Name: Mohd Talha Ansari , Class: TY (AIML) – D1

Experiment no:1 Use of named entity recognition information extraction technique

Program:

```
pip install spacy

python -m spacy download en_core_web_sm import

spacy

# Load the SpaCy model nlp =

spacy.load("en_core_web_sm")

# Sample text text

= """

Apple Inc. is looking at buying U.K. startup for $1 billion.

Steve Jobs founded Apple in Cupertino, California.

"""

# Process the text with SpaCy

doc = nlp(text) for ent in

doc.ents:

  print(f"Entity: {ent.text}, Label: {ent.label_}")
```

Output:

```
Entity: Apple Inc., Label: ORG

Entity: U.K., Label: GPE

Entity: $1 billion, Label: MONEY

Entity: Steve Jobs, Label: PERSON

Entity: Apple, Label: ORG

Entity: Cupertino, Label: GPE

Entity: California, Label: GPE
```

Name: Mohd Talha Ansari
Class: TY (AIML)
Batch: A3
Roll No: 47

Experiment No:2   Impement sentiment ANALYSIS techniques for classifying the data into positie,negative or neutral

Program:

```
pip install textblob python -m

textblob.download_corpora from

textblob import TextBlob


# Function to classify sentiment def

classify_sentiment(text):     blob =

TextBlob(text)     # Get the polarity

score     polarity =

blob.sentiment.polarity


   # Classify based on polarity     if

polarity > 0:

       return "Positive"     elif

polarity < 0:

       return "Negative"     else:

       return "Neutral"


# Sample texts texts

= [

   "I love this product! It's amazing.",

   "I'm really unhappy with the service.",
    "It's an average experience.",
    "This is the best day ever!",
    "I'm not sure how I feel about this."
]

# Analyze sentiment for each text for
text in texts:
    sentiment = classify_sentiment(text)     print(f"Text:
{text}\nSentiment: {sentiment}\n")
```

Output:

```
Text: I love this product! It's amazing.
Sentiment: Positive

Text: I'm really unhappy with the service.
Sentiment: Negative

Text: It's an average experience.
Sentiment: Neutral

Text: This is the best day ever!
Sentiment: Positive

Text: I'm not sure how I feel about this.
Sentiment: Neutral
```

Name: Mohd Talha Ansari

Class: TY (AIML)

Batch: A3

Roll No: 47

Experiment No: 3   Use of nlp techniques for text summarization Program:

Extractive summary

```
pip install nltk transformers torch
import nltk from nltk.tokenize import
sent_tokenize from nltk.corpus import
stopwords from collections import
defaultdict import numpy as np
# Download required NLTK resources
nltk.download('punkt') nltk.download('stopwords')
# Function for extractive summarization def
extractive_summary(text, num_sentences=2):
 # Tokenize the text into sentences    sentences =
sent_tokenize(text) # Preprocess and calculate
word frequencies     stop_words =
set(stopwords.words("english"))
word_frequencies = defaultdict(int)     for word in
nltk.word_tokenize(text.lower()):
    if word not in stop_words and word.isalpha():
word_frequencies[word] += 1
```

```
    # Calculate sentence scores based on word frequencies
sentence_scores = defaultdict(int)     for i, sentence in
enumerate(sentences):
        for word in nltk.word_tokenize(sentence.lower()):            if
word in word_frequencies:                       sentence_scores[i] +=
word_frequencies[word]


    # Select the top sentences
    summarized_sentences = sorted(sentence_scores, key=sentence_scores.get,
reverse=True)[:num_sentences]


    # Return the summary     return ' '.join([sentences[j] for j in
sorted(summarized_sentences)])


# Sample text text
= """
Natural Language Processing (NLP) is a subfield of artificial intelligence that focuses on the interaction
between computers and humans through natural language. The ultimate objective of NLP is to read,
understand, and derive meaning from human languages in a valuable way. NLP is used in various
applications,  such as chatbots, translation services, and sentiment analysis. """


# Generate extractive summary summary
= extractive_summary(text)
print("Extractive Summary:")
print(summary)
```

output:

```
Extractive Summary:

NLP is used in various applications, such as chatbots, translation services, and sentiment analysis.

Natural Language Processing (NLP) is a subfield of artificial intelligence that focuses on the interaction
between computers and humans through natural language.
```

