Celebrity Profiling using TF-IDF, Logistic Regression, and SVM
Notebook for PAN at CLEF 2019

Victor Radivchev, Alex Nikolov, Alexandrina Lambova

FMI at University of Sofia
victor.radivcev@gmail.com, alexnickolow@gmail.com, sanilambova@gmail.com

Abstract. This paper aims to describe a TF-IDF approach based on word bigrams and n-grams at a character level used in the Celebrity Profiling competition at PAN CLEF 2019.

1 Introduction

This paper will take a look at different approaches towards the 2019 Celebrity Profiling competition at PAN CLEF[1]. The task is defined in the following way: Given a set of tweets for a user, one has to classify the user in four categories: gender, fame, occupation and age. There are three classes for gender and fame, eight for occupation and the age range is from 1940 to 2012.

2 Related Work

Previous related work on the subject of gender classification includes an SVM classifier with different types of word and character n-grams as features, along with dimensionality reduction using Latent Semantic Analysis (LSA) [4]. On the subject of age classification, numerous techniques have been attempted such as concatenating text mining features, sociolinguistic and content-based features and classifying them using the Random Forest algorithm [5].

3 Preprocessing

The applied preprocessing involved the following techniques:

- Removing all retweets of the user
- Removing all symbols except for letters, numbers, @, and #
- Replacing all hyperlinks with a special <*url*> token
- Replacing all user tagging with a special <*user*> token
- Substituting multiple consecutive spaces with a single one
- Adding a special <*sep*> token at the end of each tweet in order to distinguish between each tweet's beginning and end.

Two additional ways of preprocessing were tested but proved to have poorer results on a baseline model, which used logistic regression, taking TF-IDF vectors with 10,000 features, the first one tried preserving the retweets of the users, the second one substituted happy and sad emojis, such as ':-)', ':)', ':D', ':S', ':(', ':<' amongst others, with special <*happy*> and <*sad*> tokens.

We chose to split the provided data into a training and validation set in a ratio of 80:20.

	Gender F1 score on training set	Gender F1 score on validation set	Fame F1 score on training set	Fame F1 score on validation set	Occupation F1 score on training set	Occupation F1 score on validation set
Discarding retweets and emojis	0.5483	0.5514	0.3818	0.3687	0.3597	0.3529
Preserving retweets	0.5237	0.5331	0.3675	0.3600	0.3564	0.3488
Replacing emojis	0.5233	0.5326	0.3670	0.3600	0.3569	0.3491

Table 1 - Different preprocessing results on a baseline tf-idf model with logistic regression

4 Submitted model

The submitted approach used the described above preprocessing on a subset of all tweets per user. The subset for each user was chosen randomly and had a maximum cardinality of 500. The users' tweets were vectorized with a TF-IDF vectorizer, taking into account the top 10,000 features from word bigrams. A combination of logistic regression and SVM were used as models for each different task. Different class weights were used for each task based on the number of labeled examples from each class – the more examples a particular class has, the lower the class weight assigned to that class will be. The computed class weights were directly supplied to each tested algorithm as a parameter (*class_weights*). For the task of identifying the users' gender, the following class weights were used:

Male – 0.46586179, Female – 1.17494337, Nonbinary – 428.12698413

The class weights used for the fame task are as follows:

Rising – 7.52987159, Star – 0.44780927, Superstar – 1.57703327

For the occupation task the following class weights were used:

Sports – 0.31435894, Performer – 0.42521125, Creator – 0.7731025, Politics – 1.49844444, Manager – 5.57272727, Science – 5.10060514, Professional – 8.08513189, Religious – 140.4791667

The results from tests on different models using different hyperparameters are described below.

	Gender training set F1	Gender validation set F1
Logreg, multiclass=ovr	0.94160	0.61237
Logreg,multiclass=multi, solver=newton_cg	**0.92809**	**0.71714**
SVM, default	0.98185	0.62486
SVM, multiclass=hinge	0.97299	0.62283
SVM, default, c=1.25	0.99161	0.62583
SVM, default, c=1.5	0.99280	0.62503
SVM, default, c=0.75	0.97917	0.62575
SVM, default, c=1	0.94627	0.62031

Table 2 - Results from experiments on gender using different models and hyperparameters

We can see that logistic regression with hyperparameters *multiclass=multi* and *solver=newton_cg* achieved the best results on the test set – 0.71714

	Fame training set F1	Gender validation set F1
Logreg, multiclass=ovr	0.75532	0.60251
Logreg, solver=lbfgs, multiclass=multinomial	0.71992	0.58420
Logreg, solver=sag, multiclass=multinomial	0.66106	0.55128
Logreg, solver=newton_cg, multiclass=multinomial	0.66454	0.56460
Logreg, solver=lbfgs, multiclass=multinomial	0.66423	0.55954
Logreg, multiclass=ovr, c=1.5	0.79114	0.60687
Logreg, multiclass=ovr, c=1.25	0.77595	0.60297
Logreg, multiclass=ovr, c=1.75	0.80434	0.60462
SVM, default	0.91779	0.59683
SVM, multiclass=crammer_singer	0.81824	0.58902
SVM, c=0.75	0.90750	0.60500
SVM, c=0.5	0.88860	0.61160
SVM, c=0.1	**0.76788**	**0.61987**

Table 3- Results from experiments on fame using different models and hyperparameters

	Occupation training set F1	Occupation validation set F1
Logreg, default	0.73271	0.49652
Logreg, solver=newton_cg, multiclass=multinomial	0.76639	0.49946
Logreg, solver=sag, multiclass=multinomial	0.52213	0.40665

