# Author Profiling on Twitter

T.J. Kreutz
t.j.kreutz@student.rug.nl

## Introduction

Author profiling is the task of automatically determining author characteristics from textual data. PAN organizes this as a shared task each year in which participants are challenged to predict demographics (gender and age) and personality traits of Twitter users based on their tweets. They had to do this for four languages: Dutch, English, Italian and Spanish (3). The fourth edition of this task will be organized in 2016, and the final project for the course Learning from Data mirrors this challenge.

One difference from the PAN task is that our project aimed to only predict the mentioned demographics of users. However, inspiration could still be taken from contributions to the PAN task made in the previous years. Further, it was central to the problem to apply the techniques and considerations in Machine Learning (ML) that were taught in the course. This report is an overview of my attempt at the task, with a specific focus on the choices I made and the reasons behind it, and a reflection on the results and lessons learnt.

## Data

The data consists of xml files for each author. 24 for Dutch, 106 for English, 27 for Italian and 70 for Spanish, totalling 227. All of these files had the characteristics available in a separate truth-file. The xml files contained a total of 21,249 tweets; about 94 tweets on average per author.

## Approach

In my opinion I took away a pretty decent understanding of basic ML techniques and their underlying considerations from the course. A global notion I took away is that theoretically planning a system is more important than the implementation of a system, because it can save a lot of effort (trail and error) and time later on.

Taking this notion to heart, I selected two systems which ranked in the top five contestants in 2015 and read the related articles. One system had the overall best performance (Álvarez-Carmona et al.), and the other used the widest range of features (Kiprov et al.). They shared their best feature (function words) and ML algorithm (Support Vector Machines). I took both as basic pillars of my system, and started implementing it in Scikit-learn, an ML-module for Python.

A consideration that was less informed by previous work, and more by intuition was to start by classifying individual tweets rather than authors. Since the data had not a lot of authors (especially for Dutch and Italian), classifying per tweet would give more consistent results in testing the system.

## Features

Function words were understood to be an important feature for determining the gender and age of an author(Álvarez-Carmona et al.). However, online sources (4) have divergent definitions of the category. For this task, function words denote the words specific to a certain category of authors. In other words, they are the words that are more frequently used in one category relative to the rest of the categories. Examples include sport-words for male users versus female users, or popular texism for the youngest category of uses versus the older categories of users.

I calculated the relative frequency of a term as below, where t denotes the term and c the category, then used a ranked list of the top thousand relatively most frequent words for each category.

$$rf_{tc} = \frac{tf_{tc}}{tf_{t!c}}$$

This resulted in decent accuracies per tweet (table 1). Other features were tested from (Kiprov et al.) and ones that made a decent improvement (larger than 0.5 percent) were included in the permanent features. These were character tri- and 4grams, occurrences of hashtags, mentions, retweets, links and punctuation (periods, exclamation points and question marks).

## Machine Learning Algorithm

The tweets were classified by a linear support vector machine for both gender and age. Changing the parameters from the default (C=1.0) did not improve performance, nor

| | Accuracy | Recall | Precision | F-score |
|---|---|---|---|---|
| Baseline | 0.526 | 1.000 | 0.526 | 0.689 |
| Dutch | 0.596 | 0.596 | 0.575 | 0.540 |
| English | 0.573 | 0.573 | 0.598 | 0.548 |
| Italian | 0.627 | 0.627 | 0.628 | 0.625 |
| Spanish | 0.568 | 0.568 | 0.579 | 0.526 |
| **Average** | 0.591 | 0.591 | 0.595 | 0.593 |

Table 1: Results with function words feature for gender.

| | Accuracy | Recall | Precision | F-score |
|---|---|---|---|---|
| Baseline | 0.331 | 1.000 | 0.331 | 0.497 |
| English | 0.520 | 0.520 | 0.526 | 0.456 |
| Spanish | 0.449 | 0.449 | 0.457 | 0.395 |
| **Average** | 0.485 | 0.485 | 0.492 | 0.487 |

Table 2: Results with function words feature for age.

did changing the kernel from linear to one of the others. Using these settings and the added features, I 5-fold cross validated the performance which yielded the results in table 3 and table 4. For the age classification, using character n-grams had a positive impact, so I let it weigh three times as heavy as the other features.

