A brief review of User Profiling Techniques from Textual Data

Georgios Michoulis*
gmichoul@csd.auth.gr
Aristotle University of Thessaloniki
Greece

Styliani Kyrama*
kyrastyl@csd.auth.gr
Aristotle University of Thessaloniki
Greece

Vasileios Moschopoulos*
vmoschop@csd.auth.gr
Aristotle University of Thessaloniki
Greece

Dimitrios Tourgaidis*
tourgaidi@csd.auth.gr
Aristotle University of Thessaloniki
Greece

Abstract

In this era of big data, where there is so much information available, there is a need for personalized recommendations, suggestions that correspond to each user's choices, needs, and interests. For this purpose, building a representative enough profile for each user is considered necessary. Even though the user profiling procedure involves obstacles, it is being used in several real-life applications such as news articles recommender systems or recommender systems in general, user identification applications, etc. As user profiling is a complex task, in this paper we focus only on presenting the existing approaches for the profile building phase exclusively from textual data, while we explain briefly the data collection methods, some commonly used data sources, and processing steps. Furthermore, we explain our approach, in which we plan to utilize the user profiling procedure in order to discover gender biases that may exist, by processing and analyzing profiles of various academics across various European universities.

Keywords: user profiling, user modeling, textual data, data collection

ACM Reference Format:
Georgios Michoulis, Styliani Kyrama, Vasileios Moschopoulos, and Dimitrios Tourgaidis. 2021. A brief review of User Profiling Techniques from Textual Data. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

*All authors contributed equally to this research.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

In this age of "Big Data", where the amount of information available is constantly increasing in every way, Internet users can get lost in this information abundance. Consider the Amazon's website, which consists of nearly 12 million products. A user could be "lost" among all these products, not being able to find exactly what he wants. Yet, Amazon is one of the most popular online marketplaces, and one of the reasons is the ease of recommendations. Being able to recommend to users products similar or correlated to what they have searched before, is a huge advantage for the company, as users get more attracted when the recommendations and offers correspond to their preferences and needs.

From all the aforementioned, we can understand that personalization is necessary to be and remain competitive in an increasingly savvy marketplace, not just a "nice to have". To be able to make personalized offers, recommendations, etc, we need to understand each user regarding his needs, preferences, choices, behavior, and build a comprehensive profile for each one individually.

1.1 User profiling

The existence of profiles for users is the main priority of every information system. Besides e-commerce, that previously mentioned as an example, user profiling has a crucial role in multiple domains such as banking, social media and social networking, healthcare, and cybersecurity among others.

Based on the research conducted in this field, there are numerous definitions of what exactly a user profile is. Some claim that a user profile is a set of certain characteristics of a user, such as demographic information (age, gender, country, etc.), while others define a profile as a narration of the user's behavior, intentions, preferences (obtained via questionnaires), or as information that gives us an insight of user's future needs, and can predict their intentions. A user profile can either be any of the above, or all of them together, as it is something fully related and is inextricably linked to the type and domain of the application in which it will be used.

According to Wikipedia[1], a user's profile is a visualization of personal data, like a person's digital identity. It can also be considered as the computer representation of a user model, i.e. a data structure used to capture some characteristics of the user. The process of obtaining the user profile is called **user modeling or profiling**.

1.2 Main challenges

Obtaining information for a user in order to build a representative profile comprehends obstacles that we need to be aware of.

One of the biggest challenges is the **dynamic nature of the users' profiles**. Users tend to change their information, interests, or even behavior, quite often, while the profiles we build to represent them, are mainly static, or low-level dynamic at best. This may for example lead to wrong recommendations in a recommender system if we don't regularly update users' profiles, as the users' interests may have changed.

Another difficulty, which is also mainly related to the recommender systems mentioned above, is the so-called "cold start" problem; the system doesn't know what to recommend to users for whom it has not collected a sufficient amount of information, so it is not yet able to make personalized suggestions based on user's interests.

In addition to the problems mentioned above, there are also some key points that we need to pay attention to, which mainly concern the way in which we collect the necessary data-information for users. Users interact with many different systems, whether these are search engines, social media, and news sites, among others. We, therefore, note that there are multiple sources from which we can, and should, gather the necessary information in order for the user profile to be complete and representative, but keeping in mind that not all information is useful. However, we must always respect the privacy of users, not violating their rights, and always comply with the established regulation of ***GDPR***[2].

1.3 Outline

The rest of this paper is organized as follows. In section 2, we intend to present some basic information about the user profile, including the categorization of data collection methods and profile types, among others, concepts that we are going to use, and on which we will rely, in later sections. In section 3 we present the techniques used by the authors to extract textual data from specific sources, in order to create their dataset, but also the overall processing of the data. Section 4 is dedicated to the user profiling techniques and approaches that have been proposed in the last years, and are a novelty in the relevant field, while section 5 presents conclusions drawn for the methods discussed in more detail in the above sections, mainly at a theoretical level, but also in practice, by comparing their results for specific applications, to the extent possible. Last, in section 6 we briefly describe the approach we are going to follow for building profiles of academics, in order to study the relevance of gender with the evolution of a person in academia.

[1] User Profile [Wikipedia]
[2] General Data Protection Regulation

2 Background

User profiling may seem like a simple and straightforward task, but it is actually a complex process that consists of many basic steps. The whole process is presented in Figure 1, which illustrates the fundamental sub-tasks that must be performed in order for the process to be considered successfully completed. Although in this paper we focus on presenting exclusively the modeling techniques and approaches, in this section we are going to explain further the previous subtasks in the pipeline, exhibiting in the meantime some of the key concepts we are going to use later.

A great amount of research has taken place in the field of user profiling, leading to the solidification of abstractions, which contribute to the exploitation and comprehension of the field. These abstractions concern *users' profile types*, *profile modeling types*, *information collection methods* and *profile filtering*. All the aforementioned abstractions are analyzed to a greater extent in the following subsections, as well as their correlation.

