Department of Information Engineering and Computer Science

Bachelor's degree in
Computer Science

Final Dissertation

EXTRACTION OF USERS' BEHAVIOURAL INSIGHTS FROM SOCIAL MEDIA

Supervisor
Alberto Montresor
Co-Supervisors
Daniele Miorandi
Carlo Caprini

Student
Stefano Perenzoni

Academic year 2019/2020

Contents

Abstract

The recent rise of social media inside the life of our society caused a sharp increment of data availability at the user's level. Almost everyone relies daily on this type of platform to share their experiences, their thoughts, to interact with friends, to stay up to date with the latest news and also to find new career opportunities. All these activities carried out by a user leave publicly accessible a great amount of relevant information. The frequency a person logs into a social, the way he or she interacts, the network created by her connections allow extracting several personal aspects of the single individual. These characteristics can include personality traits, habits and particular attitudes. So, if identified, they can provide a huge business advantage in terms of knowledge of your own customers.

Social media result perfect for this type of analysis because people feel free to post whatever and whenever they want, often giving a strong personal opinion which reveals the behavioural aspects introduced before. Moreover, these services has been growing exponentially in the number of active users in the last decade. For example, the last *Digital 2020 Report*, carried out by *wearesocial.com*, shows that worldwide there are more than 3.8 billion social media users[1].

Even though the current literature has been covering the extraction of behaviours from social media, the majority of studies do not focus on the application of the result in order to get a marketing advantage. Therefore, there is no software system able to identify, and then let companies use, this information. For example, none of the research observed took into consideration users' rights in terms of privacy and data protection. However, since regulations such as the GDPR are becoming mandatory all around the world, compliance to their rules should not be neglected. Therefore, the final goal of this thesis is the development of a system for the extraction of personal habits and attitudes from social media that are immediately relevant and usable by a company's marketeers. This project has been realized at U-Hopper, a small enterprise located in Trento specialized in big data analytics solutions.

The proposed system follows the whole process, from the download of raw data from social media to the conclusive behavioural insights. Each phase was developed taking into consideration different aspects. In particular, with respect to the state of the art, this solution follows the main requirements of the GDPR regarding the authorization to access personal data and then the correct treatment of the same. The prototype is designed to interact with many social networks using their public API to download user's content. The part of insight generation is realized applying many different techniques. It uses both machine learning models for the extraction of personality traits of the MBTI personality model and non-machine learning algorithm for the computation of better-defined parameters, such as the language used or the periods of activities through the day. All these algorithms rely on information obtained through different analyses on the downloaded social profiles. For example, the natural language processing of the activities' text represents an important component, especially for the identification of the personality characteristics. The system was then developed as a web service accessible and exploitable thanks to a series of well-defined endpoints.

Finally, a web dashboard was realized to help the evaluation of the system. Thanks to some architectural choices made, it also allows to observe how the identified insights varied over time.

[1] `https://wearesocial.com`

1 Introduction

During my internship at U-Hopper, I had the opportunity to develop this thesis as a result of my experience inside the company. *U-Hopper is a research-intensive deep-tech SME, headquartered in Trento, providing big data-enabled solutions and technologies for the government, retail and manufacturing sectors. U-Hopper has received numerous awards for its innovative solutions, including, among the others, the Lamarck prize (2013), a EC Seal of Excellence (2015), the Innov@Retail prize (2016) and a nomination for the 2017 EC Innovation Radar Awards.*. The company is active in many different domains such as retail and tourism and offers a variety of competences including chatbots, analytics, and machine learning. Thanks to Tapoi[2], an innovative data intelligence solution, U-Hopper is also into the sector of user profiling. It allows businesses to deliver personalized experiences to their customers through the mining and analysis of their activities on social networks. Thus, the extraction of behavioural insights can be a valuable aspect since being aware of how an individual comes to a decision helps to provide each customer with the right tailored content.

1.1 Motivation and business requirements

Dissatisfied customers represent a dangerous threat for companies and their brands. Thus, it is fundamental for a business to track audience satisfaction and do whatever it can to fulfil their want. Dissatisfaction can impact a company in two different ways. First, those who are not completely satisfied would behave passively towards the business, reducing the number of purchases, and therefore stop being consumers of its products and services. Moreover, those who are more active and extroverted could interact with others and convey their disappointment. Overall, a large number of unhappy customers will entail a significant loss of customers.

This problem is of particular interest to those typologies of companies that follow a *business-to-customer (B2C)* sales process, with a wide customer base and which interactions with their audience are characterized by online relationships. This relation can be purely telematic, as in the case of e-commerce, or it can support a physical one where the material interaction is unavoidable, as in the case of banking and insurance sectors.

For this kind of businesses, customers' satisfaction is not trivial to accomplish since each one of them has different needs and requests and standard methodologies do not adapt well for everyone. Thus, over the past few years, personalization of customer experience has become vital in order to inspire an honest and natural emotional response. It is then important to be able to access information which allows marketeers to offer fully tailored contents, through a specific mean of communication and with personalized messages to meet each individual's requirements.

While, thanks to Customer Relationship Management (CRM), data related to the direct interaction between customer and company has already been deeply explored, social media networks gave access to more personal information allowing a deeper understanding of the person. The system discussed in this thesis proposes a solution that goes further than the diffused purchase history-based personalization. It aims to provide companies with the ability to extract readable and valuable insights about singular individuals from their activities online. The final goal is to make available actionable insights about users' behaviour, demographics and attitudes. In particular, this dissertation focusses on the extraction of personality traits

[2] `www.tapoi.me`

to obtain a detailed description of a person's behaviour and reaction to a number of observed solicitations

1.2 Extraction of personality models

According to neuroscientists Adelstein et al., personality describes human behavioural responses to wide classes of external stimuli [2]. It works as an adaptive system for taking in, organizing information and driving the response to inner and outside demands [5]. The parameters of the adaptive system represent the variation of the same from person to person and, therefore, characterize uniquely every individual. These parameters are also referred to as personality traits in several different personality models studied over the years. Each model includes its range of traits which combinations describe several personality types. Researchers have shown clear connections between general personality traits and many types of behaviour.

Some fundamental traits describe the type of relationship a person has with the outside world and the way he or she communicates [19]. Thus, to facilitate communication, recently, businesses are using personality models to gain a better understanding of what drives the interests of a person. This approach is showing clear benefits in many different applications. In the field of Human-Computer Interaction, users prefer interfaces designed to represent personalities that most closely matched their own [26]. Some studies have also suggested connections between customer personality and marketing. Through techniques more focused on the target audience, it is possible to profile individuals, and tailor advertisement automatically displayed based on their personality [3]. Therefore, the ability to identify people's personality or, even better, details of their personality traits through well-defined models is a significant competitive advantage since we would have a precise representation of the customer's reasoning process.

1.2.1 Big Five personal traits

While several models exist, the *Big Five*, also known as the *five-factor model* and the *OCEAN model* is one of the most well-researched and widely accepted taxonomies among scientists [23, 22]. It formalizes personality along 5 domains, namely Openness, Conscientiousness, Extroversion, Agreeableness, and Neuroticism. Each one of these traits is continuous and usually ranges on a scale from 1 to 5. High openness marks imagination, creativity, and curiosity in learning and exploring new things. Conscientiousness represents self-discipline and attention to details. Extroversion measures preferences for interacting with other people. Agreeableness reflects the extent to which a person is generous, trustworthy and always willing to help others. Finally, a high score on neuroticism indicates a tendency to get stuck in negative emotions. At the two extremes of each trait, two separate aspects reflect a particular behaviour. For example, conscientiousness is bounded by carelessly at the lowest end and by organization and efficiency at the greatest one.

Since its first definition, this model rapidly became one of the standards in the psychological community, largely accepted by the most share of scientists since it allows to describe accurately the traits of a singular. However, concerning the exploitation of personality information in the work and marketing environments, it received some critics about the extraction of actionable insights [14, 28]. Indeed, since each trait is represented by a real number between two extremes, it has been argued to be hardly readable and therefore less valuable for fields such as marketing and business. Thus, structures based on clearer distinctions are often preferred.

1.2.2 Myers-Briggs Type Indicator

The Myers-Briggs model, also called *Myers-Briggs Type Indicator, or MBTI*, is the most common alternative to the Big-Five model. Contrarily to the former, there are discussions about the MBTI and its limitations in reflecting the whole personality system. Boyle and Barbuto are two of the scientists that presented a number of psychometric limitations pertaining to the validity and reliability of this model [6, 4]. However, many of their arguments have been proved wrong

by Furnham who demonstrated several correlations between the dimensions defined by Myers and the big five factors [12].

The MBTI is a categorical model, based on the conceptual theory of Jung and developed by Katharine Briggs and Isabel Myers who used four different dichotomies to evaluate the personality of people [15]. A first one differentiates a person's attitude in either extraversion (E) or introversion (I).These two preferences describe if one focusses on external stimuli, such as action and interaction with other people or internal ones like self-reflection. Two perceiving functions, sensation (S) and intuition (N) describe the process of gathering new information. On the one hand, people who trust tangible and concrete facts; on the other hand, those who tend to find patterns and meaning also regarding future possibilities. The third cognitive function is that of decision-making which can be thinking (T) or feeling (F). While thinkers make reasonable and consistent choices and reflect over consequences applying a rigid set of rules, feelers tend to emphasize with the situation considering the needs of people involved Finally, there is the lifestyle preference function dichotomy, judging (J) or perceiving (P). Judging types like the outside world to be structured; according to Myers, they prefer to "have maters settled". On the contrary, perceiving personalities like it flexible and spontaneous and tend to "keep decisions open" [25]. There are 16 different types of personality given by the combination of these 4 cognitive functions identified by 4-characters codes such as "INFJ" or "ENFP".

