Personality prediction from tweets: a language independent approach ~ DRAFT

Angelo Basile - s3275655

Abstract

Author profiling is the task of predicting some aspects (e.g., gender, age, personality) of the author of a given text. In this paper we describe a system that predicts the personality of twitter users. We trained a neural network on the TwiSty corpus and we reached a [TODO ADD] x.xx in RMSE score.

1 Introduction

It has been shown that given a text, it is possible to say a lot about its author: its gender, wheter he or she is old or young, if he is depressed, what is his dominant personality trait. This study focuses on this exact last thing: given some text — tweets, in this case — we want to predict the personality of the author.[TODO expand]

2 Related Work

The work of [TODOLIU] shows that neural models can successfully model author personality traits from text. The authors suggest at the end of the paper that "the lack of feature engineering should support language independence". For this work we attempted to replicate the exact same model that [TODO fix][liu et al] built: we wanted to test wether it was possible to use it across languages.

For the baseline: we considered the submissions to the PAN forum [TODO ADD references and footnote explaining what PAN is] in the last years and we saw that many participants proposed an SVM-based model with a sparse feature representation. Such systems achieved good performances: we decided to use them as a baseline.

3 Model

Our system consists of a recurrent and compositional neural-network. A first layer builds embeddings at the character level; the output is then used to build word embeddings; finally, a feed forward network uses both these representations to predict each personality trait individually.

4 Experiments

4.1 Data

For this task we trained and tested our model on the TwiSty corpus [TODO add reference]. Additionally, we worked on the PAN 2015 dataset in order to compare our results with those reported by [TODOLIU]: Table 1 shows a sample of the data.

4.2 Pre-processing

For the baseline system we used the default preprocessor included in the scikit-learn tf-idf vectorizer, which lowercases all the words and .tokenizes the text.

4.3 Evaluation

5 Results

5.1 Baseline

6 Conclusions and Future Work

TODO

All the code used to obtain the results presented in this paper is available at `https://github.com/anbasile/perspred`.

Table 1: Sample instances from the PAN 2015 dataset

author	text	ext	sta	agr	con	opn
e5b59ccc	@username @username ay friend, q te fumasteSSS…	0.0	0.2	0.2	0.3	0.2
ed970294	“@username: @username "you can’t have your cak…	0.1	0.2	0.2	0.0	0.1
4b05f4e0	I should probably go to bed considering I have…	0.5	0.0	0.3	0.3	0.4
de7f0515	@username the sameee@username Great!!@u…	0.2	-0.1	0.2	0.0	0.1
a71c93ed	On my very last Nerve!am nothing and I hav…	0.2	0.0	0.0	0.3	0.4

Table 2: Results (negative MSE and standard deviation, CV-10) for the baseline system on the PAN 2015 dataset using the SVM classifier and unigrams with tf-idf normalization.

	agr	con	ext	opn	sta
en	-0.02 (0.02)	-0.02 (0.02)	-0.02 (0.02)	-0.02 (0.01)	-0.04 (0.04)
es	-0.02 (0.02)	-0.02 (0.03)	-0.02 (0.03)	-0.02 (0.03)	-0.03 (0.03)
it	-0.02 (0.04)	-0.01 (0.01)	-0.02 (0.05)	-0.02 (0.03)	-0.03 (0.04)
nl	-0.03 (0.04)	-0.01 (0.03)	-0.02 (0.03)	-0.01 (0.02)	-0.03 (0.06)