| | Accuracy | Recall | Precision | F-score |
|---|---|---|---|---|
| Baseline | 0.526 | 1.000 | 0.526 | 0.689 |
| Dutch | 0.596 | 0.596 | 0.575 | 0.540 |
| English | 0.573 | 0.573 | 0.598 | 0.548 |
| Italian | 0.627 | 0.627 | 0.628 | 0.625 |
| Spanish | 0.568 | 0.568 | 0.579 | 0.526 |
| **Average** | 0.591 | 0.591 | 0.595 | 0.593 |

Table 3: Results with all features for gender.

| | Accuracy | Recall | Precision | F-score |
|---|---|---|---|---|
| Baseline | 0.331 | 1.000 | 0.331 | 0.497 |
| English | 0.520 | 0.520 | 0.526 | 0.456 |
| Spanish | 0.449 | 0.449 | 0.457 | 0.395 |
| **Average** | 0.485 | 0.485 | 0.492 | 0.487 |

Table 4: Results with all features for age.

## Classifying authors

The biggest problems arose when trying to generalize tweets to the author level. This was a problem both for the framework, which I will discuss here, and for the biggest flaw in the system, which will be discussed in the discussion section.

In the framework it was hard to test the system on the author level because the splits yielded very different results each time. I hypothesized that generalizing labels from predicted tweets would yield higher results on the author level compared to the classification of tweets, but this was not reflected in the inconsistent results.

To generalize to the author level, the tweets were again grouped per author and the author received the label most common in his tweets. I implemented two different methods. One method predicted not a label, but a probability for a tweet to belong to a certain category. The probabilities were then added per category and compared, but this yielded visibly lower results.

Another method took the list of probabilities for each category and used these as features for another linear support vector machines (SVM) classifier. The previous SVMs thus worked on the tweet level, while this layer attempted to classify authors. But this method also was outperformed by the simpler method that took the most common label.

## Results

Using this, the system was then evaluated on test data and yielded the results in table 5 and table 6.

| | Accuracy | Recall | Precision | F-score |
|---|---|---|---|---|
| Baseline | 0.526 | 1.000 | 0.526 | 0.689 |
| Dutch | 0.900 | 1.000 | 0.750 | 0.857 |
| English | 0.652 | 0.708 | 0.654 | 0.680 |
| Italian | 0.545 | 0.167 | 1.000 | 0.286 |
| Spanish | 0.867 | 0.765 | 1.000 | 0.867 |
| **Average** | 0.732 | 0.844 | 0.667 | 0.745 |

Table 5: Results on test data for gender.

| | Accuracy | Recall | Precision | F-score |
|---|---|---|---|---|
| Baseline | 0.331 | 1.000 | 0.331 | 0.497 |
| English | 0.826 | 0.826 | 0.730 | 0.769 |
| Spanish | 0.767 | 0.767 | 0.835 | 0.737 |
| **Average** | 0.797 | 0.797 | 0.783 | 0.790 |

Table 6: Results on test data for age.

## Conclusion and discussion

I was content with these results, although the Italian gender classification was lower than expected. My most important feature really came into play in the age classification task since different age groups tend to use different words. Overall I performed the best out of the other system in the age classification for English and Spanish.

If this system were to be adapted for the PAN 2016 shared task I would implement a more robust way to determine function words, and train these lists on external sources rather than the training data only. I would also not start with classifying tweets since it does not give a realistic representation of the system's performance, nor was there any reason to start out like this. In fact, it could be that some of

the performance was weighed down by the way labels were determined on the author level. Further, there are some features that can be better determined on the author level, such as vocabulary size and verbosity.

## References


Álvarez-Carmona, M. A., López-Monroy, A. P., Montes-y Gómez, M., Villaseñor-Pineda, L., and Jair-Escalante, H. Inaoes participation at pan15: Author profiling task.

Kiprov, Y., Hardalov, M., Nakov, P., and Koychev, I. Su@ pan2015: Experiments in author profiling.

PAN (2016). Tasks. [Online; accessed 24-January-2016].

Wikipedia (2016). Function word — Wikipedia, the free encyclopedia. [Online; accessed 24-January-2016].