1.3 Research objectives

Extraction of behavioural insights from social media has recently attracted the attention of both researchers and businesses. Even though the latter has released a couple of solutions, these fit better for personal and psychological use rather than a commercial one. The main objective of this thesis is to design and develop a solution that can be used by a company to personalize customer experience with respect to individual abstract preferences. Therefore, the question it answers is: *is it possible to understand costumers behaviour from their online profiles and activities?*.

Using a personality model to catch these behavioural aspects, the extraction of personality from social media activities is a *machine learning* problem. Precisely, with a categorical model, such as the MBTI, it consists of numerous classification tasks, one for each variable of the taxonomy. Machine learning is one of the most talked-about fields of computer science and many sources give their own definition. Basically, ML deals with allowing a computer system to "learn with data, without being explicitly programmed" [34]. It has been applied in many contexts, such as decision making, optimization problems, forecasts, and predictions. Nowadays, we face ourselves with machine learning in everyday life: home assistants, security surveillance, music and shopping suggestions, customer services are strongly powered by artificial intelligence. These services rely on data to learn how to work as good as possible: they are trained with samples of data similar to what they expect to receive by their users: the more accurate, exhaustive and in large quantities they are, the better the system learns. Therefore, data have a very central role in machine learning problems.

A classification task has the goal of assigning a belonging class to a given object. The input is composed by a tuple of *features* that characterize the object, usually made by numbers, and the output is a categorical variable, such as a "yes/no" label. In other words, it can be seen as a mathematical function, that maps a vector $\boldsymbol{x} \in \mathbb{R}^n$ to an answer $y \in C$

$$f \colon \mathbb{R}^n \to C$$
$$f \colon \boldsymbol{x} \mapsto y$$

where C is a set of possible categories. For example, in one of the four classifiers for this problem, $\boldsymbol{x}$ represents a user and her activities on the social media, and $C = \{\texttt{Introvert}, \texttt{Extrovert}\}$

The designed system should be able to work with numerous social media platforms to have a wide variety of data sources. Finally, the principal aspect that it must always satisfy is the

ability to use the result. Indeed, extracted insights need to be actually actionable, directly by the marketing department or in conjunction with further analysis, to represent a competitive advantage.

1.4 Outline

Chapter 2 describes the state of the art. Chapter 3 introduces the design of the solution. It focuses on used components and algorithms, their logic and their interfaces. Chapter 4 shows how the mentioned components are implemented and integrated. It follows the implementation of the algorithm and the evaluation of a general prototype of the proposed system. Chapter 5 concludes the thesis with some observations and future work proposals.

2 State of the Art

This chapter presents the current state of the art regarding insights extraction on social media. Many aspects of online users have been explored in order to profile customers. "Then, there is a focus on what has been done in terms of providing actionable personality insights."

Some studies aimed to identify clear demographic characteristics based on both the analysis of a user's activities and her network inside the social media. Twitter is commonly used for the extraction of gender [24], age or age groups [9]. Also, a person's family status is inferred through the detection of life events such as the birth of a child and a marriage [10].

The literature also presents many examples of latent attributes extraction. Some of the most remarkable research has been carried out by the *World Well-Being Project* [3]; a research center which used social media to measure attitudes and personal characteristics such as optimism and pessimism [33], temporal orientation [35], etc. Many different social networks have been explored as well as many aspects that are not limited only to text but also include images and social interactions. Finally, it is a common practice inferring behaviour through a variety of personality models.

However, what has been done is almost completely focused only on the feasibility of extracting attitudes' insights from online activities rather than a commercial use of the obtained information to generate a marketing advantage. So, the literature presents only a few systems which satisfy the right requirements for an application in the real world, such as those imposed by the GDPR[4].

2.1 Customers Profiling

A precise and detailed description of social media users requires the analysis of many aspects of social media. Indeed, understanding the users means being able to quantify and qualify how they present themselves [36].

Many of the systems proposed for social media analysis use as fundamental component features that describe interactions of users, such as the number of followers, mentions, likes, and comments. This type of analyses has been largely explored since studies about user influence and social engagement. First, raw measures publicly available on social media were used to calculate metrics to represent effectively the user's influence [27]. Further research proved that simply observing ground numbers of a profile can lead to a misunderstanding. Cha stated that the indegree alone (number of followers) reveals little and suggested to consider shares and mentions from other users [7]. D. Romero et al. observed influence analysing the propagation of web links over time using both the structural properties of the network as well as the diffusion behaviour among users [32]. They also regarded the *passivity of a user*, a measure of how difficult it is for other users to influence him, and used it to weigh the tweets propagation network. Many different networks can be explored on social media in order to identify influence, communities, and trend topics applying the myriad of network concepts and analyses such as degree centrality and modularity [8]. The nature of these graphs can change regarding the platform's characteristics and the aspect we are looking at. Li complained about undirected networks, such as the Facebook friends graph and proposed a method based on the *Share/Reply/Mention* directed network to

[3] https://wwbp.org/

[4] `https://gdpr.eu/`

capture user influence [18]. These observations are usually used to profile a person's social environment and to assess his or her role inside it.

A second fundamental point carried out by literature on social media is the analysis of the context the user is talking about. Obviously, being aware of what topics drive someone's interactions is essential to profile his or her interest. Moreover, they can be used to reduce other types of analyses to a specific field of interest. For example, focussing on users' influence in sports discussions. To understand context, it is necessary to observe the content of the messages which is usually composed by text and images or videos. Firstly, keywords in the activities were used to identify topics [7]. This methodology shows some clear issues, especially when used for social media when messages tend to be extremely abbreviated through acronyms and slang words. Other approaches, feasible in a limited number of platforms, proposed to use most used hashtags to obtain linguistic content starting from the activities [29]. Finally, a more general technique is using the tree of Wikipedia categories to characterize the user's interests. This method fits well with both text and multimedia content thanks to a number of services that apply semantic analysis techniques to extract relevant entities [39].

2.2 Behavioural insights

"Psychometric profiling is the process by which your actions are used to infer your personality."

The literature presents many different techniques for the extraction of behavioural information which are all based on the most used personality models to study specific traits of an individual. Each model is specialized to a single specific personal characteristic. The models proposed are classifiers or regression one depending on which personality taxonomy is being applied. The Big Five model is the most spread the most used one for the automatic extraction of personal attitudes. Commonly, each one of the five traits composes a regression task because of their continuous nature [17] Even though, Sumner experimented a binary classification for each aspect using as classes the two extremes of the trait [37]. On the other hand, the MBTI model requires the application of binary classifiers. Generally, each cognitive function is inferred separately since it has been showed that multi-class classification on the sixteen personality types bring to poor performance [20] Also, a few studies worked on characteristics that do not belong to any personality model. For example, researchers from the World Well-Being Project explored Facebook and Twitter to infer optimism, pessimism, empathy, and trustfulness [33, 1, 43] Almost all models presented work on social user composed by the totality, or a portion, of their timeline rather than single activities since linguistic information contained by a single short activity is not enough to accurately predict personality aspects [21].

The feature extraction shares some fundamental aspects in the majority of systems. The results of the analysis seen before represent two essential groups. Indeed, understanding a user's network helps understand how he or she reacts to external stimuli. Therefore, it plays a crucial role in the extraction of behavioural insights from online activities. Also, research has shown a strong correlation between discussed topics and personality aspects of a person [16]. Guntuku et al. proved that studying semantic concepts contained in posted images can give a significant performance gain in predicting personality traits with respect to the *Big Five model* [13]. However, the literature contains a very few number of proposals that considered the content of the activity and are usually confined to hashtags and key words in the text [33].

Regarding features that describe the social presence of a person. These are usually included by the majority of models. Although some are limited to basic information such as the number of followers, following or friends, the number of activities, and their frequency [31]. Over time, the literature presented the application of more complex features, obtained as results from further analysis of the user's network such as interaction patterns by a person towards the author of the post [10]. For example, significative patterns could be a high retweet ratio by users who do not retweet much other sources by or an elevate number of interactions by users with many followers. However, these last observations need the permission of each person belonging to the

analysed network to be respectful of GDPR requirements. Thus, even though they could give great results, their lawful application in the market is quite intricate.

Then, there is a third fundamental group of features which is probably the most important one. Since psychological studies proved that there is an effective relationship between linguistic style and personality aspects, understanding detailly how an individual writes is a crucial step [30]. Some of the most common and basic features are word counts, sentences per activity, word per sentence, and punctuation count. These have been applied by the majority of models with great results in many different environments. For example, Farnadi recognized personality of YouTube vloggers using the script of their videos to extract this linguistic information [11]. Furthermore, more recent studies have tested features from specialized and complex tools for text analysis. These can reveal precisely thoughts, feelings, and motivations of the text's author. The *Linguistic Inquire and Word Count (LIWC)* developed by Tausczik and Pennebaker is certainly the most used one [38]. Other services that have been tested are the *MRC Psycholinguistic Database* and the *NLPRO*, developed by NLPLAB [42, 40]. Lima et al. tested the three of them concluding with the first one as the most performing one [20]

2.3 GDPR Compliance

A big issue that emerged recently in dealing with user profiling is the new regulation adopted by the European Union (EU) on the protection of personal data of individuals. The *General Data Protection Regulation (GDPR)* became enforceable in May 2018 after being adopted in April 2016. Its validity spread around the European Economic Area. It deals with the privacy of natural persons with particular regard to the processing of personal data and on the free movement of such data outside the EU area. A goal of the GDPR is to harmonise the rules for all the Member States in order to reduce the legal complexities and uncertainties and to reinforce the data subjects' rights. However, it is a regulation and not a directive. Therefore, even though it has to be applied, it provides flexibility for certain aspects to be adjusted by singular Member States

The GDPR does not completely truncate the freedom of business for the benefit of the single person. It aims to balance the right of the physical individual and the right to do business of the enterprise. Its rules are valid for businesses, also called juridical people, which treats personal data with market and professional purposes. The GDPR rules how data shall be processed. Firstly, it specifies that data processing is lawful when at least one out of six criteria, called lawful bases, is met. Foe example, personal data can be processed when the data subject give consent to specific processing or when data processing represents a vital interest for the person. Then, the regulation define a list of fundamental principles about the processing of personal and sensitive data [41]. The principle of **purpose limitation** impose the formal definition of each singular purpose of the treatment and the corresponding legal basis. The principle of **data minimisation** states that only data strictly necessary for the final purpose should be collected and should not be further used for reasons that were not stated. Finally, data is required to be **accurate**. Data that is inaccurate or incomplete must be erased or rectified.

