

Experiment No.5
Perform Natural language Entity Extraction from medical reports
Date of Performance: 04/09/2024
Date of Submission: 11/09/2024

Aim: Perform Natural language Entity Extraction from medical reports

Objective: The objective of this experiment is to perform named entity recognition (NER) using SpaCy and related libraries on MIMIC III dataset.

Theory:

NER stands for Named Entity Recognition, which is a subtask of Natural Language Processing (NLP). It involves identifying and classifying named entities (such as names of people, organizations, locations, dates, and more) within a text. The goal of NER is to extract structured information from unstructured text data and to recognize specific entities mentioned in the text.

For example, given the sentence: "Apple Inc. was founded by Steve Jobs in Cupertino on April 1, 1976," a named entity recognition system would identify and categorize "Apple Inc." as an organization, "Steve Jobs" as a person, "Cupertino" as a location, and "April 1, 1976" as a date.

NER has various applications in NLP, including information retrieval, question answering, sentiment analysis, text summarization, and more. It plays a crucial role in understanding the context and semantics of a text by identifying and categorizing entities, which can help in extracting meaningful insights from large amounts of text data.

spaCy:

spaCy is a fast and efficient NLP library that provides pre-trained models for NER. It's known for its speed and accuracy.

```
python
import spacy
```

```
nlp = spacy.load("en_core_web_sm")
doc = nlp("Barack Obama was born in Hawaii.")
```

```
for ent in doc.ents:
```

```
    print(ent.text, ent.label_)
```

Using Named Entity Recognition (NER) in Natural Language Processing (NLP) offers several advantages that enhance the understanding and processing of text data:

Information Extraction: NER allows you to extract structured information from unstructured text. This is especially valuable for tasks like populating databases, creating summaries, or generating reports from large amounts of text data.

Entity Categorization: NER categorizes entities into predefined classes such as persons, organizations, locations, dates, and more. This categorization provides context and semantic meaning, enabling more sophisticated analysis of the text.

Improved Search and Retrieval: By identifying and tagging named entities, NER can improve the accuracy and relevance of search results in applications like search engines, document retrieval systems, and recommendation systems.

Question Answering: NER is essential for question answering systems, as it helps identify relevant entities in the text that are related to the question being asked. This can lead to more accurate and informative answers.

Entity Linking: NER can be used to link recognized entities to knowledge bases, such as Wikipedia or other domain-specific databases, enriching the information by connecting it to external resources.

Named Entity Disambiguation: NER can help in disambiguating the

context of an entity. For instance, the same name "Apple" could refer to a fruit or a technology company. NER can help distinguish between these different meanings based on the context.

Sentiment Analysis: In sentiment analysis, recognizing entities can help determine the sentiment towards specific entities. This is useful for understanding public opinion about companies, products, or individuals.

Event Extraction: NER can aid in extracting events and relationships between entities in a text. This is useful for tasks like event detection, timeline generation, and understanding connections between entities.

Regulatory Compliance and Data Security: In industries like finance and healthcare, NER can assist in identifying sensitive information, like personal names, medical terms, and financial figures, ensuring compliance with data protection regulations.

