Gender classification of Twitter comments using supervised methods

1st Sima Rezaeipourfarsangi
Faculty of Computer Science
Dalhousie University
Halifax, Canada
sima.rezaei@dal.ca

2nd Evangelos Milios
Faculty of Computer Science
Dalhousie University
Halifax, Canada
eem@cs.dal.ca

***Abstract*—In recent years, the undeniable use of social networks has given rise to numerous motivations for exploring posts, comments, and so on. However, getting some accurate and actual information about users is a complicated task because many users post incorrect information about their age, appearance, and especially their gender in their profile. Information extraction specialists are therefore keen to discover if these characteristics can be detected automatically. In this study, we try to determine the gender of people on Twitter with the help of a novel classification method based on Firefly Algorithm (FA). We tested several classifications of existing algorithms alongside our customized classifier for this problem. By optimizing the basic parameters of the proposed classification, it showed a higher accuracy of 94%.**

***Index Terms*—Gender Classification, Random Forest, Extra Trees, Ensemble Methods**

I. Introduction

Many people around the world spend several hours a day on social media. This has made social networks, especially Twitter, a great and rich source for information extraction. Twitter, a micro-blogging service supporting more than 35 languages, has more than 300 million active users per month. Daily, these users post close to 500 million tweets. Users can use Twitter to share events, daily activities, and information, as well as connect with friends [1]. One of the problems with this important source of information is that some user profiles are unclear or incorrect. The reason why many researchers and organizations need to know the true gender of users in social networks is its application in areas such as Psychological analysis, Legal investigation, Marketing analysis, Forensics, Recommendation Systems, and Advertising. Therefore, in this study, we decided to provide a modified classifier to determine the gender of users based on the text of comments on Twitter.

II. Related Works

One of the first works in this field was done in 2010 in [3]. This work presents an innovative research of stacked-SVM-based classification algorithms applied to identifying these four user attributes: regional origin, age, gender, and political inclination, using a rich set of original information. This paper reported an accuracy of 71.8 percent when utilizing sociolinguistic characteristics, but only 67.7 percent when employing n-grams. Using the stacked Support Vector Machine support-vector-based classification model, they attained an accuracy of 72.3 percent when integrating n-gram features with sociolinguistic data.

Another application of SVM proposed in [4] makes use of attributes linked to the homophily concept. This involves inferring user features based on the attributes of the user's immediate neighbors utilizing tweet content and profile information. The studies were carried out using an support vector classifier, and the prediction model's accuracy was 80.2 percent when using neighborhood data and 79.5 percent when using simply user data. There is also a name-based and image-based method in [5] where authors measured accuracy based on user location. In some countries, this method predicted the gender of users with near-perfect accuracy but error rates are highly dependent on an individual's country of residence. Ensemble classifiers have been used in in [6] for both bot detection and gender identification. this approach predicts genders with 78 percent accuracy. This problem is also addressed in the context of gender detection by [7] , which uses multi-model deep learning architectures to generate specialized understanding from different feature spaces and achieves an accuracy of up to 86%.

Gender detection from tweet text has been used in [8] using character embeddings and attention-based Convolutional Neural Networks (CNNs) without any preprocessing. This model had a 75 percent accuracy.

III. Data

In this study, we used the same data set that was used to learn a CrowdFlower AI gender predictor. Contributors were simply asked to look at a Twitter profile and identify whether the user was male, female, or a brand (non-individual). Each row in the dataset contains a user name, a random tweet, an account profile and image, a location, and even the color of the link and sidebar [9].

IV. Method

In this section, we first present a 5-step framework for determining gender. This framework is presented in order to use different classification algorithms along with the proposed algorithm. In other words, since the main innovation in this research has been done in the classification stage, by defining this framework, we can accurately compare the results

obtained with other classification algorithms by just changing the classification section to obtain and analyze their results. Then, in order to obtain the best combination of supervised learning models and preprocessing, we have examined the implementation results of different scenarios. in Figure 1 the general architecture of how to determine gender is shown.

