**NLP Sentiment Analysis through ANN and ALBERT Architectures**

*German Baltazar, Anyinssan Nava, David Lizama*

For this project, a text dataset was selected to determine the type of conversation each phrase denoted, classifying it into three different labels. Two neural network models were implemented to achieve this goal: the regular Artificial Neural Network (ANN) and the small Transformer architecture of ALBERT.

The selected dataset was a subcategory of the Sequence labellIng evaLuatIon benChmark fOr spoken laNguagE (SILICONE), a public benchmark with multiple scripted scenarios that cover daily life situations. This dataset consisted of a series of daily dialog acts, classified into four acts: commissive, directive, inform, and question. The data distribution can be seen in Table I, proving that the information for the training, validation, and test sets are enough to train an adequate model. However, due to the number of phrases categorized as commissive and directive, these sentences were combined into a general category of chat, resulting in the data distribution shown in Figure 1. For the rest of the project, each label will follow the next classification:

* Class 0. Regular chat (commissive and directive)
* Class 1. Inform
* Class 2. Question

After properly categorizing the observations of the dataset, each sentence of the training set were evaluated to observe the length distribution. In this case, the length analysis was made before and after preprocessing the sentences, as shown in Figure 2. The preprocessing consisted of deleting every non-ASCII character and any punctuation mark and stopword in each sentence; finally, each phrase was transformed into its lower case, lemmatizing the verbs and removing single letters.

Table I. Dataset observations

|  | **Training Set** | **Validation Set** | **Test Set** |
| --- | --- | --- | --- |
| *Original size* | 87,170 | 8,069 | 7,740 |
| *Deleting duplicates* | 72,391 | 7,682 | 7,469 |

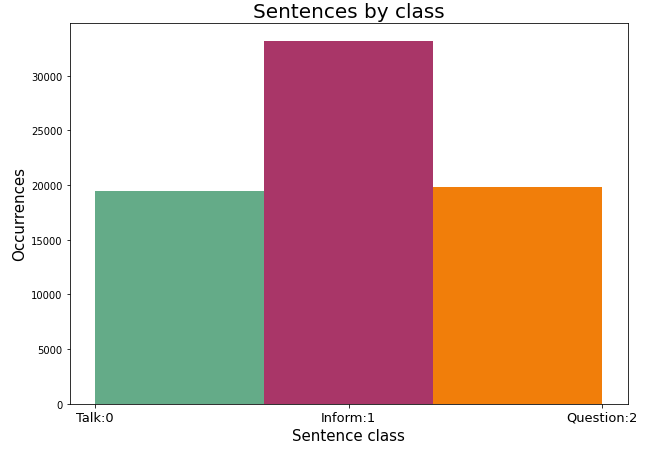
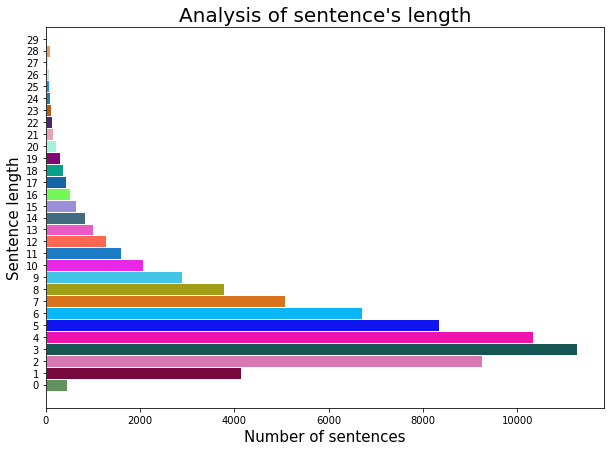
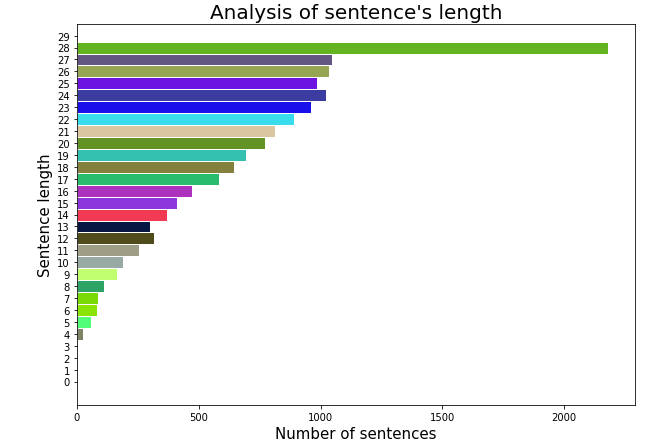


Figure 1. Class distribution



a) b)

