Examining Multiple Features for Author Profiling

Edson R. D. Weren, Anderson U. Kauer, Lucas Mizusaki, Viviane P. Moreira,
J. Palazzo M. de Oliveira, Leandro K. Wives

Institute of Informatics, Universidade Federal do Rio Grande do Sul, Porto Alegre–Brazil
{erdweren,aukauer,lepmizusaki,viviane,palazzo,wives}@inf.ufrgs.br

Abstract. Authorship analysis aims at classifying texts based on the stylistic choices of their authors. The idea is to discover characteristics of the authors of the texts. This task has a growing importance in forensics, security, and marketing. In this work, we focus on discovering age and gender from blog authors. With this goal in mind, we analyzed a large number of features – ranging from Information Retrieval to Sentiment Analysis. This paper reports on the usefulness of these features. Experiments on a corpus of over 236K blogs show that a classifier using the features explored here have outperformed the state-of-the art. More importantly, the experiments show that the Information Retrieval features proposed in our work are the most discriminative and yield the best class predictions.

Categories and Subject Descriptors: H.3 [**Information Storage and Retrieval**]: Miscellaneous; I.7 [**Document and Text Processing**]: Miscellaneous

Keywords: author profiling, classification, text mining, information retrieval

1. INTRODUCTION

Author profiling aims at finding out as much information as possible about a person just by analyzing texts written by that person. The analysis can focus on identifying age, gender, mother language, level of education, or other social-economic categories. This is a prolific field, with a wide range of applications in forensics, marketing, and Internet security. For example, companies may analyze the texts of product reviews to identify which type of customer likes or dislikes their products, or the police may identify the perpetrator of a crime by analyzing suspects' writing profiles.

This area has been gaining greater attention in the last couple of years. As a consequence, in 2013 the PAN Workshop Series on uncovering plagiarism, authorship, and social software misuse[1] has designed an evaluation lab focusing on author profiling [Gollub et al. 2013; Rangel et al. 2013]. This task required participants to come up with approaches to identify the gender (male/female) and the age group (10s, 20s, 30s) of the authors of a set of texts of blogs given as input. The *profiles* are the classes which need to be predicted. The task raised a lot of interest – over 70 groups signed up and 21 effectively took part by submitting a software. PAN organizers provide participants with training data (*i.e.*, texts for which the age and gender of the authors is known) and then evaluate the submitted softwares on a new unseen dataset. The submissions are ranked according to the accuracy of their predictions. The best scoring system [Meina et al. 2013] achieved 0.59 accuracy in predicting gender and 0.65 accuracy for age. This shows that author profiling is a challenging task and there is still room for improvement.

The aim of this article is to contribute to the topic of author profiling by experimenting with

[1] http://pan.webis.de/

This work has been partially funded by CNPq project 478979/2012-6.
Copyright©2014 Permission to copy without fee all or part of the material printed in JIDM is granted provided that the copies are not made or distributed for commercial advantage, and that notice is given that copying is by permission of the Sociedade Brasileira de Computação.

various features and classification algorithms. Our goal is to evaluate which features are more helpful in discriminating the characteristics of the authors and which classification algorithms are most suited to this task. In our previous work [Weren et al. 2013], we have explored features based on Information Retrieval (*i.e.* similarity metrics such as cosine and Okapi BM25) and on readability tests. Here we expand on that study by using more IR-based features, analyzing a total of 61 features which were submitted to 55 classification algorithms. The results of the experiments outperform the state-of-the-art and surpass the top scoring team at PAN 2013. In addition, they show that the IR-based features are the most discriminative and yield the best classification results.

The remainder of this article is organized as follows: Section 2 discusses the related literature, Section 3 introduces the features we used, Section 4 presents our experimental evaluation, Section 5 discusses further relevant issues on the topic of author profiling, and Section 6 concludes this work.

2. RELATED WORK

Koppel et al. [2003] is one of the pioneers in author profiling. They showed that it is possible to find patterns in writing styles which may be used to classify the author into a specific demographic group. In that study, a group of function words (i.e., words that do not have meaning but serve to express grammar relationships with other words – e.g. prepositions, pronouns, auxiliary verbs) and part-of-speech analysis was used in order to identify the gender of the author and whether the text was fiction or nonfiction. A total of 1081 features were used, resulting in an accuracy higher than 0.80 on 920 documents from the British National Corpus. In a later work, Argamon et al. [2009] define two basic groups of features that can be used for author profiling: content dependent features (*i.e.*, certain subjects, keywords, and key phrases that are mainly used by one of the groups), and style-based features, which are content independent (e.g. average length of words). Experiments on texts from 19K blogs using Bayesian Multinomial Regression as the learning algorithm have reached accuracy of about 0.76. The authors observed that determiners and prepositions are markers of male writing while pronouns are markers of female writing. Contractions without apostrophes were found to be discriminative of younger writing, while prepositions are stronger in older writing.

Mukherjee and Liu [2010] address the problem of identifying the gender of blog authors. They mine patterns in part-of-speech (POS) tags and propose a feature selection method. An experimental evaluation shows that features selection brings up to 10% improvement and the POS patterns yield gains of about 6%.

Otterbacher [2010] also focused on gender identification, but on another dataset: movie reviews. That study concentrated on evaluating differences between male and female writing with respect to style, content, and metadata (*i.e.*, date posted, review rating, number of reviews contributed). These features were used in a logistic regression classifier. Best results were obtained when all types of features were used. Interestingly, they found that a metadata attribute, *i.e.*, the utility of the review, yielded good results as reviews written by men were usually found to be more useful.

