CELEBRITY PROFILING PROGRAM:WHOAMI

Semih Tereci

Department of Computer Engineering

Hacettepe University

Ankara,Turkey

stereci@gmail.com

Ali Alperen İslikaye

Department of Computer Engineering

Hacettepe University

Ankara,Turkey

alialpereen@gmail.com

ABSTRACT

Context: Celebrity profiling is a sub topic of authorship analysis. Authorship analysis is a challenging area that has been developed through centuries and with research done widely scattered across multiple disciples of mainly computational linguistics, text mining, data mining, stylometry and machine learning. Conventional techniques from the past relied heavily on stylometry and text-based content analysis of document text for authorship analysis. More recent developments use network embedding training focus heavily on document attributes to build a network and predict the author. We propose a system that incorporated the strengths of both text mining and network embedding methods and utilizes both the document text and document attributes fully when available.

Objectives: In this paper, we describe the system overview and implementation in detail and discuss how by supporting a more multi-faceted information based network embedding, it can be possible to get improved results. Finally we discuss our results and suggest some future improvements in terms of results, speed and performance for our system to handle larger corpus.

Methods: A sequential exploratory strategy characterized by the collection and analysis of qualitative data followed by the collection and analysis of quantitative data, will be used in this work. To achieve objectives, firstly a thorough systematic literature review has been conducted. After that, the quantitative data obtained was subjected to statistical analysis for the further interpretation of the data and draw results based on it. Also research question methodology used to identify problem in detail and find solutions for questions.

Results: We implement our program for really big corpus from twitter as 8 gb. That’s why when we were trying to implement codes to get some results it had been really long times. All traits that we get from our program WHOAMI are judged by their respective F1, Precision and recall of birthyear are calculated leniently. After these things with some combined metrics we got better predictions.

Conclusions: In this work we address the issue of profiling authors in twitter. We describe in detail the approach used to solve celebrity profiling. We select a set features shown to have relatively high predictive power in terms of age and gender accuracy grouped into content-based and stylistic-based. We collect a big data from twitter. Then, we report the accuracies obtained.

Keywords: Author profiling, text mining, feature engineering, machine learning, network modeling, celebrity profiling

# Introduction

Despite the significant impact of social media on our daily life, we observe a lack of information about those who create the contents. In authorship attribution, we are asked to identify the author of an anonymous text from a closed set of candidate. However,in author profiling, we have to distinguish between different classes of authors, rather than individual ones. Indeed, author profiling can be used to determine the gender, age, native language, or personality type of authors by analysing their published texts. With machine learning algorithms it is now possible to create profile of authors based on their writings. And, nowadays author profiling is of growing importance in a variety of areas, including marketing, security and forensics. In marketing, for example, companies may be interested in knowing the demographics of their target group in order to achieve a better market segmentation.By analysing blogs and online product reviews and using author profiling techniques, they can know what types of people like or dislike their products. And, using those information, they will have a sort of first feedback from their clients and, so, they can adapt their strategy concerning this product. Similarly, author profiling can be used to determine the linguistic profile of the author of a suspicious text which may provide valuable background information useful in the phase of evaluating suspects.

To profile authors, we will classify text documents according to a set of predefined classes, using a machine learning technique. The classification will be performed based on features extracted from the text document. This features will be used later to train the classifier. The classifier will assign classes to new data based on the statistics learned from the extracted features from the trained data.

In this project,we are interested in the problem of author profiling more precisely, we will focus on the prediction of gender and age of authors. We used a naiıve Bayes classifier as a machine learning technique and we have trained it with the PAN@CLEF2019 dataset. We have investigated and implemented features related to author profiling such as content based features and stylistic features.

* 1. Definition

Authorship analysis problems can be classified and defined as follows [1][2]:

● Authorship identification/ authorship attribution: Finding the most likely author(s) for an article or document or the likelihood of an author writing a piece.

● Author profiling or characterization: Finding author profile or characteristics like gender, socio-cultural background and language familiarity, etc.

● Similarity detection: Analyzing the similarity between pieces of work to determine if possibly produced by single author without necessarily determining the author. Used often in plagiarism detection.

The terms “author identification”,” authorship attribution” and author profiling” is often used interchangeably in some cases and have a heavy overlap as author identification usually relies on author profiling and attribution as the first step. Author identification problem can be extended to check for anonymity testing where we check whether the author is fairly anonymous and hard to predict or is predictable and hence makes the author subject to bias sometimes (eg in scenarios of double-blind reviews).

In this paper we primarily explore author identification, but due to overlap in the areas we explore the same in brief below.

# BACKGROUND

Development in this area of author identification/attribution have roots in stylometry and its study. Stylometry is the study of linguistic style often with the objective of author profiling, author attribution and authorship identification/verification.

Stylometry was used as early as the 4th century to determine document authenticity [3]. A notable very recent work from 2015 is where a play “Double Falsehood” was identified as being William Shakespeare’s work [4].

As stylometry paved the path towards author profiling/ attribution and identification, there is one historically famous case for author identification from the 1780s being the authorship of the disputed federalist papers, that has often been used as an example problem for many earlier works in author identification works [5][6] and used often by researchers to further improve and evaluate their methods.

