NLP - Description for Students

November 11, 2024

1 Natural Language Processing

This project will give you practical experience using Natural Language Processing techniques. This project is in three parts: - in part 1) you will use a traditional dataset in a CSV file - in part 2) you will use the Wikipedia API to directly access content on Wikipedia. - in part 3) you will make your notebook interactive

1.0.1 Part 1)

- The CSV file is available at https://ddc-datascience.s3.amazonaws.com/Projects/Project.5-NLP/Data/NLP.csv
- The file contains a list of famous people and a brief overview.
- The goal of part 1) is provide the capability to
  - Take one person from the list as input and output the 10 other people who's overview are "closest" to the person in a Natural Language Processing sense
  - Also output the sentiment of the overview of the person

1.0.2 Part 2)

- For the same person from step 1), use the Wikipedia API to access the whole content of that person's Wikipedia page.
- The goal of part 2) is to produce the capability to:
  1. For that Wikipedia page determine the sentiment of the entire page
  2. Print out the Wikipedia article
  3. Collect the Wikipedia pages from the 10 nearest neighbors in Step 1)
  4. Determine the nearness ranking of these 10 to your main subject based on their entire Wikipedia page
  5. Compare the nearest ranking from Step 1) with the Wikipedia page nearness ranking

1.0.3 Part 3)

Make an interactive notebook.

In addition to presenting the project slides, at the end of the presentation each student will demonstrate their code using a famous person suggested by the other students that exists in the DBpedia set.

1.1 Imports

[15]:
```python
import nltk
import textblob

# Download required NLTK data
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('wordnet')

# Your existing imports
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import NearestNeighbors
from textblob import TextBlob
```

```
[nltk_data] Downloading package punkt to /root/nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data…
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package wordnet to /root/nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
```

[16]:
```
!curl -s https://ddc-datascience.s3.amazonaws.com/Projects/Project.5-NLP/Data/
 ↪NLP.csv | wc -l
```

```
42786
```

1.2 First DF pt1

[18]:
```python
# Load the CSV file
url = "https://ddc-datascience.s3.amazonaws.com/Projects/Project.5-NLP/Data/NLP.
 ↪csv"
df = pd.read_csv(url)

# Display the first few rows of the DataFrame
print(df.head())

# Display the column names
print(df.columns)

# Display the shape of the DataFrame
print(df.shape)
```

```
                                             URI                  name  \
0        <http://dbpedia.org/resource/Digby_Morrell>        Digby Morrell
```

```
1       <http://dbpedia.org/resource/Alfred_J._Lewy>       Alfred J. Lewy
2        <http://dbpedia.org/resource/Harpdog_Brown>        Harpdog Brown
3  <http://dbpedia.org/resource/Franz_Rottensteiner>  Franz Rottensteiner
4               <http://dbpedia.org/resource/G-Enka>               G-Enka

                                                text
0  digby morrell born 10 october 1979 is a former…
1  alfred j lewy aka sandy lewy graduated from un…
2  harpdog brown is a singer and harmonica player…
3  franz rottensteiner born in waidmannsfeld lowe…
4  henry krvits born 30 december 1974 in tallinn …
Index(['URI', 'name', 'text'], dtype='object')
(42786, 3)
```

[21]:

```
# Clean and preprocess the data
df['name'] = df['name'].str.lower()
df['name'] = df['name'].str.replace(r'[^\w\s]', '', regex=True)
```

[23]:

```
# Create a TF-IDF vectorizer and transform the data
vectorizer = TfidfVectorizer(stop_words='english')
X = vectorizer.fit_transform(df['text'])
```

[24]:

```
# Implement the K-Nearest Neighbors algorithm
nn = NearestNeighbors(n_neighbors=11, metric='cosine')
nn.fit(X)
```

[24]:

```
NearestNeighbors(metric='cosine', n_neighbors=11)
```

[25]:

```
# Create a function to find nearest neighbors and sentiment
def find_nearest_neighbors_and_sentiment(person_name):
    # Find the index of the person
    person_index = df[df['name'] == person_name].index[0]

    # Get the nearest neighbors
    distances, indices = nn.kneighbors(X[person_index].reshape(1, -1))

    # Get the names of the nearest neighbors (excluding the person itself)
    nearest_neighbors = df.iloc[indices[0][1:]]['name'].tolist()

    # Calculate sentiment for the person
    person_text = df.loc[person_index, 'text']
    sentiment = TextBlob(person_text).sentiment

    return nearest_neighbors, sentiment
```

[28]:

```
# Test the function
person_name = df['name'].iloc[0]  # Use the first person's name as an example
nearest_neighbors, sentiment = find_nearest_neighbors_and_sentiment(person_name)
```

```python
print(f"Nearest neighbors to {person_name}:")
for neighbor in nearest_neighbors:
    print(f"- {neighbor}")
```

```
Nearest neighbors to digby morrell:
- steven browne
- peter freeman footballer
- lindsay smith australian footballer
- earl spalding
- relton roberts
- mark austin footballer
- todd curley
- daniel harris footballer
- richard ambrose
- darren pfeiffer
```

[29]:

```python
print(f"\nSentiment of {person_name}'s text:")
print(f"Polarity: {sentiment.polarity}")
print(f"Subjectivity: {sentiment.subjectivity}")
```

```

Sentiment of digby morrell's text:
Polarity: -0.041666666666666664
Subjectivity: 0.17896825396825394
```