María Leva Jiménez and Eloi Singla i Milian
Natural Language Processing

Name generation using a MLP

Introduction

For this task, we have trained two character-level language models based on the code provided by Andrej Karpathy in his *Neural Networks: From Zero to Hero Series* to generate person names in Catalan and Spanish (task 2A). We have also assessed the transfer between the two models to see how they perform when generating names in the language they were not trained with. Our expectations would be that the loss for the transfer would not be significantly higher than the one for the testing sets, due to the similarity between the two languages names.

Material and methods

The data we have used to train the models were essentially two lists of common names in Catalan in Spanish. The Catalan list is composed of 1594 names and was extracted from the webpage "Noms del Mon". After processing it, 4 of those names were dropped, so the final training input was 1590 names. The Spanish list was taken from the webpage of the "Instituto Nacional de Estadística", and it contains names with a frequency equal or higher than 20. The list was shortened so that both models had the same number of input during training.

The only preprocessing the data received was converting the original texts into lists and turning the names into lowercase. All characters that occurred in the lists were used, including spaces and characters with diacritics. The special start-end character we chose was "#", because the standard "." occurred in another list we used for development.

We developed a function that converted all characters that were non-alphabetic (in the literal sense) into letters found in the alphabet. However, we ended up not simplifying the data this way. The data was split into the training-validation-test sets in an 8:1:1 ratio.

The two models had only two layers. Model 1 (M1, Catalan names generator) had 50 hidden dimensions, while model 2 (M2, Spanish names generator) had 100. Batch size was set at 32 and block size at 3. Both models ran for 30000 epochs. The learning rate changed through the training: before epoch 15000 it was 0.1, after and before 25000 0.05, and after and until the end it was 0.01. For reproducibility of results, both models used set seeds. Tanh was used to introduce nonlinearity following Karpathy's code. The loss function used to assess the quality of the models' predictions is cross-entropy. We optimized the parameters to try to reduce overfitting while also minimizing the validation loss.

Results

After training, we first inspected the output of the models. We had each one generate 25 names. They ranged from 3 to 16 characters in Catalan and from 3 to 23 in Spanish (Tables 1 and 2 in Appendix). As the output in both tables show, it seems M2 performed better than M1, even though its generated names are still not acceptable. Some weirdness in M1's names could be explained by the

fact that a lot of the names in the list have a biblical origin and are unusual names for people nowadays.

The metrics used to quantitatively evaluate the models' performance (loss computed by means of cross-entropy, Table 3 in Appendix) confirm this in each step of the models' training: cf. a train loss of 2.26 for M1 and of 1.49 for M1, a validation loss of 2.33 for M1 and of 1.77 for M2, and a testing loss of 2.45 for M1 and of 1.84 for M2. Even though validation loss is usually expected to increase slightly with respect to training loss, M2 shows a greater difference, which can be accounted for by overfitting on the training data. Still, it seems that the lack of data made the models reach a limit and a significantly lower loss could not be achieved by tweaking the parameters.

M1 shows a greater than expected increase between validation and testing loss, which proves the model has difficulties in capturing the patterns in the data. This is also the intuition behind the embedding plot of M1, which shows how the model has trouble distinguishing between some characters such as letters with diacritics and seemingly less frequent consonants (cf. with the embedding plot of M2, Figures 1 and 2 in Appendix). Since both models have a similar architecture and were fine-tuned by the training data itself, these differences in performance can only be explained by the data. The dataset of Catalan names employs more diacritics and is more heterogeneous than the dataset of Spanish names, which repeated names in double names (hence the validation loss' value). Also, M2 had more examples in practice due to its training dataset including double names (which the Catalan dataset did not include at all), which made the names longer and therefore with more examples per name. The code was designed to optimize the number of examples given to each of the models as input, but double names still gave an advantage to M2.

The loss values for the transfer task (exercise 2A) were surprising, since Catalan and Spanish names tend to be quite similar. Both models' performance was suboptimal (cf. 3.17 for M1 and 3.69 for M2, Table 4 in Appendix), with M2 being worse than M1 in predicting names despite the former model's performance being better during training. Again, the input data accounts for this, since the Spanish dataset is more homogeneous than the Catalan one and contains fewer diacritics.

If anything, this exercise brings home a key issue in Machine Learning: the quality and quantity of the training data. The limitations for this task revolve precisely around the data used to train the models, since neither their architecture nor their parameters can be further improved. Perhaps results could be optimized by turning to the data (e.g. homogenizing the Catalan dataset or controlling the split datasets to make them more balanced), using more data or a different architecture relying on attention.

