Experiment No.5

Code:

```
import re
import random
from collections import defaultdict

class NGramModel:
    def __init__(self, n):
        self.n = n
        self.ngrams = defaultdict(list)

    def preprocess(self, text):
        # Clean and tokenize the text
        text = text.lower()
        text = re.sub(r'[^a-zA-Z0-9\s]', '', text)
        tokens = text.split()
        return tokens

    def train(self, text):
        tokens = self.preprocess(text)
        for i in range(len(tokens) - self.n):
            key = tuple(tokens[i:i + self.n - 1])
            next_word = tokens[i + self.n - 1]
            self.ngrams[key].append(next_word)

    def generate_text(self, seed, num_words=20):
        seed_tokens = self.preprocess(seed)
        if len(seed_tokens) < self.n - 1:
            raise ValueError(f"Seed text must be at least {self.n -
1} words long.")

        current = tuple(seed_tokens[-(self.n - 1):])
```

```
        output = list(current)

        for _ in range(num_words):
            next_words = self.ngrams.get(current)
            if not next_words:
                break
            next_word = random.choice(next_words)
            output.append(next_word)
            current = tuple(output[-(self.n - 1):])
        return ' '.join(output)

text = """
Deep learning models like CNN, RNN, and transformers are powerful
tools for solving AI problems.
They learn patterns from data and generalize to unseen examples.
"""

# Create and train a bigram model (2-gram)
model = NGramModel(n=2)
model.train(text)

# Generate text
seed = "deep learning"
print("Generated:", model.generate_text(seed, num_words=15))
```

Output:

```
Generated: learning models like cnn rnn and transformers are
powerful tools for solving ai problems they learn
```

Experiment No.6

Code:

```
import spacy
from spacy import displacy
from sklearn.metrics import classification_report

# Load spaCy's small English model
nlp = spacy.load("en_core_web_sm")

# Sample texts
texts = [
    "Apple is looking at buying U.K. startup for $1 billion.",
    "Barack Obama was born in Hawaii.",
    "Elon Musk is the CEO of SpaceX and Tesla."
]

# Extract entities and their labels
for text in texts:
    doc = nlp(text)
    print(f"\nText: {text}")
    print("Entities:")
    for ent in doc.ents:
        print(f"{ent.text:<20} -> {ent.label_}")
    # Optionally visualize it in browser
    # displacy.serve(doc, style="ent")

# Actual vs Predicted Entities for 1 sample
y_true = ['ORG', 'GPE', 'MONEY']  # Ground truth labels
y_pred = ['ORG', 'GPE', 'MONEY']  # Predicted labels from spaCy

print(classification_report(y_true, y_pred))
```

Output:

```
Text: Apple is looking at buying U.K. startup for $1 billion.
Entities:
Apple                -> ORG
U.K.                 -> GPE
$1 billion           -> MONEY

Text: Barack Obama was born in Hawaii.
Entities:
Barack Obama         -> PERSON
Hawaii               -> GPE

Text: Elon Musk is the CEO of SpaceX and Tesla.
Entities:
Elon Musk            -> PERSON
Tesla                -> NORP
              precision    recall  f1-score   support

         GPE       1.00      1.00      1.00         1
       MONEY       1.00      1.00      1.00         1
         ORG       1.00      1.00      1.00         1

    accuracy                           1.00         3
   macro avg       1.00      1.00      1.00         3
weighted avg       1.00      1.00      1.00         3
```

Experiment No.7

Code:

```
import spacy


# Load spaCy's small English model

nlp = spacy.load("en_core_web_sm")


# Your input text

text = """Elon Musk, the CEO of Tesla and SpaceX, announced a new AI
company in San Francisco on April 5, 2025."""


# Process the text

doc = nlp(text)


# Print named entities

print("Named Entities, their labels, and explanation:")

for ent in doc.ents:

    print(f"{ent.text:<25} -> {ent.label_:<10}
({spacy.explain(ent.label_)})")
```

Output:

```
Named Entities, their labels, and explanation:
Elon Musk                 -> PERSON     (People, including
fictional)
Tesla                     -> ORG        (Companies, agencies,
institutions, etc.)
SpaceX                    -> PERSON     (People, including
fictional)
AI                        -> ORG        (Companies, agencies,
institutions, etc.)
San Francisco             -> GPE        (Countries, cities, states)
April 5, 2025             -> DATE       (Absolute or relative dates
or periods)
```