Figure 2. Sentence length a) before and b) after pre-processing data

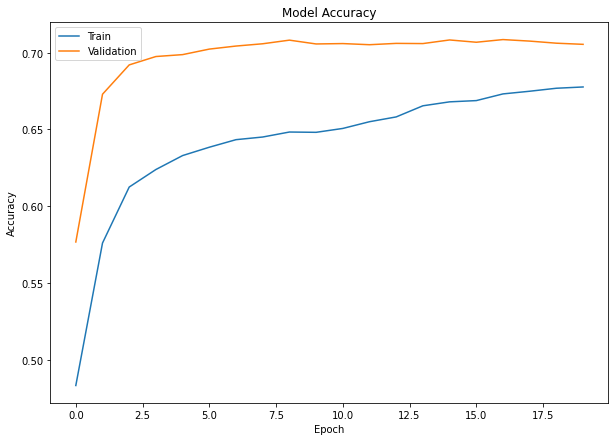
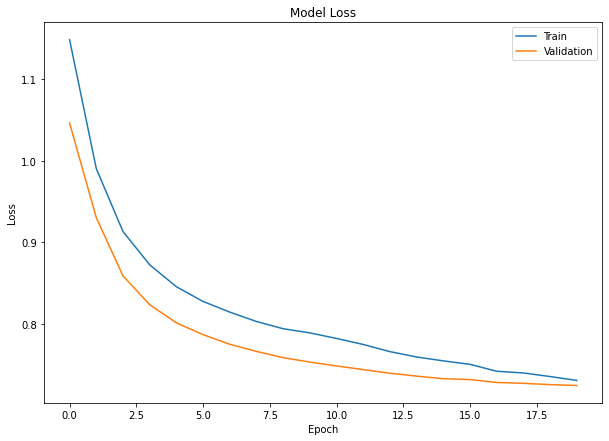
Once the dataset was pre-processed, two different neural networks were trained to generate the classification of the given phrases. The first model consisted of a regular ANN architecture, with one hidden layer of four neurons with a ReLU activation function. A 20% dropout was used on the model as well. The output layer of the model had a total of three neurons (one per class), with its corresponding Softmax activation function. The NNLM embedder was used to parse the string inputs of the sentences into their corresponding embeddings for the model's training; this embedder was trained on a 7B corpus of Google News to generate 50-dimensional embeddings of English words. Finally, the model was trained for 20 epochs with the Adam optimization algorithm and a batch size of 512 observations.

As shown in Figure 3, even though the model's loss function was converging to a lower value, the model's accuracy followed an underfitting behavior, meaning that the model required much more information to generalize the data correctly. This behavior could be solved by augmenting the complexity of the model, increasing the number of hidden layers and neurons, or using a different architecture. Table II shows the general results of each of the sets’ accuracies.

The low accuracy of the model is better described in Figure 4, where the confusion matrix of the model is expressed. In this image, it can be seen that it is pretty difficult for the model to correctly classify the conversations of regular chats, easily confusing them with just customary informative phrases.

Table II. ANN accuracy comparison

| **Training Set** | **Validation Set** | **Test Set** |
| --- | --- | --- |
| 67.77% | 70.54% | 73.08% |



a) b)

Figure 3. a) Loss function and b) accuracy of the ANN model

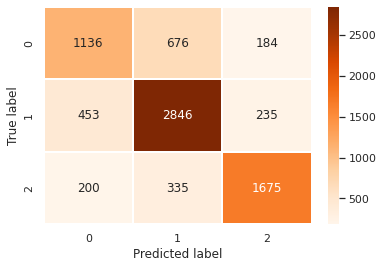


Figure 4. Confusion matrix for the ANN model

Now, the process of applying the ALBERT model is described. ALBERT is an algorithm of NLP that allows one to analyze large amounts of text in a self-supervised way. This model is one of the best algorithms for NLP tasks thanks to its higher accuracy and fast classification capacities compared to regular neural networks, like the LSTM model.

To apply the algorithm is necessary first to do the embedding of the words of the sentences into a vectorized numeric form. A word embedding is a learned representation of a text, where words with the same meaning have a similar representation. Each word is mapped to a vector, and vector values are learned in a way that resembles a vector.

One of the reasons why the ALBERT algorithm is one of the state-of-the-art models is due to the enormous corpus that it uses for training; this quality allows the model to generalize enough the structure of any sentence, making unnecessary the pre-processing of the sentences.

