### Assignment 6 Walkthrough

#### 1. Tokenize Text

**Overview:** This function uses NLTK to tokenize the provided text into sentences and words. Tokenization is a critical preprocessing step in many NLP tasks, breaking down text into smaller units for easier analysis.

**Detailed Pseudo Code:**

FUNCTION tokenize\_text(text)  
 USE sent\_tokenize to divide text into a list of sentences  
 USE word\_tokenize to divide text into a list of words  
 CREATE a dictionary named 'tokens' with keys 'sentences' and 'words'  
 - 'sentences' stores the list of sentences  
 - 'words' stores the list of words  
 RETURN 'tokens'  
END FUNCTION

**Implementation Guide:** - Ensure NLTK is installed and import sent\_tokenize and word\_tokenize from nltk.tokenize. - Apply sent\_tokenize to split the text into sentences. - Apply word\_tokenize to split the text into words. - Compile the results into a dictionary with keys for sentences and words, facilitating further NLP tasks such as POS tagging or NER.

#### 2. POS Tagging

**Overview:** Part-of-speech (POS) tagging assigns word types (nouns, verbs, adjectives, etc.) to each word in the input text. This function leverages NLTK for POS tagging, aiding in understanding the grammatical structure of sentences.

**Detailed Pseudo Code:**

FUNCTION pos\_tagging(text)  
 TOKENIZE the text into words using word\_tokenize  
 APPLY nltk.pos\_tag to the tokenized words, assigning POS tags  
 RETURN the list of tuples containing (word, POS tag)  
END FUNCTION

**Implementation Guide:** - Use word\_tokenize to break the text into individual words. - Utilize nltk.pos\_tag, which takes a list of words and returns a list of tuples, each containing a word and its corresponding POS tag. - Return this list, which can be used for syntactic analysis or to improve the performance of other NLP tasks.

#### 3. Named Entity Recognition

**Overview:** Named Entity Recognition identifies and classifies entities within text into predefined categories such as persons, organizations, and locations. This method uses Spacy, a powerful NLP library, to perform NER, enhancing text analysis by extracting specific information.

**Detailed Pseudo Code:**

FUNCTION named\_entity\_recognition(text)  
 PROCESS the text using Spacy's NLP model to obtain a document  
 INITIALIZE an empty list for entities  
 FOR each entity in the document's entities  
 APPEND a tuple of (entity text, entity label) to the entities list  
 RETURN the list of entities  
END FUNCTION

**Implementation Guide:** - Initialize Spacy’s NLP model (en\_core\_web\_sm). If the model isn’t available, download it using Spacy’s CLI command (spacy.cli.download("en\_core\_web\_sm")). - Pass the text to the NLP model, which processes it and identifies entities. - Iterate through these entities, extracting the text (the actual word or phrase) and the entity type (e.g., person, location). - Compile these into a list of tuples and return it, providing a structured view of the named entities present in the input text.