**Ramco Systems - Text Data Embedding System Standard Operating Procedure (SOP)**

**Scope:**

This SOP defines the scope of procedures for creating embeddings from JSON and CSV data at Ramco Systems. The outlined activities involve using a custom Python application, integrating various data formats, and employing machine learning models for embedding generation.

**1. Purpose**

The purpose of this SOP is to provide a detailed guide on the steps involved in producing embeddings from JSON and CSV data using the designated Python application at Ramco Systems. The emphasis is on leveraging the BERT model for JSON data and the GloVe model for CSV data. By following these procedures, employees can gain a thorough understanding of the embedding generation process, promoting transparency and dependability in data processing operations.

**2. Procedure**

**JSON Data:**

* **Loading JSON Data:**
* JSON (JavaScript Object Notation) is a light-weight data interchange format extensively used for storing and transmitting structured statistics.
* Use the `json.Load()` feature to load JSON data into the Python environment.
* **Extracting Text from Data:**
* Text extraction from JSON records entails recursively traversing the JSON structure and figuring out relevant text fields to extract significant text.
* Recursively traverse the JSON structure to identify relevant text fields. Extract meaningful text from nested key-value pairs.
* **Text Preprocessing:**
* Text preprocessing is important to clean and put together textual statistics for similarly evaluation.
* Perform text preprocessing using common techniques such as tokenization, lowercasing, punctuation removal, and stop word elimination.
* Utilize the `basic\_preprocess()` function from Gensim for fundamental text preprocessing.
* **Loading the Pre-Trained Model (BERT):**
* The BERT (Bidirectional Encoder Representations from Transformers) model is employed for tokenization and embedding in the SOP, utilizing the 'bert-base-uncased' model for extracting contextualized word representations.
* Use the 'bert-base-uncased' model for tokenization and embedding.
* **Generating Embeddings:**
* Word embeddings are numerical representations of phrases that seize semantic statistics about the phrases' meanings and relationships.
* Apply the pre-trained BERT model to generate embeddings for preprocessed text data.
* Represent words as dense vectors in a continuous space for semantic similarity.
* **Saving the file into JSON file:**
* Converting the embedded file into array or inserting the vector data into a list.
* Saving the list in a JSON Format.
* **Retrieving from JSON file:**
* To retrieve the data from the JSON File, extract the data from the JSON File as array format.
* Now, from the array retrieve the vector form of the data.

**CSV Data:**

* **Loading CSV Data:**
* CSV (Comma-Separated Values) is a simple file format used to store tabular statistics in simple textual content.
* Read CSV data into a pandas DataFrame for convenient data manipulation.
* **Loading the Pre-Trained Model (GPT-2):**
* The GloVe (Global Vectors for Word Representation) model is utilized for generating word embeddings in the SOP, employing the 'glove.6B.100d.txt' pre-trained model to capture semantic meaning and contextual information from textual data.
* Utilize the 'glove.6B.100d.txt' GloVe model for generating word embeddings.
* **Tokenization and Model Inference:**
* Tokenization is the manner of breaking textual content into smaller units, which includes words or subwords, for in addition processing.
* Tokenize text from the CSV data using the GPT-2 tokenizer. Pass tokenized sequences through the GPT-2 model to obtain embeddings.
* **Calculating Embeddings:**
* Generate embeddings by processing tokenized text and extracting the last hidden state of the model output

**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.  
  
I have attached my GitHub Link, check my code here : [github](https://github.com/ponjose004/Intern-Assessment)