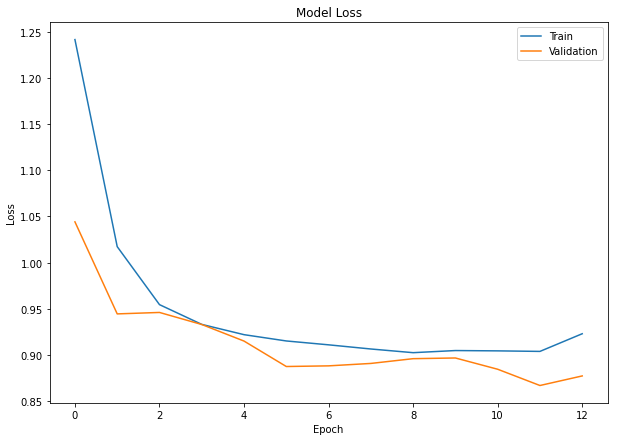
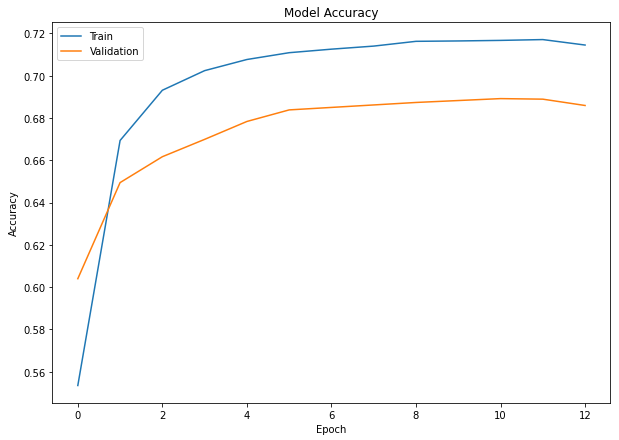
For this case, we used transfer learning of the ALBERT model as the initial phase of the model. A 10% dropout of the neurons was also used to avoid overfitting. The final layer used three neurons for the output; since there are three different classes, using the softmax function is necessary for solving the classification problem with more than two labels.

The different accuracies of the model can be seen in Table III. This table shows that the model had a good performance since the accuracy of the train, validation, and test data accuracy are very similar. Also, the accuracy incremented on each epoch. However, due to the computing power difficulties, the model was not capable of further training to increase the accuracy further.

Figure 6 shows that the ALBERT model has classification problems between informative texts and regular chats. This is a reasonable outcome considering that there was not enough training time for the model to start generalizing enough to separate these two labels.

Table III. ALBERT accuracy comparison

| **Training Set** | **Validation Set** | **Test Set** |
| --- | --- | --- |
| 71.45% | 68.59% | 73.12% |



a) b)

Figure 5. a) Loss function and b) accuracy of the ALBERT model

As a final implementation of the model, an API was designed to prove the functionality of the classification of both algorithms. However, due to technical difficulties, only the ANN model could be implemented on the demo code. Figure 7 shows the general structure of the demonstration page.

Even though the available accuracy of the selected models was not high enough, it can still be demonstrated that the models can differentiate between different types of sentences, being informative or questions. For general use, these models could be implemented as the first phase of a narrative algorithm or a chatbot, detecting how a human user communicates and creating a more natural and fluent conversation with him.

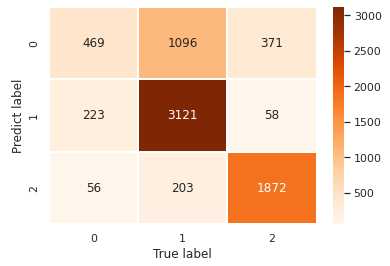


Figure 6. Confusion matrix for the ALBERT model

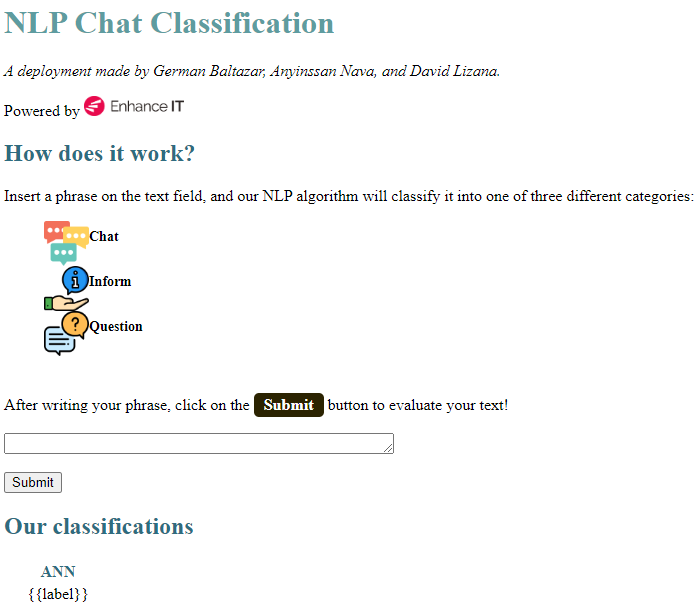


Figure 7. API demo for the NLP use case