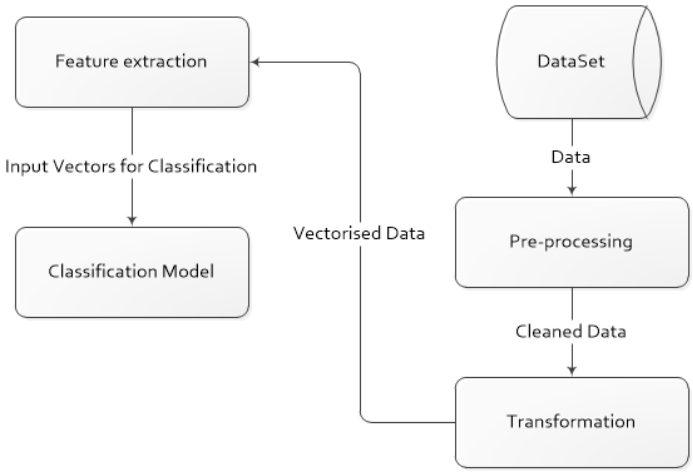

Fig. 1. General Flowchart for gender detection.

A. Pre-Processing

In order to turn the problem of gender prediction into a manageable problem of classification, we first need to remove some useless data such as numbers, punctuations, special character, hashtags, smileys, emojis, extra blank space.

B. Data transformation

The technique of translating text into numerical form is known as text vectorization. in other word, it is important to represent the papers and the text within the documents as vectors. At this point we need to decide how to do this. Three models including TF-IDF, W2V, and Glove have been reviewed for this purpose, which will be briefly reviewed below.

1) TF-IDF: The term frequency–inverse document frequency (tf-idf) approach is a numerical statistic supposed to measure the importance of a word in a corpus of documents. The Tf-idf vectorizer compares the frequency of tokens in the document to their frequency in other documents [11].

2) Glove: Matrix factorization and neural embeddings are two types of dense vectors. GloVe, along with another prominent neural approach known as Word2vec, falls under the latter type [15]. In a nutshell, GloVe is an unsupervised learning algorithm that prioritizes word-word co-occurences over other techniques like skip-gram or bag of words when extracting meaning. The concept is that a given term co-occurs more frequently with one word than with another. For example, the word ice is more likely to appear alongside the phrase water.

3) W2V: Glove and word2vec are word-learning methods that take into account the presence and co-occurrence of words in vectors. Word2vec generates one vector for each word, whereas tf-idf generates a score. Word2vec is excellent for delving deeper into our documents and identifying content and subsets of content. Its vectors represent the context of each word. (i.e. the n-gram that it is a part of) [16]

C. Classification Model

After normalizing the data and discovering the properties, we now have a classification tool that in the stage we apply different models of classifiers, which we will briefly introduce in the following, to the data.

In addition to testing different models, we need to test the various parameters of the models themselves to achieve the best configuration for each. For this purpose, we will optimize each of the hyper-parameters with the grid search method.

1) Logistic Regression (LR): Logistic regression is a statistical model that uses a logistic function to represent a binary dependent variable in its most basic form, however many more advanced extensions exist [10]. A Logistic function is a common S shape with equation $f(x) = \frac{L}{1+e^{-k(x-x_0)}}$. where L shows the curves maximum value, K is the steepness of the curve and x is value of the sigmoid's point. Standard logistic function is given by $f(x) = \frac{1}{1+e^{-x}}$

2) SVM: SVM is a well-known classification technique that has applications in fraud detection, distinguishing cancer cells from healthy cells, facial recognition, weather prediction, and so on. SVM is designed to identify a binary classifier using training data that has already been labeled by the supervisor; thus, it is referred to as a supervised learning classifier. There are several variants of this problem in the literature, but binary SVM classification is the most common [12].

3) Random Forest (RF): The random forest is an ensemble classification algorithm that means it is made of many decision trees and for final result it aggregates the prediction of any single decision tree. In other words, its Random Because uses bagging and feature randomness when building each individual tree and it is Forest because try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree [10].

Overfitting in the decision tree can be avoided using the Random Forest. To train each tree, a random fraction of the training set is sampled, and then a decision tree is built, with each node splitting on a feature selected at random from the entire feature set. Because each tree is trained independently of the others, random forest training is extremely fast, even for large data sets with many attributes and data instances. The Random Forest method avoids overfitting and provides a good approximation of the generalization error [17] .