2.1 Data Collection Methods

A crucial task in user profiling regards the techniques used, in order to extract users' information. As mentioned in section 1.2, there are multiple sources from which one can obtain information for each user, such as Twitter, or other social media, online forums, etc. The information can be gathered *explicitly*, *implicitly*, or using both methods combined (*hybrid*).

Explicit collection. In these methods, explicit information that is asked from the system, and provided from the users, is utilized to build users' profiles. This can be the data that is provided in online forms, from which, information regarding users' name, address, phone, job status, hobbies, and interests, could be extracted. Furthermore, users' textual or non-textual reviews on items, that are explicitly asked from the users, could also be considered as explicit information. Explicit information collection methods are considered insufficient, due to the fact that users prioritize their privacy, a truth that prevents the users to share their personal information as public.

Implicit collection. Contrary to the previous techniques, in implicit collection methods, information about users is extracted from their implicit feedback to the system. For example, online transactions and browsing history among others, which are implicit feedback from users, could be used in order to build users' profiles. For those cases, data

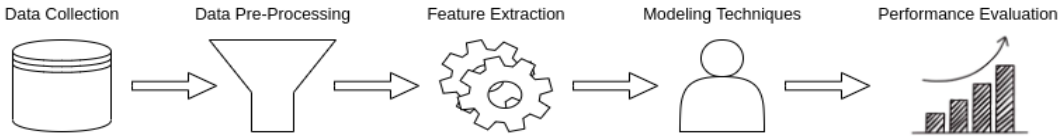

Figure 1. Pipeline of user profile modeling

mining techniques and machine learning models are used to apprehend users' patterns and interests. Implicit information collection methods are a more efficient approach, compared to explicit ones, because in order to extract information further effort is not needed from the users, as for example filling forms. On the other hand, a drawback of these methods is that due to the machine learning models and in order to build an efficient user profile, users must have interacted with the system adequately, contrary to explicit methods where there is no such requirement.

Hybrid collection. In hybrid methods, the two previously mentioned approaches are combined, in order to collect as much data as possible for each user, and exploit every available source, for the profile to be more complete and representative. Usually, both explicit and implicit methods are used for users' information collection, so we can conclude that hybrid collection is the most widely used technique.

2.2 Profile Types

An important distinction in users' profiles concerns their type. User profiles are distinguished into *static* and *dynamic*, depending on the technique used to build them and whether they get updated regularly, occasionally, or never.

Static Profiling is the procedure of collecting and analyzing users' static and predictable data, usually provided by them. A static profile gives an immediate apprehension of users' interests, but it comes with some drawbacks. Due to the fact that a static profile contains information obtained only once, and not got updated, we can say that it lacks quality over time. Thus, the profile may provide inadequate, or even outdated information about a user, due to its static nature, leading to wrong predictions and recommendations. Static profiles are mainly built with explicitly gathered information.

Dynamic Profiling on the other hand, regards the constant evolution of users' profiles, based on their continuous inputs over time. Users' inputs are captured and analyzed in real- or near-real- time, leading to immediate and automated user profile updates. Dynamic profiling leads to a more substantial and accurate understanding of users' interests, due to the exploitation of more data, as well the constant adjustment of user's profiles, in response to users' altering of interests over time. A dynamic profile is usually related to and created using implicitly collected information, as it is easier to obtain and does not require continuous explicit interaction with the user, demanding from him to participate in the profile update procedure.

2.3 Modeling Types

In user profiling, the main point of interest is to interpret and model the different aspects of a user's character. Consequently, user profile modeling is distinguished into three types: *Behavioral*, *Interest* and *Intention* modeling.

Behavioral Modeling concerns the apprehension of user's behavior patterns in a system. The intentions of the users' actions are interpreted based on their previous ones, in order to acquire a global picture of a user's general behavior. The actions of the users are observed, leading to constant updates of their profiles and therefore, behavioral modeling is exclusively correlated with dynamic profiling.

Interest Modeling attempts to quantify the degree of a user's interest in regard to material and non-material assets. In order to form a user's interest model, implicitly and explicitly collected information may be used, and consequently, we can say that can be related to both static and dynamic profiles.

Intention Modeling tries to comprehend users' intentions behind their actions, with the objective of understanding the purpose of the user's interaction with the system. Users can be grouped into categories, based on the intentions of their actions. An example would be categorizing users of an online casino into two categories, the ones who are willing to spend a lot of money and the ones who are not. Intention modeling relies on behavioral and interest modeling, building its form solely on top of them.

2.4 User Profile Filtering

Before using any more specialized techniques for the user profile building process, it is considered necessary to clean and pre-process the data so that it is in a convenient form for the type of application to be created. One of the steps is that irrelevant information for users must be removed, and this could be achieved through filtering techniques. It is worth noting that there is not just one way to filter data, and also, that different filtering techniques are used in different situations, depending on the type of modeling used. Filtering techniques are divided, based on how they work, into:*rule-based*, *content-based*, *collaborative-based* and *hybrid* filtering, and are described in more detail below.

Rule-Based Filtering is an approach that relies exclusively on "if-then" rules, which are used to select useful

information for each user, from the gathered ones. These rules may be rules exclusively written by experts or rules created by a system in a more automated way, using a system for rule generation, based on the input information currently available. The effectiveness of this method depends on the quality of the rules, and also the knowledge they provide. The main drawback of the rule-based filtering approach is that it cannot be maintained or expanded easily.

