

# Diplomado Inteligencia Artificial y Ciencia de Datos: Métodos fundamentales, PrograLógica, Ciencia de Computación e Ingeniería de Computación.

---

## Report - Transformer Translation

**Student:** García Martínez Zoé Ariel

**E-mail:** zoe.garcma@gmail.com

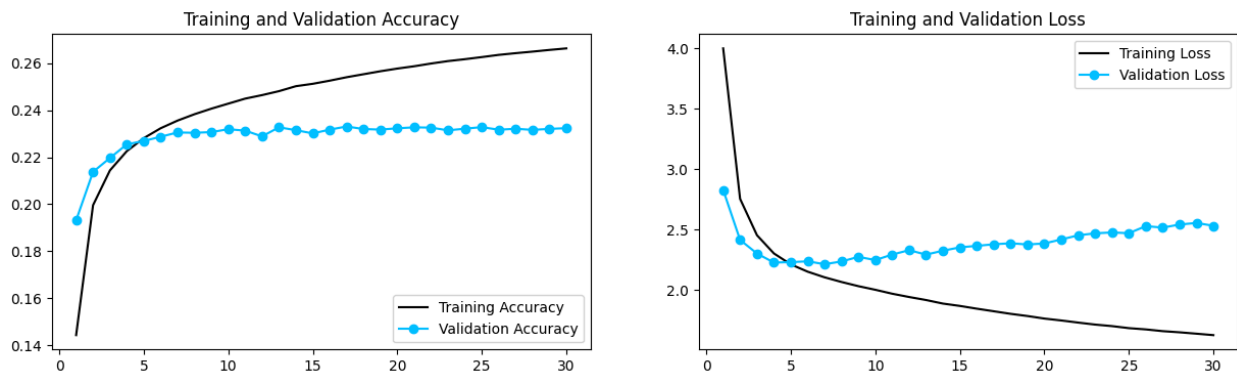
**TL;DR:** In this report I show the results while training a transformer (encoder - decoder) model to perform translations from english to spanish, using generated embedding words, and pre-trained embedding words.

### Start: Simple Embedding Words

I used the embedding method suggested in this reference from jchollet ([here](#)). The challenge here was to understand how pairs and embedding works.

After 30 epochs the model performed quite well, being able to perform translations. The Acc value reached  $\sim 23\%$  and loss function was  $\sim 2.5$ .

Inspecting the graph we can see that the model stopped learning at



IMG1: Transformer training Acc and Loss curves

### Example sentences translated

```
You are the only one who can protect them. ---> [start] eres el único que puede abrir el que puede que los [UNK] [end]
You know this. ---> [start] tú sabes esto [end]
The dog went away. ---> [start] el perro se salió [end]
```

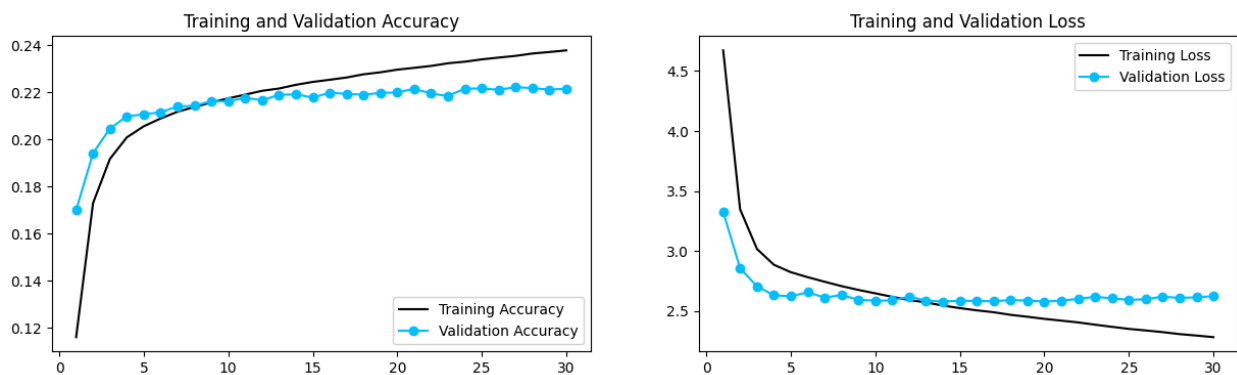
## Second Step: GloVe Embedding Words

In this configuration the challenge was to modify the embedding class previously defined in the code to allow us passing it fixed embeddings, as the one from GloVe ([Stanford Embeddings](#)).

