EmotivITA Sentence Classification

Yusupha Juwara
Sapienza University of Rome
yusuphajuwara@gmail.com

Abstract

This report walks through the key details of the implementation of multiple models on the EmotivITA dataset for sentence classification task for the Italian language. This work aims to establish baseline models and then implement more robust models from the RNN family that can outperform them. I experimented with 3 statistics-based baselines, 2 Logistics Regressions (one with embedding layer), and some combinations of **BiLSTM** and **BiGRU** models. I was able to achieve varying but good results as elaborated in the Results section 4. I also experimented with **triple head** – like **Siamese networks** – where, instead of having a single output layer, it has a triple. This is the same as having 3 different networks. However, 3 different networks require more compute but more stable and faster to converge, thus I used that.

1 The Dataset (EmotivITA

EmoITA (Gafà et al., 2023) is the Italian version of the EmoBank dataset, annotated based on the dimensional model of emotions, where each of the 3 dimensions (Valence, Arousal, and Dominance) is a real-value associated with a sentence. Note: in this task, we are required to map the real-values to categorical labels and separate the dimensions.

In this work, I did some preprocessing steps, such as removing **punctuations**, **stopworks**, **tokenizing** the texts, **lemmatizing** them (see lemmatization vs. stemming), etc.

2 Baselines

I experimented with 5 baselines: **Random Sampling**, **Weighted Random Sampling**, **Majority Voting/Class**, two simple **Logisitics Regressions** one with **Bag-of-Words (BoW)** representation and the other with **Embeddings**. The baselines serve as benchmarks for the more robust models, such as the **BiLSTM**, **BiGRU**. **Note that I consider the BiGRU to be my extra baseline as required!**

3 Design Choices of the Models

It should be noted that many of my final design choices were based on experimentation with the limited resources at my disposal.

During the design of the models, I had many solid design choices, such as a (good) model should have **feature extraction** layer(s), **classification** layer, have enough **capacity** to model the data distribution well enough to be generalizable. With these in mind, I did a **Grid Search** technique to find some of the **Hyperparameters**, including varying the number of layers, different **Learning Rate Schedulers** and **Optimizers**, different model choices from the RNN family (LSTM, GRU), etc.

I also experimented with **pretrained** embeddings. However, there were no significant performance gains and they require a lot of compute and are really slow.

To mitigate **underfitting**, I experimented with different models of varying numbers of layers to ensure that the models have enough capacity for the data sets and the task at hand, and then observed how the performances vary over the capacities.

For the **overfitting** mitigation, techniques such as **Batch Normalization**, **Early Stopping**, **Learning Rate Annealing**, etc were employed.

4 Results

Below are the results obtained for this task. Note that instead of reporting the performances based only on the test sets, I also report the performances on the validation set. In this way, the reader can have a broader view on the difference in performance and whether the models are over-fitted without the need to look at the **notebook** itself. However, to gain more insights into the performances such as the **classification report**, **confusion matrix**, etc, have a look at the notebook. Everything is reported there!

4.1 Baseline Results

Random Sampling: After averaging over 1000 runs, it has an average accuracy of ($\approx 33.3\%$) same as the theoretical random sampling value between three classes $\approx \frac{1}{3}$.

Weighted/Stratified Random Sampling: This is just slightly better than the non-weighted random sampling above. It has an accuracy of $\approx 36.68\%$

Majority Classifier/Voting: With an accuracy of $\approx$ 47.7%, it outperforms the previous two by more than 10%. This is not surprising though, since the data distribution is skewed towards it.

For the two Logistics Regression models, the one with BoW obtained an accuracy of $\approx 48.2\%$ on the validation set and $\approx 48.15\%$ on the test set, while the other with embedding has $\approx 56.66\%$ on the validation set and $\approx 50.49\%$ on the test set.

4.2 BiLSTM

Experimenting with varying model capacities, the **BiLSTM** model obtains an accuracy of 60.2% with 6 bilstm layers on the validation set and $\approx 50.1\%$ on the test set.

4.3 Triple heads

The accuracies on the validation set of each model of the triplet:

- **Arousal**: $\approx 51.5\%$ on the validation set and $\approx 49.64\%$ on the test set.
- **Dominance**: $\approx 54.58\%$ on the validation set and $\approx 43.1\%$ on the test set
- **Valence**: $\approx 50.1\%$ on the validation set and $\approx 46.3\%$ on the test set

To aggregate the results from the triplets on the test set:

- By averaging the probabilities: $\approx 42.4\%$
- Prediction of the model with **Max prob**: $\approx 42.73\%$
- By **Majority voting**: $\approx 42.4\%$

4.4 Extra Baseline: BiGRU

Note that the **BiGRU** is similar to the **BiLSTM** architecture, except that it replaces the lstm layer(s) with the gru layer(s). In fact, in my implementation, it inherits from the lstm model. The **BiGRU** has an accuracy of $\approx 60.61\%$ on the validation set and $\approx 50.1\%$ on the test set, on par with the **BiLSTM** counterpart.