Content-Based Filtering hypothesizes that a user behaves the same under the same conditions. Users' current behavioral patterns are calculated based on their past information. Those type of filtering methods, are heavily combined with explicit information collection techniques. Hence, content-based method calculates the content correlation of the elements with the users' profile, returning the elements with the highest content correlation score. A substantial drawback occurs in case where the elements' content is limited, therefore, they cannot be analyzed. Furthermore, those methods adapt too much in past behavior, leading to correlate users with elements of the same type, excluding elements of other types that users may be interested in. Moreover, as mentioned in previous sections, users tend to hide explicit information.

Collaborative Filtering is a technique, in which users are grouped into categories, based on their personal characteristics (e.g. age, social class, hobbies, interests). This type of filtering works under the assumption, that users belonging to the same group have similar characteristics. Consequently, if a user is correlated to an element, and there exist other users that belong to the same group as that user, then those users will be correlated to that element as well. Collaborative filtering has two drawbacks. Firstly, the sparsity problem [1], which occurs when a new element is introduced, and derives from the fact that not much analysis has occurred in respect to that element, in order to correlate it to specific groups. Secondly, the cold-start problem [2] exists, which is caused when users' profile cannot be built, due to lack of data, and, therefore, cannot be categorized into groups.

Hybrid Filtering is a method that combines both contend-based and collaborative filtering methods, in order to leverage the advantages of both and exceed their drawbacks. The hybrid approach guarantees that a user profile will be assigned to a user from the beginning, and in general, the efficacy regarding users' profile building exceeds the limitations of the two discrete types of filtering methods.

3 Data Collection and Processing

In this section, we describe the novel web mining and data processing techniques of our literature review, which use textual data for the user profiling phase. This section is further split into two subsections, based on the origin of the data. Hence, we categorize the papers in literature into Twitter-based and web-services-based. In this survey we focus only on implicit data collection methods, because as we mentioned in section 2.1, implicit methods are more often but also more efficient.

3.1 Twitter-based data

This subsection is dedicated to works using Twitter as their data source. Twitter is one of the most popular social media platforms and its easy-to-use API has led many people to harness the large and diverse amount of data it can provide. Due to this, many of the applications in the literature were found to use Twitter-based data (e.g., tweets, user's info, hashtags).

In 2013, **Liu et al.** [3] introduced their work with the aim of how to incorporate a user's name into a gender classifier and how the quality of the inferred labels is improved. The authors developed their own canonical gender-labeled Twitter dataset. A novelty of the paper is the use of the Amazon Mechanical Turk workers to classify the gender from the users' profile picture (male, female and unknown). Following the labeling of the users, the authors collected the most recent tweets generated by each user. Those tweets are combined with the user profile information in order to create the user's textual content. Afterwards, the authors validate the quality of their dataset with the real Twitter statistics and other similar datasets (e.g., Burger).

A more recent approach is the one of **Guimarães et al.** [4]. The authors address two tasks: **i)** they aim to measure the importance of the age group information in sentiment analysis, and **ii)** they build a machine learning algorithm to quantify the effect of a predicted age group in sentiment analysis when there is none provided.

For the first task, the authors queried 76 assessors, with a variety in regions of birth, age, educational level and gender, and split them into two groups; teenagers and adults. The authors asked the assessors to provide a number of tweets, in two subjective tests, providing in the meantime the estimated sentiment scores of their tweets, ranging from -5 to 5. This way, the author collected 6,387 tweets, along with their estimated sentiment target scores.

For the second task, the authors used the Twitter API to collect 6,280 tweets, equally split into 7 different topics. Using these tweets, the authors then built a dataset containing 13 different features. Features were split into three categories, with 7 features extracted from tweets, 5 features extracted from tweet authors' profiles, and 1 feature being the age group.

Another interesting approach of this category is the one presented in **Cui et al.** [5], in which the authors designed an automatic process which use the reported location of the tweets to assign Location–activity labels (e.g., user is traveling, dining etc).

Their model is used to study the followers' behaviors from a number of well-known Twitter users, with the results

suggesting that the findings it provides are highly related to the predicted characteristics of these accounts.

Firstly, using the Google Map API, they collect specific places and more specifically, their detailed coordinates, focusing on specific categories and activities. These coordinates were used to collect tweets from the Twitter API. Moreover, they kept only the tweets which indicated that an activity may be concluded at a preferred location. Afterwards, they normalized, cleaned and removed hashtags signs from the dataset. Due to an imbalanced dataset, they used different weights for each class. Finally, randomly was the division of the training, the development, and the test sets in the ratios of 6:2:2.

Moreover, user profile building is a popular topic in recommendation systems. **Alshammari et al.** [6] proposed a recommendation system that constructs users' profile by utilizing Twitter. The system's objective is to suggest tweets to the users, that are more likely to be re-tweeted by them. In order to build the users' profile, the authors extract information from the content of the users' personal tweets, as well the content of their friends' tweets.

The content of the tweets that has been selected, is preprocessed, based on the work suggested by Micarelli et al. [7], resulting to a key-word representation (bag-of-words) of it, that afterwards can be used to start the recommendation phase.

In order to decide, which tweets are going to be selected to construct the users' profile, both user tweets and friends' tweets are used to quantify the influence scores of the users, in correlation to their friends. Hence, the level of influence each friend of every user has on that user is calculated. Features like re-tweeting, following, favoring, and replying are used to quantify the influence scores. Consequently, the final content that is collected from the users' friends to build the users' profile is in reference to the influence scores.

After the calculation of the influence scores, the friend list of the users are split up into 3 groups, using the K-means clustering algorithm; influential users, less influential users, and non-influential users. Regarding the user profiles construction, all the tweets of the influential users are used, where on the other hand the tweets of the non-influential users are not used. As for the tweets of the less influential users, they are categorized even further into representative (re-tweeted by the user) or not representative (not re-tweeted by the user). Then, the representative (re-tweeted) tweets of the less influential users are added to the tweets of the influential users, which now constitutes the final dataset that will be used to construct the users' profile.

