # Read the contents of the first file
6     with open(file1_path, 'r') as file1:
7         text1 = file1.read()
8
9     # Read the contents of the second file
10    with open(file2_path, 'r') as file2:
11        text2 = file2.read()
12
13    # Combine the text from both files into a single list
14    combined_text = [text1, text2]
15
16    # Create an instance of CountVectorizer
17    vectorizer = CountVectorizer()
18
19    # Fit the vectorizer to the combined text and transform it into a word-term matrix
20    word_term_matrix = vectorizer.fit_transform(combined_text)
21
22    # Convert the matrix to a NumPy array
23    word_term_matrix = word_term_matrix.toarray()
24
25    # Get the feature names (words) from the vectorizer's vocabulary
26    words = vectorizer.get_feature_names_out()
27    return word_term_matrix, words

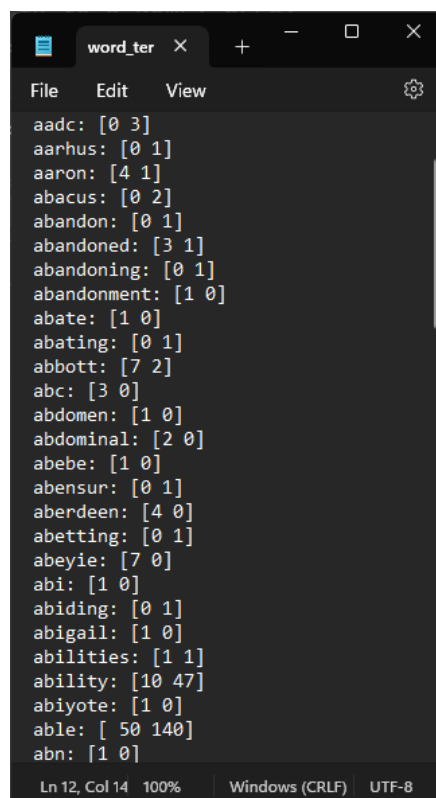
```

سپس با استفاده از این تابع، خروجی مورد نظر را در فایل `word_term_matrix.txt` ذخیره می کنیم که در فایل تحویلی موجود است. المان اول (چپ) تعداد کلمه در فایل `sport` و المان دوم (راست) تعداد در فایل `tech` است.

```

28
29 file1_path = 'sport4.txt'
30 file2_path = 'tech4.txt'
31 matrix, words = create_word_term_matrix(file1_path, file2_path)
32
33 # Print the word-term matrix
34 print("Word-Term Matrix:")
35
36 with open('word_term_matrix.txt', 'w') as file:
37     for i, word in enumerate(words):
38         print(f"{word}: {matrix[:, i]}")
39         file.write(f"{word}: {matrix[:, i]} \n")
40 # first element is word count in sport.txt and second element is count in tech.txt

```



```

aadc: [0 3]
aarhus: [0 1]
aaron: [4 1]
abacus: [0 2]
abandon: [0 1]
abandoned: [3 1]
abandoning: [0 1]
abandonment: [1 0]
abate: [1 0]
abating: [0 1]
abbott: [7 2]
abc: [3 0]
abdomen: [1 0]
abdominal: [2 0]
abebe: [1 0]
abensur: [0 1]
aberdeen: [4 0]
abetting: [0 1]
abeyie: [7 0]
abi: [1 0]
abiding: [0 1]
abigail: [1 0]
abilities: [1 1]
ability: [10 47]
abiyote: [1 0]
able: [ 50 140]
abn: [1 0]

```


تمرین f (شباهت کسینوسی):

با استفاده از کتابخانه sklearn و متد های countvectorizer و cosine_similarity تابعی می سازیم تا شباهت کسینوسی را محاسبه کند:

```
1 from sklearn.feature_extraction.text import CountVectorizer
2 from sklearn.metrics.pairwise import cosine_similarity
```