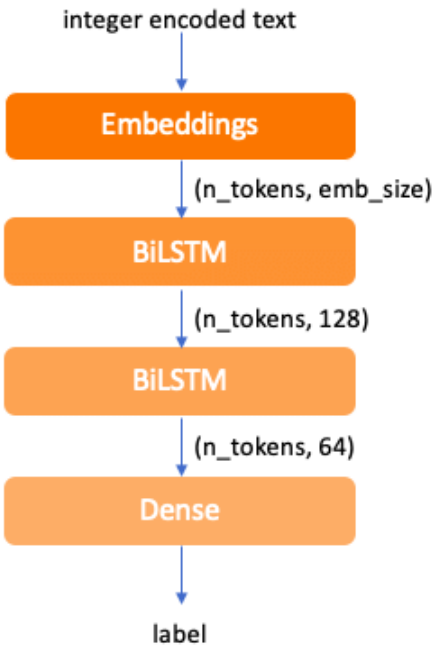
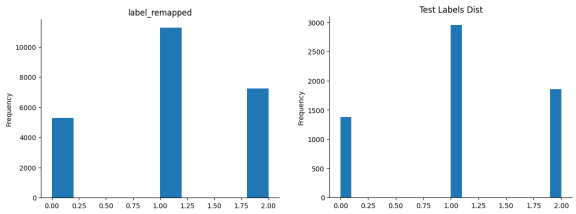
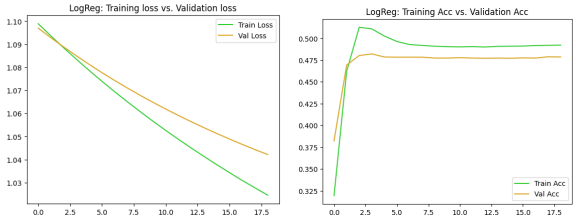


Figure 1: BiLSTM architecture with embedding and classification layers



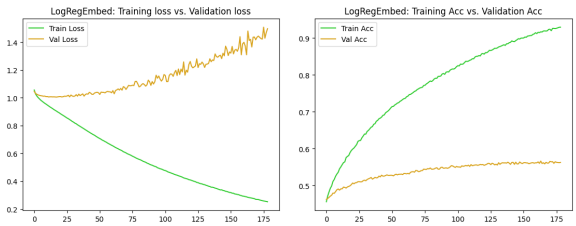
(a) Training Data Distribution (b) Testing Data Distribution

Figure 2: The datasets distributions



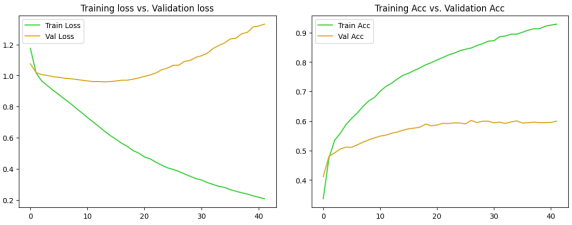
(a) training vs. validation losses (b) training vs. validation accuracies

Figure 3: Logistics Regression with BoW representation



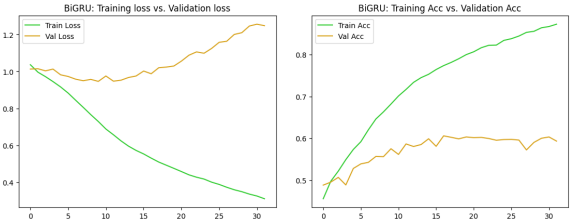
(a) training vs. validation losses (b) training vs. validation accuracies

Figure 4: Logistics Regression with Embedding representation



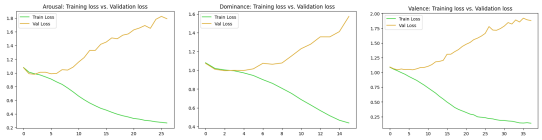
(a) training vs. validation losses (b) training vs. validation accuracies

Figure 5: Bidirectional Long-Short Term Memory (BiLSTM)



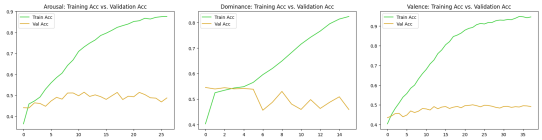
(a) training vs. validation losses (b) training vs. validation accuracies

Figure 6: Bidirectional Gated Recurrent Units (BiGRU)



(a) Arousal (b) Dominance (c) Valence

Figure 7: training vs. validation losses of each triple head



(a) Arousal (b) Dominance (c) Valence

Figure 8: training vs. validation accuracies of each triple head

References

Giovanni Gafà, Francesco Cutugno, and Marco Venuti. 2023. Emotivita at evalita2023: Overview of the dimensional and multidimensional emotion analysis task. In *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023)*. CEUR.org.