References

Instituto Nacional de Estadística. (2024). *Apellidos y nombres más frecuentes: Todos los nombres con frecuencia igual o mayor a 20 personas.* https://www.ine.es/daco/daco42/nombyapel/nombres_por_edad_media.xlsx

Karpathy, A. (2022). *Neural Networks: From Zero to Hero. Lecture 3: Building makemore Part 2: MLP.* [Computer software]. Github. https://github.com/karpathy/makemore/blob/master/makemore.py

Nomsdelmon.cat. (2025). *Noms Catalans (tots)*. https://nomsdelmon.cat/catalans/?lletra=tots

Appendix

Table 1: Catalan names generated by M1

dlra
glúbia
alo
zen·le
belia
son
mansòa
bei
periand
nala
calba
eleni
mol·l
amam
car
builla
cabacboricia
mexbestaconis
naj
aercarreda
sanos
vamk
ari
ocfesitinaïdore
sisangaramiet

Table 2: Spanish names generated by M2

dro javier davier
dan
bkai
alejandrian carlos dano
san
csa
cesarmoned
arkah
bpancisco sabi
romerto
crge luis
ror jose
en
lussio
car
juan
cevadam
atinay
jaote
francisco alfonsoloo
manuel
pet
maros
luis andres
mur

Table 3: Performance metrics for M1 and M2 during training

	Model 1	Model 2
Training loss	2.26	1.49
Validation loss	2.33	1.77
Testing loss	2.45	1.84

Table 4: Performance metrics for M1 and M2 in transfer learning

	Loss value
Model 1	3.17
Model 2	3.69

Figure 1: Embedding plot for model 1

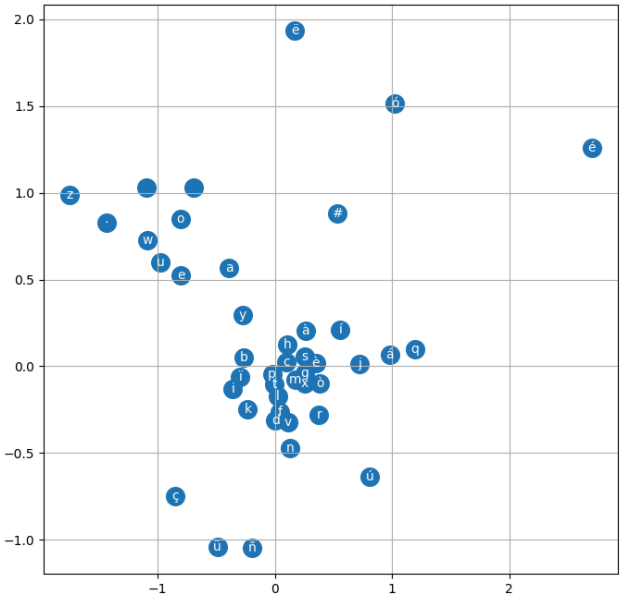
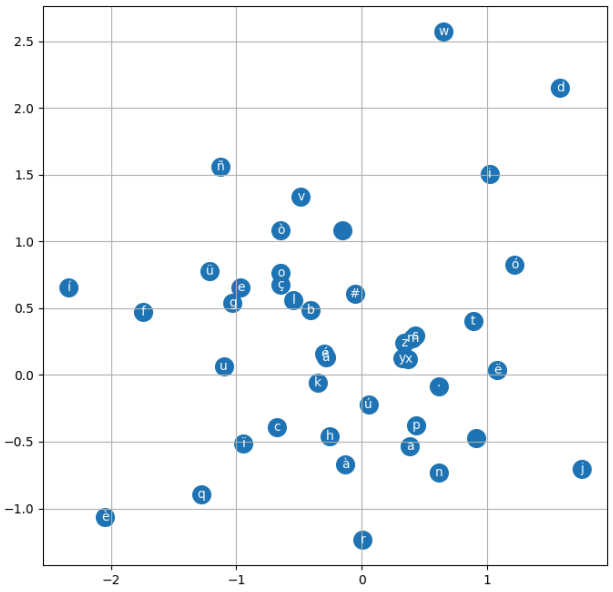


Figure 2: Embedding plot for model 2



List of contributions

Coding / checking the writing: Eloi Singla i Milian

Writing / checking the code: Maria Leva Jiménez

Link to code and data repository

https://github.com/EloiSinglaMilian/NLP-Exercise-2/