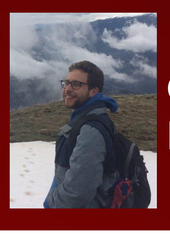
English Text Normalisation Using Attention Based Mechanisms


Carlos Arenas

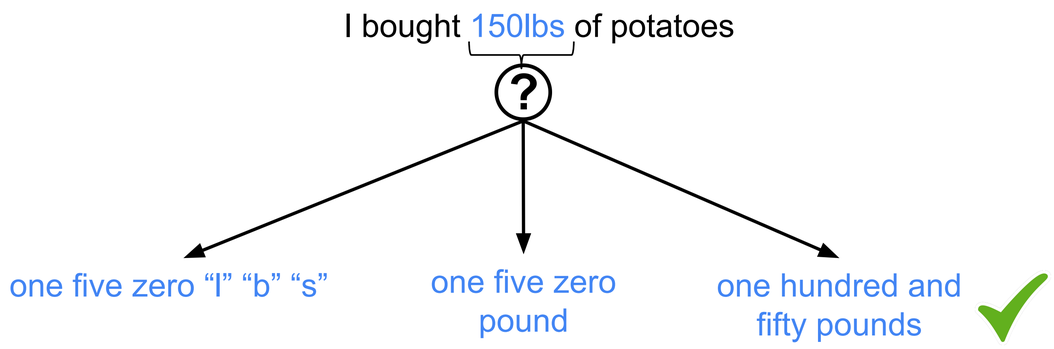

Marc Combalia


Daniel Moreno

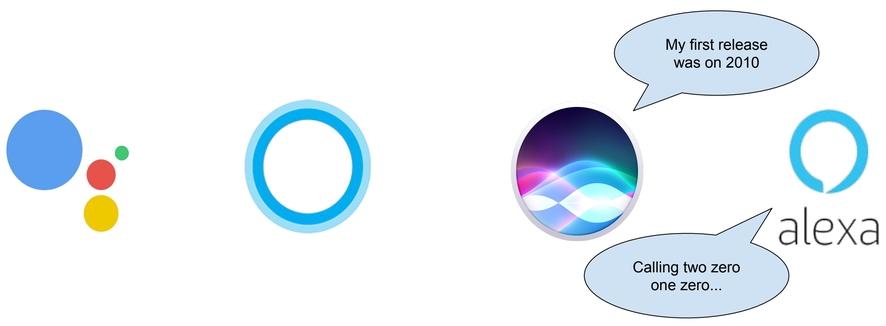

Guillem Lahuerta

Motivation

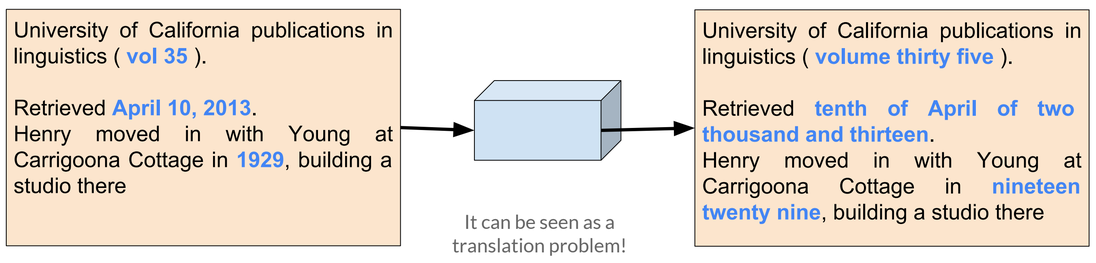
Writen or spoken english is far from being understandable for an AI



This allows easier user-machine interaction

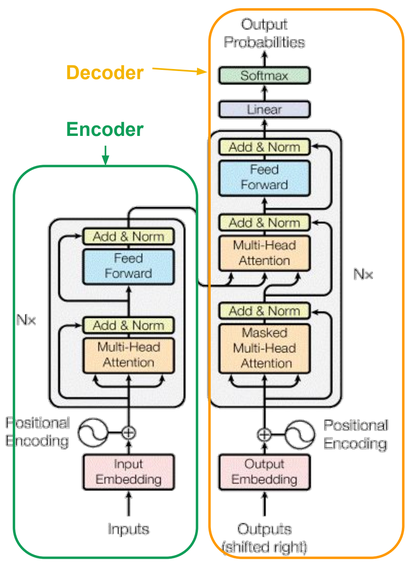


Desired behavior



Attention

We have based our work on the Transformer[1]



As this is a problem of Natural Language Processing, we could have chosen a RNN or an Attention network, however, we decided Attention as RNNs show:

- Vanishing gradients: the network can’t see the whole text
- Difficult to fully compact a sentence on a single vector
- Slow to train
- Not the way humans read a text