Program:Abstractive summary

```
from transformers import pipeline #

Initialize the summarization pipeline

summarizer = pipeline("summarization")

# Sample text text

= """

Natural Language Processing (NLP) is a subfield of artificial intelligence that focuses on the interaction

between computers and humans through natural language. The ultimate objective of NLP is to read,

understand, and derive meaning from human languages in a valuable way. NLP is used in various
applications,  such as chatbots, translation services, and sentiment analysis.
"""

# Generate abstractive summary abstractive_summary = summarizer(text, max_length=50,

min_length=25, do_sample=False) print("Abstractive Summary:")

print(abstractive_summary[0]['summary_text'])
```

Output:

```
Abstractive Summary:

Natural Language Processing (NLP) is an AI subfield that focuses on human-computer interaction through
language.

NLP is applied in various fields such as chatbots and sentiment analysis.
```

Name: Mohd Talha Ansari

Class: TY (AIML)

Batch: A3

Roll No: 47

Experiment no:5 Implement a code for Aspect mining and topic modeling

Program

```
pip install nltk gensim sklearn pandas


import pandas as pd import nltk from

nltk.tokenize import word_tokenize from

nltk.corpus import stopwords from gensim

import corpora from gensim.models import

LdaModel import re import nltk

nltk.download('averaged_perceptron_tagger')


# Download necessary NLTK resources

nltk.download('punkt') nltk.download('stopwords')


# Sample data: replace this with your dataset data

= {

    'reviews': [

        "The battery life is amazing, but the camera is mediocre.",

        "I love the design of the phone, but the performance is disappointing.",

        "Great value for money! The screen quality is excellent.",
        "The battery drains quickly, but the software is user-friendly."
    ]
}

# Create a DataFrame df
= pd.DataFrame(data)

# Preprocessing function
def preprocess_text(text):
# Lowercase     text =
text.lower()
    # Remove punctuation
    text = re.sub(r'[^\w\s]', '', text)
    # Tokenize     tokens = word_tokenize(text)     # Remove
stopwords     stop_words = set(stopwords.words('english'))
tokens = [word for word in tokens if word not in stop_words]
return tokens

# Preprocess reviews df['tokens'] =
df['reviews'].apply(preprocess_text)

# Aspect Mining: Extracting potential aspects (nouns)
aspects = [] for tokens in df['tokens']:     for token in
tokens:
        if nltk.pos_tag([token])[0][1] in ['NN', 'NNS']:  # Noun            aspects.append(token)
```

```
# Get unique aspects unique_aspects
= set(aspects)

print("Extracted Aspects:") print(unique_aspects)

# Topic Modeling with LDA
# Create a dictionary and corpus for LDA dictionary =
corpora.Dictionary(df['tokens']) corpus =
[dictionary.doc2bow(tokens) for tokens in df['tokens']]

# Train LDA model lda_model = LdaModel(corpus, num_topics=2,
id2word=dictionary, passes=10)

# Print topics print("\nTopics found:") for
idx, topic in lda_model.print_topics(-1):
   print(f"Topic {idx}: {topic}")
```

output:

```
 Extracted Aspects:
{'money', 'screen', 'excellent', 'design', 'quality', 'battery', 'value', 'phone', 'life', 'software', 'performance',
'mediocre', 'love', 'camera', 'drains'}
Topics found:
Topic 0: 0.125*"battery" + 0.075*"quickly" + 0.075*"userfriendly" + 0.075*"software" + 0.075*"drains" +
0.075*"life" + 0.075*"mediocre" + 0.075*"amazing" + 0.075*"camera" + 0.025*"love"
Topic 1: 0.071*"quality" + 0.071*"screen" + 0.071*"value" + 0.071*"money" + 0.071*"great" +
0.071*"excellent" + 0.071*"phone" + 0.071*"performance" + 0.071*"design" + 0.071*"disappointing"
```

Name: Mohd Talha Ansari

Class: TY (AIML)

Batch: A3

Roll No: 47

Experiment No 4: Implemnt a simple machine translation from one language to another

Program:

```
pip install transformers torch from

transformers import pipeline


# Initialize the translation pipeline for English to French translator

= pipeline("translation_en_to_fr")

# Sample text to translate text = "Machine translation is a fascinating

field of artificial intelligence."

# Perform translation translated_text =

translator(text, max_length=40)


# Output the translated text print("Translated

Text:")

print(translated_text[0]['translation_text'])
```

Output:

```
Translated Text:

La traduction automatique est un domaine fascinant de l'intelligence artificielle.
```