Word Distance Approach for Celebrity Profiling

Muhammad Usman Asif, Muhammad Naeem, Zeeshan Ramzan, and Fahad Najib

Department of Computer Science, University of Engineering and Technology, Lahore, KSK Campus, Pakistan
usmanasifweb@gmail.com, naeemshahzad7075@gmail.com, zramzan@uet.edu.pk, fahad.najib@uet.edu.pk

Abstract This paper describes and evaluates a model for Celebrity Profiling 2019 dataset. The training data set contain 33,836 celebrities' text with 50 different languages. The task was to create a model for this textual complex dataset which predict gender (male, female, nonbinary), fame (star, superstar, rising), occupation (sports, performer, creator, professional, manager, science, politics, religious) and birthyear (1940-2011) of celebrity. We use word distance features as input to different classifiers for different aspects (gender, fame, occupation and birthyear) of celebrity to create models. Results showed that word distance-based features outperformed the PAN baseline results.

Keywords: Celebrity Profiling · Text Classification · Natural Language Processing.

1 Introduction

Celebrity profiling task [14] offerd by PAN'19 [3] is to predict the celebrity predict gender (male, female, nonbinary), degree of fame (star, superstar, rising), occupation (sports, performer, creator, professional, manager, science, politics, religious) and birthyear (1940-2011) from celebrities' tweets written in 50 different languages. This task was offered by PAN [3]. The dataset [13] for both training and testing of models was given by PAN. The complete dataset contains tweets of 48,335 celebrity users. The training dataset consists of tweets of 33,836 users, and rest of the users' tweets were included in test dataset. The prediction of properties containing many labels e.g., birthyear contain 71 label classes and occupation contain 8 label classes, makes the task more challenging.

Almost all celebrities use the twitter and tweets there. The task has importance in social media and in celebrity industry for predicting the celebrity properties like gender, birthyear, occupation, fame by using their tweets. To measure these properties of celebrities from their tweets is significant for the celebrity fans, social media and industry. Knowing users' demographics from their written text has also applications in marketing as brands could increase reach of their message to more relevant audience [10,12]. The problem of celebrity traits predictions has also applications in forensic

[8,4] because of increasing cases of cyber crime including sexual harassment, threatening, identity theft etc.

2 Related Work

The problem of predicting different personality traits from text, due to it's applications in various other problem, have gain lot attention from community. Previously doc2vec document embedding technique used to train SVM and logistic regression classifier [2,5,1]. RUSProfiling (Cross-Genre Gender Identification in Russian texts) used character n-grams, word n-grams and gender specific Russian grammatical features to train multinomial Naive Bayes, logistic regression, random forest, and ensemble classifier for gender identification [7]. The problem to identification of author's traits from his/her written text have been addressed by using stylistic features to train different Machine Learning classifiers e.g., J48, Logistic Regression, Random Forest and Naive Bayes [11]. Different feature representations including raw frequency, binary, normalized frequency, tf-idf and second order attributes (SOA) have been used in combination with different machine learning algorithms including multinomial naive Bayes, Support Vector Machines (SVM), logistic regression [6].

3 Corpus

The PAN'19 [3] Celebrity Profiling [14] Dataset [13] contains twitter data of total 48,335 User Profiles. These tweets belongs to 50 different languages. A subset of this dataset, tweets of 33,836 users, used for the purpose of training models, whereas, remaining dataset consisting of 14,499 user profiles is used for testing of trained models. The complete training datasets consists of a single ndjson file in which tweets of all 33,836 user profiles/celebrities are present.

The corpus contains tweets, grouped by user/celebrity and labeled with gender (male, female, nonbinary), degree of fame (star, superstar, rising), occupation (sports, performer, creator, professional, manager, science, politics, religious) and birthyear (1940-2011).

The corpus was not balanced (See Figure 1). In case of gender, more than 50% profiles are of male celebrities, whereas, only 32 users belongs to nonbinary. Similarly, huge proportion of user profiles are stars, whereas, the frequency of rising and superstars is very low. Same is the case with occupation, where, there are sufficient instances of sports, performer and creator, whereas, remaining categories are in minority. The corpus is also unbalanced in case of birthyear (See Figure 2).

