|  |  |
| --- | --- |
| **Name** | **ID** |
| Malk Haitham Mostafa | **20-01137** |
| Sara Rabie Ahmed | **20-01186** |
| Aya Yasser Helmy | **20-01114** |
| Rana Khaled MohyEl-dien | **20-01182** |
| Sara Ayman Mahmoud | **20-01133** |

* **Documentation about machine translation system (between English and Arabic(**
* **Algorithm used in the project**

**1-Logistic Regression :**

Logistic Regression is a statistical algorithm that is commonly used in classification problems, including problems in natural language processing such as machine translation. The logistic regression algorithm is used to predict a binary output variable based on one or more input variables, where the output variable represents the probability of a particular event occurring.

In the context of machine translation, the input to the logistic regression algorithm might be a sentence in one language and the output might be the corresponding sentence in another language. The input sentence is typically represented as a vector of numerical features, such as word frequencies or embeddings, and the output sentence is represented as a sequence of tokens.

During training, the logistic regression algorithm learns the weights of the input features that are most predictive of the output sentence. These weights are iteratively adjusted using a training set of input/output pairs until the algorithm converges to a set of weights that best fit the training data. Once the model is trained, it can be used to predict the output sentence for new input sentences.

In the context of machine translation, logistic regression is often used in conjunction with other natural language processing techniques.

**Aacuracy (0.004)**

**2-Decision Tree Classifier :**

The DecisionTreeClassifier algorithm is a popular machine learning algorithm used for classification tasks. It builds a decision tree model that makes decisions by recursively splitting the data into subsets based on the values of the input features.

The basic idea behind the DecisionTreeClassifier algorithm is to create a tree-like model that predicts the class of a given input by making a series of sequential decisions based on the input features. Each decision corresponds to a split in the data, where the algorithm chooses the feature that best separates the data into different classes.

The algorithm starts by selecting the feature that provides the best split in the data, based on a criterion such as the Gini index or information gain. The data is then split into two subsets based on the values of that feature. This process is repeated recursively for each subset, until a stopping criterion is met, such as a maximum depth of the tree or a minimum number of data points in each leaf node.

The resulting decision tree can be used to make predictions by following the path from the root node to a leaf node that corresponds to the predicted class. Each internal node of the tree corresponds to a decision based on a feature, and each leaf node corresponds to a predicted class.

The DecisionTreeClassifier algorithm has several hyperparameters that can be tuned to optimize performance, such as the maximum depth of the tree, the minimum number of data points required to split a node, and the criterion used to evaluate the quality of a split. The algorithm is also prone to overfitting, so it is important to use regularization techniques such as pruning or setting a minimum number of samples per leaf to avoid overfitting.

**Accuracy(0.012)**

**3- SVM:** **Support Vector Machines (SVM) is a popular machine** learning algorithm used for classification and regression problems, including problems in natural language processing such as machine translation. SVMs are particularly useful in tasks where there is a clear margin of separation between different classes of data.

In the context of machine translation, the input to the SVM algorithm might be a sentence in one language and the output might be the corresponding sentence in another language. The input sentence is typically represented as a vector of numerical features, such as word frequencies or embeddings, and the output sentence is represented as a sequence of tokens.

During training, the SVM algorithm learns a hyperplane that best separates the input feature vectors into their corresponding output labels. The hyperplane is chosen to maximize the margin between the feature vectors of different classes. Once the hyperplane has been learned, the algorithm can use it to predict the output label of a new input feature vector based on which side of the hyperplane it falls on.

In the context of machine translation, SVM can be used as a standalone algorithm or in combination with other natural language processing techniques to create more accurate and effective translation systems. For example, SVM can be used in conjunction with a language model or a neural machine translation system to improve the accuracy of the translations. One potential advantage of SVM is that it can handle high-dimensional feature spaces and is relatively robust to overfitting. However, its performance can be sensitive to the choice of kernel function and the regularization parameter, and training can be computationally expensive for large datasets.

**Accuracy(0.013)**

**4-KNeighborsClassifier:**

K-Nearest Neighbors (KNN) is a machine learning algorithm used for classification and regression problems, including problems in natural language processing such as machine translation. In KNN, the input to the algorithm is a set of labeled data points, and the output is the label of a new input data point based on the labels of its k closest neighbors in the training data.

In the context of machine translation, the input to the KNN algorithm might be a sentence in one language and the output might be the corresponding sentence in another language. The input sentence is typically represented as a vector of numerical features, such as word frequencies or embeddings, and the output sentence is represented as a sequence of tokens.

During training, the KNN algorithm stores the input feature vectors and their corresponding output labels. When a new input feature vector is presented to the algorithm during testing, it calculates the distances between the new vector and all the stored feature vectors in the training data. It then selects the k closest neighbors (i.e., the k training feature vectors with the smallest distance to the new input vector), and assigns the output label of the new input vector to the most common label among its k nearest neighbors.

In the context of machine translation, KNN can be used as a standalone algorithm or in combination with other natural language processing techniques to create more accurate and effective translation systems. For example, KNN can be used in conjunction with a language model or a neural machine translation system to improve the accuracy of the translations. One potential advantage of KNN is that it is relatively simple to implement and can be trained quickly with small amounts of data. However, its performance can be sensitive to the choice of distance metric and the value of k.

**Accuracy(0.009)**

**5-GaussianNB :**

Logistic Regression is a statistical algorithm that is commonly used in classification problems, including problems in natural language processing such as machine translation. The logistic regression algorithm is used to predict a binary output variable based on one or more input variables, where the output variable represents the probability of a particular event occurring.

In the context of machine translation, the input to the logistic regression algorithm might be a sentence in one language and the output might be the corresponding sentence in another language. The input sentence is typically represented as a vector of numerical features, such as word frequencies or embeddings, and the output sentence is represented as a sequence of tokens.

**Accuracy(0.014(**