**Architecture**

**EVALUATION MODULE**

Precision, Recall, F1, Support

* **Model: Random Forest**
* **Train on Input Data**
* **Predict/Classify**

**MODEL BLOCK**

Classification Result

**E + Y2**

**E**

**PIPELINE**

**MODEL BLOCK Y4**

**MODEL BLOCK Y2**

**MODEL BLOCK Y3**

**FEATURE COMBINER**

**EMBEDDINGS**

**+**

**MODEL OUTPUT**

**DATA PREPROCESSOR**

Generate clean Data (DataFrames)

* **Clean Duplicate Texts**
* **Remove Word Noise**
* **Translate Text to English**

**EMBEDDING**

Generate Embeddings using TF-IDF Vectorization

**DATA LOADER**

Loads CSV Files and returns unified DataFrames

**AppGallery.csv | Purchasing.csv**

**E**

**E + Y2 + Y3**

**Y3 TONE**

**Y2 INTENT**

**Y4 RESOLUTION**

This architecture design will classify emails into multiple inter-dependent categories such as Intent (Type 2), Tone (Type 3) and Resolution (Type 4). A sequential pipeline is created where the prediction of each model is fed into the next model. The system begins with the Data Loader block where the customer messages from CSV files are loaded. This is followed by a Data Preprocessor block that cleans the data, normalizes it by removing duplicate texts, word noise and then translates the cleaned text into English language. The Embedding block transforms the cleaned text into numerical embeddings using TF-IDF technique, a popular one in NLP. A pipeline is created which manages the chaining of Random Forest Classification models and ensures proper data flow. The first model is fed only with embedding data to predict Intent (Type 2). The output (Y2) thus generated is then sent to Feature Combiner block which combines embeddings and the Y2, which is fed into the second model to predict Tone (Type 3). The result (Y3) generated is then again combined with embeddings, Y2 and Y3 and fed into the third model to predict Resolution (Type 4). Between each model stage, the Feature Combiner block will merge the embeddings with the previous model outputs to make sure the models have the context from previous outcomes. Finally, the Evaluation Module will compute performance metrics such as Precision, Recall, F1 Score and Support.

**Modifiability**

Since the architecture consists of multiple modules such as data loading, preprocessing, embedding, and also a pipeline, the architecture is modular and hence easily modifiable. It is easy to replace embedding techniques such as TF0IDF or experiment on different machine learning models without affecting other blocks. Additional layers can also be added to the pipeline with minimal modification.

**Performance**

Chained model architecture is easy to implement, and the accuracy can be greatly improved as the errors get filtered as the context passes down the model chain. However, misclassification in earlier stages will propagate the errors down the chain, thus affecting the prediction of later models. Running several models in a chain can increase computing time.

**Scalability**

The architecture can be greatly scalable since the models operate individually in the pipeline. With minimal modification, additional model blocks can be added to the pipeline without much effort. This will help in increasing the functionalities provided by the architecture. As the chain grows larger, it becomes difficult to maintain as failure in one model block will affect the rest following the chain.