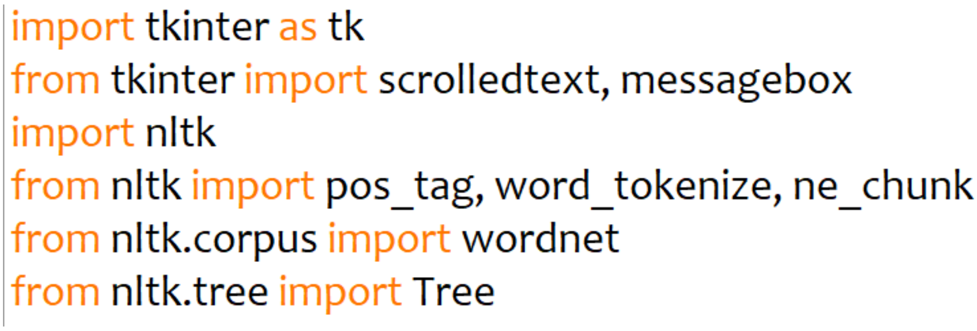
3 NLP Submission: upto WordNet

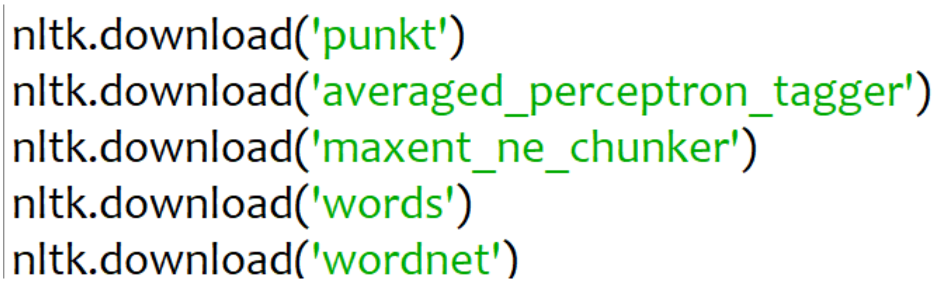
The provided Python script is a graphical application that leverages Natural Language Processing (NLP) techniques to perform various linguistic analyses on user-inputted text. Specifically, it allows users to conduct Part-of-Speech (PoS) tagging, chunking, Named Entity Recognition (NER), and WordNet lookups through an intuitive graphical user interface (GUI) built using the tkinter library. Below is a comprehensive breakdown of the script's components and functionalities.

1. Importing Necessary Libraries



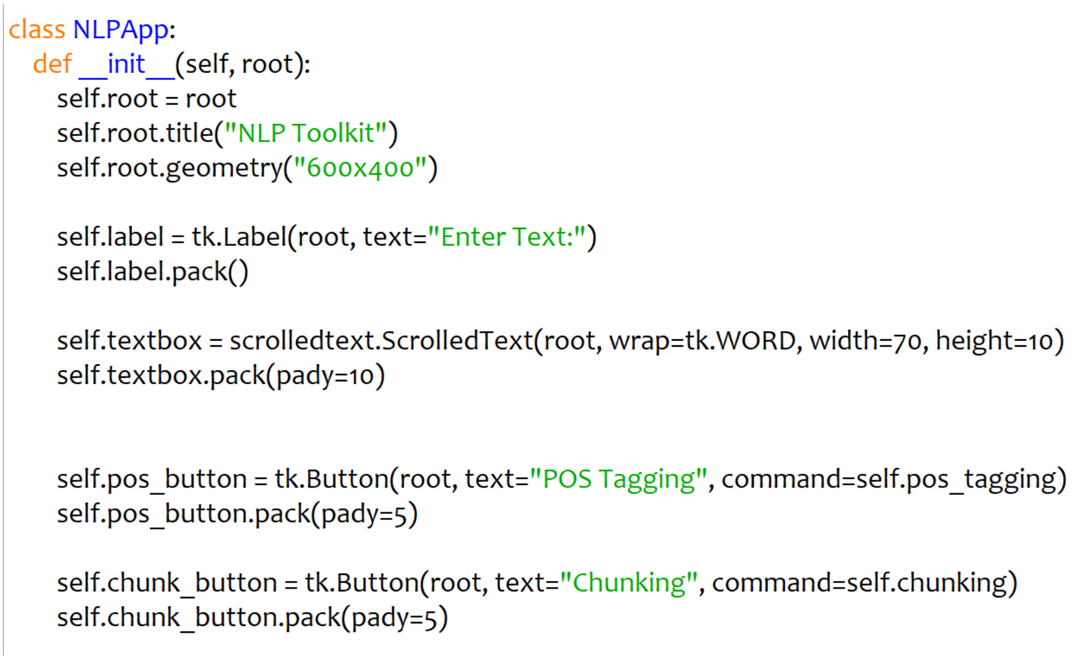
```
import tkinter as tk
from tkinter import scrolledtext, messagebox
import nltk
from nltk import pos_tag, word_tokenize, ne_chunk
from nltk.corpus import wordnet
from nltk.tree import Tree
```

