Twitter Feeds Profiling With TF-IDF

Notebook for PAN at CLEF 2019

Juraj Petrik, Daniela Chuda

Slovak University of Technology in Bratislava, Slovakia
{juraj.petrik, daniela.chuda}@stuba.sk

Abstract. Paper describes our approach in celebrity profiling task at CLEF 2019 conference. Our method is based on TF-IDF feature extraction method combined with random forest classifier. We were mainly focused on preprocessing phase, where we implemented multiple methods for a text normalization such as emoji transformation, lemmatization, URL replacing. The biggest problem was class imbalance, which we tried to resolve by using synthetic oversampling techniques.

1 Introduction

This notebook describes our approach in celebrity profiling task [1] [3]. We were trying to adjust our method used for source code authorship attribution [4]. However, we were not able to achieve good consistent results. Thus, we took our baseline method based on TF-IDF and random forest and tune it for purposes of this challenge and our results were consistently better with this approach. Our solutions were tested and evaluated by TIRA evaluation platform [2].

Author profiling is subtask in stylometry which deals with text analysis to extract various characteristics of the author. For instance, nationality, age, political or religion believes, gender or occupation. We can use such traits to determine who is "on the other end" of chat communication. To know if we are talking with real person, with the person which is acting as or to adapt way of communication to this specific person.

2 Task Description

Task is to profile given celebrity Twitter feed. Our task is to predict four traits - author's occupation, birthyear, fame and gender.

An average F1 macro score amongst all traits was chosen by organizers as a final evaluation score. Classes in training dataset are heavily imbalanced, especially nonbinary gender class. Since, birthyear prediction is extremely difficult, score of birthyear trait is calculated leniently.

3 Related Work

In recent years there has been reborn of stylometry and authorship detection, multiple papers are dealing with authorship attribution or stylometry in different contexts. We can distinguish between two common types of stylometry: linguistic stylometry and source code stylometry.

The survey [6] describes five subtasks in a linguistic (textual) stylometry – authorship attribution, authorship verification, authorship profiling, stylochronometry and adversarial stylometry. Combination of lexical, syntactic, semantic, structural, domain-specific features and topic models has best results in authorship attribution in combination with machine learning techniques, which outperforms probabilistic methods and string distance methods.

Additionally, 14 open source algorithms for authorship attribution were benchmarked on a corpus with 1000 authors [6]. It turned out that lower-level representations (mainly character-level features) are more important than high-level features (word-level features).

Representing texts as complex networks based on a word adjacency model look promising [7]. Working with graphs and comparing similarity of graphs of multiple documents is a computational complex problem, so authors extracted features as accessibility, degrees, assortativity, betweenness of nodes from these graphs. Hybrid approaches outperformed traditional methods.

Other authors decided to deal with a problem of multiple authors of one document – multi label classification [8]. Dataset is composed from early revisions of Wikipedia pages. Results were quite usable when there were 2 authors of one document but with 3 and 4 authors of one document, the accuracy was pretty low and not ready for a real-world usage. Although thinking about this, we need to take into account that there is a huge difficulty jump, because of possible authors combinations.

4 Our Method

As stated above, provided training dataset consists of unprocessed Twitter feeds [5]. A feed consists of maximum 3000 single tweets from the celebrity. One tweet is usually ~100 characters long or in terms of words, it is 30 words (the maximum length is 180 characters, but it depends on language). An average number of tweets per celebrity in training dataset is 2181. Given these statistics, we got relatively huge number of texts per sample (celebrity).

One of our first approaches was convolutional recurrent neural network [4] modified for purposes of this task with hierarchical tweets processing. Unfortunately, this approach was surprisingly inferior to our baseline approach leveraging TF-IDF as features extraction method. Therefore, we used our baseline as the base of the method for this task and tuned it for better results.

We used 10-fold stratified cross validation as testing strategy for our solution. Stratification is especially important in this task, because classes are heavily imbalanced.

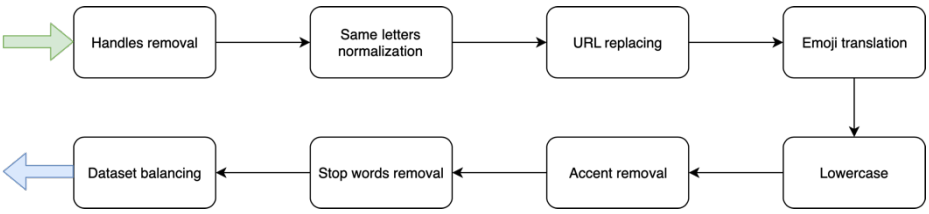

Figure 1. Preprocessing pipeline

Preprocessing

Vastly majority of tweets contain handles. A handle is unique personal id in Twitter social network. People are commonly referencing other profiles by this id, however for our purposes this information is just some kind of highly dimensional feature.

Unfortunately, we were not able to use handles to improve performance of our method. Therefore, we decided to remove all handles from dataset.

Another common trait in tweets is a multiple usage of the same letters in a row to emphasize something. Our approach is based on a word (sub word) frequency. We are reducing dimensionality in the next steps, so we need to deal with this kind of synonyms. Fortunately, solution is simple – squeeze multiple occurrences of the same letters (more than 2).

Next thing how to reduce dimensionality of texts was replacing URLs with sequence "URL_TOKEN". Due to Twitter is using own URL shortener service, all URLs in dataset (in tweets) are starting with string "*https://t.co/*". On the one hand URL info could be good feature, we could be able to cluster similar webpages and then we can replace original URL with cluster representation. But on the other hand, because of mentioned shortener, we need to resolve all target URLs, which is time and resources consuming. As because of the deadline, we decided just to replace URLS with token, as stated above.

