Using Machine Learning Algorithms for Author Profiling In Social Media

Notebook for PAN at CLEF 2016

Daniel Dichiu[1], Irina Rancea[1]

[1] Bitdefender Romania
ddichiu@bitdefender.com, irancea@bitdefender.com

Abstract. In this paper we present our approach of solving the PAN 2016 Author Profiling Task. It involves classifying users' gender and age using social media posts. We used SVM classifiers and neural networks on TF-IDF and verbosity features. Results showed that SVM classifiers are better for English datasets and neural networks perform better for Dutch and Spanish datasets.

1 Introduction

Due to the huge amount of text information on the Internet, both the academia and industry have developed an interest in author profiling. It consists of discovering as much insight as possible about an unknown author by analyzing his data posted on-line. The PAN Author Profiling task is focusing this year on gender and age classification. The training documents consist of tweets, while the evaluation is performed on blogs or other social media documents, except tweets. Similar contributions on classifying age and gender on short texts obtained from tweets has been developed in [1], using TIRA platform ([2], [3]). Training documents are provided for three languages: English, Spanish and Dutch.

2 Approach Description

Our approach for the classification tasks implies using the scikit-learn LinearSVC [4] and a neural network based on nolearn Lasagne module [5] as distinct predictors. For the feature extraction part we used vectorizers from scikit-learn module for python.

For features we tried a *tf-idf* matrix at both character and word level with various n-gram ranges and fine tuning for the rest of parameters depending on the language and subtask.[6] We computed the *tf-idf* matrix using TfidfVectorizer from scikit-learn Python module. Before vectorizing data we concatenate all tweets for each user.

The authors in [7] obtained good results in PAN 2015 Author Profiling competition with SVM classifiers on *tf-idf* matrices at character level. However, the training and testing datasets were based on the same type of social media, while PAN 2016 Author

Profiling competition's training and testing datasets were based on different types of social media (e.g. Twitter for training dataset and blogs for testing dataset). Taking this into consideration, we thought a tf-idf matrix at word level would better generalize the classification model and so we trained models based on both types of tf-idf matrices.

We combined, in a scikit-learn FeatureUnion structure, the *tf-idf* scores with a *verbosity* rate computed as a type/token ratio, as was done in [7].

There were 3 types of classifiers:

1. Support Vector Machine (SVM_1 hereinafter), based on verbosity and features extracted with *tf-idf* at character level;
2. Support Vector Machine (SVM_2 hereinafter), based on verbosity and features extracted with *tf-idf* at word level;
3. Neural Network (NN hereinafter), based on features extracted with *tf-idf* at word level.

To find good parameters that do not overfit, we used scikit-learn's StratifiedKFold [4] for the cross-validation phase of the SVMs.

For SVM_1 the LinearSVC parameters common for all running tests were: *dual* = False, *loss* = squared_hinge, *penalty* = l2. Table 1, on page 2, and table 2, on page 3, summarizes the parameters we found as optimal for SVM_1. Parameters which are missing in the table have the default value.

For SVM_2 the LinearSVC algorithm was used with default parameters. Table 3 on page 3 summarizes the parameters we found as optimal for this classifier. Parameters which are missing in the table have the default value.

Table 1: English Gender Classification SVM_1 Parameters

Algorithm	**Parameter Name**	**Parameter Value**
TfidfVectorizer	sublinear_tf	True
	max_df	0.75077
	ngram_range	1,1
	min_df	0.17785
LinearSVC	C	3.0
	fit_intercept	True

NN is a neural network classifier, with 2 hidden layers, each hidden layer having 50 nodes. The input features were based on a *tf-idf* matrix at word level, reduced to 50-dimensional feature space using scikit-learn's TruncatedSVD. Table 4 on page 3 summarizes the parameters used with this neural network.

To reduce the impact of overfitting, we used a dropout layer [8] (with a dropout probability of 50%) between the hidden layers. We also made use of early stopping [9], and the maximum number of epochs for each classifier is reported in tables 5, 6, and 7, on page 4.

Table 2: English Age Classification SVM_1 Parameters

Algorithm	Parameter Name	Parameter Value
TfidfVectorizer	sublinear_tf	False
	max_df	0.976896
	ngram_range	1,1
	min_df	0.142695
LinearSVC	C	3.0
	fit_intercept	False

Table 3: English Gender and Age Classification SVM_2 Parameters

Subtask	Algorithm	Parameter Name	Parameter Value
Gender	TfidfVectorizer	max_df	0.7
		ngram_range	1,1
	LinearSVC	all	defaults
Age	TfidfVectorizer	max_df	0.7
		ngram_range	1,1
	LinearSVC	all	defaults

Table 4: Neural Network Parameters

Parameter Name	Parameter Value
layers	dense, dense
layer_1_num_units	50
layer_1_dropout	0.5
layer_2_num_units	50
output_nonlinearity	softmax
update	nesterov_momentum
update_learning_rate	0.001
update_momentum	0.9
eval_size	0.2

Table 5: Spanish Gender Neural Network (NN) Approach Parameters

Algorithm	Parameter Name	Parameter Value
TfidfVectorizer	max_df	0.7
	analyzer	Word
	ngram_range	1,1
	min_df	0.3
LinearSVC	all	defaults

Table 6: Spanish Age Neural Network (NN) Approach Parameters

Algorithm	Parameter Name	Parameter Value
TfidfVectorizer	analyzer	word
	ngram_range	1,1
TruncatedSVD	n_components	50
NN	max_epochs	4200

Table 7: Dutch Gender Neural Network (NN) Approach Parameters

Algorithm	Parameter Name	Parameter Value
TfidfVectorizer	analyzer	word
	ngram_range	1,2
TruncatedSVD	n_components	50
NN	max_epochs	1600

3 Results

Table 8 on page 5 shows the results of the classifiers on test dataset 1.