4) XGBoost: XGBoost is a distributed gradient boosting algorithm that has been tuned for efficiency, flexibility, and portability. It uses the Gradient Boosting framework to create machine learning algorithms. XGBoost is a parallel tree boosting (also known as GBDT, GBM) algorithm that solves a variety of data science issues quickly and accurately [13].

5) Extremely Randomized Trees (ET): Extremely Randomized Trees also known as Extra-Trees algorithm [14], unlike other tree-based ensemble methods such as RF, employs the entire training sample rather than a bootstrap replica subset of features and separates nodes by selecting cut-points at random.

The distinction between ET and the random forest approach is that the extreme random tree method, unlike the random forest method, obtains the branching value fully at random in order to execute classification tree branching. On a random subset, find the forest's best fork attribute. In addition, the extreme random tree approach employs all of the training data in each regression tree.

6) Proposed Classification Model: Unlike numerical data in textual data, what is particularly important is the similarity of the texts to each other. Based on this fact, the main idea of this article is to present a personalized classification based on the similarity of the vectors extracted from the text. For this purpose, we are inspired by center-based methods in machine learning. Although most of these methods are used for unsupervised learning, in the proposed method as described below, we present a supervised center-based algorithm for classifying comments by gender.

In identifying the gender of the author of a sentence, we can define a concept called the central writing pattern. A central writing pattern is a sentence that has a repetitive pattern among its peers. These centralized writing patterns are essentially the same centers in centralized methods with differences to adapt the concept to textual data. So we need a simple criterion for similarity between points (vector sets). We sleceted Jaccard similarity coefficient of the texts that is calculated in pairs. If $S(A)$ is the set of vectors of one text and $S(A)$ is the set of vectors of other one. The Jaccard similarity coefficient is defined as:

$$J(A,B) = \frac{|S(A) \cap S(B)|}{|S(A) \cap S(B)|} \tag{1}$$

this criterion had shown good performance for similarity of vectored texts [2].

In the continuation of our model, for each class, we need several central patterns that have the low similarity to each other in the same class and the most similarity to other points in similar class. In fact, by doing this, each central pattern of the text represents only a specific group of the desired class.

Therefore, the set of central patterns of each class (male or female) is defined as follows.

$$P_{class} = \{V | \sum_{T \in class} J(V,T) \leq \Delta \wedge \sum_{C \in P_{class}} J(V,C) \geq \delta\} \tag{2}$$

$$\Delta = \frac{\alpha \sum_{T \in D} J(V,T)}{|D|} \tag{3}$$

$$\delta = \frac{\beta \sum_{T \in class} J(V,T)}{|class|} \tag{4}$$

Where α and β are two hyper-parameters of our proposed model. They determine the coefficient of selection of the central pattern thresholds based on the average of the similarities in the data set as well as each of the classes. Also D is the whole training data.

Based on this background algorithm 1 shows the training phase of proposed classification model. From now on, we will call this model VTCC, which stands for "Vectorized Text Center-based Classifier".

Algorithm 1 Learning algorithm of VTCC

Input:
- D: Set of vectors (from texts) as training data.
- α, β and λ hyper parameters

for all $class$ **do** $class \leftarrow \phi$
end for
for $V \in D$ **do**
  if $class_V = \phi$ **then**
    add V to P_{class_V}.
    initialize Δ and δ: $\Delta = \frac{\alpha \sum_{T \in D} J(V,T)}{|D|}$ $\delta = \frac{\beta \sum_{T \in class} J(V,T)}{|class|}$
  else
    if $\sum_{T \in class_V} J(V,T) \leq \Delta \wedge \sum_{C \in P_{class_V}} J(V,C) \geq \delta$ **then**
      add V to P_{class_V}.
      update Δ and δ.
      if $|class_V| > \lambda$ **then**
        sort P_{class_V} on $\sum_{T \notin class_V} J(V,T)$
        remove least one from $|class_V|$.
      end if
    end if
  end if
end for

In this algorithm, we define another Hyper-parameter that represents the maximum number of central patterns that we know as λ. For prediction of a new sample point in this model, we calculate the similarity with central patterns of each class and assign it to more similar one.

D. Optimize Hyper-paramaters

An important activity in finding the optimal method for regression problems is finding the optimal value for the Hypter-parameters. Tuning may be manual [18] or using Grid Search, Random Search, GA or other evolutionary algorithms [19]. This is especially important for tree-based methods such as random forest and ET.

