# Experiment 08 : Implement a Psychological Model in cognitive computing.

**Learning Objective :** Students should be able to apply a Psychological Model for Cognitive Computing and solve problems.

**Tool :** Python

**Theory :**

Implementing a psychological model in cognitive computing involves using concepts from psychology to design and develop algorithms that can mimic human thought processes. Here are the steps involved in implementing a psychological model in cognitive computing:

Identify the Problem: The first step is to identify a specific problem that can be solved using cognitive computing. The problem should be related to human behavior or cognition, and should have a clear goal.

Select a Psychological Model: Once the problem is identified, select a psychological model that can help to understand the underlying cognitive processes involved in the problem. The model can be based on theories of perception, attention, memory, or decision-making.

Develop the Algorithm: Based on the selected psychological model, develop an algorithm that can simulate the cognitive processes involved in the problem. The algorithm can be developed using various techniques such as artificial neural networks, decision trees, or rule-based systems.

Train the Algorithm: Once the algorithm is developed, it needs to be trained using data that reflects the cognitive processes involved in the problem. The data can be collected from human subjects or simulated using computer models.

Test the Algorithm: After training, the algorithm needs to be tested to evaluate its performance. The testing can be done using a separate set of data that was not used for training. The performance can be evaluated using metrics such as accuracy, precision, and recall.

Refine the Algorithm: Based on the testing results, refine the algorithm to improve its performance. This step may involve changing the model, adjusting the parameters, or collecting more data.

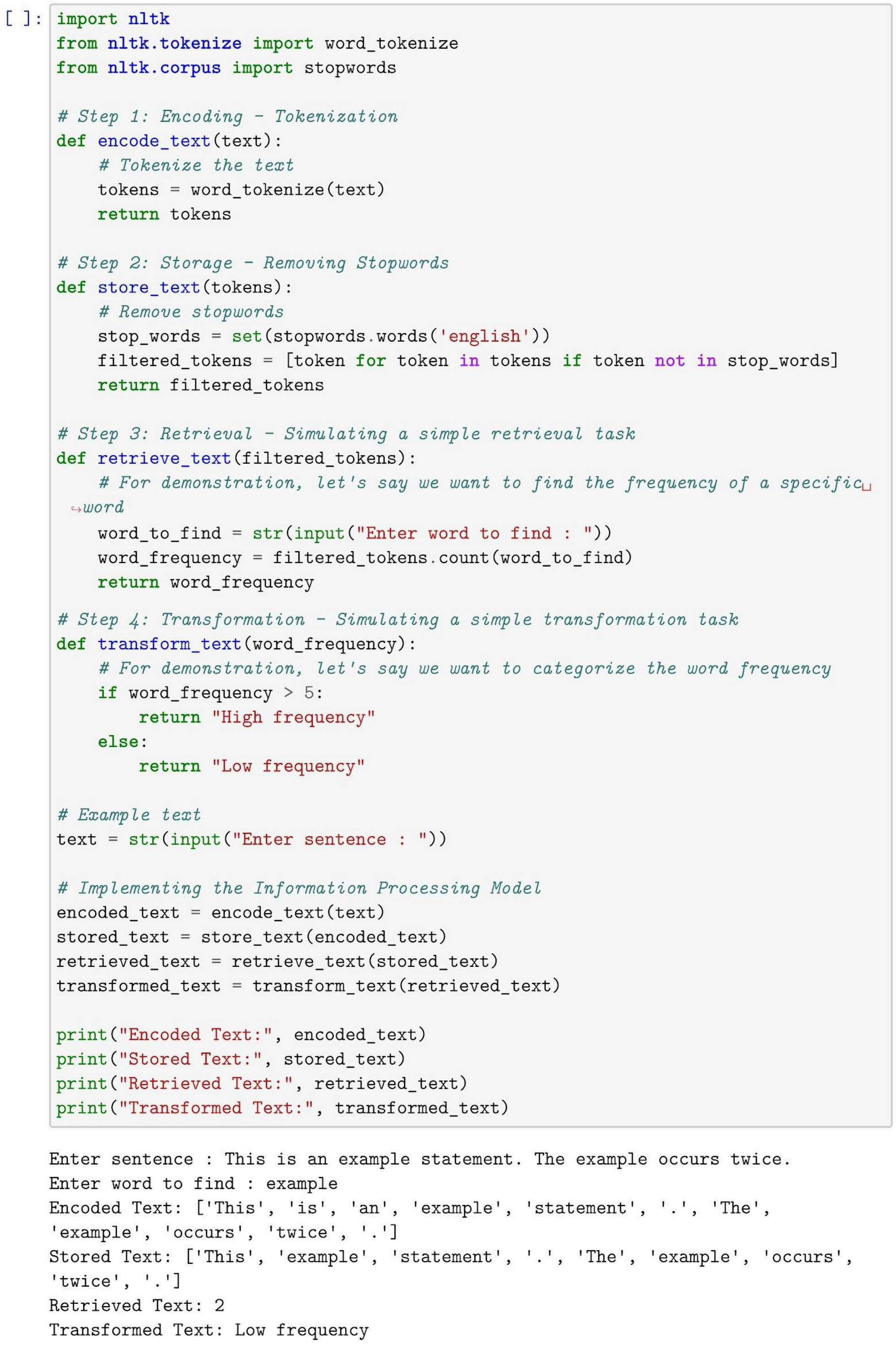
Deploy the Algorithm: Once the algorithm is refined and tested, it can be deployed in a real-world setting to solve the problem.