The GDPR also considers scientific research as a specific context of personal data processing. Here, the equilibrium between individual freedom and the freedom of research must allow both personal data processing and sharing in the pursuit of the public interest. So, even though some specific rules are applied more liberally to scientific research, the general principles must be respected. However, the literature does not contain significant studies that considered the limitations imposed by this regulation. The GDPR is mentioned rarely probably because the majority of the research focuses more on performance aspects of the the result, such as its reliability and accuracy rather than the application of the result itself for commercial purposes which would require a strict compliance of the previous requisites.

3 Design and methodology

This chapter presents the architecture of the system. Starting with a basic schematic view, it shows what fundamental components compose it, their role, how they work and how they communicate each other. Then, the architecture is detailed more. Each step, starting with the download of raw activities and ending with the final user's attitudes, is described specifying what choices have been done and why.

At a high level, the system takes as input an ID representing a user in the system. This user is associated with many IDs, one for each social network he or she is registered in and has provided access to. These are used to download her profiles which, together with their respective activities, are used to classify the user's behavioural aspects included in the analysis.

3.1 Requirements

The main functional requirement is that the system has to be able to extract a set of behavioural aspect that characterise the user taken as input. All the social network the user has registered must be analysed for both the social profile and the activities. The architecture should allow the addition of new users into the system and the removal of existing ones. Then, once a system user is created, it must be possible to register and remove new social media account. During the download of new activities, the system should let use a filter or a counter to exclude unnecessary activities. Finally, the set of attitudes that can be classified must be defined a priori. For example, it can be equal, but not limited, to the 4 cognitive functions of the MBTI previously introduced.

3.2 Process logic

Here, the fundamental process followed by the system is introduced. Each step covers a section of the process required to go from the simple user ID to the final insights that the system aims to provide. The whole flow is sequential. Each activity takes as input the output of the previous one There are 4 essential steps which can not be excluded despite specific design choices:

1. Download profiles and activities from social networks.
2. Analyse the downloaded data to extract significant features.
3. Classify the profiles.
4. Put together the partial results from each classifier to generate the complete user image.

The steps are executed the same number of times and each one immediately follows the previous one with the requirement that the former must have ended for the latter to start. The only exception can be found for the first two activities which could be grouped together. Indeed, once an activity is downloaded this could be immediately sent to the following action while a new one is put on download. However, before the classification could be performed the whole profiles should be completely downloaded and analysed. Finally, the final result can be stored. So, all the process is coordinated by a single call which start with the download phase and end the annotated user.

These steps give a general description of all the jobs required to go from the raw user identifier to its classification. These functionalities are subsequently realized thanks to different components with the purpose of following this abstract process. The final components of the system may vary according to the architectural choices made.

3.3 Classificator's architecture

While the first two steps are quite common and do not present significant choices that could modify the system, the third one deserve to be observed in more detail. In the extraction of insights at user level, it is essential to understand precisely how the social user is defined and by what data it is characterised. The state of the art rarely took into consideration different approaches. Usually, all the information obtainable by the totality of the activities downloaded are put together to represent the user ready to be classified. It means that, at each request, all the downloaded activities are merged into a single user that summarise what has been provided by the social media. At first, a solution of this type might work if the goal is to verify the feasibility of a specific classification in order to be able to measure its performance quickly and easily. On the other hand, when it comes to the production environment, this technique presents some evident lacks that need to be considered. For example, one main issue concerns the integration of the insight once new activities are posted by a user. The system should only download and classify the new activities and then integrate the result with the one previously stored.

Starting from this last problem I worked on three different alternatives. These three proposals are the feasible solutions that answered to a series of questions and critical points that emerged during this design process.

- Architecture **per aggregation**: the classification is performed on the so-called user aggregates. User aggregates represent the totality of the feature extracted by the profile and its activities. In the moment new activities are downloaded from the social profile, the previous aggregates are updated and then stored ready for the classifications.

- Architecture **per activities**: each activity is classified singularly. Instead of the features, the result of each activity is stored and then put together to generate the insight at a user-level.

- Architecture **per batch**: the classification is performed on the aggregates obtained from a batch of activities. Batches are disjoint sets of downloaded activities. Each is classified independently and its result is stored. These partial results of a specific user are merged together to get the final result.

I evaluated and compared these three alternatives against six criteria considered fundamental with respect to the initial research question. The six bases are: computation complexity, result's accuracy, GDPR compliance, result's flexibility, storing, and scalability.

3.3.1 Computation Complexity

The performance of each propose strictly depends on the trade-off between a single classification and the aggregation of features from many activities. Assume that the cost of a single classification is equal to the number of features M, $\Theta(M)$. Estimate now, for each solution, the computational complexity of the download of N activities and the user's classification. Firstly, It must be noticed that the first two steps of the process, the activities download and their analysis, are not affected by the architectural choice. Therefore, to keep everything clearer, their contribution will be omitted from the following estimations.

Assuming that the aggregation algorithm just iterate on each post of the timeline a single time observing each one of the M extracted features. So, its computational cost is $\Theta(N * M)$. As already said, using user's aggregates implies to compute the classification on its totality at

every request to the system. So, this method's complexity considers the number R of requests which is obviously constantly growing. It can be formalized as $\Theta(N * M) + \Theta(M * R)$.

Regarding the second alternative, while it is no longer necessary to aggregate the analysis of each activity, it is required to merge together each singular result in order to obtain the final insight. Moreover, the classification is performed N times on the exact number M of features. So, the complexity of the solution activity-based is $\Theta(N * M) + \Theta(N)$.

To analyse the last solution, the batch-based one, it is not necessary to define how batches are composed in detail. The N downloaded activities are somehow partitioned into B batches with $B \leq N$. Each batch's content is aggregated with the same algorithm introduced before. In general, it is not important how many batches there are, the aggregator will still work on M features of the N activities. Finally, the B batches are classified singularly and their results are then joined together. So, the complexity is $\Theta(N * M) + \Theta(B * M) + \Theta(B)$.

To conclude, since the architecture aggregation-based considers in its formula the number of requests to the system, it is not scalable in terms of increasing loads. On the other hand, the other two alternatives are computationally similar. The differences between the two stay in the number of classifications done and the final aggregation of partial results.

3.3.2 Accuracy of the result

Regarding the final result obtained by each of these three methods, even though it would be better to carry out a detailed evaluation process, some general observations can be done without any implementation test.

Starting with the architecture per batch, user aggregates are easy to calculate since they generally consist in the sum of many values or in their average. Thus, they are precise and represent correctly the user's values. However, even though until now I have been talked only about features extracted by the online activities of the user, other useful information is represented by that related to the social profile. For example, fields such as the number of followers, that of people followed, the user's location, and many other help describing the individual, her social presence and connections. In this case, at every download request, this information contained in the user aggregate is updated with the new one. Consequently, the snapshot of the old status is lost. This means that the activities are not associated with the exact moment they were written but they all treated the same way without differentiating the social situation of the user.

This last issue is partially solved by the activity-based methodology because in the features used to classify each activity can be included that information about the social user. However, this data is taken in the moment the social network's API are consulted so it is not perfectly accurate for each post but it can represent a good approximation. On the other hand, this alternative involves two major obstacles. Firstly, it has been proved that, especially in the case of psychological studies, text represent the first source of information. Typically, on the social media, users tend to write short posts composed by just a pair of sentences. Thus, what has been argued is that the quantity and the quality of the extracted features do not allow a precise classification of behavioural aspects. Secondly, once every single activity is classified, they must be merged together to generate the user's insight. In the case of categorical results it is not clear how singular results should be merged and how each one should be weighted with respect to the others.

Finally, the architecture per batch has the same advantage of the one per activities because each batch would be characterised by the features that describe the social status of the user. So, each batch of activities, depending on when its was downloaded, includes some data describing that information mentioned before. Moreover, the problems of the previous technique here are less pronounced. Indeed, batches can include the right number of activities in order to get the right amount of features. Also, during the aggregation of partial results, each one could be weighted both dimensionally and temporally giving so more freedom to adapt the implementation.

3.3.3 GDPR Compliance

Design a system that respect the last regulations about the protection of personal data is one of the goals of this thesis. In this design phase, some requisites from the GDPR emerged as crucial aspects. In this chapter it is explained how the three different architectures satisfy or not the requisite of data minimisation and that of data accuracy already introduces in chapter 2 "State of the Art".

Starting as always from the solution per aggregation, it implies to store the user aggregates so that they can be updated or classified whenever the user wants. Even though this data is composed by the relative features extracted from the activities and not by the activities themselves, they still contain personal and sensitive that need to be protected. Then, as discussed is the previous section, info about the social status are unique and dated to the last upload so they are less accurate with respect to the whole timeline.