4 Methodology

We use the word distance approach for training models to predict different personality traits of celebrities. We made 200 (4 * 50) models as corpus contains tweets of 50 different languages and we have to predict fours aspects of user profile for each user profile. Each model predicts the specific class / personality trait for specific language such as

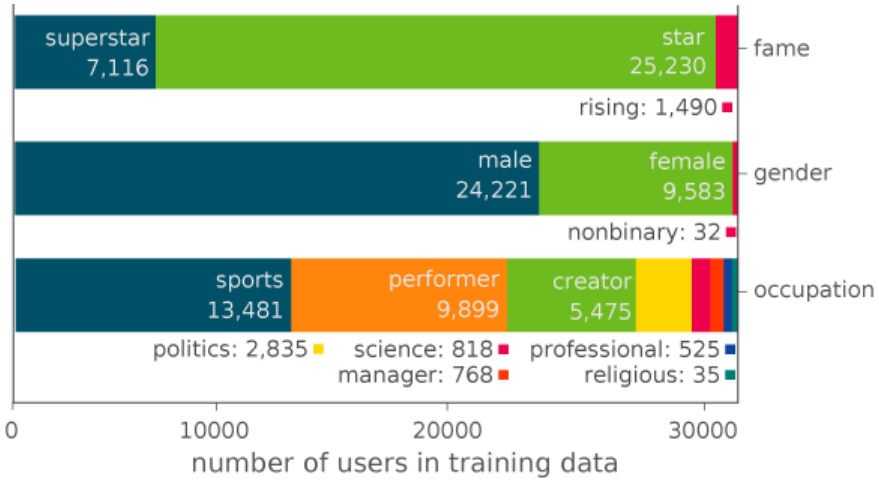

Figure 1. Overview of PAN19 Celebrity Profiling Dataset (Source: [14])

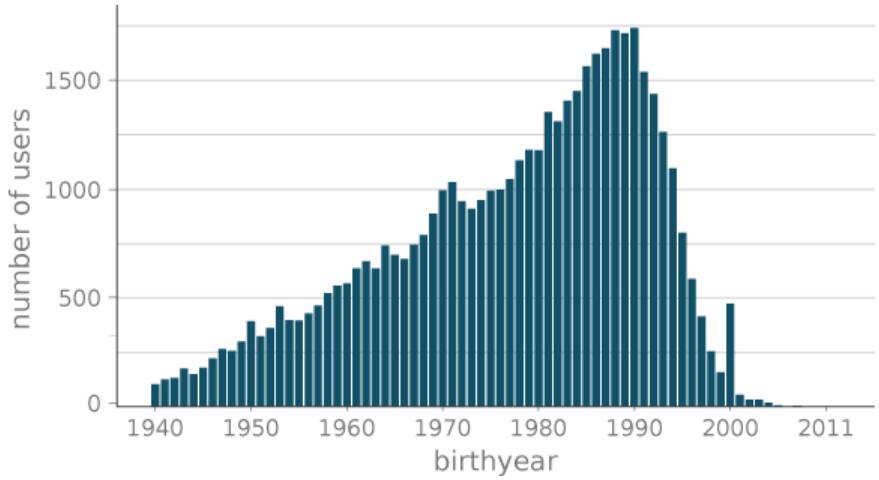

Figure 2. Frequency of user profiles in each birthyear (Source: [14])

English gender model predicts gender of user whose tweets are in English language. Each model has been trained using different sets of features and classifier.

4.1 Pre-processing

As corpus contains tweets written in 50 different languages, we put the same language tweets in same file by using langdetect module of Python. In this way, whole corpus was divided into 50 ndjson files such as en.ndjson, ar.ndjson files for languages English and Arabic. After separation we examine that almost 93% tweets are of English language

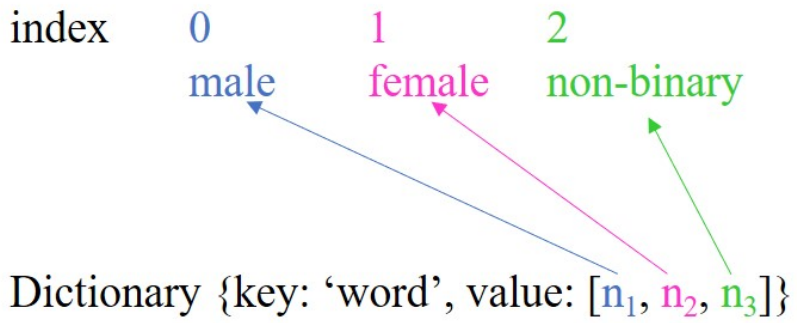

Figure 3. Example of dictionary used for Features Selection

and remaining 7% are non-English so made two categories English and non-English corpus.