2. Downloading NLTK Data

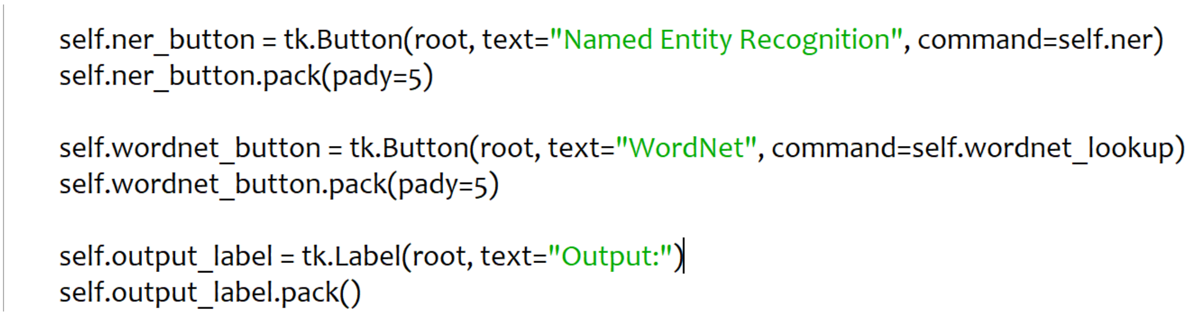


```
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('maxent_ne_chunker')
nltk.download('words')
nltk.download('wordnet')
```

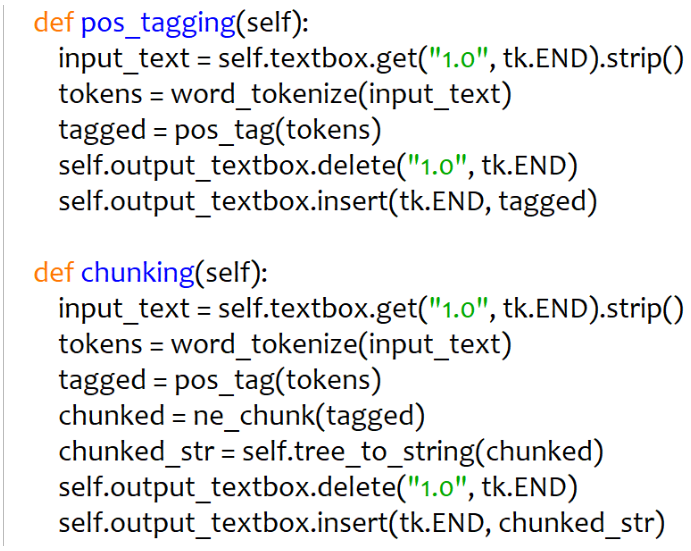
3. Creating a GUI Layout



```python
class NLPApp:
    def __init__(self, root):
        self.root = root
        self.root.title("NLP Toolkit")
        self.root.geometry("600x400")

        self.label = tk.Label(root, text="Enter Text:")
        self.label.pack()

        self.textbox = scrolledtext.ScrolledText(root, wrap=tk.WORD, width=70, height=10)
        self.textbox.pack(pady=10)


        self.pos_button = tk.Button(root, text="POS Tagging", command=self.pos_tagging)
        self.pos_button.pack(pady=5)

        self.chunk_button = tk.Button(root, text="Chunking", command=self.chunking)
        self.chunk_button.pack(pady=5)
```