The model was trained with two configurations:

### Config - 1: First english embedding Freezed and Second embedding Trainable

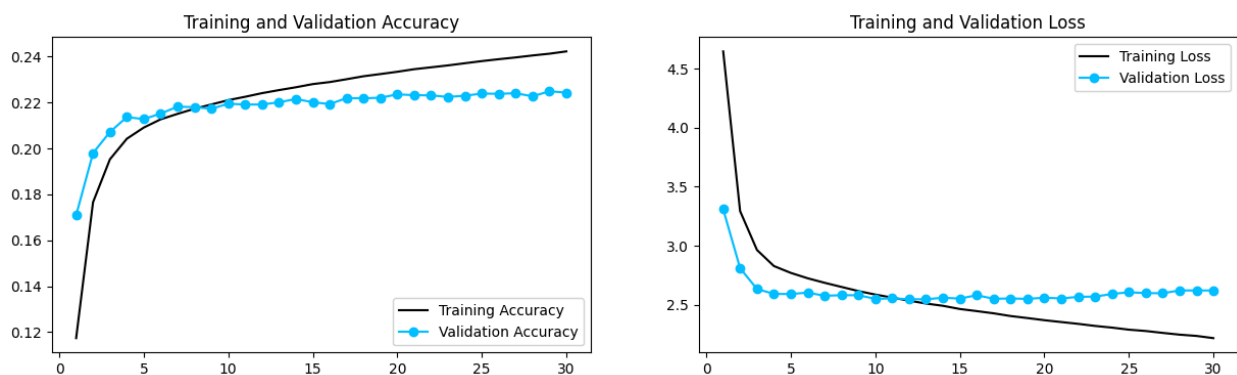
In this configuration the model performed worse compared vs the original configuration, the training converged faster and reached  $\sim 22\%$  of Acc, and  $\sim 2.6$  loss.



IMG2: Transformer training Acc and Loss curves for Glove Config-1

### Config - 2: First english embedding Trainable and Second embedding Trainable

In this configuration the model performed worse compared vs the original configuration, the training converged faster and reached  $\sim 22\%$  of Acc, and  $\sim 2.6$  loss.

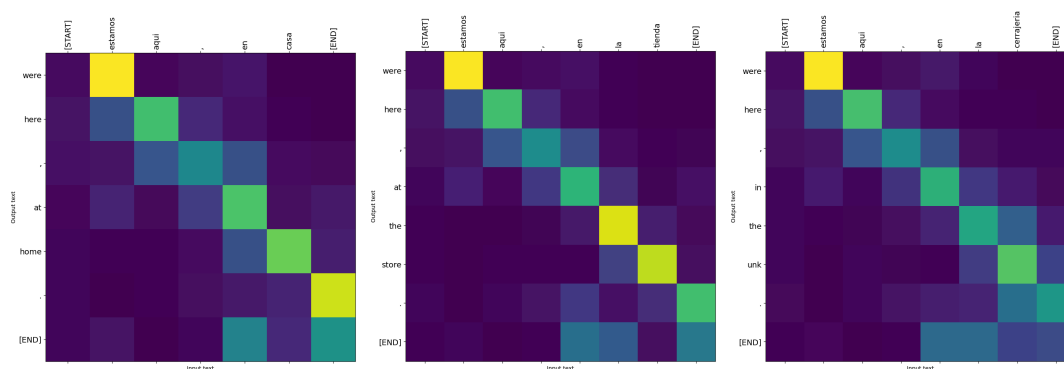


IMG3: Transformer training Acc and Loss curves for Glove Config-2

*In both Configurations the translations were as good as the ones obtained in the original one.*

## Third Step: Visualizing Attention

In this section of the challenge the main issue for me was while defining the model to pull out the weights of the MultiHeadAttention layer without messing up the training. Once fixed the following graphs were obtained:



IMG4: Visualization of weights with similar entries

It's interesting to see how the query works here, what is happening is that the weights indicate whenever a high probability of having the most accurate value for the query.

## Extra Step: Translating los Miserables!

I performed this translation from French to Spanish using the Marian Model from the transformers package in TF/Keras.

I got pretty good results with it:

Ex:

"[...]las costumbres una condenación social que crea artificialmente en plena civilización del infierno y complicando de una fatalidad humana el destino que es divina como los tres problemas del siglo la degradación de el proletariado la pérdida de la mujer por el hambre latitrofia de la niñez por la noche no será no resueltas mientras que en algunas regiones la asfixia social será[...]"

I annexed the code.