After separating tweets of different languages, we applied different technique for data cleaning and features extraction for training models. There were lots of emojis, tag words, stop words, punctuation words, numbers, alphanumeric words, links, URLs, short form words, repeating characters words with punctuation's marks and escape characters. First of all, we removed all links / URLs from corpus by using a regular expression. Then, we extracted words from the text by tokenizing text using word tokenizers. We used the language specific word tokenizers for the purpose of tokenizing text into words. If, we could not find any word tokenizer for any language then we tokenized the text with space delimiter. We made a set of unique tokens. After that, we excluded all punctuation marks, stop words, numbers, alphanumeric words, URLs. Then we remove all escape characters and hash tag (#), @, spaces, brackets etc., from the word string and made the words clean. That's how we get the cleaned set of words.

4.2 Features Selection

After data cleaning and pre-processing, we made a dictionary for each personality trait for each language, which yielded 200 (4 * 50) dictionaries. The key of each of these dictionaries is a word and value of this dictionary is a list. Length of list depends upon the number of labels in the class (gender, fame, occupation or birthyear) to predict. Let if we want to make a model to predict gender then the list's first index (0 index) give count of male users in corpus who used this word (key in the dictionary) in their tweets. Same, second index for female and third index for non-binary (See Figure 3).

- n_1: Number of males in corpus who uses this word in their tweet
- n_2: Number of females in corpus who uses this word in their tweet
- n_3: Number of non-binaries in corpus who uses this word in their tweet

Same process was followed to create dictionaries for the fame, occupation and birthyear classes. For example, the range of birthyear is from 1940 to 2011. It contains 71 possibilities, so the list length would be 71.

The most important and tricky part of feature selection was to filter most distinguishing features from the dictionaries created in last step. For each word, we checked that which label / class (male, female, non-binary) is using this word most. If all classes are using a word with almost comparable frequency, then it is common word and we will not choose it as feature to train model. But if one label / class is using this word most and others are using it least, then we will choose this word as feature. We can say that, we will choose such words, for which, one label class has greater distance in count from other classes. For this we designed strategy to calculate how much maximum distance, a word is creating. As in total corpus we have let say 60% men, 30% women and 10% non-binary tweets so men will always dominate the women and non-binaries. For this we multiplied the count (n_1, n_2, n_3) of list with the corresponding ratio. Equations 1, 2 and 3 shows the formulas to calculate the ratio to be multiplied with the count.

$$ratio\_male = \frac{\text{total number of tweets in all corpora}}{\text{total number of tweets of male users}} \tag{1}$$

$$ratio\_female = \frac{\text{total number of tweets in all corpora}}{\text{total number of tweets of female users}} \tag{2}$$

$$ratio\_nonbinary = \frac{\text{total number of tweets in all corpora}}{\text{total number of tweets of non-binary users}} \tag{3}$$

After calculating the ratio, we multiplied the ratio number with each word's count list (n_1, n_2, n_3). New structure of key-value pair of dictionary is represented below:

$Word : [n_1 * ratio_{male}, n_2 * ratio_{female}, n_3 * ratio_{non-binary}]$

After multiplying the ratio, the problem created because of unbalanced dataset or dominating class is somehow solved. After this we calculated the difference created by the highest value of count with other counts in the list. For this we picked the highest count value in the list let say n_1 has the highest value in list, then calculated it's difference with other values in the list. At the end add the all differences. After adding we get a number which is the distance of that word. Let's say n_1 has highest value in list (n_1, n_2, n_3), the word distance would be calculated using Equation 4.

$$Word\_Distance = (n_1 - n_2) + (n_1 - n_3) \tag{4}$$

After calculating distance of each word, the dictionary would now contain words as keys and their respective distance as value

$Dictionary\{word_1 : word_1\_distance, word_2 : word_2\_distance\}$

Now, we sorted this dictionary in reverse direction. The large size of dictionary made it challenging to sort dictionary. Therefore, we get the list of all values form dictionary and sorted it in reverse order and then deleted the low distance values. After sorting list, we picked top scoring values and got the corresponding words from the dictionary and selected them as features.