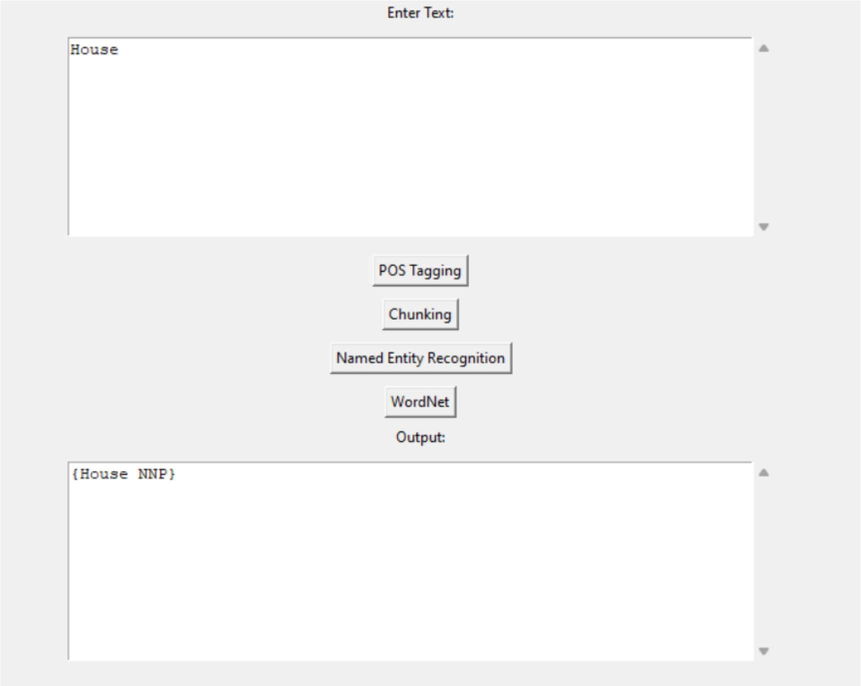
```python
        self.ner_button = tk.Button(root, text="Named Entity Recognition", command=self.ner)
        self.ner_button.pack(pady=5)

        self.wordnet_button = tk.Button(root, text="WordNet", command=self.wordnet_lookup)
        self.wordnet_button.pack(pady=5)

        self.output_label = tk.Label(root, text="Output:")
        self.output_label.pack()
```



```python
    def pos_tagging(self):
        input_text = self.textbox.get("1.0", tk.END).strip()
        tokens = word_tokenize(input_text)
        tagged = pos_tag(tokens)
        self.output_textbox.delete("1.0", tk.END)
        self.output_textbox.insert(tk.END, tagged)

    def chunking(self):
        input_text = self.textbox.get("1.0", tk.END).strip()
        tokens = word_tokenize(input_text)
        tagged = pos_tag(tokens)
        chunked = ne_chunk(tagged)
        chunked_str = self.tree_to_string(chunked)
        self.output_textbox.delete("1.0", tk.END)
        self.output_textbox.insert(tk.END, chunked_str)
    def ner(self):
        input_text = self.textbox.get("1.0", tk.END).strip()
        tokens = word_tokenize(input_text)
        tagged = pos_tag(tokens)
        named_entities = ne_chunk(tagged)
        named_entities_str = self.tree_to_string(named_entities)
        self.output_textbox.delete("1.0", tk.END)
        self.output_textbox.insert(tk.END, named_entities_str)

    def wordnet_lookup(self):
        input_text = self.textbox.get("1.0", tk.END).strip()
        synsets = wordnet.synsets(input_text)
        self.output_textbox.delete("1.0", tk.END)
        if synsets:
            for synset in synsets:
                self.output_textbox.insert(tk.END, f"Synset: {synset.name()}\nDefinition: {synset.definition()}\n\n")
        else:
            self.output_textbox.insert(tk.END, "No synsets found.\n")

    @staticmethod
    def tree_to_string(tree):
        return ' '.join([str(subtree) if isinstance(subtree, Tree) else subtree[0] for subtree in tree])

if __name__ == "__main__":
    root = tk.Tk()
    app = NLPApp(root)
    root.mainloop()
```