Using one of the other two alternatives, the situation changes markedly because they do not require to store the activities nor the extracted features. It is enough to memorise all the partial results and some descriptive information, such as the number of activities and their date, in order to be able to join them together. The solution activities-based, compared to the last one, brings more profit with respect to the accuracy issue. Indeed, it gives the opportunity to complement every single activity with the social status in the moment it was posted while batches still implies a minimum approximation.

3.3.4 Flexibility of the result

This small section gives a focus on the benefits regarding the result's flexibility rather than criteria that need to be met. Depending on this architectural choice the system could allow the composition of the final results with different options. For example, by taking into consideration only a subset of the whole user's timeline

Using the user aggregates, it is quite difficult to act on the final result once it has been classified. Indeed, the result is the direct output of the subset of aggregates the classification algorithm uses. So, the only way to affect the result is by working on these aggregates which include the whole timeline and are unique for each user.

The architecture activities-based is for sure the one with the highest flexibility. Indeed, it is immediate excluding some activities that do not meet a specific requirement. For example, it would be possible to compute the user only for her social presence during the summer periods or during the weekends. So, memorising a flag together with the result would be enough to subsequently filter the final result. These flags are not limited only to datetime aspects. For instance, they can be used to differentiate posts that talk about sports from those that talk about politics.

With the solution based on batches, it becomes possible to treat each batch differently. It all depends on how the batches are composed. For example, batches that cover specific time periods give the opportunity to observe temporal variations. Then, they could be filtered depending on a exact term. Also, each batch can be normalized regarding the number of activities it contains in order to weigh its contribute on the final result.

3.3.5 Storing

Here is briefly discussed where, in the system, data persistence needs to be implemented; then, it is observed how the data model affects the system and, in particular, its evolution.

To compute the user aggregates, persistence must be implemented on the component that perform the aggregation so that they can be updated at every new download. Then, the results of the classifications need to be stored. Also, the activities collector should know which is the last activity downloaded for each user in order to not count them multiple times. The features the system memorise are always known a priori depending on what classifications the system performs. Consequently, the removal or the addition of some features, for example to improve

or add a new model, involve adapting the whole data model and the download of the whole user's timeline.

For the other two proposals, it is necessary to have data persistence on the component that deals with the classifications so that the result of each batch or activities can be stored. Obviously, the activities collector always need to know which is the last downloaded post. It is quite immediate to add new services at the system since the features extracted do not need to be memorised but are immediately used from the next component. So, the data model is affected only on the result aspect in the case new classifications are added.

3.3.6 Scalability

This last aspect does not vary much from one solution to another. All three architectures allow the parallelization of the independent classifiers. In addition, for the two architectures based on multiple classifications, these can be parallelized since both each batch and each activity are independent of the others.

3.4 Components's logic and interaction

After this analysis, it was decided to focus on the third proposal, the batch-based architecture. The one based on the user aggregates is quite basic has several shortcomings in many of the observed aspects. The other two share many advantages but regarding the accuracy of the result, the activity-based solution has some complications not included on the other one. However, as said before, that aspect was analysed through general observations and testing both the architecture would give a more detailed overview.

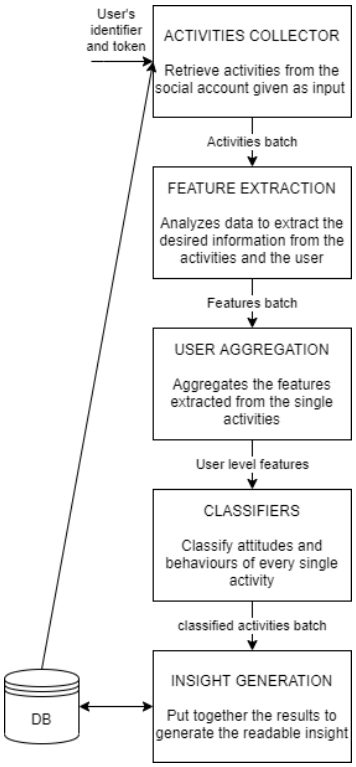

Figure 3.1: The architecture components

In this section, the architecture is decomposed into its components. Each component works on the input of the previous one to carry out a step of the whole process described previously. Totally, five components have been identified: activities collector, features extractor, user aggregator, classifiers and the insight generator. They are shown in Figure 3.1:

Next, the logic of each component is provided, illustrating what each one does, its input, and output. For each component, it is also presented its API. Internal APIs are used by the components to facilitate and standardise their interaction. These components are presented as web interfaces that takes as input and returns as output JSON objects. Only the parameters used to identify the object itself are used. For example, a postID and a socialID are enough to indicate an activity.

3.4.1 Activities collector

This part deals with the downloaded of user's content from social networks through the services offered by the public APIs of each platform. The data includes information about the user, such as social name, number of followers and friends, location, and then her activities. An activity is usually characterised by some text, optional attachments such as media or files, a creation date, the number of likes, shares and comments, and, sometimes, information derived from the geolocation.

This component takes as input a list of IDs, one for each social network connected to the user, that identify all her profiles. Some social media, like Facebook and Instagram, also require an access token necessary, together with the user ID, to consult, through the API, the content of that profile. These tokens are released directly at the organization that is handling the user authorization.

For each social network, the system need to know at which point the data has been previously downloaded. Usually, the IDs identifying the posts are ordered and the API allows you to specify the point the download should start from using an ID. As output, the components returns a social profile composed by significant user's information and the list of new activities. With new, it is meant all the content that was not previously downloaded by past requests. Finally, the last activity of the timeline is used to update the point the data has been fetched.

The collector interacts with many different social networks. To do it, this service should run in a dedicated process and the communications with each platform should be parallelized. This is necessary because many socials provide free API plans which limit the number of requests in a given time window. So, in case one of the limits is reached, it should not cause a critical bottleneck that freezes the whole system.

```
//input
[
    {
        "socialID": 1234,
        "userID": 8564
    }
]
```

```
//output
[
    {
        "postID": 123456,
        "socialID": 1234
    }
]
```

3.4.2 features extractor

This component works on the downloaded content to extract significant features for the following step. From the previous part it receives data both about the user's information and about her activities. While the first one is quite structured and does not need deep analyses, the activities carry raw texts and images which have to be processed to extract the desired information.

First of all, the text of each activity is observed and analysed to extract significant features for the classification models, in particular for the psychological ones. A standard *Natural Language Processing NPL* library is used. It allows to observe some superficial characteristics such as the number of sentences, words, characters, capital letters and the use of punctuation. Then, giving the environment we are working with, that of social media, it is important to handle properly hashtags, external links and emojis.

The first two, as well as any mention or @tag, are usually returned by the social network's API. So, it is not necessary to extract them from the raw text. On the other hand, emojis are

treated like any other character by the API. Therefore, they need to be parsed carefully. Taking care of emojis is extremely important because on social media they are abused by the majority of users and, thanks to their variety and intuitiveness, they can reveal a lot about the traits of a person.

Another significant step of the natural language process is the *Part-of-speech (POS)* tagging. It consists in assigning a particular part of speech,, such as nouns, verbs, and adjectives, to a specific word in the corpus. Doing that, it is possible to observe, for example, how much a person tends to detail her messages by using adjectives or whether someone writes in the first or third person.

Then, we want to understand what the user has talked and posted about. Both text and images can be used to accomplish this goal; thanks to some external semantic analysers that can extract entities, topics, and a sentiment value. These services usually use the Wikipedia tree to map both entities and topics. The first set refers to every concept that has its own page on Wikipedia. Differently, a topic is a Wikipedia category. In the Wikipedia structure, each page can be characterised by many different categories. The sentiment is returned as a label that can be *negative, neutral, or positive* or as an integer ranging between two values. For example, between -1 and 1 where -1 means extremely negative and 1 extremely positive.

At the end, this component returns as output the same user taken as input where each of her activities has been analysed. So, each post is replaced by the list of features extracted from it. Every activity obviously has the same general feature names. In some cases, for example, if a text did not contain any Wikipedia entities, some may be empty but they are still contained in the returned object.

The feature extraction is not limited to the analyses mentioned here. The idea is that this component should be integrable with new services and functions.

```
//input
[
    {
        "postID": 123456,
        "socialID": 1234
    }
]
```

```
//output
[
    {
        "postID": 123456,
        "socialID": 1234
    }
]
```

3.4.3 Aggregator

This part has to task to create a single object, characterised by a series of features, that is ready for the following classifications. It receives as input the social profile and its analysed activities. By simply iterating over all the activities received, it is primarily responsible for dividing them in batches and then calculating their aggregated values.

The way batches are composed can not be decided a priori. Two different techniques for their composition have been identified. A first one that works on a fixed number of activities for each batch and an alternative which divides posts into batches covering fixed time periods.

The dimensionality-based one is the most basic. Simply, if 180 new activities are downloaded and it is decided to compose batches of 50 activities each, three full and one-half batches will be returned. This methodology is in contrast with the result's flexibility criteria seen above because any type of control over the individual activities is lost.

On the other hand, the temporality-based alternative divides batches in a more organized way. In fact, there is an additional degree of freedom that allows you to decide the length of the time periods. This solution helps visualizing the result and giving the opportunity of observe how significant insights varies over time. However, this period must be common for every set. In Chapter 5 a prototype of the system is implemented with batches covering one month.

After the various activities are divided, it is necessary to run the same aggregation algorithm on every batch. At first, static information about the user's profiles, such as creation date and

the number of friends are copied in each batch. Then, for each set of features, the significant features are aggregated in the best way to represent them globally So, for example, it is counted how many activities of each type there are. These can be differentiated into original posts, shares, comments and replies. Then, text features are summed together treating the batch as a single text document.

