Semantic Similarity

Index

[1. Overview 1](#_Toc87868735)

[1.1. Context 1](#_Toc87868736)

[1.2. Input/Output Files 1](#_Toc87868737)

[1.3. Repository 2](#_Toc87868738)

[2. Environments 2](#_Toc87868739)

[2.1. Pre-requisites 2](#_Toc87868742)

[2.2. Environment 2](#_Toc87868743)

[3. Process Flow 3](#_Toc87868744)

# Overview

## Context

*Semantic Similarity helps in finding the most similar words in a dataset. Given a query(n-gram) the code can generate the top N similar words from the corpus along with a score. The score ranges from 0-1 and higher the score, higher the similarity.*

*The code inputs a clean data hence text extraction and preprocessing are recommended before using the code. The query must be related to the corpus to get better results.*

## Input/Output Files

**Input File name** - Semantic\_Similarity\_Input\_US\_7Jun22

**Output File name** – Semantic\_Similarity\_<today’s date>

The output file contains nearest\_words, score, and input query of top 5(user input) nearest words.

## Repository

|  |  |
| --- | --- |
| **Type** | **Path** |
| **Input** | Modeling-Semantic\_Similarity/Input\_Data |
| **Output** | Modeling-Semantic\_Similarity/Output\_Data |
| **Code** | Modeling-Semantic\_Similarity |

# 

# Environments



## Pre-requisites

* Python environment
* Python libraries: pandas, SentenceTransformer, gensim.utils,scipy, openpyxl

## Environment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Environment** | **Location** | **OS** | **RAM** | **Hard Disk** |
| Development | Local machine | Windows | 8GB | 128GB |
| Testing | Local machine | Windows | 8GB | 128GB |

# Process Flow

Select the input columns - corpus and query



Set your working directory

Input the excel

Enter the top N required number

Corresponding Eigen vector of corpus and query gets created

Cosine distance is used to calculate the distance

Output dataframe with nearest\_words, score, and input query

Lower the distance, higher the score