OUTPUT:-

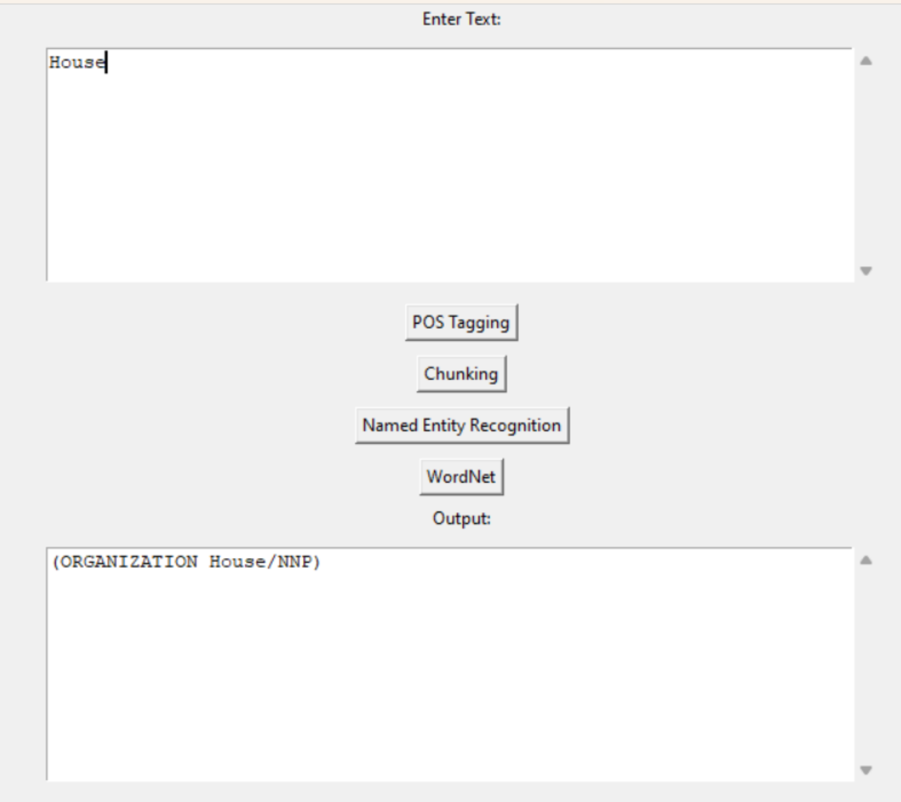
1. POS Tagging



```
House
```

```
(House NNP)
```

