**Text Data Embedding System Standard Operating Procedure (SOP)**

**Scope:**

This SOP describes the procedures for creating embeddings from JSON and CSV data at Ramco Systems. Activities include the use of the Python programming language, integration into various data formats, and the use of machine learning models to drive generation.

**1. Purpose**

This SOP outlines a streamlined process for producing embeddings from JSON and CSV data using a dedicated Python application at Ramco Systems. By incorporating technologies like the Universal Sentence Encoder, the procedure ensures precision and transparency in data processing. Following these steps not only empowers employees with a clear understanding of embedding generation but also upholds a commitment to efficient and dependable data practices at Ramco Systems.

**2. Procedure**

**CSV Data:**

* **Import Libraries**: -Import the Necessary libraries
* **Loading CSV Data**:
  + The CSV data containing text lines is loaded into a pandas DataFrame. The file path is specified, and the CSV data is read into the dataset variable.
* **Loading the Pre-Trained Model for CSV Data (USE)**:
  + The Universal Sentence Encoder (USE) is specifically designed for generating embeddings from sentences. It transforms variable-length text inputs into fixed-size vectors, commonly referred to as embeddings, that capture semantic information about the input sentences.
* **USE Embeddings for Text**: -The implemented function, get\_use\_embeddings, showcases a streamlined approach to harnessing the power of the Universal Sentence Encoder (USE) for text embedding generation.
* **Generating Embeddings**:
  + Generating embeddings for a CSV file unfolds seamlessly through the integration of the Universal Sentence Encoder (USE). Leveraging TensorFlow and TensorFlow Hub, this methodology facilitates the transformation of textual data within the CSV into high-dimensional vectors, encapsulating semantic information.

**JSON Data:**

* **Import Libraries**: -Import the Necessary libraries
* **Loading JSON Data**:
  + The JSON data containing textual entries is loaded from the specified file path into the Python environment using the json.load() function.
* **Loading the Pre-Trained Model for JSON Data**:
  + The GloVe (Global Vectors for Word Representation) model is a pre-trained word embedding model. Pre-training involves training the model on a large corpus of text data to learn vector representations (embeddings) for words based on their semantic relationships.
* **Text Preprocessing**:
  + Text preprocessing is performed using the simple\_preprocess() function from Gensim. This step tokenizes the text and performs basic preprocessing tasks such as lowercasing and punctuation removal.
* **Generating Embeddings for JSON Data**:
  + Word embeddings are generated for the preprocessed text using the pre-trained GloVE model. The embeddings capture the semantic meaning of words in a continuous vector space.
* **Convert the Embedded Vectors to Save in JSON Forms**: -This process involves converting embedded vectors into a JSON format, facilitating efficient storage and retrieval. The script employs the json module to serialize embedded vectors, providing a versatile and structured representation for further use in various applications.
* **Retrieving from the Saved File**: -To retrieve embedded vectors from the saved JSON file, a simple script utilizing Python's json module is employed, allowing seamless integration and utilization of previously stored embeddings for downstream tasks.

**3. Conclusion**

This SOP provides a comprehensive guide for generating embeddings from JSON and CSV data at Ramco Systems. By following these procedures, employees can ensure streamlined operations and accurate embedding generation for various natural language processing endeavors.