Finally, a list of batches is returned. Each one includes all the data needed for the classifications implemented by the system. Each batch should also contain some descriptive information such as the ID of the oldest activity it contains, the number of activities, and the time period that it covers.

```
//input
[
    {
        "postID": 123456,
        "socialID": 1234
    }
]
```

```
//output
[
    {
        "UserID": 8564,
        "batchID": 001
    }
]
```

3.4.4 Classifiers

At this point the situation is that each batch represent a partial user who can be classified. This component has the goal to assign a value for each aspect implemented by the system.

This decision can be made both with machine learning and non-machine learning algorithms. Section 3.5 gives a more detailed description of the classifiers included in the system. In this thesis, as discussed later in Chapter 5, machine learning models have been used for psychological aspects while better-defined characteristics, such as the users' activity periods over the day, are inferred by looking singularly at the features.

As showed in Figure 3.1 all the classifiers are treated as a single component. Actually, there are many different classifiers. Some may work on the same features but, in general, they are all independent from each other. So, their execution can be viewed in parallel where, as it is shown in Figure 3.2 all the subcomponents use the same input and each one contributes to the generation of the final output

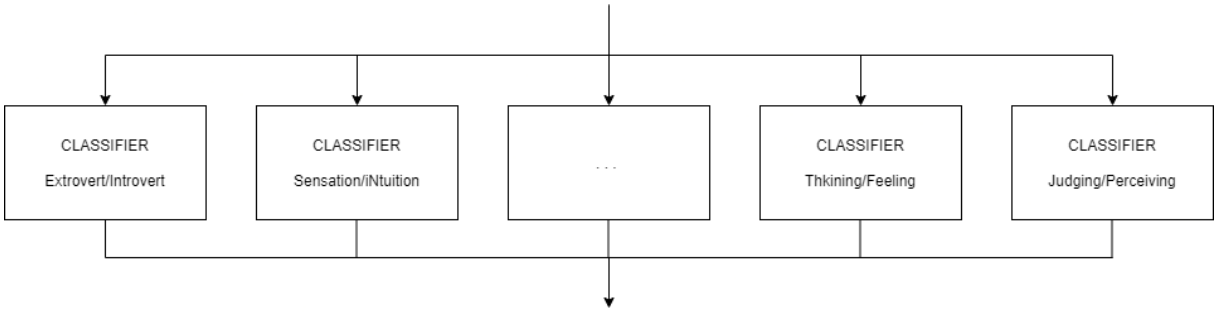

Figure 3.2: Parallel view of some of the classifiers

In the case of classifiers that use text features, it must be remembered that all the analyses done on natural language depend strictly on the language on the same. Therefore, to handle multiple languages, multiple classifiers are required.

As output, the same list of input batches is returned where for each implemented aspect the corresponding class is returned.

```
//input
[
    {
        "UserID": 8564,
        "batchID": 001
    }
```

```
]

//output
[
    {
        "UserID": 8564,
        "batchID": 001
    }
]
```

3.4.5 Insight generator

This component has to final task to merge together all the partial results of each batch to generate the general user's insights. To do it, an algorithm was defined. The input of this algorithm is a list of **classified batches**. Each batch is an object that contains the descriptive information necessary to map it into the whole user's timeline and results of all the classifications performed by the system. As explained in more details later in Section 3.2, all the results are integers that represent the corresponding class. The algorithm is here defined in pseudocode. This results aggregation consider two characteristics of each batch: the number of activities it contained and the time period it covers. These are used to weigh the contribution each batch gives to the final result. Indeed, for each batch, two parameters are calculated. γ that is the weight coming from the number of activities and δ, that coming from the batch period. The algorithm iterates only one on each batch. An iteration is composed by two main parts. In the first one, the two parameters are computed. In the second one, each result is taken in consideration for the final result. The method used to choose the final label is the same for all the classification. For each class of each classification a score is computed. So, each result contributes to its class of $s = \gamma/\delta$ For each classification, the class with the highest final score is assigned. Algorithm 1 explain the method.

Algorithm 1 Merge together partial results of each batch

```
1: function mergeBatches(batches)
2:     scores ← 0
3:     MAXs ← 0
4:     for all b ∈ batches do
5:         γ ← activitiesContribute(b.actitivitiesNum)
6:         δ ← periodContribute(b.period)
7:         for all i ∈ b.results.length do
8:             scores[i][b.results[i]] ← scores[i][b.results[i]] + γ/δ
9:             MAXs[i] ← max(MAXs[i], scores[i][b.results[i]])
       return MAXs
```

Section 4.1.2 explains how the two contributes are calculated in the implemented prototype.

3.5 Classifiers

In this section, each classifier supported by the system is described. There are both binary models, where the output is either 0 or 1, and multi-class classifiers that classify instances into one of N classes, with $N > 2$. So, here the output ranges from 0 to $N - 1$ and each integer represents a class. In this way, all the models share a common structure for the representation of their results. The mapping from the integer to the descriptive category is carried out later when the user's result is requested.

A big issue in dealing with user profiling from social media is the lack of data. Indeed, even though many different aspects of a person have been studied, only a limited number of papers published the datasets they used or built for that research. Since not enough data to train and evaluate a machine learning model is available, artificial intelligence solutions are not always viable. For this reason, as introduced before, many of the algorithms presented here do not follow a machine learning approach but a more standard one.

3.5.1 MBTI Personality traits

As largely introduced in Chapter 1 the *Meyers-Briggs Type Indicator* model was chosen to extract the psychological characteristics of people. Thanks to the availability of a labelled dataset, later presented in Section **??**, the extraction of each trait was mapped to a binary classification problem. This model is organized in four different personality traits which consist in four binary functions: Extrovert/Introvert, Sensor/Intuitive, Thinker/Feeler and Judger/Perceiver. The meaning of each class is explained in Section 1.2.2

Among the extracted features, these four classifiers use especially the textual ones. A clear drawback is that, since the majority of these features are language-dependent, the models are trained on a specific language and their application on profiles that do not use that specific one would obviously bring to unreliable results. To solve this issue, a model for each supported language should be implemented and then results for each one should be merged together. Thanks to the algorithm showed in Section 3.4.5 this last aggregation is not a problem. So, the language-dependence of these classifiers is mainly due to the lack of usable data.

An important choice made for the classification of the MBTI personality model is that each one of the four cognitive functions is classified independently. Different approaches tried a multi-class classification on the six-teen personality types given by the combination of the 4 dichotomies. However, following the methodologies shown in the current state of the art, the 4 models consist in the same ML algorithm and are trained on the same set of features.

The output of each classifier is an integer representing the assigned value. For example, for the dychotomy *Extrovert/Introvert*, the value 0 means that the person is extrovert while 1 indicates introversion.

3.5.2 Daily Usage

This classifier says in which period of the day a person is more active on social networks. To infer social presence the posted activities are observed. These do not include only original posts wrote by the user but also every kind of retweets and replies.

To do so, the twenty-four daily hours are divided into time slots which can have a fixed and common duration or a variable one. Since the result of each classifier is a readable and actionable insight, each slot must be assigned a label indicating the specific activity period.

Once the temporal bands are defined, the algorithm that attributes one to each batch is quite simple. For each temporal slot is computed the percentage of tweets posted in that time frame. The one with the highest quantity of activities is kept and if its tweet's percentage is higher than a threshold α, the value corresponding to that slot is assigned. Else, if the threshold is not exceeded, a special value that classifies the person as a whole-day user is assigned.

3.5.3 Influence Role

This model classifies the users for the role they assume on the social in terms of influence, communication, and social engagement. As explained in Chapter 2, much has been done to infer how much a person is listened and influence with her activities on the social networks. Many different studies developed their own metrics which uses many measurements provided directly by the social to evaluate a general influence value. However, this approach does classify the user with a usable label but rather with a real value that needs to be interpreted.

So, this classifier aims to define well-defined classes of users with respect to their influence role inside their social network. Once the categories are chosen, the observation of the aforementioned measurements is necessary to classify the user into the right one.

The approach followed here was to identify a number N of categories and sort them from the one that represents less influence users to the one dedicated to influencers. Then, it is necessary to define which metrics the algorithm will take into account and a condition that each one may satisfy or not. For example, it could observe the number of followers and check if it greater or lower than ten thousand. Depending on the check over this condition, the total score of the user that is being classified is updated. When all checks have been done, the users is assigned the

label depending on its final score. Indeed, each class covers a subset of the values the score can assume. So, observing the score, the class is finally inferred.

3.5.4 Language and Communication Style

These two classifiers cover different aspects but their logic is extremely similar. The language one classifies the user with respect to the language he or she most used on their posts. The second one observes how a person tends to compose his tweets in terms of which means are used to communicate messages and emotions. For example, it can identifies users that communicate with images rather than raw text.

For the former, it is not necessary to define a list of classes because it directly uses the *IETF language tags* to decide which language is the most used one. A language tag that identifies human languages. Anyhow, this algorithm includes an extra category in case it is not possible to identify a dominant language in the user's communication. In this case, the user is classified as *polyglot*. To verify if a person prefers a specific language, their usage percentage is checked. The higher one is chosen as the *candidate*; if this exceeds a minimum threshold β, then its label is assigned. Else, the polyglot one is.

The latter works similarly but it requires the definition of the classes. Once the classes are defined and it is decided which parameters observe, for example, if a person uses photos or videos, the label can be easily assigned observing the usage percentage of these parameters.

3.5.5 Sentiment

This last model regards the interpretation and classification of emotions. It allows to identify users sentiment in their daily communication on the social networks. Conducting this type of analysis on social media allows using both the text component and the visual one to infer someone's polarity. There is a number of different services that take as input a piece of text or an image and return a corresponding sentiment value. Section 4.1.1 explains how this system implemented it, which activity's components have been used and the reasons of these choices.

