

INFO5731 Assignment 2

In this assignment, you will work on gathering text data from an open data source via web scraping or API. Following this, you will need to clean the text data and perform syntactic analysis on the data. Follow the instructions carefully and design well-structured Python programs to address each question.

Expectations:

- Use the provided *.ipynb* document to write your code & respond to the questions. Avoid generating a new file.
- Write complete answers and run all the cells before submission.
- Make sure the submission is "clean"; *i.e.*, no unnecessary code cells.
- Once finished, allow shared rights from top right corner (see *Canvas for details*).
- **Make sure to submit the cleaned data CSV in the comment section - 10 points**

Total points: 100

Deadline: Wednesday, at 11:59 PM.

Late Submission will have a penalty of 10% reduction for each day after the deadline.

✓ Question 1 (40 points)

Write a python program to collect text data from **either of the following sources** and save the data into a **csv file**:

- (1) Collect all the customer reviews of a product (you can choose any product) on amazon. [atleast 1000 reviews]
- (2) Collect the top 1000 User Reviews of a movie recently in 2023 or 2024 (you can choose any movie) from IMDB. [If one movie doesn't have sufficient reviews, collect reviews of atleast 2 or 3 movies]
- (3) Collect all the reviews of the top 1000 most popular software from G2 or Capterra.
- (4) Collect the **abstracts** of the top 10000 research papers by using the query "machine learning", "data science", "artificial intelligence", or "information extraction" from Semantic Scholar.
- (5) Collect all the information of the 904 narrators in the Densho Digital Repository.

```
1 # Your code here
2 '''
3 2) Collect the top 1000 User Reviews of a movie recently in 2023 or 2024 (you can choose any movie) from
4 '''
5 #importing the required modules
6 import requests
7 from bs4 import BeautifulSoup
8 import pandas as pd
```

```

1 #creating function name with scrape_imdb_reviews
2 def scrape_imdb_reviews(number_of_reviews=1000):
3     base_url = f'https://www.imdb.com/title/tt15398776/reviews?ref_=tt_urv'
4     # creating an empty dictionary for reviews
5     reviews = []
6     page_num = 1
7     while len(reviews) < number_of_reviews:
8         url = f'{base_url}&start={page_num}'
9         response = requests.get(url)
10        soup = BeautifulSoup(response.content, 'html.parser')
11        review_containers = soup.find_all('div', class_='text show-more__control')
12        reviews.extend([container.text.strip() for container in review_containers])
13        page_num += 25 # 25 reviews per the page
14    return reviews[:number_of_reviews]
15 # Scraping top 1000 user reviews
16 Oppenheimer_reviews = scrape_imdb_reviews(number_of_reviews=1000)

1 # Creating a DataFrame
2 reviews_df = pd.DataFrame(Oppenheimer_reviews, columns=['reviews'])
3 reviews_df

```

	reviews
0	One of the most anticipated films of the year ...
1	You'll have to have your wits about you and yo...
2	I'm a big fan of Nolan's work so was really lo...
3	"Oppenheimer" is a biographical thriller film ...
4	This movie is just... wow! I don't think I hav...
...	...
995	It's isn't a masterpiece. It's a decent biopic...
996	I'm a big Nolan fan. Maybe this one just wasn'...
997	My Review - Oppenheimer\nMy Rating Ten plus 10...
998	Nolan is good at constructing complicated timi...
999	It saddens me that so many people are mistakin...

1000 rows × 1 columns

```

1 # Saving to the CSV file
2 reviews_df.to_csv('Oppenheimer_reviews.csv', index=False)
3 print(' Reviews are collected, and saved successfully ')

```

Reviews are collected, and saved successfully

✓ Question 2 (30 points)

Write a python program to **clean the text data** you collected in the previous question and save the clean data in a new column in the csv file. The data cleaning steps include: [Code and output is required for each part]

(1) Remove noise, such as special characters and punctuations.