After extracting features, we created CSV files to pass it to the Machine Learning algorithms to train model.

Table 1. Classifiers used for creating models

index	Language	Gender	Fame	Occupation	Birthyear
1	af	SVC	SVC	SVC	SVC
2	ar	Logistic Regression	K Neighbors	Random Forest	Logistic Regression
3	bg	SVC	SVC	SVC	SVC
4	bn	SVC	SVC	SVC	SVC
5	ca	SVC	SVC	SVC	SVC
6	cs	Logistic Regression	SVC	Logistic Regression	Logistic Regression
7	cy	SVC	SVC	SVC	SVC
8	da	Decision Tree	Logistic Regression	Logistic Regression	Logistic Regression
9	de	Logistic Regression	SVC	Logistic Regression	Logistic Regression
10	en	Logistic Regression	Random Forest	Random Forest	Logistic Regression
11	el	SVC	SVC	SVC	SVC
12	es	Logistic Regression	Logistic Regression	Random Forest	Logistic Regression
13	et	SVC	SVC	SVC	SVC
14	fa	SVC	Decision Tree	Decision Tree	Logistic Regression
15	fi	SVC	Decision Tree	Decision Tree	Logistic Regression
16	fr	Decision Tree	SVC	Random Forest	SVC
17	gu	SVC	SVC	SVC	SVC
18	he	Gaussian NB	SVC	Gaussian NB	Logistic Regression
19	hi	Logistic Regression	Random Forest	K Neighbors	Logistic Regression
20	hr	Gaussian NB	SVC	Decision Tree	Logistic Regression
21	hu	SVC	SVC	SVC	SVC
22	id	Gaussian NB	SVC	Gaussian NB	Logistic Regression
23	it	Gaussian NB	Decision Tree	Decision Tree	Logistic Regression
24	ja	Decision Tree	Logistic Regression	K Neighbors	Logistic Regression
25	kn	SVC	SVC	SVC	SVC
26	ko	SVC	SVC	SVC	SVC
27	lv	SVC	SVC	SVC	SVC
28	mk	SVC	SVC	SVC	SVC
29	mr	SVC	SVC	Decision Tree	Logistic Regression
30	ne	SVC	SVC	SVC	SVC
31	nl	Decision Tree	SVC	Decision Tree	SVC
32	no	Logistic Regression	Decision Tree	Logistic Regression	Logistic Regression
33	pl	Random Forest	Gaussian NB	Logistic Regression	Logistic Regression
34	pt	Decision Tree	SVC	Random Forest	Decision Tree
35	ro	SVC	SVC	SVC	SVC
36	ru	Logistic Regression	K Neighbors	Decision Tree	Logistic Regression
37	sk	SVC	SVC	SVC	SVC
38	sl	Logistic Regression	Logistic Regression	Decision Tree	Logistic Regression
39	so	Random Forest	Gaussian NB	Decision Tree	Logistic Regression
40	sq	SVC	SVC	SVC	SVC
41	sv	Decision Tree	Decision Tree	Logistic Regression	Logistic Regression
42	sw	Random Forest	Logistic Regression	Logistic Regression	Logistic Regression
43	ta	SVC	SVC	SVC	SVC
44	te	SVC	SVC	SVC	SVC
45	th	SVC	SVC	SVC	SVC
46	tl	SVC	SVC	SVC	SVC
47	tr	Gaussian NB	Random Forest	K Neighbors	K Neighbors
48	uk	SVC	SVC	SVC	SVC
49	ur	Logistic Regression	Logistic Regression	Logistic Regression	Logistic Regression
50	vi	SVC	SVC	SVC	SVC

Table 2. Results on Training Dataset

Traits	F-measure
Gender	0.746
Fame	0.989
Occupation	0.995
Birthyear	0.307
cRank = 0.604653	

4.3 Classifiers

The sklearn implementations of various Machine Learning algorithms were applied on CSV files created in last step. We used 80% data for training the model and 20% for testing. We applied six different algorithms (See Table 1) to train models. Then tested the models with 20% testing data. We selected highest scoring algorithms for training the model using 100% available data.