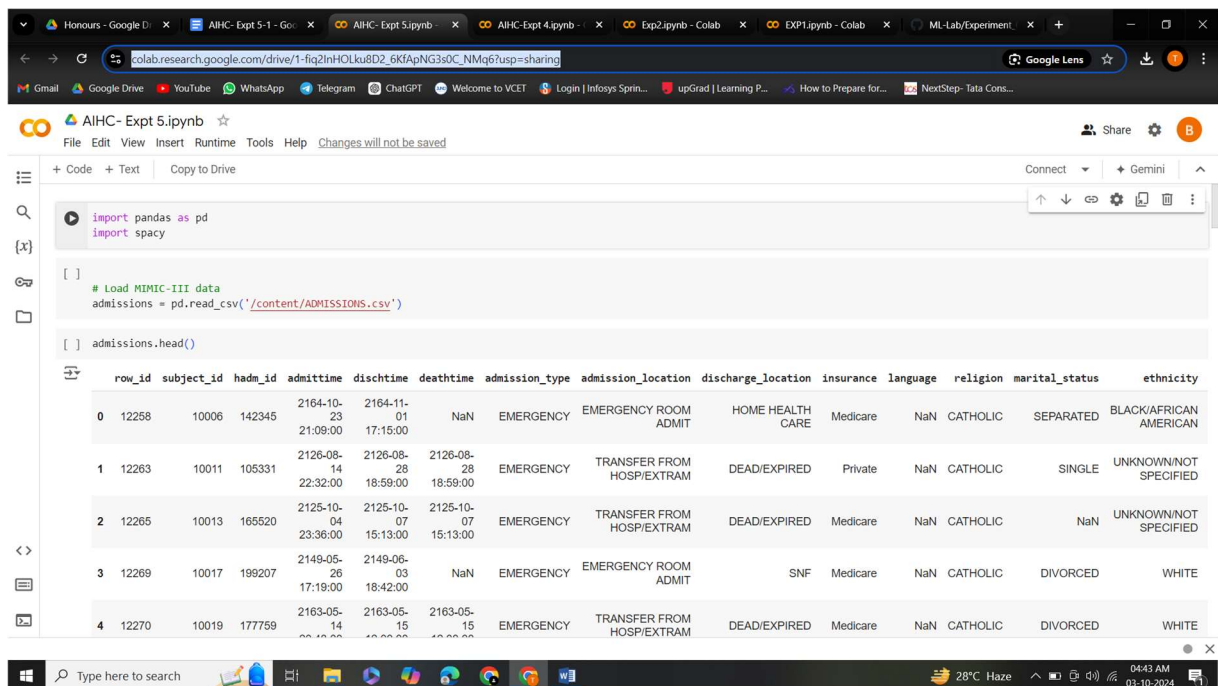
Language Translation and Generation: NER can improve the accuracy of machine translation and text generation by preserving the names of entities, resulting in more coherent and contextually relevant output.

Data Analysis and Visualization: By extracting and categorizing entities, NER can facilitate data analysis and visualization, making it easier to identify trends, patterns, and relationships in large text datasets.

Automated Document Summarization: NER can be used to identify key entities in a document, which in turn can be used to generate informative and concise document summaries.

Overall, NER is a foundational tool in NLP that adds structure, context, and meaning to text data, enabling a wide range of applications that require understanding and processing human language.

Code: -



The screenshot shows a Google Colab notebook titled "AIHC- Expt 5.ipynb". The code cell contains the following Python code:

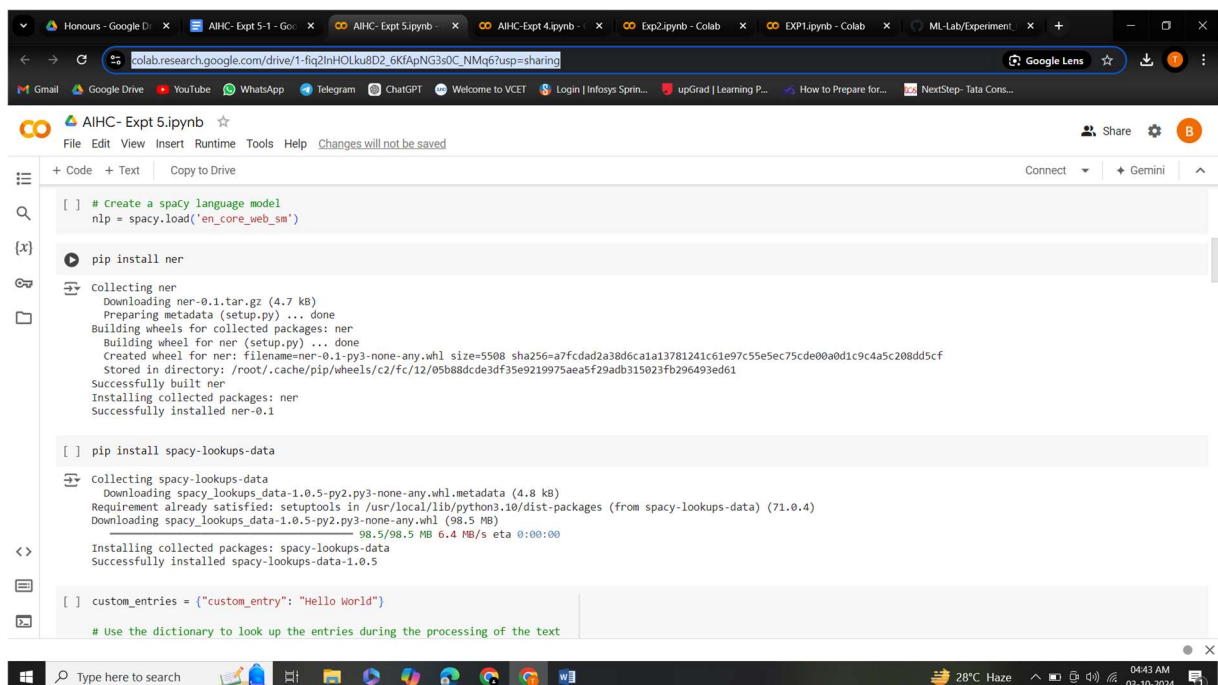
```
import pandas as pd
import spacy

# Load MIMIC-III data
admissions = pd.read_csv('/content/ADMISSIONS.csv')

admissions.head()
```