- (2) Remove numbers.
- (3) Remove stopwords by using the stopwords list.
- (4) Lowercase all texts
- (5) Stemming.
- (6) Lemmatization.

```

1 # Write code for each of the sub parts with proper comments.
2 #importing all the required modules
3 import pandas as pd
4 import nltk
5 from nltk.corpus import stopwords
6 from nltk.stem import PorterStemmer, WordNetLemmatizer
7 import string
8 import re
9
10 # Download NLTK resources
11 nltk.download('stopwords')
12 nltk.download('punkt')
13 nltk.download('wordnet')

```

```

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
True

```

```

1 data_url="https://raw.githubusercontent.com/Sowmika26/INFO_5731/main/Assignment_2/Oppenheimer_reviews.cs
2 df = pd.read_table(data_url,names=[ 'text' ])
3 df

```

	text
0	reviews
1	One of the most anticipated films of the year ...
2	You'll have to have your wits about you and yo...
3	I'm a big fan of Nolan's work so was really lo...
4	"Oppenheimer" is a biographical thriller film ...
...	...
996	It's isn't a masterpiece. It's a decent biopic...
997	I'm a big Nolan fan. Maybe this one just wasn'...
998	My Review - Oppenheimer\nMy Rating Ten plus 10...
999	Nolan is good at constructing complicated timi...
1000	It saddens me that so many people are mistakin...

1001 rows × 1 columns

```
1 #1: Remove Noise function (special characters and punctuations)
2 def remove_noise(text):
3     clean_text = re.sub('[^a-zA-Z0-9]', ' ', text)
4     return clean_text
5 df['clean_text'] = df['text'].apply(remove_noise)
6 # Displaying the Data Frame after removing noise
7 print("\nData Frame after removing noise:")
8 df
```

Data Frame after removing noise:

	text	clean_text
0	reviews	reviews
1	One of the most anticipated films of the year ...	One of the most anticipated films of the year ...
2	You'll have to have your wits about you and yo...	You ll have to have your wits about you and yo...
3	I'm a big fan of Nolan's work so was really lo...	I m a big fan of Nolan s work so was really lo...
4	"Oppenheimer" is a biographical thriller film ...	Oppenheimer is a biographical thriller film ...
...
996	It's isn't a masterpiece. It's a decent biopic...	It s isn t a masterpiece It s a decent biopic...
997	I'm a big Nolan fan. Maybe this one just wasn'...	I m a big Nolan fan Maybe this one just wasn ...
998	My Review - Oppenheimer\nMy Rating Ten plus 10...	My Review Oppenheimer My Rating Ten plus 10 ...
		Nolan is good at constructing complicated

```
1 #2:- Remove Numbers
2 #creating remove_number function
3 def remove_numbers(text):
4     clean_text = re.sub(r'\d+', ' ', text)
5     return clean_text
6 df['clean_text_remove_numbers'] = df['clean_text'].apply(remove_numbers)
7 print("\nData Frame after removing numbers:")
8 df
```

Data Frame after removing numbers:

	text	clean_text	clean_text_remove_numbers
0	reviews	reviews	reviews
1	One of the most anticipated films of the year ...	One of the most anticipated films of the year ...	One of the most anticipated films of the year ...
2	You'll have to have your wits about you and yo...	You ll have to have your wits about you and yo...	You ll have to have your wits about you and yo...
3	I'm a big fan of Nolan's work so was really lo...	I m a big fan of Nolan s work so was really lo...	I m a big fan of Nolan s work so was really lo...
4	"Oppenheimer" is a biographical thriller film ...	Oppenheimer is a biographical thriller film ...	Oppenheimer is a biographical thriller film ...
...
996	It's isn't a masterpiece. It's a decent biopic...	It s isn t a masterpiece It s a decent biopic...	It s isn t a masterpiece It s a decent biopic...
997	I'm a big Nolan fan. Maybe this one just wasn'...	I m a big Nolan fan Maybe this one just wasn ...	I m a big Nolan fan Maybe this one just wasn ...
998	My Review - Oppenheimer\nMy Rating Ten plus 10...	My Review Oppenheimer My Rating Ten plus 10 ...	My Review Oppenheimer My Rating Ten plus I...
999	Nolan is good at constructing complicated timi...	Nolan is good at constructing complicated timi...	Nolan is good at constructing complicated timi...

```

1 # 3: Remove stopwords by using the stopwords List
2 #creating remove_stopwords function
3 def remove_stopwords(text):
4     stop_words = set(stopwords.words('english'))
5     words = nltk.word_tokenize(text)
6     filter_words = [word for word in words if word.lower() not in stop_words]
7     return ' '.join(filter_words)
8 df['clean_text_remove_stopwords'] = df['clean_text_remove_numbers'].apply(remove_stopwords)
9 print("\nData Frame after removing stopwords without lowercase:")
10 df