Still on gender prediction, Sarawgi et al. [2011] considered blogs and scientific papers. The authors used probabilistic context-free grammars [Raghavan et al. 2010] to capture syntactic regularities beyond shallow ngram-based patterns. They found character-level language models performed better than word-level models. As expected, predicting gender on scientific papers was harder than on blogs. However, the results show an accuracy of 61% which is well above a random prediction (*i.e.*,50%). This demonstrates that gender can be identified even in formal writing.

Nguyen et al. [2011] focused on age prediction on different corpora: blogs, transcribed telephone calls, and posts from an online forum. The authors used gender as a feature combined with unigrams, POS tags, and word classes obtained by LIWC [Pennebaker et al. 2001]. Experiments using linear regression have shown that the combination of features yields the best results. Gender was found to be a useful feature for discriminating the age of younger authors.

Peersman et al. [2011] studied age and gender prediction on social networks. The key aspect here is the short length of the text (12 words on average). The features used include emoticons, sequences of words (unigrams, bigrams and trigrams) and characters (bigrams, trigrams and tetragrams). Those were submitted to a SVM classifier. Word features and emoticons were found to be more useful. Best results were obtained when the training dataset was balanced.

Last year, PAN 2013 provided a common benchmark for evaluating author profiling. The 21 participants who submitted softwares reported their experiments in notebook papers which were summarized by Rangel et al. [2013]. The three best scoring systems for English, ranked by their accuracies, were by Meina et al. [2013], López-Monroy et al. [2013], and Mechti et al. [2013].

Meina et al. [2013] used a wide range of features including structural features (number of sentences, words, paragraphs), part-of-speech analysis, readability, emotion words, emoticons, and statistical topics derived from Latent Semantic Analysis (LSA) [Deerwester et al. 1990]. LSA employs a dimension reduction technique aiming at noise reduction. There were 311 features for age and 476 for gender, which were used by a Random Forest classifier [Breiman 2001] (*i.e.,* an ensemble classifier which builds several decision trees and outputs the class which is the mode of the classes predicted by the individual trees). The authors also included a preprocessing step to identify spam-like blog posts and discard them from the dataset. Notice that, while having the best accuracy scores, this was one of the slowest systems, taking 4.44 days to process the test data (which does not include training). This system got the best accuracy for gender and the second best for age.

While most approaches represent documents as vectors using each term as a feature (i.e., bag-of-words), López-Monroy et al. [2013] propose a different approach for document representation: the Second Order Attribute. The key idea is to compute how each term relates to each *profile*. The profiles are the classes that we wish to predict, namely: 10s_female, 10s_male, 20s_female, 20s_male, 30s_female, and 30s_male. The relationship between a term and a profile is based on term frequency. Once the term vectors are computed, the relationship between document vectors and profiles is calculated. Their approach performs classification into 6 classes. A LibLinear [Fan et al. 2008] classifier was used with the 50K most frequent terms as features. The speed performance of this system was also good – it took 38 minutes to classify the test data. This system got the best accuracy for age and the third best for gender.

Mechti et al. [2013] computed the 200 most frequent terms per profile and grouped them into classes such as: determiner, prepositions, pronouns, words related to love, words used by teens, etc. There were 25 such classes for English. The features were used to train a decision tree classifier (J48). This was the slowest system, taking over 11 days to classify the test data. It ranked second for gender prediction, however, age prediction was ranked much lower.

We also took part in this competition using ten features based on Information Retrieval and two features based on readability tests [Weren et al. 2013]. That was a preliminary study in which we indexed just a small sample from the training data.

In this work, we aim at further analyzing the validity of the Information Retrieval based features, considering a total of 30 such attributes. Also, unlike related work, we do not make use of POS tagging as this is a costly operation. Another difference is that we do not use the terms from the text as features.

3. IDENTIFYING CHARACTERISTICS OF THE AUTHOR OF AN ANONYMOUS TEXT

In this section, we detail the features used in our investigation as well as the classification approach that we adopted.

3.1 Features

The collection of texts from each author were represented by a set of 61 features (or attributes), which were divided into six groups. Next, we explain each of these groups.

3.1.1 *Length.* These are simple features that calculate the absolute length of the text.

—`Number of Characters`;
—`Number of Words`; and
—`Number of Sentences`.

3.1.2 *Information Retrieval.* This is the group of features that encode our assumption that authors from the same gender or age group tend to use similar terms and that the distribution of these terms would be different across genders and age groups. The complete set of texts (without stemming or stopword removal) is indexed by an Information Retrieval (IR) System. Then, the post (*i.e.* blog document) that we wish to classify is used as a query and the k most similar posts are retrieved. The idea is that the posts that will be retrieved (i.e., the most similar to the query) will be the ones from the same gender and age group. The ranking is given by the cosine or Okapi metrics as explained below. These metrics were chosen as they are the most widely used in IR for ranking documents in response to queries. The cosine is used in the Vector-Space Model and Okapi BM25 is a well established probabilistic model [Manning et al. 2008]. We employ a total of 30 IR-based features.