Current author identification techniques go beyond analyzing stylometric features and focus on modeling other document information using network modeling or other advanced data/text mining and machine learning techniques and for digital document use various information like graphics, emoticons, colors, layouts and so on and handles multiple information sources of heterogeneous nature[7][8].

# RELATED WORK

We find a large amount of work in the area of age and gender prediction based on textual data extracted from social media, most of them were performed in response to the PAN 2013 author profiling task[5]. A recent work concerning profiling authors has demonstrated the ability to extract hidden information about authors of social media with accuracies in excess of 80% for attributes such as gender[7] . Works such as these, however, tend to focus on collections of lengthy text posts. Similar work has been done on inferring latent user attributes such as gender, age, regional origin, and political orientation from much shorter social media posts, such as Netlog chat mesages[4] and Twitter microblogs[6] . It is interesting to note that in these works age identification is often treated as a binary classification problem (e.g. distinguishing between users who are below 30 and users who are above 30), while in this work age prediction is defined as a more challenging multi-class prediction problem with three classes (13-17, 23-27, 33-47).

# DATASET

The data used for the training of our system is the shared task dataset of the PAN workshop 2019 that consists of chat-room conversations texts written in both English and Spanish. Each document in this corpus represents of a collection of posts made by a single user. The PAN dataset is formed by 48,335 user profiles and tweets from these profiles that counted 2181 average per user. Novel traits are mainly fame and occupation. New attributes that appended birthyears and nonbinary gender. Training data is splitted two part: input and output json files. That all allocates in memory 8 gb.

# RESEARCH METHODOLOGY

# RESULTS OF STUDY

abc

# VALIDITY OF RESULTS

ass

# DISCUSSION OF RESULTS

ass

# CONCLUSION

In this paper we did bla bla

##### References

1. D. Kester, M. Mwebesa, and J. S. Bradbury, “How Good is Static Analysis at Finding Concurrency Bugs?,” in 2010 10th IEEE Working Conference on Source Code Analysis and Manipulation, Timisoara, Romania, 2010, pp. 115-124.
2. N. Rutar, C. B. Almazan, and J. S. Foster, “A Comparison of Bug Finding Tools for Java,” in Software Reliability Engineering, International Symposium on, Los Alamitos, CA, USA, 2004, vol. 0, pp. 245-256.
3. G. Almasi and A. Gottlieb, Highly parallel computing. Redwood City, CA, USA: Benjamin-Cummings Publ. Co.,Inc., 1989.
4. D. Hovemeyer and W. Pugh, “Finding concurrency bugs in java,” in Proceedings of the PODC Workshop on Concurrency and Synchronization in Java Programs, St. John’s, Newfoundland,Canada, 2004.
5. S. Lu, S. Park, E. Seo, and Y. Zhou,”Learning from mistakes: a comprehensive study on real world concurrency bug characteristics,” in ACM SIGPLAN Notices, 2008, vol. 43, p. 329􀌽339.
6. Y. Ben-Asher, E. Farchi, and Y. Eytani, ”Heuristics for finding concurrent bugs,” in Parallel and Distributed Processing Symposium, 2003.
7. J. R. Ruthruff, J. Penix, J. D. Morgenthaler, S. Elbaum, and G. Rothermel, “Predicting accurate and actionable static analysis warnings: an experimental approach,” in Proceedings of the 30th international conference on Software engineering, 2008, p. 341􀌽350.
8. K. Stobie, “Too Darned Big to Test,” Queue, vol. 3, p. 30􀌽37, Feb. 2005.
9. A. Aurum, H. Petersson, and C. Wohlin, “State-of-the-art: software inspections after 25 years,” Software Testing, Verification and Reliability, vol. 12, no. 3, pp. 133–154, 2002.
10. “IEEE Standard for Software Reviews and Audits,” IEEE STD 1028-2008, pp. 1–52, Aug. 2008.
11. M. E. Fagan, “Design and Code Inspections to Reduce Errors in Program Development,” IBM Systems Journal, vol. 15, no. 3, pp. 182–211, Sep. 1976.
12. M. Ciolkowski, O. Laitenberger, and S. Biffl, “Software Reviews: The State of the Practice,” IEEE Software, vol. 20, no. 6, pp. 46–51, Nov. 2003.
13. O. Laitenberger, S. Vegas, and M. Ciolkowski, "The State of the Practice of Review and Inspection Technologies in Germany," Technical Report ViSEK/010/E, ViSEK, 2002.
14. S. Heckman and L. Williams, “A systematic literature review of actionable alert identification techniques for automated static code analysis,” Information and Software Technology, vol. 53, no. 4, pp. 363–387, Apr. 2011.
15. M. Ciolkowski, “What Do We Know About Perspective-based Reading? An Approach for Quantitative Aggregation in Software Engineering,” in Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement, Washington, DC, USA, 2009, pp. 133–144.
16. V. D’silva, D. Kroening, and G. Weissenbacher, “A Survey of Automated Techniques for Formal Software Verification,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 7, pp. 1165–1178, 2008.
17. M. Ivarsson and T. Gorschek, “A method for evaluating rigor and industrial relevance of technology evaluations,” Empirical Software Engineering, vol. 16, no. 3, pp. 365–395, Oct. 2010.
18. https://en.wikipedia.org/wiki/SonarQube