```

Data Frame after removing stopwords without lowercase:

	text	clean_text	clean_text_remove_numbers	clean_text_remove_stopwo
0	reviews	reviews	reviews	revi
1	One of the most anticipated films of the year ...	One of the most anticipated films of the year ...	One of the most anticipated films of the year ...	One anticipated films year m people inclu
2	You'll have to have your wits about you and yo...	You ll have to have your wits about you and yo...	You ll have to have your wits about you and yo...	wits brain fully switched watcl Oppenheim
3	I'm a big fan of Nolan's work so was really lo...	I m a big fan of Nolan s work so was really lo...	I m a big fan of Nolan s work so was really lo...	big fan Nolan work really loo forward unc
4	"Oppenheimer" is a biographical thriller film ...	Oppenheimer is a biographical thriller film ...	Oppenheimer is a biographical thriller film ...	Oppenheimer biographical thi film writte
...
996	It's isn't a masterpiece. It's a decent biopic...	It s isn t a masterpiece It s a decent biopic...	It s isn t a masterpiece It s a decent biopic...	masterpiece decent bi interesting person
997	I'm a big Nolan fan. Maybe this one just wasn't...	I m a big Nolan fan Maybe this one just wasn ...	I m a big Nolan fan Maybe this one just wasn ...	big Nolan fan Maybe one m promoted stor
998	My Review - Oppenheimer\nMy Rating Ten plus	My Review Oppenheimer My Rating Ten plus 10	My Review Oppenheimer My Rating Ten plus I...	Review Oppenheimer Rating plus Cinemas nc

```

1 # 4: Lowercase all texts
2 df['clean_text_lowercase'] = df['clean_text_remove_stopwords'].apply(lambda x: x.lower())
3 print("\nData Frame after converting texts to lowercase:")
4 df

```

Data Frame after converting texts to lowercase:

	text	clean_text	clean_text_remove_numbers	clean_text_remove_stopwords
0	reviews	reviews	reviews	review
1	One of the most anticipated films of the year ...	One of the most anticipated films of the year ...	One of the most anticipated films of the year ...	One anticipated films year mai people include
2	You'll have to have your wits about you and yo...	You ll have to have your wits about you and yo...	You ll have to have your wits about you and yo...	wits brain fully switched watchir Oppenheimer
3	I'm a big fan of Nolan's work so was really lo...	I m a big fan of Nolan s work so was really lo...	I m a big fan of Nolan s work so was really lo...	big fan Nolan work really lookir forward unde
4	"Oppenheimer" is a biographical thriller film ...	Oppenheimer is a biographical thriller film ...	Oppenheimer is a biographical thriller film ...	Oppenheimer biographical thrill film written
...
996	It's isn't a masterpiece. It's a decent biopic...	It s isn t a masterpiece It s a decent biopic...	It s isn t a masterpiece It s a decent biopic...	masterpiece decent biop interesting person g
997	I'm a big Nolan fan. Maybe this one just wasn'...	I m a big Nolan fan Maybe this one just wasn ...	I m a big Nolan fan Maybe this one just wasn ...	big Nolan fan Maybe one mov promoted story i
998	My Review - Oppenheimer\nMy Rating Ten plus 10...	My Review Oppenheimer My Rating Ten plus 10 ...	My Review Oppenheimer My Rating Ten plus I...	Review Oppenheimer Rating T plus Cinemas now
999	Nolan is good at constructing complicated timi...	Nolan is good at constructing complicated timi...	Nolan is good at constructing complicated timi...	Nolan good constructir complicated timing eve
1000	It saddens me that so many people are mistakin...	It saddens me that so many people are mistakin...	It saddens me that so many people are mistakin...	saddens many people mistaki bigger louder qu

001 rows × 5 columns

```

1 #5. Stemming
2 stemmer = PorterStemmer()
3 def apply_stemming(text):
4     words = nltk.word_tokenize(text)
5     stemmed_words = [stemmer.stem(word) for word in words]

