Name : Automatic Text Suggestion system

Scope: Autocomplete, or word completion, is a feature in which an application predicts the rest of a word a user is typing ....... Autocomplete speeds up human computer interactions when it correctly predicts the words a user intends to enter after only a few characters have been typed into a text input field. A word completion user interface allows the user to customize each suggestions list with user- defined name- completion pairs on an on- going basis. This work proposes a autocomplete word completion system detection method using NLP. This method of autocomplete of text using ML can be very useful to the employees/ representatives to have a better text completion system.

Software Requirements :

1: Os Module: The OS module in python provides functions for interacting with the operating system. OS, comes under Python’s standard utility modules. This module provides a portable way of using operating system dependent functionality. The \*os\* and \*os.path\* modules include many functions to interact with the file system. Following are some functions in OS module: osname: This function gives the name of the operating system dependent module imported.

2: NumPy: Numpy is a python library used for working with arrays. Numpy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of highlevel mathematical functions to operate on these arrays. MoreoverNumpy forms the foundation of the Machine Learning stack.

3: Pandas: Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

4: Scikit : Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. It is used to build machine learning models .

5: Metrics: The sklearn. metrics module implements several loss, score, and utility functions to measure classification performance. Some metrics might require probability estimates of the positive class, confidence values, or binary decisions values.

6: Jupyter Notebook : Project Jupyter is a project and community whose goal is to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages".

Language used : Python 3

Steps involved :

1: 1: First and foremost is to load all the required and important libraries to work with, importing libraries such as -The OS module in Python provides functions for interacting with the operating system. importing libraries such as Pandas/NumPy is mainly used for data analysis importing data from various allowing various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features. The JSON module is mainly used to convert the python dictionary above into a JSON string that can be written into a file. While the JSON module will convert strings to Python datatypes, normally the JSON functions are used to read and write directly from JSON files.

2: importing all the important pairs to work with such as - mostly the feature extraction and metrics pairwise, we need these feature extraction techniques to convert text into a matrix and vector of features, and we will be using these pairs to work with our model .

3: loading our corpus- Our corpus is all about history of sentences written by a customer and a representative in the past. The data is stored in a json file and using certain regex rules we will clean this data and convert this data that can be fed into our model.

4: process and clean our data using certain python functions and regex rules we split the data , split it on punctuations , split text to rows using certain inbuilt python functions such as split text or split text to rows and after processing the data we will count the no of sentences from the representative perspective , remove duplicates and use sentences which have at least 2 words .

5: After processing and cleaning our data, we define our model, and we are using the tfidf model - which stands for term frequency–inverse document frequency, is a numerical model that is intended to reflect how important a word is to a document in a collection or corpus. Typically, the tf-idf model is composed by two terms-

Normalized Term Frequency (tf) - no of times term t occurs in a document/total no of terms in a document Inverse Document Frequency (idf) - log (total number of documents)/no of documents with term t in it after calculating both term frequency and inverse document frequency, we merge their results and higher the tfidf score more frequent and signific that word or sentence is to a document/corpus.

6: now we will use a tool known as tfidf vectorizer which comes along with the tfidf model that has something also known as the weight feature which works along the tfidf

score and gives weight to the most important sentences. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. This weight feature is directionally proportional to the importance/significance of the sentence

7:Last but not the least we feed all of this to a matrix known as cosine similarity matrix that is a metric which is used to determine how similar two entities are irrespective of their size. It generally works with vector weights and After the calculation of a similarity matrix based on the sklearn tfidf tool (frequency and normalization of words), we use this matrix to calculate the similarity between the new few words written by the representative and the history of messages written in the past.

8:Finally, after applying all the 8 steps, we create a user interface to take multiple inputs from the user, the autocomplete is calculating the similarity between the sentence in the data and the prefix of the sentence written by the representative. As a

weight feature, we chose to reorder using the frequency of the most common similar sentence. The Autocomplete will recognize the closest sentences and rank 3 final proposals based on the respective inputs.