Multilingual Author Profiling using LSTMs

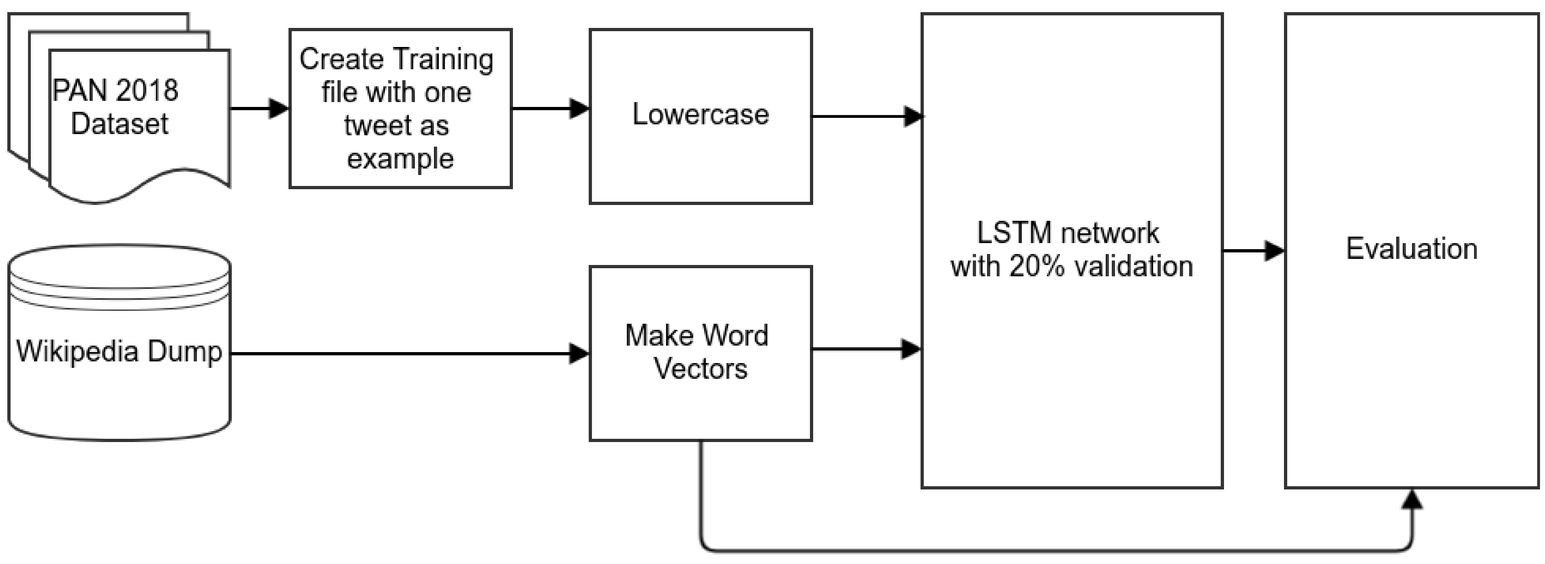
Notebook for PAN at CLEF 2018

Roy Bayot and Teresa Gonçalves
Department of Informatics, University of Évora, Portugal
d11668@alunos.uevora.pt, tcg@uevora.pt

1. Abstract

Author profiling has been one of the thrusts in PAN and CLEF. It is quite useful especially in today's communication methods where social media contains a lot of fake profiles. In this year's PAN edition, an author's profile is determined using texts in the form of tweets as well as pictures. Our approach has been solely on tweets. We use a pipeline of word embeddings, a simple LSTM, a dense layer and an output layer for this task. We achieved an accuracy of 67.60% for Arabic, 77.16% for English, and 68.73% for Spanish gender classification. However, there needs more improvement in the approach.