If we use Grid Search to find optimal parameters so that we can analyses the impact of each single one on accuracy increasing or decreasing. As part of the results, the amount of prediction error for different values in ET is shown in Figures 3 and 2.

Based on Figures 3, 2 finding optimal value for hyper-parameters of these models can increase the performance but it seems not enough to gain the maximum performance.

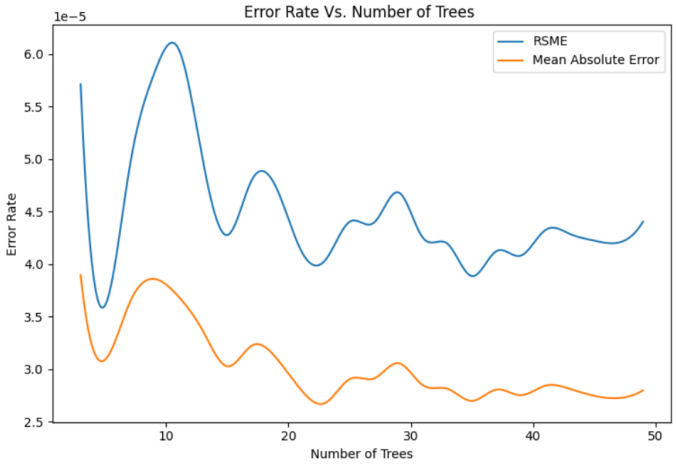

Fig. 2. Impact of Number of trees on Error rate.

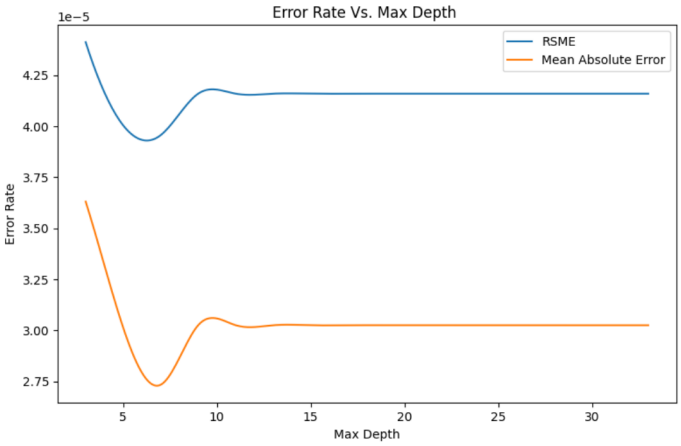

Fig. 3. Impact of Max Depth on Error rate.

Therefore, the idea of using Firefly Algorithm (FA) to make a novel voting tree-based classifier involved here.

V. Results

As it mentioned above to find a final robust configuration for our method, we tested more than 20 combinations. In Table I the percentage of true classification for each method combinations are shown.

TABLE I
ACCURACY OF ALL COMBINATIONS

Classifier / Vectorise	**TF-IDF**	**W2V**	**Glove**
LR	53.65	57.14	54.43
SVM	52.67	52.67	52.67
RF	47.7	47.72	48.46
XGBoost	54.93	55.38	52.39
ET	57.42	54.88	56.12
VTCC	57.14	55.17	55.90

As can be seen, the simple case of the classifiers used predicts almost half of the cases correctly. In order to increase accuracy, the classification hyper-parameters were optimized as described in Section IV-C. The results after this optimization are given in Table III. Also, the list of parameters considered for each of these models is shown in Table II.

TABLE II
HYPER-PARAMETERS OF MODELS

Classifier	Hyper-parameters
LR	Maximum number of iterations - Penalty
SVM	Regularization parameter - Kernel
RF	Number of trees - Max Depth - Criterion
XGBoost	Number of boosting stages - Criterion
ET	Number of trees - Max Depth - Criterion
VTCC	α , β and λ

TABLE III
ACCURACY OF ALL COMBINATIONS AFTER HYPER-PARAMETERS ADJUSTMENT

Classifier / Vectorise	TF-IDF	W2V	Glove
LR	55.52	55.91	54.21
SVM	66.07	52.28	52.07
RF	63.94	65.19	63.9
XGBoost	80.04	75.22	74.25
ET	92.05	88.13	89.99
VTCC	94.22	89.74.13	90.98

VI. Conclusion

In this study, several classification models were developed on data related to gender identification on Twitter by optimizing the hyper-parameters of these models. In addition to the usual methods for classification, a proposed classification method was proposed that was personalized for the present problem. This method is a center-based method based on the characteristics of textual data.