Some common psychological models used in cognitive computing include:

- Information Processing Model
- Signal Detection Theory
- Working Memory Model
- Dual-Process Model
- Prospect Theory

The choice of model depends on the problem to be solved and the availability of data. These models can be implemented using various programming languages such as Python, R, and Java. The choice of programming language depends on the specific requirements of the project and the availability of libraries for implementing the algorithms.

**Implementation :**



```
[ ]: import nltk
     from nltk.tokenize import word_tokenize
     from nltk.corpus import stopwords

     # Step 1: Encoding - Tokenization
     def encode_text(text):
         # Tokenize the text
         tokens = word_tokenize(text)
         return tokens

     # Step 2: Storage - Removing Stopwords
     def store_text(tokens):
         # Remove stopwords
         stop_words = set(stopwords.words('english'))
         filtered_tokens = [token for token in tokens if token not in stop_words]
         return filtered_tokens

     # Step 3: Retrieval - Simulating a simple retrieval task
     def retrieve_text(filtered_tokens):
         # For demonstration, let's say we want to find the frequency of a specific␣
      ↪word
         word_to_find = str(input("Enter word to find : "))
         word_frequency = filtered_tokens.count(word_to_find)
         return word_frequency

     # Step 4: Transformation - Simulating a simple transformation task
     def transform_text(word_frequency):
         # For demonstration, let's say we want to categorize the word frequency
         if word_frequency > 5:
             return "High frequency"
         else:
             return "Low frequency"

     # Example text
     text = str(input("Enter sentence : "))

     # Implementing the Information Processing Model
     encoded_text = encode_text(text)
     stored_text = store_text(encoded_text)
     retrieved_text = retrieve_text(stored_text)
     transformed_text = transform_text(retrieved_text)

     print("Encoded Text:", encoded_text)
     print("Stored Text:", stored_text)
     print("Retrieved Text:", retrieved_text)
     print("Transformed Text:", transformed_text)
```

```
Enter sentence : This is an example statement. The example occurs twice.
Enter word to find : example
Encoded Text: ['This', 'is', 'an', 'example', 'statement', '.', 'The',
'example', 'occurs', 'twice', '.']
Stored Text: ['This', 'example', 'statement', '.', 'The', 'example', 'occurs',
'twice', '.']
Retrieved Text: 2
Transformed Text: Low frequency
```

**Result and Discussion :**

**Learning Outcomes :** Students should have the ability to

LO 1: Formulate the problem using AI and CC Approach.
LO 2: Solve the problem using a psychological model Algorithm.

**Course Outcomes :**

CO4 : Analyze Relationship between Big Data and Cognitive Computing.

**Conclusion :**

**Viva Questions :**

Q1. What is a psychological model, and how can it be translated into a computational model?
Q.2 Can you explain how model calibration and validation are performed in cognitive computing, and why are they important?
Q.3 How do you choose the appropriate programming language and tools for implementing a psychological model in cognitive computing?

**For Faculty Use**

| Correction Parameters | Formative Assessment [40%] | Timely completion of Practical [ 40%] | Attendance / Learning Attitude [20%] | Total |
|---|---|---|---|---|
| Marks Obtained | | | | |