```

```

5     stemmed_words = [stemmer.stem(word) for word in words]
6     return ' '.join(stemmed_words)
7 df['clean_text_stemmed'] = df['clean_text_lowercase'].apply(apply_stemming)
8 print("\nData Frame after applying stemming:")
9 df

```

Applying stemming:

	clean_text	clean_text_remove_numbers	clean_text_remove_stopwords	clean_text_lower
reviews	reviews	reviews	reviews	rev
st	One of the most anticipated films of the year ...	One of the most anticipated films of the year ...	One anticipated films year many people include...	one anticipated year many pe inclu
..				
o	You ll have to have your wits about you and yo...	You ll have to have your wits about you and yo...	wits brain fully switched watching Oppenheimer...	wits brain fully swit watching oppenhein
is				
d				
..				
of	I m a big fan of Nolan s work so was really lo...	I m a big fan of Nolan s work so was really lo...	big fan Nolan work really looking forward unde...	big fan nolan work r looking forward un
o				
..				
is	Oppenheimer is a biographical thriller film ...	Oppenheimer is a biographical thriller film ...	Oppenheimer biographical thriller film written...	oppenhe biographical thriller writt
al				
..				
...	
a	It s isn t a masterpiece It s a decent biopic...	It s isn t a masterpiece It s a decent biopic...	masterpiece decent biopic interesting person g...	masterpiece de biopic interesting pe
a				
..				
n	I m a big Nolan fan Maybe this one just wasn ...	I m a big Nolan fan Maybe this one just wasn ...	big Nolan fan Maybe one movie promoted story i...	big nolan fan maybe movie promoted stoi
is				
..				
-	My Review Oppenheimer My Rating Ten plus 10	My Review Oppenheimer My Rating Ten plus I...	Review Oppenheimer Rating Ten plus Cinemas now...	review oppenhe rating ten plus cine n
ly				
is				
..				
at	Nolan is good at constructing complicated timi...	Nolan is good at constructing complicated timi...	Nolan good constructing complicated timing eve...	nolan good constru complicated timing e
g				
..				
at	It saddens me that so many people are mistakin...	It saddens me that so many people are mistakin...	saddens many people mistaking bigger louder qu...	saddens many pe mistaking bigger lo
e				
..				


```
1 #6. Lemmatization
2 lemmatizer = WordNetLemmatizer()
3 def apply_lemmatization(text):
4     words = nltk.word_tokenize(text)
5     lemmatized_words = [lemmatizer.lemmatize(word) for word in words]
6     return ' '.join(lemmatized_words)
7 df['clean_text_lemmatized'] = df['clean_text_stemmed'].apply(apply_lemmatization)
8 print("\nData Frame after applying lemmatization:")
9 df
```

ext_remove_numbers	clean_text_remove_stopwords	clean_text_lowercase	clean_text_stemmer
reviews	reviews	reviews	review
of the most anticipated films of the year ...	One anticipated films year many people include...	one anticipated films year many people include...	one anticip film yea mani peopl includ oppenh..
have to have your wits about you and yo...	wits brain fully switched watching Oppenheimer...	wits brain fully switched watching oppenheimer...	wit brain fulli switch watch oppenhein could e..
big fan of Nolan s work so was really lo...	big fan Nolan work really looking forward unde...	big fan nolan work really looking forward unde...	big fan nolan worl realli look forward underst..
aimer is a biographical thriller film ...	Oppenheimer biographical thriller film written...	oppenheimer biographical thriller film written...	oppenheim biograph thriller film writer direc..
...
it a masterpiece It s a decent biopic...	masterpiece decent biopic interesting person g...	masterpiece decent biopic interesting person g...	masterpiec decen biopic interest persor good ..
Nolan fan Maybe this one just wasn ...	big Nolan fan Maybe one movie promoted story i...	big nolan fan maybe one movie promoted story i...	big nolan fan mayt one movi promot stor inven..
view Oppenheimer My Rating Ten plus I...	Review Oppenheimer Rating Ten plus Cinemas now...	review oppenheimer rating ten plus cinemas now...	review oppenhein rate ten plu cinem nowif mov..
s good at constructing complicated timi...	Nolan good constructing complicated timing eve...	nolan good constructing complicated timing eve...	nolan good construc complic time even good t..
saddens me that so many people are mistakin...	saddens many people mistaking bigger louder qu...	saddens many people mistaking bigger louder qu...	sadden mani peop mistak bigger loude qualiti..

```

1 # Save the cleaned data to a new CSV file
2 df.to_csv('cleaned_data_.csv', index=False)
3 print("\nCleaned data saved ")