—**Cosine**
The features which are based on the ranking produced by the cosine are: `Cosine_10s_count`, `Cosine_20s_count`, `Cosine_30s_count`, `Cosine_female_count`, `Cosine_male_count`, `Cosine_10s_sum`, `Cosine_20s_sum`, `Cosine_30s_sum`, `Cosine_female_sum`, `Cosine_male_sum`, `Cosine_10s_avg`, `Cosine_20s_avg`, `Cosine_30s_avg`, `Cosine_female_avg`, and `Cosine_male_avg`. These features are computed as an aggregation function over the top-k results for each age/gender group obtained in response to a query composed by the keywords in the post. We tested three types of aggregation functions, namely: count, sum, and average. For this feature-set, queries and posts were compared using the cosine similarity (Eq. 1). For example, if we retrieve 100 documents in response to a query composed by the keywords in q, and 50 of the retrieved posts were in the 10s age group, then the value for `Cosine_10s_avg` is the average of the 50 cosine scores for this class. Similarly, `Cosine_10s_sum` is the summation of such scores, and `Cosine_10s_count` simply tells how many retrieved posts fall into the 10s category. The cosine similarity is as:

$$cosine(c, q) = \frac{\overrightarrow{c} \cdot \overrightarrow{q}}{|\overrightarrow{c}||\overrightarrow{q}|} \tag{1}$$

where $\overrightarrow{c}$ and $\overrightarrow{q}$ are the vectors for the blog and the query, respectively. The vectors are composed of $tf_{i,c} \times idf_i$ weights where $tf_{i,c}$ is the frequency of term i in blog c, and $IDF_i = \log \frac{N}{n(i)}$ where N is the total number of blogs in the collection, and $n(i)$ is the number of blogs containing i.

—**Okapi BM25**
The features which are based on the ranking produced by BM25 are: `Okapi_10s_count`, `Okapi_20s_count`, `Okapi_30s_count`, `Okapi_female_count`, `Okapi_male_count`, `Okapi_10s_sum`, `Okapi_20s_sum`, `Okapi_30s_sum`, `Okapi_female_sum`, `Okapi_male_sum`, `Okapi_10s_avg`, `Okapi_20s_avg`, `Okapi_30s_avg`, `Okapi_female_avg`, and `Okapi_male_avg`. Similar to the previous featureset, these features compute an aggregation function (average, sum, and count) over the the retrieved results from each gender/age group that appeared in the top-k ranks for the query composed by the keywords in the blog. For this featureset, queries and blogs were compared using the Okapi BM25 score (Eq. 2) as follows:

$$BM25(c,q) = \sum_{i=1}^{n} IDF_i \frac{tf_{i,c} \cdot (k_1 + 1)}{tf_{i,c} + k_1(1 - b + b\frac{|D|}{avgdl})} \quad (2)$$

where $tf_{i,c}$ and IDF_i are as in Eq. 1 $|d|$ is the length (in words) of blog c, $avgdl$ is the average blog length in the collection, k_1 and b are parameters that tune the importance of the presence of each term in the query and the length of the text. In our experiments, we used $k_1 = 1.2$ and $b = 0.75$.

3.1.3 *Readability.* Readability tests indicate the comprehension difficulty of a text.

—**Flesch-Kincaid readability tests**
We employ two tests that indicate the comprehension difficulty of a text: Flesch Reading Ease (`FRE`) and Flesch-Kincaid Grade Level (`FKGL`) [Kincaid et al. 1975]. They are given by Eqs. 3 and 4. Higher FRE scores indicate a material that is easier to read. For example, a text with a `FRE` scores between 90 and 100 could be easily read by a 11 year old, while texts with scores below 30 would be best understood by undergraduates. FKGL scores indicate a grade level. A `FKGL` of 7, indicates that the text is understandable by a 7th grade student. Thus, the higher the `FKGL` score, the higher the number of years in education required to understand the text. The idea of using these scores is to help distinguish the age of the author. Younger authors are expected to use shorter words and thus have a smaller `FKGL` and a high `FRE`.

$$FRE = 206.835 - 1.015\left(\frac{\#words}{\#sentences}\right) - 84.6\left(\frac{\#syllables}{\#words}\right) \quad (3)$$

$$FKGL = 0.39\left(\frac{\#words}{\#sentences}\right) + 11.8\left(\frac{\#syllables}{\#words}\right) - 15.59 \quad (4)$$

3.1.4 *Sentiment Analysis.* These features were extracted based on the NRC Emotion Lexicon [Mohammad et al. 2013], which assigns weights to terms to reflect the type of emotion that they convey. For example, the word *efficient* is considered a positive word associated with the emotion *trust*. Figure 1 shows an excerpt of the NRC lexicon. The words in the text were lemmatized so that they match the canonical form in which the words in the dictionary are found. This process was done with the aid of the NLTK tool[2].

—**NRC emotions**
Ten of our features come from NRC, namely: `positive, negative, joy, surprise, fear, sadness, anger, disgust, trust, and anticipation`.
To calculate the sentiment analysis-based features we simply look up each word from the text into the NRC lexicon. If the word has one or more emotions associated, the scores are added up.

efficacy	positive:1	negative:0	anger:0	anticipation:0	disgust:0	fear:0	joy:0	sadness:0	surprise:0	trust:0
efficiency	positive:1	negative:0	anger:0	anticipation:0	disgust:0	fear:0	joy:0	sadness:0	surprise:0	trust:0
efficient	positive:1	negative:0	anger:0	anticipation:1	disgust:0	fear:0	joy:0	sadness:0	surprise:0	trust:1
effigy	positive:0	negative:0	anger:1	anticipation:0	disgust:0	fear:0	joy:0	sadness:0	surprise:0	trust:0
effort	positive:1	negative:0	anger:0	anticipation:0	disgust:0	fear:0	joy:0	sadness:0	surprise:0	trust:0

Fig. 1. Excerpt from NRC emotion lexicon

[2]`http://www.nltk.org/`

3.1.5 *Correctness.* This group of features aims at capturing the correctness of the text.