Logreg, solver=lbfgs, multiclass=multinomial	0.76189	0.50081
Logreg, solver=lbfgs, multiclass=ovr	0.72876	0.48533
Logreg, solver=newton_cg, multiclass=multinomial, c=2	0.82161	0.50036
SVM, default	0.93019	0.50172
SVM, multiclass=crammer_singer	0.87430	0.48280
SVM, loss=hinge	0.80346	0.48499
SVM, c=0.75	0.91441	0.50713
SVM, c=0.5	**0.88658**	**0.50865**

Table 4 - Results from experiments on occupation using different models and hyperparameters

For the task of identifying a person's age, the range of years were divided into eight classes (subranges) and each time the model predicted the mean year for an interval. The intervals were constructed in such a manner, that assuming the true age of a user lied within an interval, predicting the interval's mean would result in a correct guess due to the amount of error the contestants are allowed when predicting a user's birthyear.

Interval range	Mean year
1940-1955	1947
1965-1969	1963
1970-1980	1975
1981-1989	1985
1990-1997	1993
1998-2004	2001
2005-2009	2007
2010-2012	2011

Table 5 – Different interval ranges used with their mean year

	Age training set F1	Age validation set F1
SVM	0.90365	0.55928
Linear regression	0.52186	0.34121
Logistic regression	**0.80465**	**0.62479**

Table 6 - Results from experiments on age using different models

The following results on the test set were achieved using the models highlighted in bold:

Test set 1: C rank – 0.59259, Gender F1 – 0.72599, Fame F1 – 0.55084, Occupation F1 – 0.51539, Age F1 – 0.61845

Test set 2: C rank – 0.55893, Gender F1 – 0.60896, Fame F1 – 0.54795, Occupation F1 – 0.46128, Age F1 – 0.65727

5 Alternative Features and Methods: An Analysis of Negative Results

In this section we will discuss other attempted techniques, which did not prove to be as successful as the one mentioned above.

5.1 Using character n-grams

Our team also attempted to represent each user as a TF-IDF vector of the top 10,000 features using n-grams on a character level, the chosen range was 3- and 4-grams. Once again, we sampled 500 tweets for each user before attempting to vectorize the user. The achieved results were poorer than those achieved from the TF-DF based on word bigrams.

5.2 More complex models

This section examines the multi-layered neural network approaches we attempted.

5.2.1 Regular feed forward neural networks

We tried replacing the Linear SVM and Logistic regression with other models, having more capacity. One such is the multilayer perceptron. In order to protect against overfitting (which could be a serious problem when having 10 000/20 000 features and 27 000 examples) we used a relatively shallow model with 2 hidden layers, PRELU activation and dropout of 0.5, trained with Adam optimizer. We also used balanced class weights. We experimented with only TF-IDF features and a concatenation of TF-IDF and character n-grams. However, the inability to mid a global optimum seems to have had a more detrimental effect than a positive one from the deeper model.

	Fame training set F1	Fame validation set F1
Tf-idf	0.82197	0.58564
Tf-idf + character n-grams	**0.83256**	**0.59131**

Table 7 - Results from experiments on Fame using different MLP models

5.2.2 GloVe embeddings and 1D CNN

With the recent advances in natural language processing embeddings and deep learning become increasingly more attractive. However, due to the big size of information per person, their usage is not straightforward. In order to handle that, we used the following model:

- First we removed all retweets
- In order to equalize the number of tweets per person we selected only the first 1000 for each
- We used GloVes's default preprocessing and TweetTokenizer
- We averaged Glove's embeddings for each tweet
- For each person we constructed a 200 * 1000 matrix (padded with 0 if the person has less than 1000 tweets) which we used as the input of our model
- The model was DPCNN [6].
- We also tried training each task individually and all of them together with a multihead classifier.

Unfortunately, this did not produce satisfactory results. There was no real difference between individual models and the multihead one. Accuracy-wise the models were slightly behind the logistic regression, but their F1 score was underwhelming. Using class weights severely impacted the training process, leading to very low scores.

	Fame training set F1	Fame validation set F1
GloVe + DPCNN	0.68427	0.39955

Table 8 - Results from experiments on Fame using GloVe and DPCNN

6 References

[1] Daelemans, W., Kestemont, M., Manjavancas, E., Potthast, M., Rangel, F., Rosso, P., Specht, G., Stamatatos, E., Stein, B., Tschuggnall, M., Wiegmann, M., Zangerle, E.: Overview of PAN 2019: Author Profiling, Celebrity Profiling, Cross-domain Authorship Attribution and Style Change Detection. In: Crestani, F., Braschler, M., Savoy, J., Rauber, A., Müller, H., Losada, D., Heinatz, G., Cappellato, L., Ferro, N. (eds.) Proceedings of the Tenth International Conference of the CLEF Association (CLEF 2019). Springer (Sep 2019)

[2] Potthast, M., Gollub, T., Wiegmann, M., Stein, B.: TIRA Integrated Research Architecture. In: Ferro, N., Peters, C. (eds.) Information Retrieval Evaluation in a Changing World - Lessons Learned from 20 Years of CLEF. Springer (2019)

[3] Wiegmann, M., Stein, B., Potthast, M.: Overview of the Celebrity Profiling Task at PAN 2019. In: Cappellato, L., Ferro, N., Losada, D., Müller, H. (eds.) CLEF 2019 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2019)

[4] Daneshvar, S., Inkpen, D.: Gender identification in Twitter using N-grams and LSA, Notebook for PAN at CLEF 2018.

[5] Simaki, V., Mporas, I., Megalooikonomou, V., Age Identification of Twitter Users: Classification Methods and Sociolinguistic Analysis. (2016)

[6] Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In ACL.