Similar to the above solution, is the **Lee et al.** [8]. The authors built a news recommendation system based on users' profiles, and extracted the information they needed, using the Twitter API. For each one of them, they collected all the tweets and re-tweets they posted for a specific time interval, in order to have a satisfying amount of tweets for a more representative profile. More specifically, they studied the behavior of eight Twitter users, who agreed to help in their research, and calculated that the average user writes about 46 tweets in a month. For the purpose of having enough information for each one, they collected their tweets for about three months.

The tweets are being processed using the following procedure. Firstly, all the unnecessary information is being removed from the tweets (like emoticons, links, and metadata included), while retaining the hashtags, as they count them as valuable words. Since not all words are going to be used for the user modeling phase, some of them are removed completely (like adjectives, etc.). The final step of preprocessing is to collect from the remaining words only the nouns, giving also emphasis on the hashtags added by the user (words that are mentioned by the user right after the special character"#"). Hashtags are being kept because they provide useful information for the one who used them, by revealing whole topics this user is interested in.

Thus, they create a bag-of-words, having only nouns and hashtags, for each user. It is important for the profile to be representative enough of the user, in order for the recommendations of the news articles to be successful.

Despite the fact that the papers presented above contributed greatly on their domain of research, none of them studied textual features extraction methods in as much detail as **Pietro et al.** [9]. The research of this paper aims to identify the most likely job class for a given user based on their Twitter profile and a variety of textual features. Pietro et al. were the first who studied this topic and therefore had to create a brand-new dataset. Firstly, they adopted the 4-digit UK Standard Occupational Classification system [10] in order to create 9 sets of occupations. The authors extracted data from the description of Twitter users and transformed them into the 4-digit system. Furthermore, the authors divided the dataset into two types features:

- The UserLevel. These features were based on the general user information or aggregated statistics about the tweets [9] (e.g., number of followers or friends, proportion of retweeted tweets).
- The Textual. These are extracted from more than 10 million tweets of the collected Twitter users. Also, their novel reference corpus is required in order to represent each user as a distribution over these features.

The authors used more than four methods to obtain the textual features. The methodology they used was the word-to-word similarity that was based on the Normalized Pointwise Mutual Information (NPMI). NPMI is an information theoretic measure indicating which words co-occur in the same context, where the context is represented by a whole tweet [9]. On each method they experiment from 30 up to 200 dimensionalities for the features. So, the four methods can be summarized to the following:

1. SVD-Embeddings. The word-to-word similarity matrix is decomposed using the singular value decomposition to achieve a low-dimensional embedding of words. Each user's function representation is calculated by adding all the embedding dimensions in all words.
2. SVD-Classifier. The NPMI matrix creates hard clusters of words which represent the "topics". Then, they use the spectral clustering because it is based on applying SVD to the graph Laplacian and performs a better graph partitioning on the NPMI similarity matrix. Those clusters are preferable since a list of the most common or representative words may be used to interpret them.
3. W2V-Embeddings. This neural network is used with a skipgram model, with negative sampling, to learn word embeddings on the Twitter corpus and it factores a word-context PMI matrix.
4. W2C-Classifier. The neural embeddings are used similarly to the above, that is to obtain clusters of related words. The word-to-word similarity matrix is formulated with the cosine similarity method on the neural embeddings.

3.2 Web services

In this section we present works in the literature that use various web services as their data source. Web services have been proven to be a common vessel to extract users' data, due to the advantages which emerge from the implicit information collection methods that can applied to them, which we examine in this section.

In **Misztal-Radecka** [11] the author applies two profile modeling approaches, for users of the Polish web news service Onet, using recently browsed articles. For training the user profiling models, the author used two corpora, a custom and an external one. The custom corpus was built on 500,000 texts of Onet articles that had page-views within a two-week period and covering a wide range of topics. The external corpus was a combined corpus of the Polish Wikipedia corpus and the National Polish Language Corpora NKJP.

For user profiling, the author used the information of 103,519 anonymous registered users, split into two nearly equal gender classes, and built representation vectors of their profiles using articles they browsed within the two-week period.

The author used 6 different model variations for user profiling. The first 5 of the models were trained on the custom corpus, while the final one was trained on the external one. Data preprocessing for the first 4 models included tokenization, stopword removal and lemmatization. While for the last 2 models, only tokenization and stopword removal was applied, hence, maintaining all word forms.

A more recent approach in the literature review, using different techniques, is from **Manoharan et al.** [12]. The authors propose an inference system that intends to understand users' fields of interest, in order to recommend them viral articles from Twitter and Facebook. To achieve the aforementioned, implicit users' profiles (IUP) are analyzed, which are constructed by utilizing users' browsing history. Hence, visited URLs and search engines queries are analyzed to apprehend users' fields of interest.

More precisely, based on the two aforementioned features, two metrics are used; click frequent count (CF) and specific search query count (SSQ). For the former, the clicks for each category are counted, based on users' visited URLs, using Directory.Mozilla.Org (DMOZ). As for the latter, user searches on the search engines for each category are counted, using clustering techniques. The objective of calculating and using those two metrics, is to understand the user's level of interest in a category, constructing this way the user profiles.

4 User Profiling Techniques

In the previous section we described in detail all processes taking place before the modeling procedure. In this section, we examine user profiling techniques presented in the existing literature, and thus build the user profiles. Figure 2 presents a more general and complete picture of the existing user profile building techniques, but as the vast majority of the examined work consists of content-based techniques, we will focus on this kind of techniques for the most part. We divide the existing section in the following two subsections: **i)** Content-Based which is further split into **a)** Vector-Space Model and **b)** Machine Learning, and **ii)** Rule-Based.

4.1 Content-Based

As discussed in 2.4, content-based techniques assume that a user's behavior is the same under the same conditions. Content-based techniques rely on provided information in the past, often too much, in order to profile users, therefore regularly failing in terms of novelty in tasks such as recommendations. Content-based techniques are strongly correlated with Vector-Space Model and Machine Learning profiling algorithms.

