PRACTICAL ASSESMENT

SHARON \_

Activity 1

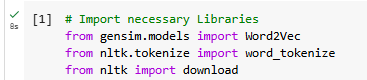
**Python Implementation of Word Embeddings using word2vec**

Requirements

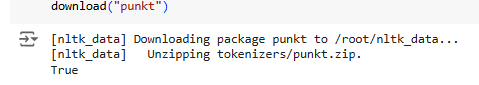
* Personal computer/laptop
* Google Collab

Procedure

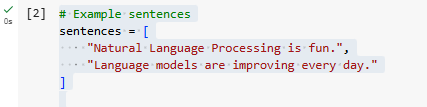
1. Import Necessary Libraries



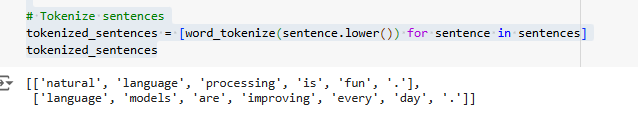
1. Download Required NLTK Data



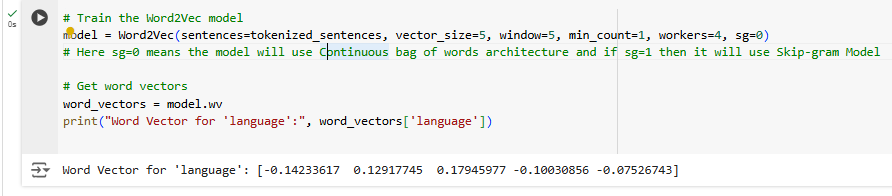
* Define Example Sentences



* Tokenize Sentences



* Train the Word2Vec Model



Activity 2

**Python Implementation of Word Embeddings using GloVe**

Requirements

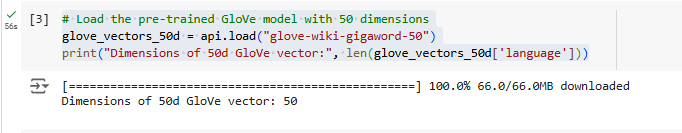
* Personal computer/laptop
* Google Collab

Procedure

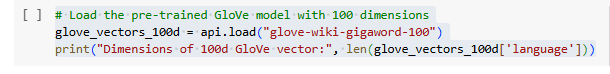
* Import the Gensim downloader



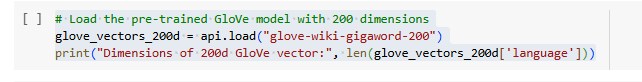
* Load the pre-trained GloVe model with 50 dimensions and Check the dimensions of a sample word vector (e.g., 'language')



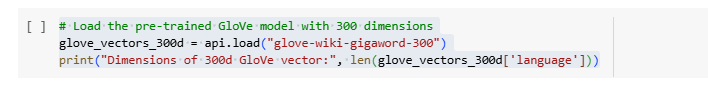
* Load the pre-trained GloVe model with 100 dimensions
* Check the dimensions of a sample word vector (e.g., 'language')



* Load the pre-trained GloVe model with 200 dimensions
* Check the dimensions of a sample word vector



* Load the pre-trained GloVe model with 300 dimensions
* Check the dimensions of a sample word vector



Activity 3

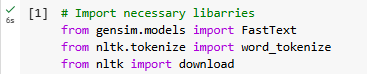
**Python Implementation of Word Embeddings using Fasttext**

Requirements

* Personal computer/laptop
* Google Collab

Procedure

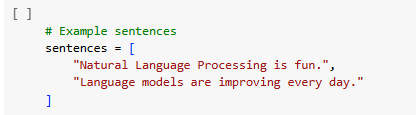
* Importing Necessary Libraries



Downloading NLTK Data



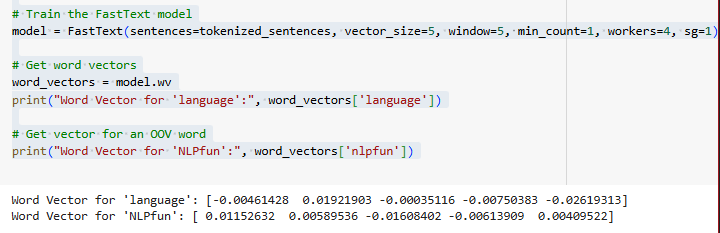
Example Sentences



Tokenizing Sentences



Training the FastText Model



output the word vectors for 'language' and 'NLPfun'. The FastText model's ability to handle OOV words through subword information is one of its key strengths, making it robust for a variety of natural language processing tasks.