In general, three label are used to describe people's polarity: *negative*, *neutral*, and *positive*. To decide which label assign to each user, it is enough to define the range covered by each class and observe the average sentiment value.

3.6 Exposed APIs

Finally, the endpoint offered at the user by the system are presented. During the design of the interfaces, the principles of the RESTful architecture have been followed. The resources publicly exposed are two: **User** and **New Activities Download**. Regarding the user, it is possible to create new instances, modify, or delete existing ones. A user is basically a list of social profiles and a name to be identified. Each social profile requires a user id and an access token. When a new user is added to the system, it results as non-classified. The **New Activities Download** allows you download and classify her social profiles. At this point, retrieving the user instance it will be classified and the extracted insights about her will be returned. The complete API specification can be found on Apiary:

`/uhopperthesis.docs.apiary.io`

4 Implementation

This chapter describes the implementation of a prototype of the system. It shows how each component was implemented to deal with the requests' limits of external services. Then, the interaction between them is explained showing the data flow starting with the user request to the final response. Finally, the techniques used for each classifier are introduced.

4.1 Components interactions

Generally, the system is composed by a main request which mobilizes the download of a user's profiles, their analysis and finally their classification. This process ends with a series of classified batches, of monthly time duration, that are stored in the database. Then, the call to obtain the final results starts the insight generator shown in Section 3.4.5 which merges together the partial results and return the final insight at the user level.

These different types of requests are handled by the request handler which is represented by the user's endpoint, the only interface to the outside of the system. As said before, it also allows to add and modify the system's users. The download request is the one that requires communication between multiple components.

This prototype allows the interaction and the download of social profiles only from Twitter. Indeed, while data from Twitter are publicly accessible through its APIs, other social media, such as Facebook and Instagram, require an access token for each consulted user. To obtain the access token, a demo video that shows what the system does must be provided, the data treatment has to be described precisely, and each person should accept to share its data.

4.1.1 Download Request

The endpoint /newactivities represents the most complex one that involved the majority of the components. In this prototype, the components execute synchronously. In the moment the request is received, the activities collector is executed and the system waits for it to end and return its results. Then, all the downloaded activities are analysed. Once all the features are extracted, they are divided in batches and finally classified and stored.

The process starts with the ID of a user which is used to search on the database the Twitter ID associated with that specific user. This is used by the activities collector to download the new content. As said before, the system is structured to download only new activities and not the whole timeline. To do it, it is used the ID of the oldest activity of the second-to-last stored batch because the last one could contain only a part of the content posted in a month. The drawback of this choice is that activities already contained in the last batches are downloaded and classified again but it is necessary to ensure a full download. Moreover, a maximum of thirty days is downloaded twice which is a limited number compared to the whole user's timeline. The collector uses *Tweepy*, a Python library, to access the Twitter API. It allows to download a user's content specifying its ID and the point from where the download of its activities should start. Once the download is complete, an object **User** is created. It contains both social information about the profile, such as creation date and description, and the list of all the fetched activities. Each one is represented by a **Post** object that contains everything about the downloaded content: creation date, number of likes and retweets, its text, list of media, hashtags and urls. The user

object is passed to the next component, the analyser, with a standard class method call by the request handler.

The analyser is the component that extracts, starting from the single activities, the significant features. Considering the various classifiers implemented, different types of features are extracted.

First of all, the information contained into the fetched tweet object that does not require any processing or further analysis is kept also in the analysed Post. This includes its id, the creation date, the number of favourites and retweets, the language of the tweet, and its type. There are 4 different types of tweets: original, reply, retweet, and quote.

The text component of the post is used to obtain fundamental textual features. This analysis is carried out using *SpaCy*, an open-source Python library for natural language processing. Before executing the Spacy pipeline, some pre-processing is done. Whitespaces are normalized, multiple spaces and new lines are treated as a single space. The number of capital letters is memorized and then the text is converted completely in small caps. Then, the text is passed through all the components of the pipeline. It starts with the *tokenizer* which segments text into tokens like words and punctuations marks. The *lemmatizer* is now involved to reduce each word to its canonical form, called **lemma**. Then, each token is assigned a part-of-speech tag thanks to the *tagger*. At this point, 2 custom components are added. First, the extension *spacymoji* is used to handle emojis efficiently. Then, the last component handles #hashtags and @mentions so that the lemma does not contain special symbols # and @. The result of this pipeline is a list of tokens that represent the bag of words extracted from the text. Each token contains information such as the raw word, its lemma, and its POS tag. Now, other more generic features are extracted. spacy allows counting automatically the number of sentences, words, and characters. Starting from this data, many averages are computed such as number of words per sentence and characters per word. To conclude, the tokens extracted before are divided into three different sets: words, stop words, emojis, and punctuation. This distinction is important because we that our classification models could treat each set differently. Stop words refer to the most common words in a language. There is no single general list of stop words and any group of words can be chosen. Spacy provides its own list for each supported language. The punctuation set includes all the standard punctuation marks but # and @ that are often special symbols on the social networks. Finally, the words set contains all the other words. Each set is memorized as a dictionary **key** : **value** used as a counter where **key** is the text of the token and **value** is its number of occurrences in the tweet. Some of the pipeline operations described, such as the lemmatizer, the tagger, and the stop words list, are language-dependent. In this prototype, the Italian module of Spacy has been used so these three specific phases are optimized only for that language. Anyway, Spacy provides plenty languages models and also a multi-language one.

Another important group of features is that describing the content of a tweet. For this purpose, *DandelionAPI*, a semantic analysis service was used. It works on unstructured text to extract its meaning. In particular, it offers two important functionalities: entity extraction and sentiment analysis. The first one can be useful to understand a person's interest and therefore is not used in this prototype which does not implement that specific classifier. The second one may have an important role with respect to psychological characteristics. It allows identifying whether an opinion contained in the tweet is negative, neutral, or positive. DandelionAPI is accessed with restAPIs and one request is necessary for each piece of text that need to be analysed. Moreover, it supports more than 40 languages, including Italian and English, and it can also compute automatically the language detection of the text given as parameter. The response is structured in JSON and contains: the detected language, a sentiment score and a sentiment type. The score is a more precise indicator ranging from -1.0 (totally negative) to 1.0 (absolutely positive) while the type is a descriptive label which can be *negative*, *neutral*, or *positive*.

Finally, the set of features regarding the media entities is created. Twitter supports three types of media: photos, videos and animated gifs. They are generally treated as *media entity*.

The TwitterAPI returns, for each entity, a exhaustive series of descriptive attributes. Of interest for this prototype are the media type and the media URL. The first one is important to understand in which way a person ten to communicate while the second one represents an important part of the tweet's content. They may be analysed in order to detect emotion, entities and, places in a way similar at that done for the text. So, even though images and videos are not used in this demo, mainly because rarity of free computer vision services, the URLs are downloaded to easily allow future improvements.

Once all the features are extracted, the same **User** object taken as input is modified replacing the list of **Post** object with a list of features which represents the analysed tweets. This new object is passed to the next component, the aggregator.

The aggregator has to goal to divide the posts in batches and then compute the aggregates for each batch. As introduced before, in this prototype the batches are temporality-based and each one ranges over a period of one month. Thanks to the creation date of the posts, it is easy to divide them. Then, the information related to the social profile is copied into each batch. At first each batch represent a partial user composed by its descriptive data and its analysed timeline. The aggregator algorithm is executed on each batch to obtain a final classifiable object. The system includes many various features structured in many different formats. So, there is no standard way in which these are aggregated together. For different typologies, different approaches were applied. For example, information about the number of favourites and retweets is given from the simple average all tweets. Textual features about the different sets of tokens are merged by summing together the single count dictionary of each activity. Then, the counters are converted to frequency with respect to the totality of words. Two lists are used to count the percentage of each typology of tweets and the time it was posted. Concerning the first one, four elements are enough. Differently, for the second one, it depends on which information is considered necessary. Here is used a list of length twenty-four to keep count at which specific hour of the day the activities are posted. In the end, the list of batches is returned and given as input to the classifiers.

The classifiers are the last step of this request. Since they share a common architecture, they take as input the same object and return the output result with a common structure. The input of each classifier is a batch, or a list of batches. The output is usually an integer that represents the class assigned by that model. In this prototype, the classification algorithms are executed sequentially one after the other. Each result is stored in an dictionary **key** : **value** where **key** is a label that identifies the classifier. For example, it can be *EI* to indicate the result that regards the classification of the Extrovert/Introvert aspect or *Sentiment* which is the name of the classifier that performed the person's general sentiment. The **value** is an integer indicating the specific assigned class. An object of this type is returned for each batch and it's integrated with other descriptive information: the month covered by the batch in the format *MM-YYYY*, the number of tweets contained, and the ID of the last posted one. As introduced in Section 3.5, lack of data is a serious problem that limits the use of machine learning techniques. So, only for a limited number of classifiers these techniques have been used. For the others, the implementation followed the algorithms explained in Section 3.5.

4.1.2 GetUser request

This section describes how the second important request was implemented. When this request is called on a classified user, its partial results of every batch are merged together applying an implementation of the algorithm 1 and then returned. As parameters, it requires only the ID of the system user. It returns in output, structured as JSON, the user characterised by its information and its final insights.

This involves only one component, the insight generator. It has to query the database to obtain all the partial results of the searched user and then run the algorithm described in Section 3.4.5. In this prototype, each batch covers a period of exactly one month. The values γ

and δ are computed as follows:

$$\gamma = \lceil((actualYear * 12 + actualMonth) - (batchYear * 12 + batchMonth)/2)\rceil \quad (4.1)$$

$$\delta = batchActivities \quad (4.2)$$

So, δ consists in only the number of activities contained in a specific batch. γ is an integer that takes into consideration the number of months between the actual one and the one covered by the batch. This value starts at one and increase of one every two months.