```

Cleaned data saved

✓ Question 3 (30 points)

Write a python program to **conduct syntax and structure analysis of the clean text** you just saved above. The syntax and structure analysis includes:

- (1) **Parts of Speech (POS) Tagging:** Tag Parts of Speech of each word in the text, and calculate the total number of N(oun), V(erb), Adj(ective), Adv(erb), respectively.
- (2) **Constituency Parsing and Dependency Parsing:** print out the constituency parsing trees and dependency parsing trees of all the sentences. Using one sentence as an example to explain your understanding about the constituency parsing tree and dependency parsing tree.
- (3) **Named Entity Recognition:** Extract all the entities such as person names, organizations, locations, product names, and date from the clean texts, calculate the count of each entity.

```
1 import pandas as pd
2 import nltk
3 from collections import Counter
4 # Download NLTK resources
5 nltk.download('punkt')
6 nltk.download('averaged_perceptron_tagger')

[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]   Unzipping taggers/averaged_perceptron_tagger.zip.
True
```

```

1 # Read the data from cleaned_data_.csv
2 df = pd.read_csv('cleaned_data_.csv')
3
4 # Part 1: Parts of Speech (POS) Tagging
5 def pos_tagging(text):
6     tokens = nltk.word_tokenize(text)
7     pos_tags = nltk.pos_tag(tokens)
8     return pos_tags
9 # Iterate through each row and print the POS tagging on a new line
10 for index, row in df.iterrows():
11     pos_tags = pos_tagging(row['clean_text'])
12     print(f"POS tagging for row {index + 1}: \n{pos_tags}\n")
13     noun_count = verb_count = adj_count = adv_count = 0
14     for _, pos in pos_tags:
15         if pos.startswith('N'):
16             noun_count += 1
17         elif pos.startswith('V'):
18             verb_count += 1
19         elif pos.startswith('JJ'):
20             adj_count += 1
21         elif pos.startswith('RB'):
22             adv_count += 1
23     print(f"Total Nouns: {noun_count}")
24     print(f"Total Verbs: {verb_count}")
25     print(f"Total Adjectives: {adj_count}")
26     print(f"Total Adverbs: {adv_count}")

```

Streaming output truncated to the last 5000 lines.

Total Adjectives: 9

Total Adverbs: 11

POS tagging for row 169: [('At', 'IN'), ('its', 'PRP\$'), ('core', 'NN'), ('Oppenheimer', 'NNP'), ('is

Total Nouns: 125

Total Verbs: 66

Total Adjectives: 47

Total Adverbs: 22

POS tagging for row 170: [('I', 'PRP'), ('just', 'RB'), ('got', 'VBD'), ('out', 'IN'), ('of', 'IN'),

Total Nouns: 87

Total Verbs: 58

Total Adjectives: 20

Total Adverbs: 22

POS tagging for row 171: [('Authentic', 'JJ'), ('audio', 'NN'), ('visual', 'JJ'), ('journey', 'NN'),

Total Nouns: 36

Total Verbs: 8

Total Adjectives: 14

Total Adverbs: 4

POS tagging for row 172: [('It', 'PRP'), ('s', 'VBZ'), ('isn', 'JJ'), ('t', 'IN'), ('a', 'DT'), ('mas

Total Nouns: 26

Total Verbs: 22

Total Adjectives: 15

Total Adverbs: 12

POS tagging for row 173: [('I', 'PRP'), ('m', 'VBP'), ('a', 'DT'), ('big', 'JJ'), ('Nolan', 'NN'), (

Total Nouns: 147

Total Verbs: 103

Total Adjectives: 51

Total Adverbs: 47

POS tagging for row 174: [('My', 'PRP\$'), ('Review', 'NNP'), ('Oppenheimer', 'NNP'), ('My', 'NNP'), (

