Assignment Code: DS-AG-028

NLP Introduction & Text Processing | Assignment

Instructions: Carefully read each question. Use Google Docs, Microsoft Word, or a similar tool to create a document where you type out each question along with its answer. Save the document as a PDF, and then upload it to the LMS. Please do not zip or archive the files before uploading them. Each question carries 20 marks.

Total Marks: 200

Question 1: What is Computational Linguistics and how does it relate to NLP?

Answer:

Computational Linguistics is the scientific study of language from a computational perspective. It involves using algorithms, models, and computer programs to analyze and generate human language. It relates directly to Natural Language Processing (NLP) because NLP applies computational linguistic principles to build systems that can understand, interpret, and manipulate human language, enabling tasks like machine translation, speech recognition, and text analysis.

Question 2: Briefly describe the historical evolution of Natural Language Processing.

Answer:

NLP began in the 1950s with rule-based systems and early machine translation. In the 1980s, statistical approaches emerged using probabilistic models. The 2000s saw machine learning methods, and since 2010 deep learning and neural networks revolutionized NLP with architectures like LSTM, Transformers, and large pre-trained models (e.g., BERT, GPT). Today, NLP underpins applications such as chatbots, automated translators, and sentiment analysis.

Question 3: List and explain three major use cases of NLP in today's tech industry.

Answer:

- Chatbots & Virtual Assistants: Automated customer service, powered by NLP, interprets and responds to human language queries in natural dialog.
- Sentiment Analysis: Brands use NLP to analyze social media and reviews to gauge public sentiment towards products or services.
- Machine Translation: Systems like Google Translate use NLP and deep learning to translate text between languages accurately and efficiently.

Question 4: What is text normalization and why is it essential in text processing tasks?

Answer:

Text normalization refers to the process of converting text into a standard, consistent format. Tasks include lowercasing, removing punctuation, expanding contractions, and correcting misspellings. It is essential because it reduces noise and inconsistencies in text data, ensuring better accuracy in downstream NLP tasks such as tokenization, parsing, and model training.

Question 5: Compare and contrast stemming and lemmatization with suitable examples.

Answer:

- Stemming reduces words to their root form by chopping off affixes (e.g., "running" → "run"). It's fast but may produce non-standard roots ("studies" → "studi").
- Lemmatization maps inflected words to their dictionary root (lemma) considering part of speech (e.g., "better" → "good" as adjective). It's slower but produces proper words.

Question 6: Write a Python program that uses regular expressions (regex) to extract all email addresses from the following block of text:

“Hello team, please contact us at support@xyz.com for technical issues, or reach out to our HR at hr@xyz.com. You can also connect with John at john.doe@xyz.org and jenny via jenny_clarke126@mail.co.us. For partnership inquiries, email partners@xyz.biz.”

(*Include your Python code and output in the code box below.*) **Answer:**

```
 import re

text = '''Hello team, please contact us at supportxyz.com for technical issues,
or reach out to our HR at hrxyz.com. You can also connect with John at john.doexyz.org
and jenny via jennyclarke126mail.co.us. For partnership inquiries, email partnersxyz.biz.'''

# Regex pattern for email
emails = re.findall(r'[\\w\\.-]+@[\\w\\.-]+\\.[\\w]+', text)
print('Extracted emails:', emails)

Output:
Extracted emails: []
(Note: None of the strings match the standard email format with an "@", so output is an empty list.)
```

Question 7: Given the sample paragraph below, perform string tokenization and frequency distribution using Python and NLTK:

“Natural Language Processing (NLP) is a fascinating field that combines linguistics, computer science, and artificial intelligence. It enables machines to understand, interpret, and generate human language. Applications of NLP include chatbots, sentiment analysis, and machine translation. As technology advances, the role of NLP in modern solutions is becoming increasingly critical.”

(*Include your Python code and output in the code box below.*)

Answer:

```
 import nltk
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist

paragraph = """Natural Language Processing NLP is a fascinating field that combines
linguistics, computer science, and artificial intelligence. It enables machines to understand,
interpret, and generate human language. Applications of NLP include chatbots, sentiment
analysis, and machine translation. As technology advances, the role of NLP in modern
solutions is becoming increasingly critical."""
```

```
tokens = word_tokenize(paragraph)
fdist = FreqDist(tokens)
print('Tokens:', tokens)
print('Frequency Distribution:', fdist.most_common(10))
```

Output:

- Shows tokenized words
- Most common tokens with their frequencies, e.g. ('NLP', 3), ('and', 2) etc.

Question 8: Create a custom annotator using spaCy or NLTK that identifies and labels proper nouns in a given text.

(*Include your Python code and output in the code box below.*) **Answer:**

```
import spacy

nlp = spacy.load('en_core_web_sm')
text = "Apple is looking at buying U.K. startup for $1 billion"

doc = nlp(text)
proper_nouns = [token.text for token in doc if token.pos_ == 'PROPN']
print('Proper Nouns:', proper_nouns)
```

Output:
Proper Nouns: ['Apple', 'U.K.']

Question 9: Using Genism, demonstrate how to train a simple Word2Vec model on the following dataset consisting of example sentences:

dataset = [

"Natural language processing enables computers to understand human language",

"Word embeddings are a type of word representation that allows words with similar meaning to have similar representation",

"Word2Vec is a popular word embedding technique used in many NLP applications",

"Text preprocessing is a critical step before training word embeddings",

"Tokenization and normalization help clean raw text for modeling"

]

Write code that tokenizes the dataset, preprocesses it, and trains a Word2Vec model using Gensim.

(*Include your Python code and output in the code box below.*)

Answer:

```
 from gensim.models import Word2Vec

sentences = [
    "Natural language processing enables computers to understand human language",
    "Word embeddings are a type of word representation that allows words with similar
meaning to have similar representation",
    "Word2Vec is a popular word embedding technique used in many NLP applications",
    "Text preprocessing is a critical step before training word embeddings",
    "Tokenization and normalization help clean raw text for modeling"
]
# Preprocessing and tokenization
tokenized = [sentence.lower().split() for sentence in sentences]
model = Word2Vec(tokenized, vector_size=50, window=2, min_count=1, workers=1)
print('Vocabulary:', list(model.wv.key_to_index.keys()))
print('Vector for \"nlp\":', model.wv['nlp'])

Output:
Shows list of vocabulary words and a sample 50-d vector for 'nlp'.
```

Question 10: Imagine you are a data scientist at a fintech startup. You've been tasked with analyzing customer feedback. Outline the steps you would take to clean, process, and extract useful insights using NLP techniques from thousands of customer reviews.

(*Include your Python code and output in the code box below.*) **Answer:**

Typical Workflow and Example Python:

- Data Cleaning: Remove duplicates, missing values, and irrelevant content.
- Text Normalization: Lowercase, remove punctuation, stopwords, special characters.
- Tokenization: Split text into words/tokens.
- Lemmatization/Stemming: Reduce words to base form.
- Sentiment Analysis: Use models to categorize reviews.
- Topic Modeling: Extract main themes using LDA/NMF.
- Visualization: Word clouds, frequency plots.

```
import pandas as pd
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

# Example reviews
reviews = pd.Series([
    "Great customer service and quick response.",
    "Poor app design and slow interface.",
    "Loved the new payment feature!"
])

# Cleaning & normalization
normalized = reviews.str.lower().str.replace(r'[^a-z ]', '', regex=True)
stop_words = set(stopwords.words('english'))
cleaned = normalized.apply(lambda x: ' '.join([word for word in word_tokenize(x) if
word not in stop_words]))

# Sample output
print('Cleaned Reviews:', cleaned.tolist())
```