Experiment: 8

AIM: Write a Program on Synset.

OBJECTIVES: To understand and perform **synset** operations using the **WordNet interface** in the **NLTK library**.

REQUIREMENTS: Python (version 3.x or above), NLTK library, WordNet.

THEORY:

WordNet is a large lexical database of English developed at Princeton University. It groups English words into sets of synonyms called synsets, provides definitions, and shows semantic relationships between them.

A **synset** (short for "synonym set") is a group of words that are interchangeable in some context. It is a special kind of a simple interface that is present in NLTK to look up words in WordNet. Synset instances are the groupings of synonymous words that express the same concept. Some of the words have only one Synset and some have several

For example, the word **"car"** belongs to a synset with words like: "automobile", "machine" and "motorcar".

Key Features of a Synset:

- Each synset represents **one unique meaning** or **concept**.
- It contains one or more **synonymous words or phrases** (called *lemmas*).
- Synsets are linked to each other through **semantic relations** like:
    - **Hypernyms** (general concepts) and **hyponyms** (specific examples),
    - **Antonyms**,
    - **Meronyms** (part-whole relationships),
    - **Entailments** (especially for verbs).

Implementation of Synset Using WordNet.

Step 1: Importing Necessary Libraries.

```
import nltk
nltk.download('wordnet')
nltk.download('omw-1.4')  # For extended multilingual WordNet
from nltk.corpus import wordnet as wn
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\tdhan\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\tdhan\AppData\Roaming\nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
```

Step 2: Retrieve all synsets for the word 'car'.

```
# Get synsets for the word 'car'
synsets = wn.synsets('car')
# Print all synsets of 'car'
print("All Synsets of 'car':")
for s in synsets:
    print(f"- {s.name()}")
```

```
All Synsets of 'car':
- car.n.01
- car.n.02
- car.n.03
- car.n.04
- cable_car.n.01
```

Step 3: Select a Synset and Print Its Details

```
# Select the first synset
car = synsets[0]
print("\nSelected Synset:")
print("Name:", car.name())
print("Definition:", car.definition())
print("Examples:", car.examples())
```

```
Selected Synset:
Name: car.n.01
Definition: a motor vehicle with four wheels; usually propelled by an internal combustion engine
Examples: ['he needs a car to get to work']
```

Step 4: Retrieve Synonyms (Lemmas).

```
# Get synonyms (Lemmas)
print("\nSynonyms (Lemmas):")
for lemma in car.lemmas():
    print("-", lemma.name())
```

```
Synonyms (Lemmas):
- car
- auto
- automobile
- machine
- motorcar
```

Step 5: Retrieve Hypernyms (More General Concepts)

```
# Get Hypernyms (more general terms)
print("\nHypernyms:")
for hyper in car.hypernyms():
    print("-", hyper.name(), ":", hyper.definition())
```

```
Hypernyms:
- motor_vehicle.n.01 : a self-propelled wheeled vehicle that does not run on rails
```

Step 6: Retrieve Hyponyms (More Specific Concepts).

```
# Get Hyponyms (more specific terms)
print("\nHyponyms:")
for hypo in car.hyponyms():
    print("-", hypo.name(), ":", hypo.definition())
```

```
Hyponyms:
- ambulance.n.01 : a vehicle that takes people to and from hospitals
- beach_wagon.n.01 : a car that has a long body and rear door with space behind rear seat
- bus.n.04 : a car that is old and unreliable
- cab.n.03 : a car driven by a person whose job is to take passengers where they want to g
- compact.n.03 : a small and economical car
- convertible.n.01 : a car that has top that can be folded or removed
- coupe.n.01 : a car with two doors and front seats and a luggage compartment
- cruiser.n.01 : a car in which policemen cruise the streets; equipped with radiotelephoni
- electric.n.01 : a car that is powered by electricity
- gas_guzzler.n.01 : a car with relatively low fuel efficiency
- hardtop.n.01 : a car that resembles a convertible but has a fixed rigid top
- hatchback.n.01 : a car having a hatchback door
- horseless_carriage.n.01 : an early term for an automobile
- hot_rod.n.01 : a car modified to increase its speed and acceleration
- jeep.n.01 : a car suitable for traveling over rough terrain
- limousine.n.01 : large luxurious car; usually driven by a chauffeur
- loaner.n.02 : a car that is lent as a replacement for one that is under repair
```

Conclusion

In this experiment, we explored the **WordNet Synset API** using the NLTK library in Python. By accessing synsets of the word "car", we extracted its synonyms, definitions, examples, hypernyms, and hyponyms. This approach is foundational in many **NLP applications** like semantic search, word sense disambiguation, and intelligent text analysis.