Total Nouns: 220

```

Total Verbs: 94
Total Adjectives: 52
Total Adverbs: 26
POS tagging for row 175:[('Nolan', 'NN'), ('is', 'VBZ'), ('good', 'JJ'), ('at', 'IN'), ('constructir

Total Nouns: 76
Total Verbs: 32
Total Adjectives: 31
Total Adverbs: 24
POS tagging for row 176:[('It', 'PRP'), ('saddens', 'VBZ'), ('me', 'PRP'), ('that', 'IN'), ('so', 'F

Total Nouns: 65
Total Verbs: 40
Total Adjectives: 35
Total Adverbs: 17
POS tagging for row 177:[('One', 'CD'), ('of', 'IN'), ('the', 'DT'), ('most', 'RBS'), ('anticipated

Total Nouns: 68
Total Verbs: 77
Total Adjectives: 42
Total Adverbs: 32

```

```

1 # Install the required modules
2 !pip install benepar
3 !pip install tensorflow
4 !pip install tensorflow==2.8.0

1 # Downloading the required models
2 import benepar
3 import spacy.cli
4 benepar.download('benepar_en3')
5 spacy.cli.download("en_core_web_sm")
6
7 # Importing the libraries
8 import sys
9 import spacy
10 from spacy import displacy
11 parser = benepar.Parser("benepar_en3")
12 nlp = spacy.load('en_core_web_sm')
13 options = {'compact': True, 'font': 'Arial black', 'distance': 100}
14 # Parsing all the sentences in clean_text
15 for sentence in df['clean_text']:
16     try:
17         tree = parser.parse(sentence)
18         print(tree)
19     except:
20         print("No Parse Tree")
21         continue
22 # Printing the parse trees using spacy module
23 for sentence in df['clean_text']:
24     doc = nlp(sentence)
25     displacy.render(doc, style='dep', options=options, jupyter=True)

```

Collecting benepar

Downloading benepar-0.2.0.tar.gz (33 kB)

Preparing metadata (setup.py) ... done

Requirement already satisfied: nltk>=3.2 in /usr/local/lib/python3.10/dist-packages (fr

Requirement already satisfied: spacy>=2.0.9 in /usr/local/lib/python3.10/dist-packages

Requirement already satisfied: torch>=1.6.0 in /usr/local/lib/python3.10/dist-packages

Collecting torch-struct>=0.5 (from benepar)

Downloading torch_struct-0.5-py3-none-any.whl (34 kB)

Requirement already satisfied: tokenizers>=0.9.4 in /usr/local/lib/python3.10/dist-pack

Requirement already satisfied: transformers[tokenizers,torch]>=4.2.2 in /usr/local/lib/

Requirement already satisfied: protobuf in /usr/local/lib/python3.10/dist-packages (fro

Requirement already satisfied: sentencepiece>=0.1.91 in /usr/local/lib/python3.10/dist-

Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from n

Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from

Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packag

Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nl

Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.10

Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.10

Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.10/d

Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.10/dist-pa

Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.10/dist-

Requirement already satisfied: thinc<8.3.0,>=8.2.2 in /usr/local/lib/python3.10/dist-pa

Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.10/dist-p

Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.10/dist-pa

Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.10/dis

Requirement already satisfied: weasel<0.4.0,>=0.1.0 in /usr/local/lib/python3.10/dist-p

Requirement already satisfied: typer<0.10.0,>=0.3.0 in /usr/local/lib/python3.10/dist-p

Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in /usr/local/lib/python3.10/di

Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dis

Requirement already satisfied: pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/p

Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from

Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (f

Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packag

Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.10/dis

Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.10/dist-packages

Requirement already satisfied: huggingface_hub<1.0,>=0.16.4 in /usr/local/lib/python3.1

Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (fro

Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-pack

Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from t

Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (fro

Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from

Requirement already satisfied: triton==2.1.0 in /usr/local/lib/python3.10/dist-packages

Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (

Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-pac

Collecting accelerate>=0.21.0 (from transformers[tokenizers,torch]>=4.2.2->benepar)

Downloading accelerate-0.27.2-py3-none-any.whl (279 kB)

280.0/280.0 kB 11.3 MB/s eta 0:00:00

Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from

Requirement already satisfied: annotated-types>=0.4.0 in /usr/local/lib/python3.10/dist