Sentiment Analysis

Sentiment Analysis can detect if a sentence or a paragraph has a positive, negative or neutral meaning.

The system is based on the Rocchio Algorithm and weighted words with an accuracy of over 85 %. This service is developed to recognize and process only English sentences.

The analysis is divided in two parts, the learning and processing.

- Machine Learning

The learning scene can be found in the following path: Assets/SentimentAnalysis/Scenes

The scene contains 2 scripts

1- Sentiment Data Training

Which has several paths to set or modify for the learning part:

a- Path to positive data set: The directory in which the positive data set are stored for reading and training. The datasets are predefined positive or negative comments / tweets. To add more comments for learning you have to add .txt files in the directory path (watch the sample dataset of 100 comments)

b- Path to negative data set: The directory in which the negative data set are stored for reading and training.

c- Path to Stop Words: Path to the file .txt containing stop words

d- Path to Contraction: Path to the file .txt containing contraction words

e- Path to save trained data: Path to save the result of the training process, a .txt file

2- Sentiment Weight Common Words

Contains paths to the common positive / negative words which will be weighted based on the training results an the path to save the result text file

In *SentimentDataTraining.cs* line 85, the method *SaveTheTrainedData* encapsulate the weighting logic, you can change it or optimize for additional precision.

- Sentiment Analysis

The processing scene can be found in the following path: Assets/SentimentAnalysis/Scenes

To use the Sentiment Analysis you must add the script “SentimentAnalysis” in the scene by attaching it to a GameObject.

The script has the following attributes:

1- Path to Positive weighted Words: Path to the file .txt that contains the weighted positive words

2- Path to Negatve weighted Words: Path to the file .txt that contains the weighted negative words

3- Path to Stop Words: Path to the file .txt that contains Stop words

4- Path to Contraction Words: Path to the file .txt that contains the Contraction words

These files must be inside the Resources Folder with a local path : Resources/SentimentDataTrained. If you have a different Resources folder, please merge them together with Analysis Resources Folder.

The System Morale Attribute advices the system to focus more on positive / negative sentences by adding extra weight on them, the default value is setted to neutral.

- How to Initialize and use the Sentiment Analysis

Every SentimentAnalysis instance is Initialized only one time by loading all the data needed for the processing. The initialization can be done in the Awake(), Start(), OnEnable() methods by calling on the SentimentAnalysis intance : ***predictionObject.Initialize();***

The Analysing process is done asynchronously, so events are provided to catch errors and results of the computation.

For error handlers, listeners to events must be provided by adding the following line:

SentimentAnalysis.OnErrorOccurs += Errors;

Where "*Errors*" is a method like the following:

private void Errors(int errorCode, string errorMessage){

Debug.Log(errorMessage + "\nCode: " + errorCode);

}

Error Codes :

- 0 : Instance of the Sentiment Analysis is already initialized
- 1 : Error Loading the .txt files (maybe the file is corrupted or missing)
- 2 : The Analysis is currently on the Thread, to ensure a Thread Safe system only one analysis can be called on the instance, after finishing, other analysis can be executed
- 3 : The .txt file is modified and the format for the text splitting is altered
- 4 : The Database is not initialized
- 5 : Empty text is sended for analysis

For the Result Callback, a listener must be added to the following event:

SentimentAnalysis.OnAnlysisFinished += GetAnalysisFromThread;

Where "*GetAnalysisFromThread"* is a method like the following:

private void GetAnalysisFromThread(Vector3 analysisResult){)

The listener is executed on the Sentiment Analysis Thread, so you cannot use the Unity API. A solution for passing the result to the main Thread is provided in the class : *"SendTextToAnalyse"*

Don't forget to unload the listener when the object is destroyed or the system will catch a null reference. The Unloading is done :

```
void OnDestroy()
{
        // Unload Listeners
        SentimentAnalysis.OnAnlysisFinished -= GetAnalysisFromThread;
        SentimentAnalysis.OnErrorOccurs -= Errors;
}
```

For text processing on the SentimentAnalysis Instance:

predictionObject.PredictSentimentText(textToSend); where "*textToSend*" is of type String