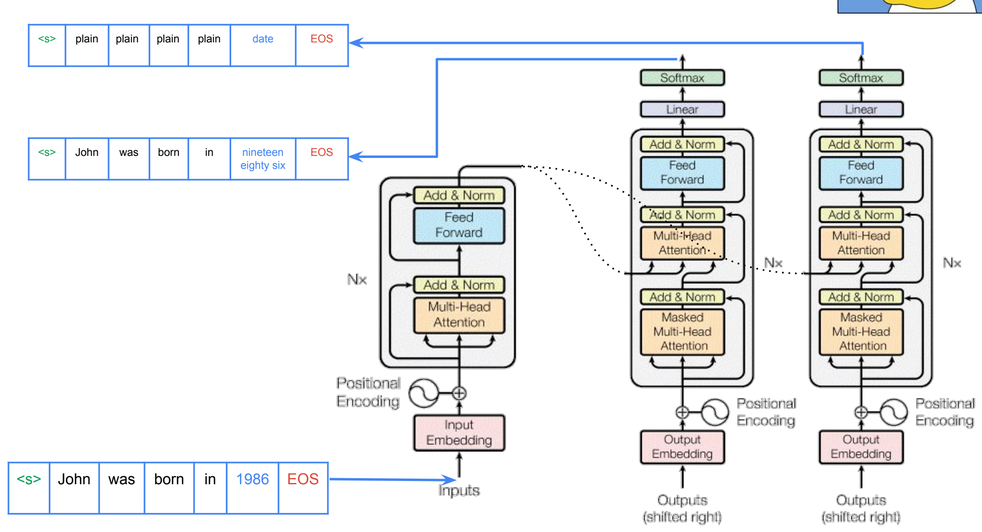
On the other hand, Attention based networks allow us to:

- Compress the input sentence into a set of vectors
- The linear combination of vectors is used to select the next word.
- Parallelize
- Focus on the most importants part of the text

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. CoRR, abs/1706.03762, 2017.

Proposed Modification

What if we could take into account the type of word to improve the translation?



Multi-Tasking

By using two decoders, we can obtain both, the translation and the class of each word in the training set and use this information to train the network. The new network absorbs the information about the classes by taking into account a mixed loss from both translations.

$$\mathcal{L}_{\text{Total}}(w, \sigma_t, \sigma_c) = \frac{1}{2\sigma_t^2}\mathcal{L}_t(w) + \frac{1}{2\sigma_c^2}\mathcal{L}_c(w) + \log \sigma_t^2 + \log \sigma_c^2$$

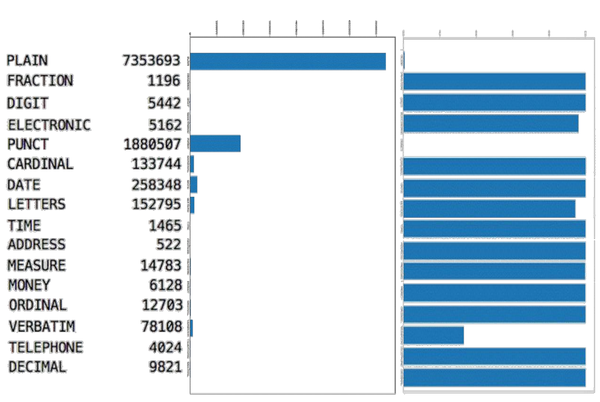
In the equation above, we define a total loss where we combine the losses of both decoders. Despite implementeing a naive approach (e.g. arithmetic mean), we implement a multi-task loss where we combine the losses of each decoder according to their parameters.

Our Data

We have used a set of 10M words distributed in 16 classes

	sentence_id	token_id	class	before	after
0	0	0	PLAIN	Brillantaisia	Brillantaisia
1	0	1	PLAIN	is	is
2	0	2	PLAIN	a	a
3	0	3	PLAIN	genus	genus
4	0	4	PLAIN	of	of

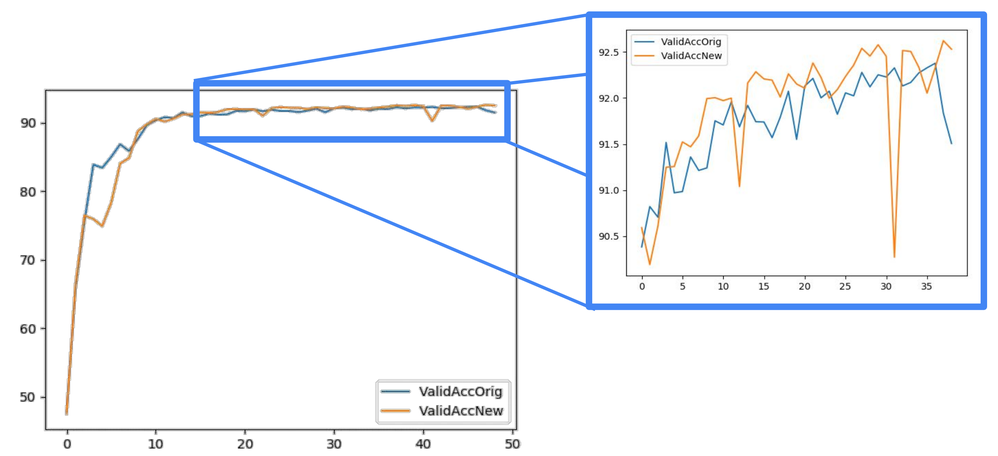
Each word was labeled with its class, its original form and the desired form after normalisation. There is a huge variation in the amount of examples from each class, being PLAIN the most common class. However, classes have some internal rules from which we can take advantage.



The figures above represent the class distribution over the database (left) and the percentual change of words from each class.

The PLAIN class, despite being the most represented class, it has one of the lowest variations. On the other hand, classes like FRACTION, DATE and many others, despite representing less than a 2.5% of the total set, they show a 100% of variation.

Experiments



We performed a first set of results by comparing both networks, the original transformed and the modified version including two decoders. The mixture of losses was done with a naive approach, both losses weighted the same when adding them. For this iteration we also used a partition of the database.

The results show that, with the naive approach, there is a slight improvement, however, it is not enough to affirm that there is a real improvement.