2. System Architecture



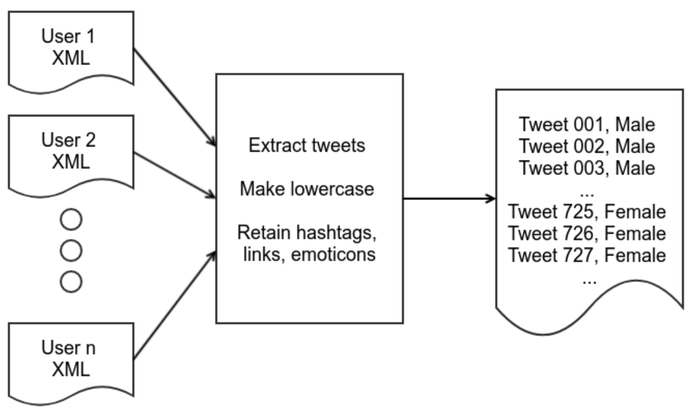
The LSTM network model is simple. The sequence input is set to 64 tokens. We pad the inputs in case it comes short. Then the inputs go to an embedding layer with 300 dimensions, followed by an LSTM layer (with dropout and recurrent dropout), then to a dense layer with 32 units, and then to the output. We used stochastic gradient descent over shuffled mini-batches with the Adam update rule. The mini-batch had 4096 examples and the number of epochs is 200 with early stopping.

3. Dataset

This year there are three different languages - English, Spanish, and Arabic. Each language has multiple users that have different tweets and images that could be used to classify. The table describes the dataset.

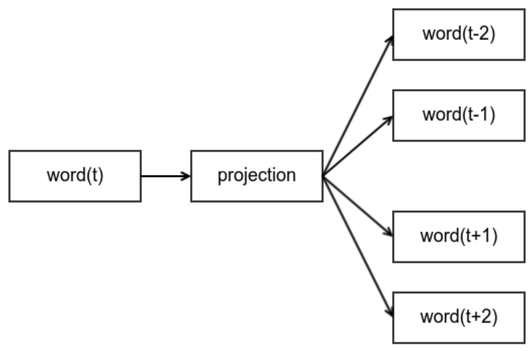
Language	Users	Tweets/user	Img/user
English	3000	100	10
Spanish	3000	100	10
Arabic	1500	100	10

4. Preprocessing



All texts were extracted from the user's xml file, then the tweets were put to lowercase. One tweet is one training example. No stop words are removed. Hash tags, numbers, mentions, shares, and retweets were not processed or transformed to anything else.

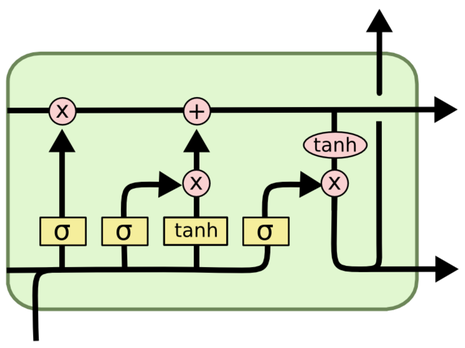
5. Word Vectors



Word embeddings were created from wikipedia dumps with 11.8Gb(EN), 2.2Gb(ES), and 600Mb(AR). The word2vec were generated from these text using Gensim. No lemmatization was done, and the window size used was 5. Skip grams was used as the method to generate the vectors with 300 dimensions.

6. LSTMs

There are various neural network architectures available. Networks with long short term memory units were the ones we used. It was developed by Hochreiter and Schmidhuber such that the units can process longer sequences.



This is done by making the units have a control that allows information to pass or not, which makes it different from other neural networks. We only used one LSTM layer with 32 units for the output, a dropout value of 0.2 and recurrent dropout of 0.5. We found these values from prior experiments.

7. Results

We submitted our runs to TIRA and we were able to get the following accuracy results on the held-out test set:

Language	**Accuracy**
English	77.16
Spanish	68.73
Arabic	67.60

Comparing with the results from other contestants, our approach was 18th globally. We ranked 20 out of 23 for Arabic, 17 out of 23 for English, and 19 out of 23 for Spanish.

8. Conclusions and Recommendations

We did a naive approach to text classification and it performed at least more than the baseline of simply outputting one class. Of course multiple hyperparameters could be tweaked to improve performance. However, it might be more interesting to see how characters vectors would perform, as well finding ways to incorporate other information such as stylometric features especially since this is twitter text.