—`Words in the dictionary`: ratio between the words from the text found in the OpenOffice US dictionary[3] and the total number of words in the text.
—`Cleanliness`: ratio between the number of characters in the preprocessed text and the number of characters in the raw text. Preprocessing is detailed in Section 4.1. The idea is to assess how "clean" the original text is.
—`Repeated Vowels`: in some cases, authors use words with repeated vowels for emphasis. e.g. "I am soo tired". This group of features counts the numbers of repeated vowels (`a`, `e`, `i`, `o`, and `u`) in sequence within a word.
—`Repeated Punctuation`: this features compute the number of repeated punctuation marks (i.e., `commas`, `semi-colons`, `full stops`, `question marks`, and `exclamation marks`) in sequence in the text.

3.1.6 *Style*

—**HTML tags**: this feature consists in counting the number of HTML tags that indicate line breaks `<br>`, images `<img>`, and links `<href>`.
—**Diversity**: this feature is calculated as the ratio between the distinct words in the text and the total number of words in the text.

3.2 Classifiers

The features described in Section 3.1 are used to train supervised machine learning algorithms. The model learned from the annotated training instances is then used to classify new unseen instances (i.e., the test data). There are several machine learning algorithms available and empirical evaluations [Caruana and Niculescu-Mizil 2006] have shown that there is no universally best classifier – even the best algorithms perform poorly for some problems. Thus, in this work we evaluate a number of algorithms with the aim of choosing the best performers for the task of author profiling.

4. EXPERIMENTS

The experiments performed in our investigation aim at analyzing the quality of the features described in Section 3.1. We also test several learning algorithms and compare our results to the best scoring systems in PAN 2013.

4.1 Experimental Setup

Datasets and Tools. The training dataset provided by PAN is composed of 236K XML files containing blog posts. The test dataset is smaller and has similar contents. Details of the datasets are given in Table I. The gender and the age of the authors is known and this enables us to use supervised classification algorithms. In a preprocessing step, we remove escape characters, tags, and the repeated occurrences of space characters. No stemming or stopword removal was performed. The aim was to keep as much as possible the writing style of the authors. Then the text is tokenized using NLTK.

Once the training data has been preprocessed, the 61 features described in Section 3.1 are calculated. In order do compute the IR-based features, we used Zettair[4], which is a compact and fast search engine developed by RMIT University (Australia). It performs a series of IR tasks, such as indexing and matching. Zettair implements several methods for ranking documents in response to queries

[3] `http://extensions.openoffice.org/en/project/english-dictionaries-apache-openoffice`
[4] `http://www.seg.rmit.edu.au/zettair/`

Table I. Details of PAN's training and test datasets

Class	Training	Test
Female	118297	12717
Male	118291	12718
10s	17200	1776
20s	85796	9213
30s	133592	14446

and has calculated cosine and Okapi BM25. For the readability features, the code available from `http://tikalon.com/blog/2012/readability.c` was used. No normalization or discretization of the values of the features was employed. All instances from the training set were used and we made no attempt to balance the classes.

Classifiers. After the features are computed, we can proceed with training the classifiers. Weka [Hall et al. 2009] was used to build the classification models. It provides a number of learning algorithms divided into seven groups [Witten and Frank 2005].

—bayes: contains bayesian classifiers, e.g., NaiveBayes;
—functions: includes Support Vector Machines, regression algorithms, neural nets;
—lazy: learning is performed at prediction time, e.g., k-nearest neighbor (k-NN);
—meta: meta-classifiers that use a base of one or more classifiers as input, e.g., boosting, bagging or stacking;
—misc: various classifiers that do not fit in any another category;
—rules: rule-based classifiers, e.g., ZeroR; and
—trees: tree classifiers, like decision trees with J48.

The complete list of classifiers used in the experiments is shown in the Appendix. For more information on how these classifiers operate, please refer to a survey by Sebastiani [2002]. Although many of the classifiers have parameters which can be tuned, we used the default parameters.

Evaluation Metrics. By comparing the class of the post with respect to gender and age from the ground truth against the classes predicted by the classifier, we are able to compute evaluation metrics. PAN used *accuracy* as a measure of quality. Accuracy is the ratio of correctly classified instances. A potential problem is that accuracy is not a reliable metric as it will yield misleading results if the dataset is unbalanced (i.e., when the number of instances in different classes vary greatly), which is the case of our dataset for age. Thus, in the results reported here, we also include scores for F-measure, which is the harmonic mean between Precision and Recall. These are computed as:

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN} \qquad F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

where TP, FP, and FN stand for true positive, false positive, and false negative, respectively. For example, let us assume we are calculating the metrics for the *female* class, TP are the female instances which were correctly classified as such, FP represents the male instances which were classified as female, and FN represents the female instances which were classified as male.

4.2 Results

Feature Selection. A widely used method for feature selection is to rank the features based on their *information gain* with respect to the class. Information gain measures the expected reduction in entropy (i.e., uncertainty) considering just the feature and the class. Table II shows the best and the worst features for age and gender as measured by their information gain. These results show that the

IR-based features are dominant among the most discriminative for both age and gender. The number of links (href) has also ranked well for age. Repeated vowels are among the least discriminative features for both age and gender. As expected, a feature that was designed specifically for age is among the most unhelpful for gender (`10s_okapi_count`). However, an attribute that was thought to be useful for age (`20s_okapi_sum`) was found to be a good discriminator for gender.

