nltk_utils.py

The code above defines a set of functions used for natural language processing (NLP) tasks.

The first function `tokenize(sentence)` uses the `nltk` library to split a given sentence into an array of individual words/tokens. The function tokenizes the sentence by breaking it into individual words and removing any punctuation, whitespace, or other non-word characters.

The second function `stem(word)` uses the Porter stemming algorithm from the `nltk.stem.porter` module to convert a given word into its root form. Stemming involves removing the suffix of a word and reducing it to its base or root form, so that words with the same root will be treated as the same word. This helps to reduce the number of unique words in a text corpus and can improve the accuracy of NLP models.

The third function `bag_of_words(tokenized_sentence, words)` is used to convert a sentence into a "bag of words" representation. This function takes a tokenized sentence and a list of all possible words in the corpus as input, and returns a numpy array with a 1 for each word in the input sentence that also appears in the list of all possible words. The bag-of-words representation is commonly used as input to NLP models because it is simple, efficient, and effective in capturing the frequency of individual words in a text corpus.