Vector-Space Model. Vector-Space is a statistical, generative type of techniques and the most common type of user profile representation. Vector-Space Model techniques use vectors of weighted terms to describe user interests, hence they model the interest distribution of users.

In **Misztal-Radecka** [11], the author attempts to represent user profiles of the Polish web news service, Onet, using Word2Vec and Latent Dirichlet Allocation (LDA) topic modelling embeddings, based on recently read articles by users, within a two-week period. The aim of this work was to evaluate the usefulness of different user profile representations in a news service context. To achieve this, the author defines two auxiliary tasks; **i)** a qualitative task for evaluating similar article retrieval and, **ii)** a quantitative task of user gender

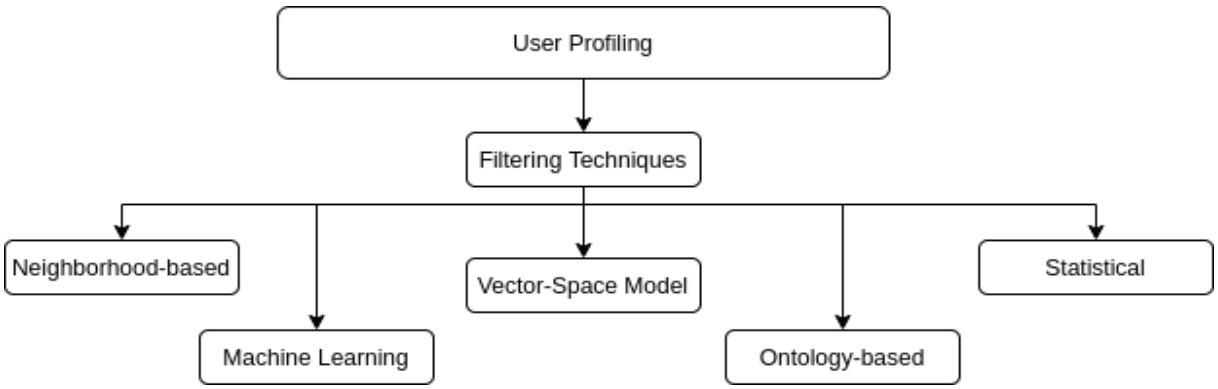

Figure 2. Profiling Techniques Classification Schema

prediction. In addition to this, the author attempts to answer a variety of research questions to discover the characteristics of the optimal model for these tasks.

The author used two corpora for training the models, an external and a custom one. The author evaluated 6 different models. Five models were trained on the custom corpus, two using the LDA method and three using Word2Vec embeddings, while the sixth model was trained on the external corpus. The first four models were applied on either the article text or the title, alternately, while the last two were applied on the article text only.

To evaluate the user profile representation ability of the models, the author used the information of 103,519 anonymous registered users, split by gender into two nearly equal classes of men and women. User profiles were described by the average of the representation vectors of their browsed articles, within the 14-day period. Due to the property of the LDA topic modeling algorithm of describing an article using a topic probability distribution vector, representation vectors of user profiles using LDA can be interpreted as the distribution of a user's interests.

Finally, the paper results in several conclusions concerning the optimal representation vector size and efficacy of each model in the tasks of similar article retrieval and gender prediction.

It is worth noting that the approach presented above concerns a specific application, such as a recommender system for news articles, but it is not the only approach for this type of application. Some of the existing techniques related to this focus mainly on obtaining information from the actual news content that users read, such as headlines, texts, and generally, click-through data. Contrary to that, there are few approaches that try to make such personalized recommendations without using information from the users' previous choices in news, like the one presented below.

In **Lee et al.** [8] the authors suggest a technique for personalized news recommendation, that relies mainly on the analysis of the content related to users. In order to find the most suitable news articles for each user, Won-Jo Lee et al., proposed a two-phase procedure that consists of i) a user profiling phase and ii) news ranking phase. Although many approaches exist in this field, that utilize user profiling in order to make news recommendations, this specific approach differentiates from the previous ones, which relate a user only with his choices in news. On the contrary, in this technique, user profiling is mainly based on Twitter, from which user tweets and re-tweets are obtained, and exploited to build a representative profile.

In more detail, authors extracted some useful key-word from each one of the collected tweets, as well as the hashtags written by the user, creating a bag-of-words. A number is assigned to each word, which reflects the frequency of this word among the user's tweets. Since hashtags are considered more important key-words, due to the fact that they are specifically selected by the user, the frequency assigned to them is higher than the usual. Nevertheless, in order to have a fair contribution of each key-word, but complementary a fair comparison of topics and also users, they authors applied normalization.

The authors used a formula that creates the user's profile, based on all the data that were collected. Thus, for each one, they created a vector of words, in which vector, each word was associated with a variable that specified its weight, i.e., the importance for the user. This weight is the *TF-IDF*[3] score for this word and for this specific user, and since the weights are normalized, they vary between 0 and 1. The formula which the authors used written in a more mathematical way is

$$P(u) = \{(Word, Weight) | Word \in \{words, hashtags\}, \\ 0 < weight \leq 1\} \quad (1)$$

To better understand, consider having 4 different users and some words like "Apple", "Samsung", "Google", etc. Depending on the user's preferences, some of these words will be more often mentioned by him, while others not so often.

[3] Tf-Idf [Wikipedia]

Thus, the importance of the word "Apple", for example, will vary among users, but also considering a specific user the importance of each word will vary from the other ones.

Have we wanted to visualize this into a table, we would have as many columns as the different words found, and a user to be represented as a row. Hence, each row would be a vector of weights, with each weight corresponding to a different word, with words that have never been mentioned by a user to have zero weight for the user.