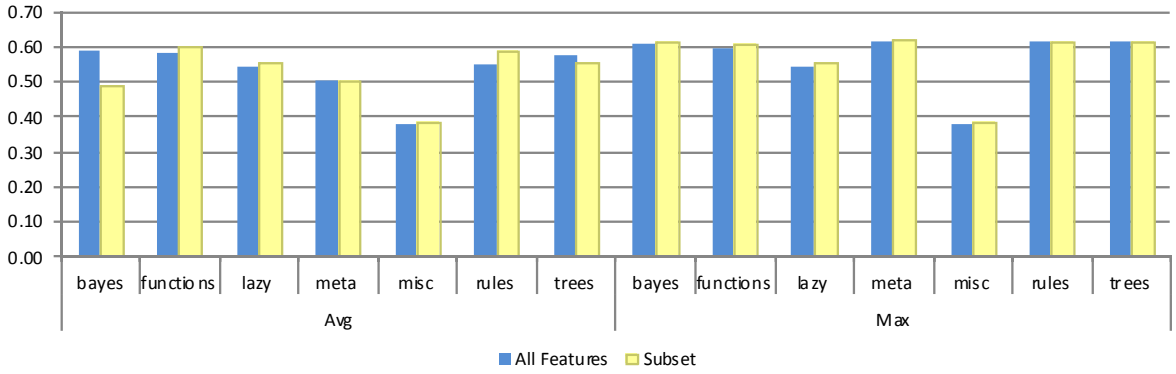
Information gain evaluates attributes independently from each other. However, when the aim is to select a good set of features, we wish to avoid redundant features by keeping features that have at the same time a high correlation with the class and a low intercorrelation. For example, female_cosine_count and male_cosine_count were the best scoring features for gender prediction. However, keeping both of them may be wasteful if they have a high intercorrelation (*i.e.,* if they tend to always agree on the predicted class). With this aim, we used Weka's subset evaluators to select good sets of features. The subset for gender has only six attributes, namely: `female_cosine_sum`, `male_cosine_count`, `male_okapi_sum`, `male_okapi_count`, `<img>`, and `cleanliness`. The subset for age has eleven attributes including `30s_okapi_avg`, `10s_cosine_sum`, `20s_cosine_sum`, `30s_cosine_sum`, `20s_cosine_count`, `30s_cosine_count`, `20s_okapi_sum`, `20s_okapi_count`, `30s_okapi_count`, `href`, and `repeated_e`.

We then compared the predictions of the classifiers run on two settings (i) using all 61 features and (ii) using just the subset of features. The results are shown in Figure 2. For the majority of the learning algorithms, the results of both runs are very close, with a slight advantage in favor of the run with the selected subset of attributes. The only exception happened with the bayesian classifiers for gender, in which using all attributes was better. A paired T-test yielded a p-value of 0.27 for gender and 0.77 for age confirming that there is no significant gain in using all attributes. In this case, using the subset is preferable since the cost in computing fewer features and training is markedly reduced.

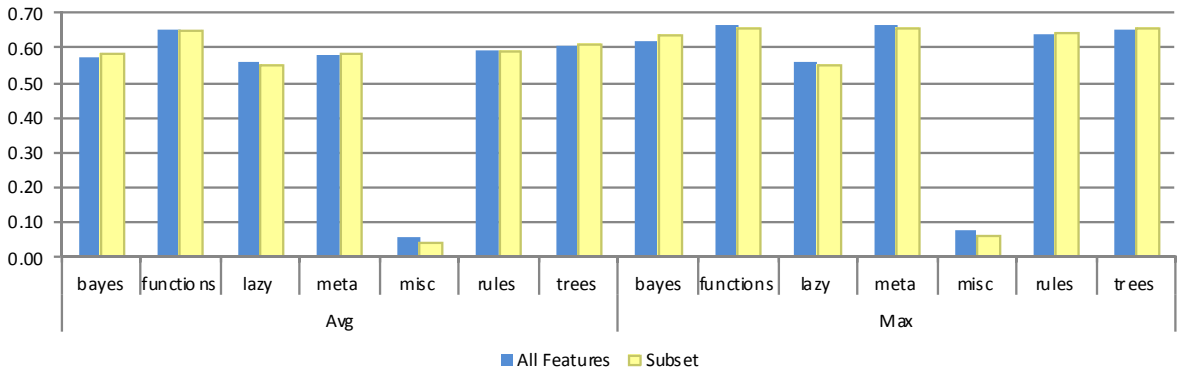
Classifier. Out of the 55 learning algorithms tested for gender, the best scoring in terms of F-measure was *meta.LogitBoost*. In addition, we observed that the results did not have big variations with respect to the type of learning algorithms employed. This can be seen in Figure 2 and Table IV. For age, the

Table II. Best and worst features for age and gender according to their information gain