The output of the `admissions.head()` command is a pandas DataFrame with 15 columns: `row_id`, `subject_id`, `hadm_id`, `admittime`, `dischtime`, `deathtime`, `admission_type`, `admission_location`, `discharge_location`, `insurance`, `language`, `religion`, `marital_status`, and `ethnicity`. The first five rows of data are displayed:

	row_id	subject_id	hadm_id	admittime	dischtime	deathtime	admission_type	admission_location	discharge_location	insurance	language	religion	marital_status	ethnicity
0	12258	10006	142345	2164-10-23 21:09:00	2164-11-01 17:15:00	NaN	EMERGENCY	EMERGENCY ROOM ADMIT	HOME HEALTH CARE	Medicare	NaN	CATHOLIC	SEPARATED	BLACK/AFRICAN AMERICAN
1	12263	10011	105331	2126-08-14 22:32:00	2126-08-28 18:59:00	2126-08-28 18:59:00	EMERGENCY	TRANSFER FROM HOSP/EXTRAM	DEAD/EXPIRED	Private	NaN	CATHOLIC	SINGLE	UNKNOWN/NOT SPECIFIED
2	12265	10013	165520	2125-10-04 23:36:00	2125-10-07 15:13:00	2125-10-07 15:13:00	EMERGENCY	TRANSFER FROM HOSP/EXTRAM	DEAD/EXPIRED	Medicare	NaN	CATHOLIC	NaN	UNKNOWN/NOT SPECIFIED
3	12269	10017	199207	2149-05-26 17:18:00	2149-06-03 18:42:00	NaN	EMERGENCY	EMERGENCY ROOM ADMIT	SNF	Medicare	NaN	CATHOLIC	DIVORCED	WHITE
4	12270	10019	177759	2163-05-14 00:00:00	2163-05-15 00:00:00	2163-05-15 00:00:00	EMERGENCY	TRANSFER FROM HOSP/EXTRAM	DEAD/EXPIRED	Medicare	NaN	CATHOLIC	DIVORCED	WHITE