Having a representative profile for each user gives the opportunity to make comparisons among users, and find correlations and similarities between them, before moving to the news ranking part and the recommendations. Especially with this representation, a simple similarity measure such as cosine similarity can give us quite good insights about the correlations between users.

The above recommending system approach uses exclusively user tweets to recommend news based on the user's interests. However, this is not the only type of recommendation system used for Twitter. Some approaches use user data and as well as friends' data in order to recommend new tweets.

In **Alshammari et al.** [6] the authors try to suggest tweets to the users, that are more likely to be re-tweeted by them. Consequently, in order to evaluate the similarity of the suggested tweets with the users' profiles, both users' profile and the suggested tweets, are declared in a vector space representation to calculate their cosine similarity. The smaller the angle is, the more relevant the suggested tweets are to the users.

In this case, the tweets that are used to construct user profiles are the aggregation of users' tweets, tweets of their influential friends and re-tweeded tweets of their less influential friends. Based on those, 6 different user profiles are built. Each of the 6 user profiles is constructed using on a different combination of types of tweets. The 6 models are compared with each other, in order to evaluate the most accurate, which constitutes the final user's profile that will be used to be vectorized. Eventually, the selected user profile vector will be used to calculate the cosine similarity between the user profile and tweets, for the recommendation to take place.

The metrics that are used to compare the models are, firstly, average precision (AP), which is used to measure the ability to retrieve the top-k most relevant tweets, and secondly, mean average precision (MAP), which is used to measure the ability to retrieve all the most relevant tweets. The results showed that the model that is constituted from tweets of a two-week's time-frame, has the best performance in both metrics.

Machine Learning. Machine learning techniques utilize Machine Learning algorithms to model user behaviour and interests. Examples of such techniques involve Decision Trees, Naive Bayes classifiers, K-Nearest neighbours, Gaussian Processes, K-means, as well as Artificial Neural Networks (ANNs). In contrast to Vector-Space Model techniques, Machine Learning techniques are discriminative, hence, they learn to classify users into different classes, instead of representing them with a vector.

Supervised Machine Learning algorithms are a class of algorithms that use labeled data, meaning for each sample in the dataset the expected output is known beforehand. Supervised Machine Learning algorithms are trained to model the relationship between sample features and output variables as accurately as possible, with the goal of correctly predicting the expected output in unknown, unforeseen data.

In the context of implementing the Word2Vec neural network model, **Misztal-Radecka** [11] can also be categorized as a Machine Learning technique. Despite the interpretability benefits of the LDA method for user profile representation, Word2Vec modelling was found to generally perform better in the task of similar article retrieval, as well as when applied to the article title only.

Guimarães et al. [4] introduce a conjoined study consisting of two different tasks on data collected from Twitter: **i)** training Machine Learning algorithms to predict a tweet's author age group, and **ii)** measuring the importance of age group information in sentiment analysis.

To measure a tweet's sentiment score, the authors utilized two metrics; the Sentimeter-Br2 score, which does not take user profile information into consideration, and the eSM score, an extension to the Sentimeter-Br2, which does. The dataset for training their Machine Learning algorithms was collected through the Twitter API and contained 6,280 tweets on 7 different topics. The authors trained 5 different models; a Multi-Layer Perceptron (MLP), a Convolutional Neural Network (CNN), a Decision Tree, a Random Forest and a Support Vector Machine (SVM). Out of all 5, the authors selected the CNN, which produced the best results overall. eSM proved to be better at predicting the sentiment score of tweets, since it takes into consideration user profile information. Using the age group as a feature produced a significant improvement in predicting the sentiment score, while using the predicted age group from the CNN produced similar results when compared to using the real age group.

Similar to the previous approach is this of **Liu et al.** [3]. This approach aims to assign a gender-label by the user name, using Twitter profiles. The authors used three methods which use SVM classifiers.

In the second one, categorized as an integrated classifier, the gender-association score of a user's name was added to the user's feature vector as a separate feature. This gender-association score is the division of the subtraction of the times the name is given to a male with the times the name is given to a female with their sum [3]. The drawback with the second method is that the gender-association score as a feature receives a uniform weight in the classification problem.

Paper	Profile filtering	Profile type	Modeling type	Profiling technique	Dataset source	Application
[11]	Content-Based	Static	Interest	Vector-Space Model	Onet	Gender prediction
[4]	Content-Based	Static	Behavioral	Machine Learning	Twitter	Age group classification / Sentiment analysis
[12]	Rule-Based	Static	Interest	Expert rules	DMOZ	News Recommendation
[6]	Content-Based	Static	Interest	Vector-Space Model	Twitter	Tweets Recommendation
[5]	Content-Based	Dynamic	Behavioral	Machine Learning	Twitter	Activity Recognition
[3]	Content-Based	Static	Interest	Machine Learning	Twitter	Gender prediction
[9]	Content-Based	Static	Interest	Machine Learning	Twitter	Job classification
[8]	Content-Based	Static	Interest	Vector-Space Model	Twitter	News Recommendation

Table 1. Categorization of user profiling literature.
All of our reviewed literature was based on Implicit data collection methods.

The third method utilizes a threshold that uses the gender-association score of the user's name to know if it needed or not to use the SVM-based classifier. This method is setting a threshold to the gender-association score and, if the score is below the threshold, only then the SVM is applied to the user. The main idea is that the higher the gender-associated names occur in population, the lower the choice of threshold can be.