Best Features			
Age		Gender	
0.089	20s_cosine_count	0.038	female_cosine_count
0.086	30s_cosine_sum	0.038	male_cosine_count
0.083	30s_cosine_count	0.038	female_cosine_sum
0.081	20s_okapi_count	0.033	male_okapi_count
0.077	30s_okapi_count	0.033	female_okapi_count
0.055	30s_okapi_sum	0.023	female_okapi_sum
0.055	20s_okapi_sum	0.019	male_okapi_sum
0.051	20s_cosine_sum	0.018	20s_okapi_sum
0.051	10s_okapi_sum	0.018	female_cosine_avg
0.050	href	0.017	female_okapi_avg
Worst Features			
Age		Gender	
0.013	female_cosine_count	0.001	10s_okapi_count
0.011	repeated_ponto	0.001	repeated_fullstop
0.007	img_html	0.001	img_html
0.003	repeated_exclamationmark	0.000	repeated_questionmark
0.001	repeated_questionmark	0.000	repeated_i
0.001	repeated_a	0.000	repeated_a
0.000	repeated_u	0.000	repeated_semicolon
0.000	repeated_i	0.000	repeated_exclamationmark
0.000	repeated_comma	0.000	repeated_comma
0.000	repeated_semicolon	0.000	repeated_u



(a) Gender



(b) Age

Fig. 2. Maximum and Average F-measures considering all and a subset of features for gender (a) and age (b)

best algorithm was *functions.Logistic* using all features. When the subset of features was used, we had a tie among *meta.ClassificatonViaRegression meta.RandomSubspace*, *functions.MultilayerPerceptron*, and *trees.SimpleCart*. This shows that meta-learners, decision-trees, and function-based learners can all be used for profiling age. Overall, the metalearning algorithms had the best prediction capabilities. For gender, trees and bayesian learners are also amongst the best performers.

Features by Group. Besides analyzing the usefulness of individual attributes, it is also interesting to assess the contribution of the six groups of features described in Section 3.1. In order to do that, we ran the classifier multiple times using two types of settings: (i) removing one group of features each time and (ii) keeping only one group of features each time. The results for these runs are shown in Table III. We used the classifiers with the highest F-measures considering a full set of attributes, namely meta.LogitBoost for Gender and functions.Logistic for Age. The worst results happened when the IR-based features were removed. In addition, we found that by using only this group of features, the classifier has better results than when all other groups were used. Surprisingly, we found that the readability features (which have been widely used in related works) not only do not help but they also degrade accuracy for age. The other groups of features had no significant impact and yield similar levels of accuracy and F-measure.

Comparison Against PAN's official results. We now compare our results with the results obtained by the participants of the Author Profiling task ran in PAN 2013 [Gollub et al. 2013; Rangel et al. 2013]. This comparison is shown in Figure 3, which reports accuracy results. Accuracy was the metric used in PAN and since we do not have access to the results of precision and recall of other participants, F-measure could not be calculated. Our results (Weren 14) outperform the best scoring

Table III. Classification results removing/keeping each group of features

Group of Features		Gender		Age	
		Accuracy	**F-measure**	**Accuracy**	**F-measure**
Without	correctness	**0.621**	**0.614**	0.681	0.663
	IR	0.586	0.580	0.616	0.590
	length	**0.621**	0.613	0.682	0.664
	readability	**0.621**	0.613	0.682	0.664
	sentiment	**0.621**	0.613	0.681	0.663
	style	0.616	**0.614**	**0.683**	**0.665**
Only	correctness	0.552	0.535	0.601	0.550
	IR	**0.616**	**0.614**	**0.680**	**0.662**
	length	0.572	0.571	0.607	0.584
	readability	0.535	0.525	0.589	0.511
	sentiment	0.554	0.533	0.572	0.462
	style	0.573	0.559	0.602	0.577
All		**0.621**	0.613	0.682	0.664

group for both age and gender and consider only the subset of attributes described. This superior results were obtained when using a very small set of attributes (six for gender and eleven for age), which is much fewer than the winner of the competition (which used 311 features for age and 476 for gender). Note that despite gender having only two classes, it was harder to predict than age, which has three classes. Most systems (including ours) had higher scores for age than for gender. This is reflected in the higher values for information gain found for age. Comparing with our PAN2013 results, we noticed that using more IR features and keeping the same classification algorithm (J48) our accuracy increased by 20%. Also, these new experiments have shown that the classifier we used is not among the top scorers for age.

Speed Performance. The speed performance of the learning algorithms is also a very important aspect. Table IV shows the time taken by the ten best classifiers to generate the model from the training data and classify all instances. The best results for time and speed are in bold. As expected, the time taken to build the model using all 61 attributes is much larger than when using just a subset. This is even more noticeable for MultilayerPerceptron for which training takes 40 times longer when using 6 times more features (for gender).

5. DISCUSSION

In this section, we discuss some relevant issues regarding our experimental evaluation.

Choice of features. In our evaluation regarding the different groups of features we noticed that any

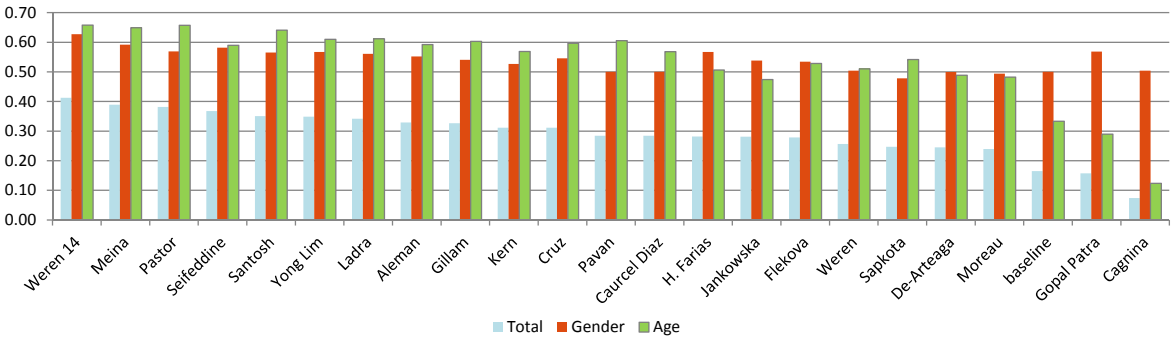


Fig. 3. Comparison against the results from PAN 2013 (accuracy)