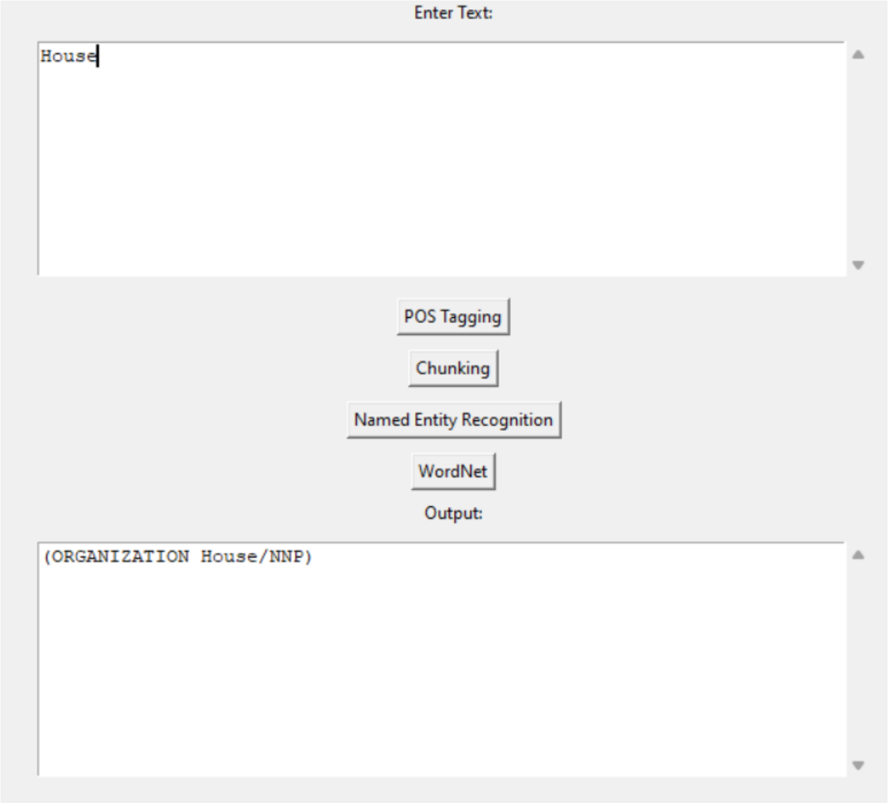
2. Chunking



```
House
```

```
(ORGANIZATION House/NNP)
```

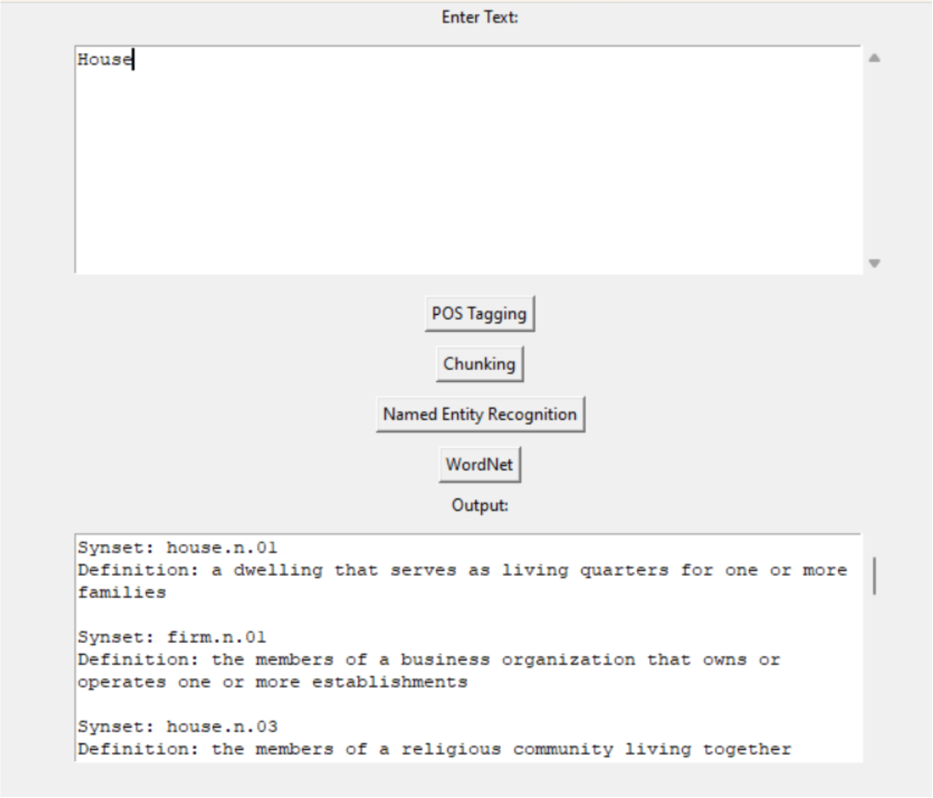
3. Name Entity Recognition



```
House
```

```
(ORGANIZATION House/NNP)
```

4. WordNet



```
House
```

```
Synset: house.n.01
Definition: a dwelling that serves as living quarters for one or more
families

Synset: firm.n.01
Definition: the members of a business organization that owns or
operates one or more establishments

Synset: house.n.03
Definition: the members of a religious community living together
```