In the baseline method, which is the first one, name information was omitted entirely, and a radial basis function was used for the SVM kernel. Due to the fact that the aim of their research was to determine the incremental value of using the user's name as a feature in gender inference, the authors had to choose nine different features for their SVM classifiers. The features for this (first) algorithm were the following:

- K-top words. The k most differentiating words used by each labeled group were included as individual features [3].
- k-top stems or k-top co-stems. The co-stem is obtained from the word minus the stem.
- k-top diagrams or k-top trigrams.
- k-top hashtags. Hashtags operate as topic labels and thus they can statistically align with gender labels.
- Frequency statistics (tweets, mentions, hashtags, links, and retweets per day).
- Retweeting tendency.
- Neighborhood size. This is the number of followers divided by the number of friends has been used to gauge a user's proclivity for generating rather than consuming content.

Finally, The highest average accuracy derived from the "threshold - 0.85" algorithm, because most names are either strongly associated with a given gender or unknown, and so, the gender-name associated score gets high scores.

In **Pietro et al.** [9] implemented the Gaussian Processes (GPs) for the classification task, using Twitter content data to identify the most likely job class of Twitter users. GPs formulate a Bayesian non-parametric machine learning framework which defines a prior on functions. GP methods aim to learn a function drawn from a GP prior. In binary classification the results are a non-Gaussian likelihood in the posterior function and therefore, they used Expectation Propagation which approximates the non-Gaussian joint posterior with a Gaussian one. Moreover, due to scalability problems it is required to use the fully independent training conditional approximation with 500 random inducing points. Finally, they perform a separate one-vs-all classification for each class and then determine a label through the occupational class that has the highest likelihood, in order to provide insight.

The authors compared the accuracy of their GP algorithm with the accuracy of the SVM with RBF kernel and the Logistic Regression with Elastic Net regularization algorithms. In most feature methods and dimensionalities, the GP's accuracy performed better than the other two algorithms. It is noticeable that the embeddings have lower performance than the clusters, while the W2V features show better accuracy than the SVD on the NPMI matrix.

Deep learning is a sub-field of machine learning that structures algorithms in layers to create an "artificial neural network" that can learn and make intelligent decisions on its own. With a deep learning model, an algorithm can determine on its own if a prediction is accurate or not through its own neural network. Also, deep learning algorithms use a higher volume of data than classical machine learning techniques. In this section, a case of a Long Short-Term Memory (LSTM) approach is presented. An extension to the Recurrent Neural Network (RNN), LSTM is deep learning algorithm

that deals with a stream of data, such as a sequence of words in a tweet.

An implementation of an LSTM-based algorithm is presented in **Cui et al.** [5]. Their LSTM algorithm aims to cover a wider range of contextual features by utilizing different modeling layers for different contextual features. Firstly, they initialized some specific tools such as, the use of transfer learning (GloVe [13]) for the LSTM embeddings with the tweet content. Then, they used POS embeddings which were randomly initialized and generated using a tweet-specific tagger. With four-time intervals in a day, post time was described as a day of the week and time of the day.

Their HD-LSTM takes text input, along with contextual features of historical information (tweets of the same author each time), POS tag sequence, and post time, in the form of embeddings. Furthermore, by concatenating their embeddings, each tweet content sequence and the direct contextual features are combined. The same type of tweets (tweet content sequences, POS tag sequences, and post time) shares the same embedding. The post time is used to indicate when the tweet was created, and the POS tag sequence is used to determine how each word was used in the tweet. The HD-LSTM produces a flat vector representation for the tweet by using the enriched sequential representation. After that, enriched fat representations for all tweets are concatenated to form a larger representation that contains the information from all inputs, which also includes the historical information for the target tweet. The result label is produced from the output layer which has as input the above concatenated vector. One more improvement on the HD-LSTM is the self-attention to all involved LSTM layers.

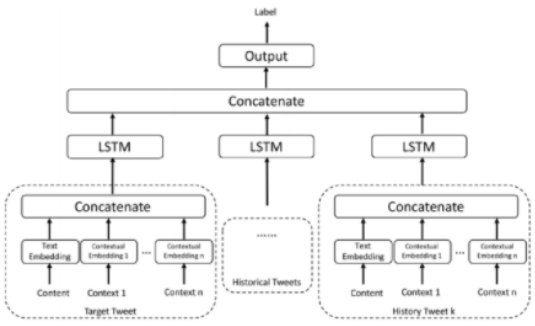

Figure 3. Hybrid LSTM Architecture [5]

The authors also applied this kind of methodology using a GRU and created a second-similar algorithm to the HD-LSTM. This structure adds direct contextual characteristics to the modeling of a single tweet sequence and blends the learned representation of each input tweet from the GRU to include all historical data.

4.2 Rule-Based

In this section we present a case of a rule-based user profiling approach using hard-coded expert rules, being the only one in the literature characterized as such. Rule-based techniques rely on "if-then" rules, using features extracted for each user, with the rules being constructed by domain experts.

In **Manoharan et al.** [12], the authors, whose objective was the recommend to the users viral articles from Twitter and Facebook, developed a proxy agent that traces users' browsing history to build implicit users' profiles (IUP). The IUP of a user, is constituted by the two features mentioned in section 3. Solely, click frequent count (CF) and specific search query count (SSQ). Both CF and SSQ, are then classified into "Low", "Medium" and "High" for a category, based on the counts. Next, CF and SSQ are given as inputs to the system, which resolves if a user is interested in a category or not, on the basis of hardcoded If-Then rules, for example, "if Cf is low and SSQ is low for X, then the user is not interest in X". These rules are differentiated for each field and implemented by corresponding domain experts. The output of the system can be either "Not interested" or "Interested" or "Highly-interested", which indicates the user's level of interest for the category. Consequently, the system proposes if an article should be recommended to a users or not, on the bases of the article's category and the user's level of interest in that category, which was quantified by the system.

5 Conclusions

The aim of this paper is to compile a list of various approaches to user profiling. Due to space constraints, we preferred to not examine techniques from all the possible categories, but rather focus on the most commonly studied and used in real-life applications.