Table IV. Speed performance of the ten best classifiers

Gender				
Classifier	**F-measure**		**Time (seconds)**	
	All (61)	**Subset (6)**	**All (61)**	**Subset (6)**
meta.LogitBoost	**0.617**	0.615	18438	1896
meta.MultiBoostAB	0.616	**0.616**	9112	946
trees.J48	0.615	0.606	4718	280
rules.ConjunctiveRule	0.613	0.615	2156	175
trees.DecisionStump	0.610	0.615	**903**	**93**
meta.RotationForest	0.609	0.607	55885	3006
functions.MultilayerPerceptron	0.609	0.597	7936	195
rules.DecisionTable	0.607	0.600	1579	113
bayes.BayesNet	0.606	0.587	984	95
trees.RandomForest	0.606	0.582	2881	1355
Age				
Classifier	**F-measure**		**Time (seconds)**	
	All (61)	**Subset (11)**	**All (61)**	**Subset (11)**
functions.Logistic	**0.665**	0.653	141	**16**
meta.MultiClassClassifier	0.664	0.653	**139**	18
functions.SimpleLogistic	0.663	0.653	12128	988
meta.ClassificationViaRegression	0.661	**0.654**	7050	1314
functions.SMO	0.657	0.649	32148	1097
meta.RandomSubSpace	0.656	**0.654**	586	79
meta.Dagging	0.655	0.649	3356	88
meta.RotationForest	0.655	0.652	58065	7169
functions.MultilayerPerceptron	0.654	**0.654**	10019	457
trees.SimpleCart	0.653	**0.654**	1835	1470

group of features yields F-measures around 0.55. Recall that since this is a binary classification (and the number of instances in the classes is balanced), a random guess would provide a result of 0.5. However, going beyond 0.6 is challenging. A visual inspection of the data shows that male and female instances are intermingled and have very similar scores for our features. This encourages us to keep searching for more discriminative attributes for gender.

Choice of classifiers. After testing 55 classification algorithms, we do not have a clear winner. For instance, gender classification using 30 out of 55 classifiers reach higher accuracy than PAN's winner. This finding corroborates the suggestion by Manning et al. [2008] that when there is a large amount of training data, the choice of classifier probably has little effect on the results and the best choice may be unclear. In those cases (*i.e.*, when the quality is similar), the choice can be made based on scalability. This choice favors *Logistic* for age and for gender both *DecisionStump* and *BayesNet*.

Noisy data. One of the factors which makes this task harder is that our data is noisy. Since the dataset is composed of blog posts, there is no requirement that the contents that are posted are generated by the author of the blog –copying full contents from other authors is a common trend. For instance, we found 11 identical posts and eight of them were tagged as male and three as female. Since we relied solely on the texts of the blog, we did not have any evidence to distinguish different classes in such cases. Perhaps a solution involves looking at other attributes, such as the title and the URL of the blog.

Limitations. Analysing our classification errors, we noticed that most of the instances from the 10s age group have been misclassified as 20s and 30s. The recall for this class was very low (about 3%). This happened because the 10s class had few instances compared to the other two, which introduced a bias in the classification models. Also, as mentioned in Section 4.1, we used the default parameters for the learning algorithms. The results we obtained here could be improved with tuning, which we

hope to do in future work.

6. CONCLUSION

In this paper, we presented an empirical evaluation of a number of features and learning algorithms for the task of identifying author profiles. More specifically the task addressed here was, given the text from a blog, identify the gender and the age group of its author. This is a challenging task given that the data is very noisy.

We performed extensive experiments on a benchmark created for PAN 2013. Their results have shown that IR-based features can lead to results that outperform the state-of-the-art, scoring higher than the winner of the PAN competition. The IR-based features are among the most discriminative for both age and gender. We found that features extracted from a dictionary created for sentiment analysis are not helpful and, surprisingly, the same goes for the readability tests.

Regarding the choice of classifiers, many of them yielded similar F-measures. In this case, speed performance and scalability may be the deciding aspects.

As future work, we plan to try different configurations for the top performing classifiers and also employ methods for selecting the instances for training. It is possible that by getting rid of some of the noise (*i.e.* by selecting the training instances) will enable us to generate models with better prediction capabilities.

REFERENCES

Argamon, S., Koppel, M., Pennebaker, J. W., and Schler, J. Automatically Profiling the Author of an Anonymous Text. *Communications of the ACM* 52 (2): 119–123, Feb., 2009.

Breiman, L. Random Forests. *Machine Learning* 45 (1): 5–32, 2001.

Caruana, R. and Niculescu-Mizil, A. An Empirical Comparison of Supervised Learning Algorithms. In *Proceedings of the International Conference on Machine Learning*. Pittsburgh, Pennsylvania, pp. 161–168, 2006.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science* 41 (6): 391–407, 1990.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. Liblinear: A library for large linear classification. *Journal of Machine Learning Research* vol. 9, pp. 1871–1874, June, 2008.