As a result, the ET model showed with an accuracy of 92% that it can accurately distinguish the gender of users in cyberspace by optimizing its parameters. The proposed classifier operates with close accuracy but higher than ET, and by optimizing its meta-parameters, gender can be accurately predicted in 94.22% of cases.

References

[1] Vicente, Marco and Batista, Fernando and Carvalho, Joao P Gender detection of Twitter users based on multiple information sources, Interactions Between Computational Intelligence and Mathematics Part 2, 2019.

[2] Mahmoodi, Maryam and Varnamkhasti, Mohammad Mahmoodi Design a Persian automated plagiarism detector (AMZPPD), arXiv:1403.1618, 2014.

[3] Rao, Delip and Yarowsky, David and Shreevats, Abhishek and Gupta, Manaswi, Classifying latent user attributes in twitter, Proceedings of the 2nd international workshop on Search and mining user-generated contents, pp. 37–44, 2010.

[4] Al Zamal, Faiyaz and Liu, Wendy and Ruths, Derek, Homophily and latent attribute inference: Inferring latent attributes of twitter users from neighbors, Sixth International AAAI Conference on Weblogs and Social Media, 2012.

[5] Karimi, Fariba and Wagner, Claudia and Lemmerich, Florian and Jadidi, Mohsen and Strohmaier, Markus, Inferring gender from names on the web: A comparative evaluation of gender detection methods, Proceedings of the 25th International conference companion on World Wide Web, pp. 53–54, 2016.

[6] Kosmajac, Dijana and Keselj, Vlado, Twitter user profiling: bot and gender identification, International Conference of the Cross-Language Evaluation Forum for European Languages, pp. 141–153, 2020.
[7] Kowsari, Kamran and Heidarysafa, Mojtaba and Odukoya, Tolu and Potter, Philip and Barnes, Laura E and Brown, Donald E, Gender detection on social networks using ensemble deep learning, Proceedings of the Future Technologies Conference, pp. 346–358, 2020.
[8] Sezerer, Erhan and Polatbilek, Ozan and Sevgili, Özge and Tekir, Selma, Gender prediction from Tweets with convolutional neural networks: Notebook for PAN at CLEF 2018, 19th Working Notes of CLEF Conference and Labs of the Evaluation Forum, CLEF 2018.
[9] Schnoebelen, T, The gender of artificial intelligence. Retrieved August 12, 2017.
[10] Hilbe, Joseph M, Logistic regression models, 2009.
[11] Rajaraman, Anand and Ullman, Jeffrey David, Mining of massive datasets, 2011.
[12] Ben-Hur, Asa and Ong, Cheng Soon and Sonnenburg, Sören and Schölkopf, Bernhard and Rätsch, Gunnar, Support vector machines and kernels for computational biology, 2008.
[13] Kiangala, Sonia Kahiomba and Wang, Zenghui, An effective adaptive customization framework for small manufacturing plants using extreme gradient boosting-XGBoost and random forest ensemble learning algorithms in an Industry 4.0 environment, 2021.
[14] Geurts, Pierre and Ernst, Damien and Wehenkel, Louis, Extremely randomized trees, Machine Learning with Applications, pp. 3–42, 2006.
[15] Pennington, Jeffrey and Socher, Richard and Manning, Christopher D, Glove: Global vectors for word representation, Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543, 2014.
[16] Church, Kenneth Ward, Word2Vec, Natural Language Engineering, pp. 155–162, 2017.
[17] Chhajer, Siddth Kumar and Satpathy, Rudra Bhanu, Deception Recognition Method Based on Machine Learning, 2017.
[18] Hutter, Frank and Lücke, Jörg and Schmidt-Thieme, Lars, Beyond manual tuning of hyperparameters, KI-Künstliche Intelligenz, pp. 329–337, 2015.
[19] Liashchynskyi, Petro and Liashchynskyi, Pavlo, Grid search, random search, genetic algorithm: a big comparison for NAS, arXiv preprint arXiv:1912.06059, 2019.