Then the algorithm is applied with these two values, that are calculated at every iteration. Finally, the final object representing the user can be returned.

4.2 User dashboard

Finally, a dashboard to see the results and evaluate the system's efficiency was realized. It is a web client that allows searching for a user and observe how his or her insights changed over time.

The application is realized with *HTML*, *JS* and uses *AJAX* to communicate with the system. The dashboard send a getUser request 4.1.2 and display the results in a readable way. For each classification, a stepped line graph is used. On the x-axis, there are the months covering the whole user's timeline. On the y-axis, there are the classes depending on which classification is being observed.

Figure 4.1 shows a view of the dashboard for a random user. The displayed classification is that of the used language. It is observable that the user tend to communicate only in Italian and English.

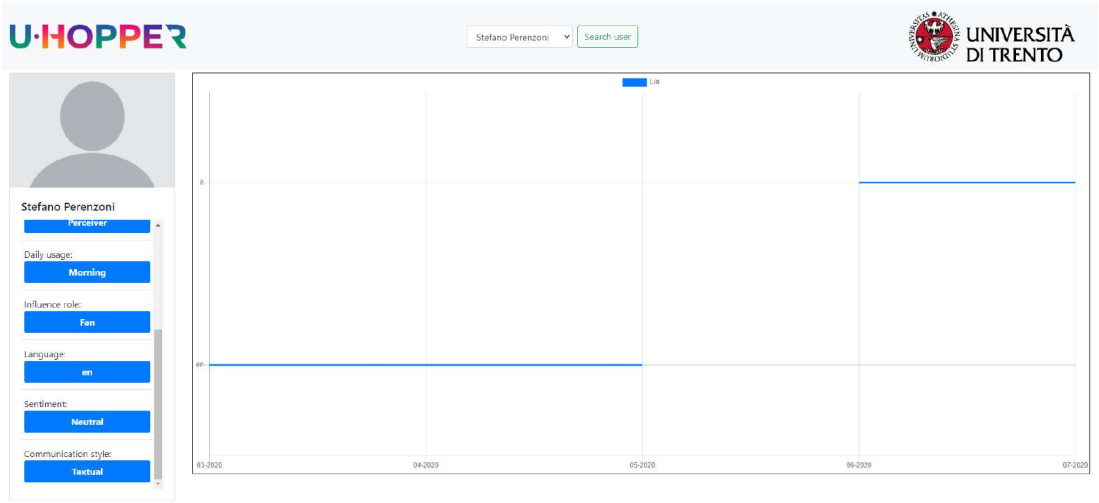

Figure 4.1: A view of the user dashboard

5 Evaluation

This last chapter presents how the prototype have been evaluated to observe if and how it meets the initial research question. Performance evaluation for the classifiers is presented. Finally, the Section 5.3 explains how the extracted insights can impact a business and help it in the interaction with its customers.

On the one hand, classifiers for the four dimensions of the MBTI personality model were evaluated using confusion matrices. A confusion matrix is a $2x2$ table used to observe the results of a binary classification algorithm. In general, along its columns it contains the values predicted by the machine learning algorithm. The rows represent the actual class of the samples. Figure 5.1 shows how the table is structured. The output of a binary classification can be either 0, or *negative* class, or 1 *positive* class. So, the 4 cells of the matrix assume these names: at top-left there is the *true positives* (TP) counter, the top-right is the counter for the *false negative* (FN), the bottom-left one for the *false positives* (FP), and bottom-right for *true negative* (TN). Totally, their sum is equal to the number of samples classified. From this table, 4 measures as usually extracted:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$
$$\text{Precision} = \frac{TP}{TP + FP}$$
$$\text{Recall} = \frac{TP}{TP + FN}$$
$$\text{F}_1\text{-Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

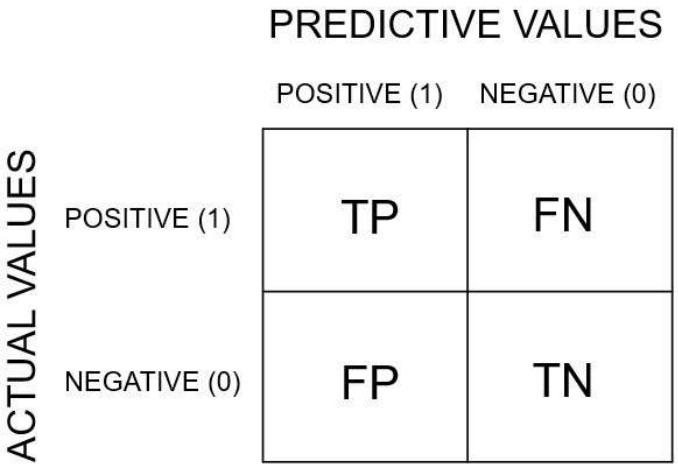

Figure 5.1: Generic schema of a confusion matrix

5.1 MBTI Classifiers evaluation

Regarding these 4 classifiers, no particular setups have been applied. Decision trees have been chosen as classifiers, according to the current state of the art. Naive bayes classifiers are also usually applied for such purposes but were discarded due to bad performance with the linguistic features.

The performance evaluations were performed on the same dataset used to train the four classifiers. The dataset was split into a training set and a test one using the default values of the scikit-learn library. So, the training set contained 75% of the samples and the test set the remaining 25%. The used features include some information from the social profile, such as the number of followers and friends, the use of hashtags, mentions, and URLs, and other information describing the text component, such as the number of sentences and word and POS tags. No particular configuration have been used for what regards *bags of words* or *n-grams*.

The measure observed to compare the performance with that of the current state of the art is the **accuracy**. Accuracy is the ratio of correct prediction to total predictions made. It is presented as a percentage. Accuracy is preferred since, for each of the four classifications, there is not discrimination between samples with a specific outcome from normal observation. So, even though, taking as example the cognitive function *Extroversion/Introversion*, extroversion is labelled as the negative class and introversion as the positive one, it is not the case where we want to reduce the number of false-positive rather than false-negative because both classes are treated the same way. So, metrics such as precision or recall tend to be avoided and not considered in the state of the art.

Even though only standard methodologies have been applied, the classifiers had discrete results. Table 5.1 compares the obtained results with that of the state of the art [20]. Compared to those obtained by Lima et al., the prototype's performances are slightly worse. It should be said that the study considered reached its best results using a particular set of psychological features extracted with services such as the *Linguistic Inquire and Word Count* (LIWC). But, as said in Section 1.3, this thesis focuses more on the actionability of the extracted insights rather than their accuracy and reliability.

Classifiers	Extr/Intr	Sensation/Intuition	Think/Feel	Judging/Perceiving
System accuracy	77.6%	81.8%	76.9%	74.3%
SOA accuracy	82.0%	88.3%	80.57%	78.26%

Table 5.1: Table.

5.2 Non-ML insights observation

While for the four classifiers of the personality traits there is an availability of data for both the training and the evaluation of ML models, this is not possible for the other classifiers which are based on non-machine learning algorithms, described in Section 3.5. To observe and evaluate their results, the user dashboard introduces in Section 4.2 has been used.

To do it, nearly twenty twitter profiles were downloaded and classified for a total of more than forty thousand activities. Their results have been displayed over time thanks to the batch-based architecture and then observed individually.

Some interesting observations can be done comparing the results for two, or more, different profiles. For example, two of the downloaded profiles, which comes from a similar environment, are the one of the **University of Trento** (`https://twitter.com/UniTrento`) and that of the **Department of Information Engineering and Computer Science** (`https://twitter.com/UniTrento_DISI`) show some interesting differences that deserve to be observed to understand better at which point the system can actually be useful and usable. For each profile, the last two thousand activities have been fetched. In Figure 5.2, the daily usage insights of these two profiles are compared.

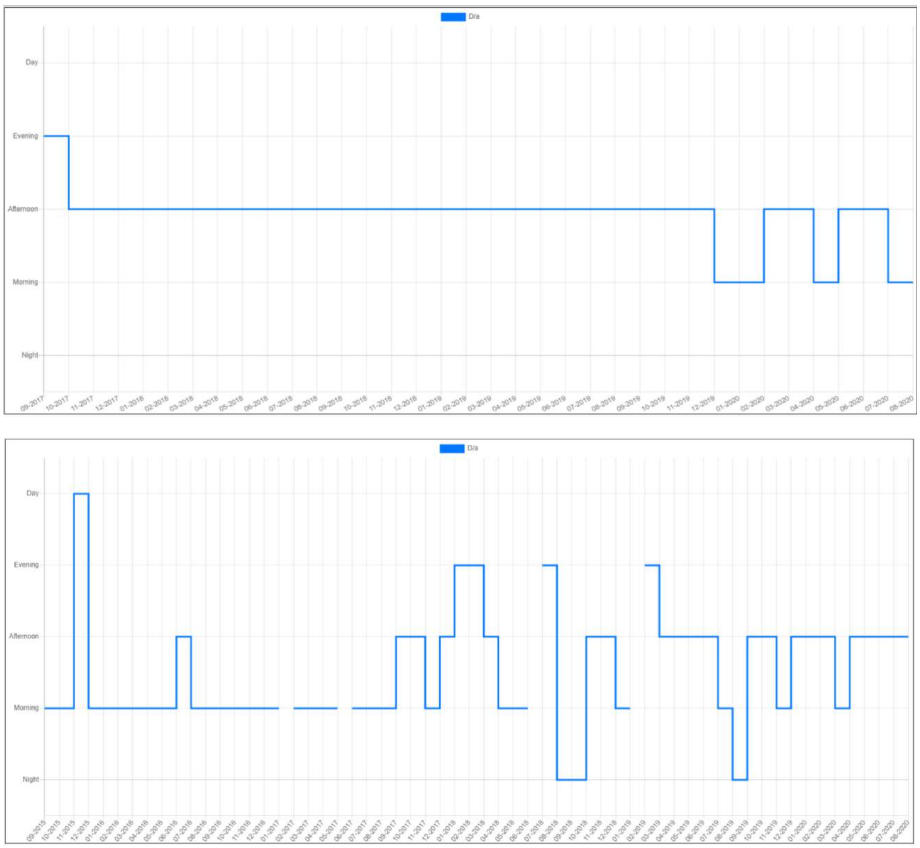

Figure 5.2: Comparison between the **daily usage** insight of **UniTN**(1) and **DISI**(2)

First of all, it can be noticed that the two profiles tweet with different frequencies. Indeed, the last two thousand activities for the profile of the University of Trento covers thirty-five months, which brings to an average of fifty-seven tweets per month, almost two per day. Differently, the same number of tweets ranges over fifty-five months for the Department of Information Engineering that corresponds to thirty-six activities per month and just over one tweet a month. The latter also shows some months of no activity at all. During these periods the graph has no value and loses its contuinity.