Gollub, T., Potthast, M., Beyer, A., Busse, M., Pardo, F. M. R., Rosso, P., Stamatatos, E., and Stein, B. Recent Trends in Digital Text Forensics and Its Evaluation - Plagiarism Detection, Author Identification, and Author Profiling. In *Notebook for PAN at Cross-Language Evaluation Forum*. Valencia, Spain, pp. 282–302, 2013.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. The Weka Data Mining Software: an Update. *SIGKDD Explorations Newsletter* 11 (1): 10–18, 2009.

Kincaid, J. P., Fishburne, R. P., Rogers, R. L., and Chissom, B. S. Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel., 1975. Naval Air Station, Memphis, TN.

Koppel, M., Argamon, S., and Shimoni, A. R. Automatically Categorizing Written Texts by Author Gender. *Literary and Linguistic Computing* vol. 17, pp. 401–412, 2003.

López-Monroy, A. P., y Gómez, M. M., Escalante, H. J., Villaseñor-Pineda, L., and Villatoro-Tello, E. INAOE's Participation at PAN'13: author Profiling task. In *Notebook for PAN at Cross-Language Evaluation Forum*. Valencia, Spain, 2013.

Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

Mechti, S., Jaoua, M., and Belguith, L. H. Author Profiling Using Style-based Features. In *Notebook for PAN at Cross-Language Evaluation Forum*. Valencia, Spain, 2013.

Meina, M., Brodzínska, K., Celmer, B., Czoków, M., Patera, M., Pezacki, J., , and Wilk, M. Ensemble-based Classification for Author Profiling Using Various Features. In *Notebook for PAN at Cross-Language Evaluation Forum*. Valencia, Spain, 2013.

Mohammad, S. M., Kiritchenko, S., and Zhu, X. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In *Proceedings of the International Workshop on Semantic Evaluation Exercises*. Atlanta, Georgia, USA, pp. 321–327, 2013.

Mukherjee, A. and Liu, B. Improving Gender Classification of Blog Authors. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Cambridge, Massachusetts, pp. 207–217, 2010.

Nguyen, D., Smith, N. A., and Rosé, C. P. Author Age Prediction from Text Using Linear Regression. In *Proceedings of the Association for Computational Linguistics Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*. Portland, Oregon, pp. 115–123, 2011.

Otterbacher, J. Inferring Gender of Movie Reviewers: Exploiting Writing Style, Content and Metadata. In *Proceedings of the International Conference on Information and Knowledge Engineering*. Toronto, ON, Canada, pp. 369–378, 2010.

Peersman, C., Daelemans, W., and Van Vaerenbergh, L. Predicting Age and Gender in Online Social Networks. In *Proceedings of the International Workshop on Search and Mining User-generated Contents*. Glasgow, Scotland, UK, pp. 37–44, 2011.

Pennebaker, J. W., Francis, M. E., and Booth, R. J. *Linguistic Inquiry and Word Count*, 2001.

Raghavan, S., Kovashka, A., and Mooney, R. Authorship Attribution Using Probabilistic Context-free Grammars. In *Association for Computational Linguistics Conference Short Papers*. Uppsala, Sweden, pp. 38–42, 2010.

Rangel, F., Rosso, P., Koppel, M., Stamatatos, E., and Inches, G. Overview of the Author Profiling Task at PAN 2013. In *Notebook for PAN at Cross-Language Evaluation Forum*. Valencia, Spain, 2013.

Sarawgi, R., Gajulapalli, K., and Choi, Y. Gender Attribution: Tracing Stylometric Evidence Beyond Topic and Genre. In *Proceedings of the Conference on Computational Natural Language Learning*. Portland, Oregon, pp. 78–86, 2011.

Sebastiani, F. Machine Learning in Automated Text Categorization. *ACM Computing Surveys* 34 (1): 1–47, Mar., 2002.

Weren, E., Moreira, V. P., and Oliveira, J. Using Simple Content Features for the Author Profiling Task. In *Notebook for PAN at Cross-Language Evaluation Forum*. Valencia, Spain, 2013.

Witten, I. H. and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.

APPENDIX A. CLASSIFIERS

Table V lists all Weka classifiers used in our experiments.

Table V. Classifiers used in the experiments

lazy.IB1	functions.Logistic
lazy.IBk	functions.MultilayerPerceptron
meta.AdaBoostM1	functions.RBFNetwork
meta.AttributeSelectedClassifier	functions.SimpleLogistic
meta.Bagging	functions.SMO
meta.ClassificationViaClustering	misc.HyperPipes
meta.ClassificationViaRegression	misc.VFI
meta.CVParameterSelection	rules.ConjunctiveRule
meta.Dagging	rules.DecisionTable
meta.Grading	rules.DTNB
meta.LogitBoost	rules.JRip
meta.MetaCost	rules.OneR
meta.MultiBoostAB	rules.ZeroR
meta.MultiClassClassifier	trees.ADTree
meta.MultiScheme	trees.DecisionStump
meta.nestedDichotomies.ClassBalancedND	trees.J48
meta.nestedDichotomies.DataNearBalancedND	trees.LADTree
meta.nestedDichotomies.ND	trees.NBTree
meta.OrdinalClassClassifier	trees.RandomForest
meta.RacedIncrementalLogitBoost	trees.RandomTree
meta.RandomCommittee	trees.REPTree
meta.RandomSubSpace	trees.SimpleCart
meta.RotationForest	bayes.BayesianLogisticRegression
meta.Stacking	bayes.BayesNet
meta.StackingC	bayes.DMNBtext
meta.ThresholdSelector	bayes.NaiveBayes
meta.Vote	bayes.NaiveBayesMultinomialUpdateable
	bayes.NaiveBayesUpdateable