```
4 def calculate_cosine_similarity(file1_path, file2_path, word1, word2):
5     with open(file1_path, 'r') as file1:
6         text1 = file1.read()
7     with open(file2_path, 'r') as file2:
8         text2 = file2.read()
9     # Combine the text from both files into a list
10    text_list = [text1.lower(), text2.lower()]
11    # Create an instance of CountVectorizer
12    vectorizer = CountVectorizer()
13    # Fit the vectorizer to the text and transform it into a word-term matrix
14    word_term_matrix = vectorizer.fit_transform(text_list)
15    # Convert the word-term matrix to an array
16    word_term_array = word_term_matrix.toarray()
17    # Get the index of the words in the feature names (vocabulary)
18    word1_index = vectorizer.vocabulary_.get(word1)
19    word2_index = vectorizer.vocabulary_.get(word2)
20    if word1_index is None or word2_index is None:
21        return "One or both words not found in the vocabulary."
22    # Get the vectors representing the occurrences of the words in each file
23    word1_vector = word_term_array[:, word1_index]
24    word2_vector = word_term_array[:, word2_index]
25    # Reshape the vectors to match the expected input shape for cosine_similarity
26    word1_vector = word1_vector.reshape(1, -1)
27    word2_vector = word2_vector.reshape(1, -1)
28    # Calculate the cosine similarity between the word vectors
29    similarity = cosine_similarity(word1_vector, word2_vector)[0, 0]
30    return similarity
```

سپس با استفاده از تابع فوق، ورودی دو فایل تکست را داده و شباهت های کسینوسی مورد نظر را به عنوان خروجی در فایل cosine_similarity.txt که در فولدر تحویل داده موجود است، درج می کنیم.

```

31
32 file1_path = 'tech4.txt'
33 file2_path = 'sport4.txt'
34
35 ref_word = {'football':{'sport': 0, 'technology': 0, 'computer': 0},
36             'sport': {'computer': 0, 'technology': 0, 'basketball': 0},
37             'computer': {'basketball': 0, 'technology': 0, 'laptop': 0},
38             'website': {'laptop': 0, 'technology': 0, 'football': 0}
39         }
40
41 for key, value in ref_word.items():
42     for word in value:
43         similarity = calculate_cosine_similarity(file1_path, file2_path, key, word)
44         value[word] = similarity
45
46 with open('cosine_similarity.txt', 'w') as file:
47     for key, value in ref_word.items():
48         print(f"{key} -> {value}")
49         print() # Print a line break after each item
50         file.write(f"{key} -> {value}\n")

```

```

cosine_similarity.txt
File Edit View
football -> {'sport': 0.9999621809685715, 'technology': 0.1319916635298333, 'computer': 0.11821288978511235}
sport -> {'computer': 0.10957246722727652, 'technology': 0.12336583033843335, 'basketball': 0.6425300444738722}
computer -> {'basketball': 0.8320502943378437, 'technology': 0.9999035633345557, 'laptop': 1.0}
website -> {'laptop': 0.9895702125995407, 'technology': 0.991475302140086, 'football': 0.2600212659000183}

Ln 4, Col 107 100% Windows (CRLF) UTF-8

```

کلمه مرجع	کلمات	شبهات کسینوسی
Football	Sport	0.9999621809685715
	Technology	0.1319916635298333
	Computer	0.1182128897851123
Sport	Computer	0.1095724672272765
	Technology	0.1233658303384333
	Basketball	0.6425300444738722
Computer	Basketball	0.8320502943378437
	Technology	0.9999035633345557
	Laptop	1.0
Website	Laptop	0.9895702125995407
	Technology	0.991475302140086
	Football	0.2600212659000183

همانگونه که نشان داده شده است، بیشترین شباهت کلمه football با کلمه ی sport، بیشترین شباهت کلمه sport با کلمه ی basketball، بیشترین شباهت computer با laptop، و بیشترین شباهت website با technology است.