4.4 Evaluation Measures

The performance of our proposed for Individual traits was judged by F_1 measure (See Equation 5). Whereas, overall performance of the system would be judged by a combined metric cRank, which is harmonic mean of each label's metric (See Equation 6).

$$F_1 = \frac{2 * precision * recall}{precision + recall} \tag{5}$$

$$\text{cRank} = \frac{4}{\frac{1}{\text{F}_{1,\text{fame}}} + \frac{1}{\text{F}_{1,\text{occupation}}} + \frac{1}{\text{F}_{1,\text{gender}}} + \frac{1}{\text{F}_{1,\text{age}}}} \tag{6}$$

5 Results and Analysis

The results of our proposed approach on Training dataset are presented in Table 2. Table 2 shows the F-measure of all traits of training dataset. The F-measure in case of occupation and fame is much higher than other two traits. The training dataset is somehow more balanced in case of occupation and fame than other two traits could be the reason for such results. Moreover, the birth year range is 1940-2011 and there are not enough user profiles for non-English to cover all these birthyears (1940-2011). These problems with the training dataset made it very challamging to correctly predict birthyear. The cRank (See Equation 6) on Training dataset, the combined score of all traits, is 0.604653.

Table 3 presents results obtained by applying our proposed technique on test dataset using TIRA [9]. These results show that our technique could not perform well on test dataset as compared with training dataset. The features, the list of words used for training, were extracted from train dataset, which were not necessarily present in test dataset

Table 3. Results on Test Dataset

Traits	F-measure
Gender	0.588
Fame	0.505
Occupation	0.427
Birthyear	0.254
cRank = 0.40181	

with comparable frequency. This limitation of this approach resulted in over-fitting by giving very promising results on training dataset but not on test dataset.

6 Conclusion and Future Work

In this paper, we have explained a technique for the prediction of the celebrities' gender, fame, occupation and birthyear from their tweets. It has applications in various fields like forensics, marketing and security. We trained models on the training data provided by the PAN organizers. The results we achieved on are pretty good. In future, more performance can e achieved by making training dataset more balanced and well representative of the population. Moreover, more sophisticated features, which are not specific to training dataset, can also improve results.

References

1. Akhtyamova, L., Cardiff, J., Ignatov, A.: Twitter author profiling using word embeddings and logistic regression. In: CLEF (Working Notes) (2017)
2. Bayot, R.K., Gonçalves, T.: Author profiling using svms and word embedding averages. In: CLEF (Working Notes). pp. 815–823 (2016)
3. 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)
4. Grant, T.: Quantifying evidence in forensic authorship analysis. International Journal of Speech, Language & the Law **14**(1) (2007)
5. Markov, I., Gómez-Adorno, H., Posadas-Durán, J.P., Sidorov, G., Gelbukh, A.: Author profiling with doc2vec neural network-based document embeddings. In: Mexican International Conference on Artificial Intelligence. pp. 117–131. Springer (2016)
6. Markov, I., Gómez-Adorno, H., Sidorov, G., Gelbukh, A.F.: Adapting cross-genre author profiling to language and corpus. In: CLEF (2016)
7. Markov, I., Gómez-Adorno, H., Sidorov, G., Gelbukh, A.F.: The winning approach to cross-genre gender identification in russian at rusprofiling 2017. In: FIRE (2017)

8. Peng, J., Choo, K.K.R., Ashman, H.: Bit-level n-gram based forensic authorship analysis on social media: Identifying individuals from linguistic profiles. Journal of Network and Computer Applications **70**, 171–182 (2016)
9. 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)
10. Rambocas, M., Gama, J., et al.: Marketing research: The role of sentiment analysis. Tech. rep., Universidade do Porto, Faculdade de Economia do Porto (2013)
11. Sittar, A., Ameer, I.: Multi-lingual author profiling using stylistic features. In: FIRE (2018)
12. Ting, T.C., Davis, J., Pettit, F.A.: Online marketing research utilizing sentiment analysis and tunable demographics analysis (2014), uS Patent 8,694,357
13. Wiegmann, M., Stein, B., Potthast, M.: Celebrity Profiling. In: Proceedings of ACL 2019 (to appear) (2019)
14. 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)