For English, the best results were obtained using a *tf-idf* at character level combined with the verbosity feature. These were then classified using an SVM.

For Spanish, the best results were obtained using a *tf-idf* at word level combined with the verbosity feature for the gender task, while for the age task just the *tf-idf* at word level was used. An SVM was used for the gender task, and a NN trained for 4200 epochs was used for the age task.

For Dutch, the best results were obtained using a *tf-idf* at word level, reduced to a 50-dimensional space and then classified with a neural network which was trained for 1600 epochs.

Table 9 on page 5 shows the results of the classifiers on test dataset 2.

For English, the best results were obtained using a *tf-idf* at word level combined with the verbosity feature. These were then classified using an SVM.

For Spanish, the best results were obtained using a *tf-idf* at word level combined with the verbosity feature for the gender task, while for the age task just the *tf-idf* at word level was used. An SVM was used for the gender task, and a NN trained for 4200 epochs was used for the age task.

Table 8: Testing Dataset 1 Results

Classifier Type	**Language**	**Gender Accuracy**	**Age Accuracy**	**Both**
SVM_1	English	0.5345	0.2989	0.1753
SVM_2+NN	Spanish	0.5469	0.2813	0.1719
NN	Dutch	0.54	N/A	N/A

Table 9: Testing Dataset 2 Results

Classifier Type	**Language**	**Gender Accuracy**	**Age Accuracy**	**Both**
SVM_2	English	0.6154	0.4103	0.2692
SVM_2+NN	Spanish	0.6429	0.4643	0.3214
NN	Dutch	0.526	N/A	N/A

For Dutch, the best results were obtained using a *tf-idf* at word level, reduced to a 50-dimensional space and then classified with a neural network which was trained for 1600 epochs.

4 Conclusions

All the classifiers suffered from overfitting. During the cross-validation phase of our training, we registered accuracies around 0.8, nowhere near the accuracy score on the test datasets. However, the types of features and models we used on English and Spanish generalize better from training dataset to testing dataset 2, while accuracies on the testing dataset 1 are, on average, about 10 percentage points lower. This could

mean that at the feature level of our choosing, training dataset and testing dataset 2 are more similar than training dataset and testing dataset 1. Based on our results, we can say that word level features are better for generalization when used with a linear SVM. Also, neural networks, when trained carefully, can outperform SVMs using the same feature set.

References

1. Rangel, F., Rosso, P., Verhoeven, B., Daelemans, W., Potthast, M., Stein, B.: Evaluations Concerning Cross-genre Author Profiling. In: Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2016)Shlomo Argamon and Anat Rachet Shimoni. Automatically categorizing written texts by author gender. Literary and Linguistic Computing, 17: 401-412, 2003
2. Potthast, M., Gollub, T., Rangel, F., Rosso, P., Stamatatos, E., Stein, B.: Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In: Kanoulas, E., Lupu, M., Clough, P., Sanderson, M., Hall, M., Hanbury, A., Toms, E. (eds.) Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14). pp. 268–299. Springer, Berlin Heidelberg New York (Sep 2014)
3. Gollub, T., Stein, B., Burrows, S., Hoppe, D.: TIRA: Configuring, Executing, and Disseminating Information Retrieval Experiments. In: Tjoa, A., Liddle, S., Schewe, K.D., Zhou, X. (eds.) 9th International Workshop on Text-based Information Retrieval (TIR 12) at DEXA. pp. 151–155. IEEE, Los Alamitos, California (Sep 2012)
4. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011
5. Dieleman, Sander, et al. "Lasagne: First Release." Zenodo: Geneva, Switzerland (2015)
6. John Houvardas, and Efstathios Stamatatos. N-gram feature selection for authorship identification. In Artificial Intelligence: Methodology, Systems and Applications, 77-86, Spinger, 2006
7. Octavia-Maria Sulea, and Daniel Dichiu. Automatic Profiling of Twitter users based on their tweets – Notebook for PAN at CLEF 2015. In Linda Cappellato, Nicola Ferro, Gareth Jones, and Eric San Juan, editors, CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers, 8-11 September 2015. CEUR-WS.org. ISSN 1613-0073
8. Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15, no. 1 (2014): 1929-1958.
9. Bengio, Yoshua. Practical recommendations for gradient-based training of deep architectures. In Neural Networks: Tricks of the Trade, pp. 437-478. Springer Berlin Heidelberg, 2012.