The screenshot shows a Google Colab notebook titled "AIHC- Expt 5.ipynb". The code cell contains the following Python code:

```
# Create a spacy language model
nlp = spacy.load('en_core_web_sm')

pip install ner

collecting ner
Downloading ner-0.1.tar.gz (4.7 kB)
Preparing metadata (setup.py) ... done
Building wheels for collected packages: ner
Building wheel for ner (setup.py) ... done
Created wheel for ner: filename=ner-0.1-py3-none-any.whl size=5508 sha256=a7fcdad2a38d6ca1a13781241c61e97c55e5ec75cde00a0d1c9c4a5c208dd5cf
Stored in directory: /root/.cache/pip/wheels/c2/fc/12/05b88dcde3df35e9219975aea5f29adb315023fb296493ed61
Successfully built ner
Installing collected packages: ner
Successfully installed ner-0.1

pip install spacy-lookups-data

collecting spacy-lookups-data
Downloading spacy_lookups_data-1.0.5-py3-none-any.whl.metadata (4.8 kB)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from spacy-lookups-data) (71.0.4)
Downloading spacy_lookups_data-1.0.5-py3-none-any.whl (98.5 MB)
98.5/98.5 MB 6.4 MB/s eta 0:00:00
Installing collected packages: spacy-lookups-data
Successfully installed spacy-lookups-data-1.0.5

custom_entries = {"custom_entry": "Hello World"}

# Use the dictionary to look up the entries during the processing of the text
```

```
colab.research.google.com/drive/1-fiq2InHOLku8D2_6KfApNG3s0C_NMq67usp=sharing

AIHC- Expt 5.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved

+ Code + Text Copy to Drive
Connect Gemini

texts = admissions['diagnosis']
labels = ["DIAGNOSIS"] # Assuming you have a list of labels

# Begin training
nlp.begin_training()
for i in range(80):
    # Train on 80% of the data
    offsets = [(t, 0, len(t)) for t in texts]
    nlp.update([(i, (list(texts), list(labels))), sents=offsets])
    # Train on the remaining 20%
    nlp.update([(i, (list(texts), list(labels))), sents=offsets])

# End training
nlp.end_training()

-----
ImportError: Traceback (most recent call last)
<ipython-input-32-d73b549ddc8d> in <cell line: 4>()
      2 from spacy import Language
      3 from spacy.pipeline import EntityRecognizer
----> 4 from spacy.lookups import LookupTables
      5
      6 nlp = spacy.load('en_core_web_sm')

ImportError: cannot import name 'LookupTables' from 'spacy.lookups' (/usr/local/lib/python3.10/dist-packages/spacy/lookups.py)

-----
NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.

To view examples of installing some common dependencies, click the
```

```
colab.research.google.com/drive/1-fiq2InHOLku8D2_6KfApNG3s0C_NMq67usp=sharing

AIHC- Expt 5.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved

+ Code + Text Copy to Drive
Connect Gemini

import pandas as pd
import spacy

# Load spaCy model
nlp = spacy.load("en_core_web_sm")

# Load the CSV file into a DataFrame
admissions_df = pd.read_csv('/content/ADMISSIONS.csv')

# Print the columns to verify their names
print("Columns in DataFrame:", admissions_df.columns)

# Inspect the first few rows of the DataFrame
print(admissions_df.head())

# Use the 'diagnosis' column for NER
text_column = 'diagnosis' # Adjust this if you want to use a different column

# Extract text data from the 'diagnosis' column
text_data = admissions_df[text_column].dropna().astype(str).tolist()

# Function to perform NER
def perform_ner(text):
    doc = nlp(text)
    entities = [(ent.text, ent.label_) for ent in doc.ents]
    return entities

# Process the first few entries for demonstration
for text in text_data[:5]: # Limiting to first 5 for demonstration
```

The screenshot shows a Google Colab notebook titled "AIHC- Expt 5.ipynb". The code cell contains a DataFrame with columns: 'row_id', 'subject_id', 'hadm_id', 'admittime', 'disctime', 'deathtime', 'admission_type', 'admission_location', 'discharge_location', 'insurance', 'language', 'religion', 'marital_status', 'ethnicity', 'edregtime', 'edouttime', 'diagnosis', 'hospital_expire_flag', and 'has_chartevents_data'. The DataFrame is displayed in a table format with 5 rows of data.

row_id	subject_id	hadm_id	admittime	disctime
0	12258	10006	2164-10-23 21:09:00	2164-11-01 17:15:00
1	12263	10011	2126-08-14 22:32:00	2126-08-28 18:59:00
2	12265	10013	2125-10-04 23:36:00	2125-10-07 15:13:00
3	12269	10017	2149-05-26 17:19:00	2149-06-03 18:42:00
4	12270	10019	2163-05-14 20:43:00	2163-05-15 12:00:00

The screenshot shows a Google Colab notebook titled "AIHC- Expt 5.ipynb". The code cell contains a DataFrame with columns: 'diagnosis', 'hospital_expire_flag', and 'has_chartevents_data'. The DataFrame is displayed in a table format with 5 rows of data.

diagnosis	hospital_expire_flag
0	0
1	1
2	1
3	0
4	1

Google Collaboratory Link: -
https://colab.research.google.com/drive/1-fiq2InHOLku8D2_6KfApNG3s0C_NMq6?usp=sharing

Conclusion: -

Comment on the role of Named Entity Recognition (NER) played in Natural Language Processing, and how it enhance the understanding and processing of unstructured text data

Named Entity Recognition (NER) plays a crucial role in the field of Natural Language Processing (NLP) by improving the understanding, analysis, and processing of unstructured text data. It enhances the ability to automatically identify and categorize important entities such as people, organizations, locations, dates, and more within large text corpora. By extracting these specific entities, NER facilitates a structured approach to analyzing vast amounts of information, transforming raw text into valuable insights that can be used for various applications.