User profiling is the practice of gathering, sorting, and reviewing information from a user profile. It's also defined as representing the need or desire of a customer. The retrieval of user data from different sources may provide useful information. User profiles can be manually retrieved using forms and polls, or automatically, using different extraction methods to retrieve data from the web or social media networks.

These approaches help user profiling programs collect useful user knowledge. This extensive study examines the principles and state-of-the-art consumer profiling techniques. It examines the modeling procedure in the area of data acquisition, pre-processing, extraction features, modeling and performance assessment, and also focuses on the novelty of each solution.

In our literature review, we found that most research papers are based on implicit data collection methods to build static user profiles. Through our review, we have also observed the following. Firstly, most works tend to favor user interest modeling. Secondly, due to the fact that not many

datasets are publicly available for user profiling, research works have to rely on the Twitter API to collect data and create custom-made datasets. Moreover, most of the works use rather simplistic user profiling methods and do not utilize advanced techniques, such as deep learning. In this context, most of the algorithms in our reviewed literature rely on Vector-Space modeling and Machine Learning techniques. For a more complete understanding of the points being made, a comprehensive overview of the details for each paper in the literature is presented in table 1.

We conclude that user profiling is a topic of increasing interest, due to the need for personalized recommendations, as well as demographic analysis. Because of this, many works have approached it in recent years, leading to a high number of innovations, as well as improved and increasingly elaborate techniques.

6 Technical implementation plan

As we have seen so far in this research, user profiling can be implemented in miscellaneous ways but also for many and different purposes, related to a specific application. We decided to utilize the user modeling process in order to study the effect gender has on one's development in academia.

Complementary to the effect, exclusively, of gender in one's academic career, we could study the impact of other factors as well. These could be either demographic, such as age, ethnicity, health and disability status, etc., or social, such as one's education, or collaborations on projects, among others. Our main purpose is to investigate whether there are any gender biases, either concerning males or females, or whether there is equal treatment of people, independent of their gender.

To do so, we are going to collect information for several academics, both male and female, from different universities across Europe. For each one of them, besides some general personal data like age, gender, and other similar, we will focus on getting information about their research work. More specifically, we aim at building a profile, which will reflect their whole progress in academia, but also, the interests and the involvement in projects and conferences they may have.

The information will be gathered from several different sources online, web services, such as LinkedIn, Research Gate, etc., for the purpose of having a more overall picture of each person's scientific background and relevant research activity. All the data will be stored in a document-based database, more specifically we are considering using the MongoDB[4] database, in which each document will correspond to someone's profile, and will contain all the necessary information.

We are also considering building a web application, using the Flask[5] web framework. In this application, we are going to present some insights about the collected data, some prejudices we may discover from this data, and in general all other findings.

[4] MongoDB
[5] Flask

References

[1] Zan Huang, Hsinchun Chen, and Daniel Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1):116–142, 2004.

[2] JesúS Bobadilla, Fernando Ortega, Antonio Hernando, and Jesús Bernal. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-based systems*, 26:225–238, 2012.

[3] Wendy Liu and Derek Ruths. What's in a name? using first names as features for gender inference in twitter. In *2013 AAAI Spring Symposium Series*. Citeseer, 2013.

[4] Rita Georgina Guimaraes, Renata L Rosa, Denise De Gaetano, Demostenes Z Rodriguez, and Graca Bressan. Age groups classification in social network using deep learning. *IEEE Access*, 5:10805–10816, 2017.

[5] Renhao Cui, Gagan Agrawal, and Rajiv Ramnath. Tweets can tell: activity recognition using hybrid gated recurrent neural networks. *Social Network Analysis and Mining*, 10, 12 2020.

[6] Abdullah Alshammari, Stelios Kapetanakis, Roger Evans, Nikolaos Polatidis, and Gharbi Alshammari. User modeling on twitter with exploiting explicit relationships for personalized recommendations. In *International Conference on Hybrid Intelligent Systems*, pages 135–145. Springer, 2018.

[7] Alessandro Micarelli and Filippo Sciarrone. Anatomy and empirical evaluation of an adaptive web-based information filtering system. *User Model. User-Adapt. Interact.*, 14:159–200, 06 2004.

[8] Won-Jo Lee, Kyo-Joong Oh, Chae-Gyun Lim, and Ho-Jin Choi. User profile extraction from twitter for personalized news recommendation. In *16th International conference on advanced communication technology*, pages 779–783. IEEE, 2014.

[9] Daniel Preoţiuc-Pietro, Vasileios Lampos, and Nikolaos Aletras. An analysis of the user occupational class through twitter content. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1754–1764, 2015.

[10] Peter Elias, Margaret Birch, et al. Soc2010: revision of the standard occupational classification. *Economic & Labour Market Review*, 4(7):48–55, 2010.

[11] Joanna Misztal-Radecka. Building semantic user profile for polish web news portal. *Computer Science*, 19:307–332, 2018.

[12] Saravanapriya Manoharan and Radha Senthilkumar. An intelligent fuzzy rule-based personalized news recommendation using social media mining. *Computational intelligence and neuroscience*, 2020, 2020.

[13] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[14] Ayse Cufoglu. User profiling-a short review. *International Journal of Computer Applications*, 108(3), 2014.

[15] Marina Farid, Rania Elgohary, Ibrahim Moawad, and Mohamed Roushdy. User profiling approaches, modeling, and personalization. In *Proceedings of the 11th International Conference on Informatics & Systems (INFOS 2018)*, 2018.

[16] Sumitkumar Kanoje, Sheetal Girase, and Debajyoti Mukhopadhyay. User profiling trends, techniques and applications. *arXiv preprint arXiv:1503.07474*, 2015.

[17] Christopher Ifeanyi Eke, Azah Anir Norman, Liyana Shuib, and Henry Friday Nweke. A survey of user profiling: State-of-the-art,

challenges, and solutions. *IEEE Access*, 7:144907–144924, 2019.