Another step is a Unicode emoji translation to their respective word description[1]. This helps us to better detect emotions (professionals and managers don't use so many positive emojis for example).

In the end we have done standard text preprocessing steps such as lowercase conversion, accent and English stop words removal.

As we mentioned above, task dataset is heavily imbalanced. Our first approach was weighting classes according to their size. This approach, however, does not improve our testing results.

Next, we tried to balance dataset using synthetic oversampling and undersampling techniques. We used Synth CSOB etic Minority Oversampling Technique (SMOTE) with combination of Tomek links to remove overlapping samples (undersampling). SMOTE combination with Tomek links shows better results than just random oversampling, time performance was quite good too.

[1] https://unicode.org/emoji/charts-12.0/full-emoji-list.html

Classification and Feature Extraction

A chosen classifier (random forest) is not able to work with text data directly and therefore we need to get features from text. We used term a frequency-inverse document frequency (TF-IDF), as it is commonly and successfully used in a high number of natural language tasks, primarily in a text classification and summarization.

TF-IDF is typically using words as input terms, when dealing with n-grams we can talk about unigrams. Additionally, we used bigrams and trigrams to capture general contextual terms (words) relations, which are beneficial for this task (higher accuracy). We were also experimenting with higher-level n-gram features (4, 5, 6 and 7 grams), unfortunately achieved results were not better. Also processing time and memory requirements were a lot higher because more features were extracted.

It is evident that there are many extracted features (tens of thousands to hundreds of thousands). We reduced a number of the features in range from 3000 to 30000 by 1000 steps, 5000 features show best trade-off in means of accuracy and processing time. A higher number of features was paradoxically crippling accuracy, this is caused by the fact that dimensionality reduction is naturally acting as a generalization helper.

A random forest was chosen as a final classification model. We used grid search in combination with random forest, decision tree and extremely randomized trees. Random forest with 200 decision trees had best f1 score. Because of the deadline (this solution was chosen few weeks before deadline), we used only a fraction of all training data for training (1/8).

5 Results

As reported multiple times above, imbalance of classes in dataset was huge problem for our approach. For instance, there were just 32 non-binary samples for gender – because of that testing f1 score was highly unstable (high standard deviation) in multiple runs. Another problem was a poor classification accuracy of some classes, namely creator, manager and professional (Table 1). Closer look on a confusion matrix shows, that the classifier was unable to properly distinguish between samples of these three classes, majority of misclassifications was within this classes. Considering all the aspects, we found out that it is extremely hard for humans a to distinguish whether it is manager's feed or professional's feed.

Table 1. Classification report of occupation

	creator	**manager**	**performer**	**politics**	**professional**	**religious**	**science**	**sports**
f1-score	**0,268**	**0,145**	0,539	0,636	**0,2**	0,666	0,366	0,682
precision	**0,305**	**0,285**	0,462	0,518	**0,297**	0,666	0,356	0,576
recall	**0,239**	**0,098**	0,648	0,825	**0,150**	0,666	0,378	0,837
support	92	102	94	86	93	6	82	86

Table 2. Training and testing results

Dataset	Birthyear (f1)	Fame (f1)	Gender (f1)	Occupation (f1)	Rank (f1)
Training	0.41	0.65	0.67	0.44	**0.543**
Testing	0.360	0.526	0.555	0.385	**0.441**

Table 2 shows our scores where is clearly visible, that problems with larger number of classes get lower f1 score (birthyear especially), because more classes equals lower chance of right guess – from statistical perspective.

6 Conclusions and Future Work

Our approach shows promising results. However, we also take into consideration that proper recurrent neural network could have better results (our first approach). Unfortunately, due to time constrains, we were not able to design and train network with better results than simple TF-IDF combined with random forest. The biggest problem was the class imbalance – we were unable to properly oversample data for training of such neural net.

Task's official results show that we struggled most with age prediction, which makes sense, since we don't use any special approach to leverage lenient age f1 score calculation, we could divide classes by age to bins and train classifier to predict these bins. With greater age, the bins will be broader, than in f1 age calculation. Alternatively, we could use this lenient age f1 score as loss function (in neural network) or target score in random forest classifier.

We could also balance classes using new data from crawling Twitter, but unfortunately, we were unable to reproduce class labels. Our expert guesses were just making training and therefore the final predictions worse.

Acknowledgments

This work was partially supported by Human Information Behavior in the Digital Space, the Slovak Research and Development Agency under the contract No. APVV-15-0508, by the Slovak Research and Development Agency under the contract No. APVV-17-0267 - Automated Recognition of Antisocial Behaviour in Online Communities and by data space based on machine learning, the Scientific Grant Agency of the Slovak Republic, grant No. VG 1/0725/19.

Bibliography

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. Juraj Petrík and Daniela Chudá. 2018. Source code authorship approaches natural language processing. In Proceedings of the 19th International Conference on Computer Systems and Technologies (CompSysTech'18), Boris Rachev and Angel Smrikarov (Eds.). ACM, New York, NY, USA, 58-61.
5. Matti Wiegmann, Benno Stein, and Martin Potthast. Celebrity Profiling. In Proceedings of ACL 2019 (to appear), 2019.
6. Neal, Tempestt, Sundararajan, Kalaivani, Fatima, Aneez, Yan, Yiming, Xiang, Yingfei And Woodard, Damon. Surveying Stylometry Techniques and Applications. ACM Comput. Surv. Article 2017. Vol. 50, no. 86.
7. Amancio, Diego Raphael. A complex network approach to stylometry. PLoS ONE. 2015. Vol. 10, no. 8, p. 1–21.
8. Hutchison, David. Brief Announcement: A Consent Management Solution for Enterprises. 2015. ISBN 978-3- 319-27238-2.