Then, looking at the values each graph has, we can observe how the two profiles differ much in terms of variability. The UniTN one is active especially during the afternoon, from 12:00 to 18:00, with some rare periods of morning activity. On the other hand, the profile of the department fluctuates much more. It spreads its tweets particularly between morning and afternoon, so between 6:00 and 18:00. However, it also shows some activities at evening and night.

One could use these type of comparisons to draw its conclusion in order to identify what's the difference between two or more profiles. For example, in this case, we can assume that the University of Trento pays more attention to social reach, a metric that measures how many users came across its tweets. Indeed, it has been proved that the best time to post on Twitter is during the afternoon [5]. Contrarily, the profile of the department, which we can assume being more international rather than national, does not have a particular time where the reach is higher; probably because of the different time zones of the followers.

The extracted attributes can then be observed in groups to find other correlations. Continuing with the example and the assumption about the internationality of the two profiles. Observing and comparing the insights about the language in Figure 5.3, it can be said that the institute tends to talk only to their Italian students since their communications are done only

[5] `https://neilpatel.com/blog/is-there-a-generic-best-time-to-post-on-social-media-platforms/`

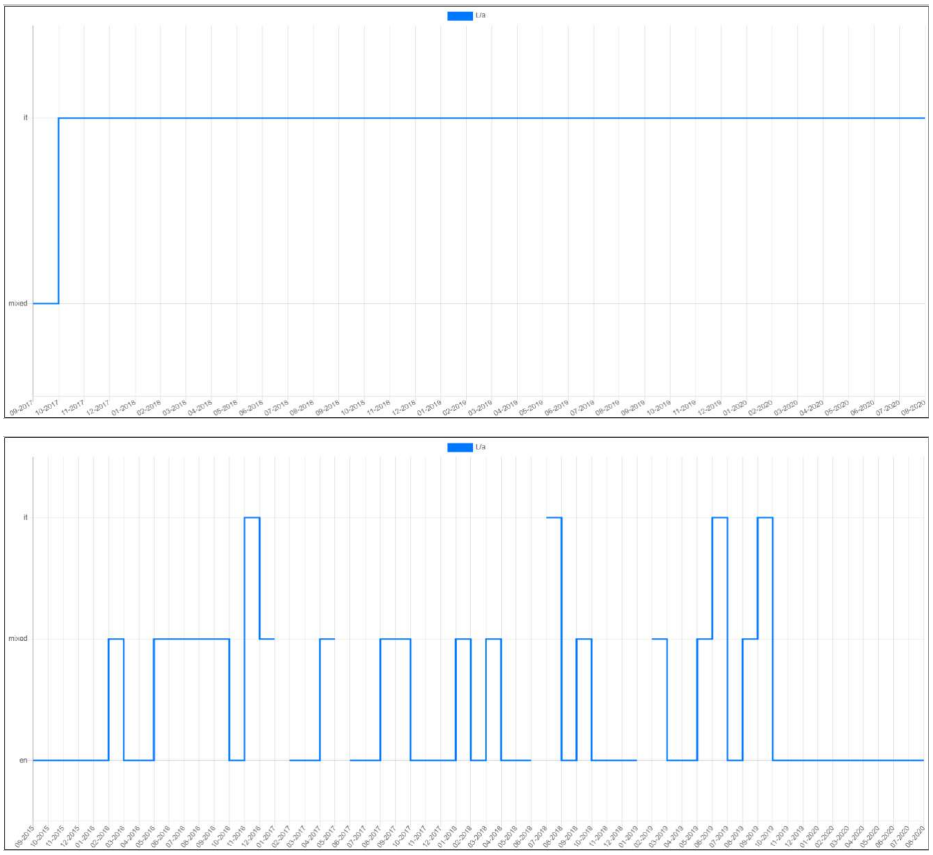

Figure 5.3: Comparison between the **language** insight of **UniTN**(1) and **DISI**(2)

in Italian while the department has to meet the language requirements of different nationalities and so English is highly preferred.

5.3 Insights' actionability

An important goal that this thesis aimed to meet was to extract actionable insights. Insights capable of helping a company during their interaction with its customers. This short section gives some examples of how the implemented classifiers can actually help a business.

Starting with one of the four personality traits, in particular with the Extrovert/Introvert dichotomy, such information can be used, for example, in any marketing campaign to propose the customer a suited offer. For instance, introvert people may prefer something that does not involve any particular human relations while extrovert ones could be more attracted by group activities.

Some more obvious aspects are those regarding the language and daily usage of a person. The first one can be trivially useful to a company to relate with their customers with a specific language. Then, the second characteristic can help the customer-company relation. Assuming that people tend to use social in their leisure, contacting them during these periods can encourage communications.

Differently from the previous ones, the insights about the communication style can also help adapting a particular service or product. For example, graphical interfaces may modify their views to completely satisfy the esigenze of each user. Indeed, someone could prefer images and videos while others like detailed pieces of text.

Moreover, the display of the results compared to time provided by the dashboard adds another degree of reading. Indeed, taking as example the daily usage, it is easy to observe if there are patterns that repeat monthly.

6 Conclusions

6.1 Limitations

6.2 Future work

References

[1] Muhammad M Abdul-Mageed et al. "Recognizing pathogenic empathy in social media". In: *Eleventh International AAAI Conference on Web and Social Media*. 2017.

[2] Jonathan S Adelstein et al. "Personality is reflected in the brain's inoptistrinsic functional architecture". In: *PloS one* (2011).

[3] Yoram Bachrach et al. "Personality and patterns of Facebook usage". In: *Proceedings of the 4th annual ACM web science conference*. 2012, pp. 24–32.

[4] John E Barbuto Jr. "A critique of the Myers-Briggs Type Indicator and its operationalization of Carl Jung's psychological types". In: *Psychological Reports* 80.2 (1997), pp. 611–625.

[5] Jack Block. *Personality as an affect-processing system: Toward an integrative theory*. Psychology Press, 2002.

[6] Gregory J Boyle. "Myers-Briggs type indicator (MBTI): some psychometric limitations". In: *Australian Psychologist* 30.1 (1995), pp. 71–74.

[7] Meeyoung Cha et al. "Measuring user influence in twitter: The million follower fallacy". In: *fourth international AAAI conference on weblogs and social media*. 2010.

[8] Bongsug Kevin Chae. "Insights from hashtag# supplychain and Twitter Analytics: Considering Twitter and Twitter data for supply chain practice and research". In: *International Journal of Production Economics* 165 (2015), pp. 247–259.

[9] Aron Culotta, Nirmal Ravi Kumar, and Jennifer Cutler. "Predicting the Demographics of Twitter Users from Website Traffic Data." In: *AAAI*. Vol. 15. Austin, TX. 2015, pp. 72–8.

[10] Thomas Dickinson et al. "Identifying prominent life events on twitter". In: *Proceedings of the 8th International Conference on Knowledge Capture*. 2015, pp. 1–8.

[11] Golnoosh Farnadi et al. "A multivariate regression approach to personality impression recognition of vloggers". In: *Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition*. 2014, pp. 1–6.

[12] Adrian Furnham. "The big five versus the big four: the relationship between the Myers-Briggs Type Indicator (MBTI) and NEO-PI five factor model of personality". In: *Personality and Individual Differences* 21.2 (1996), pp. 303–307.

[13] Sharath Chandra Guntuku et al. "Studying personality through the content of posted and liked images on Twitter". In: *Proceedings of the 2017 ACM on web science conference*. 2017, pp. 223–227.

[14] Leaetta M Hough and Adrian Furnham. "Use of personality variables in work settings". In: *Handbook of psychology* (2003), pp. 131–169.

[15] Carl G Jung. "Personality types". In: *The portable Jung* (1971), pp. 178–272.

[16] Margaret L Kern et al. "Gaining insights from social media language: Methodologies and challenges." In: *Psychological methods* 21.4 (2016), p. 507.

[17] Michal Kosinski, David Stillwell, and Thore Graepel. "Private traits and attributes are predictable from digital records of human behavior". In: *Proceedings of the national academy of sciences* 110.15 (2013), pp. 5802–5805.

[18] Jingxuan Li et al. “Social network user influence sense-making and dynamics prediction”. In: *Expert Systems with Applications* 41.11 (2014), pp. 5115–5124.

[19] Ana Carolina ES Lima and Leandro N de Castro. “Predicting temperament from Twitter data”. In: *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*. IEEE. 2016, pp. 599–604.

[20] Ana Carolina ES Lima and Leandro Nunes de Castro. “TECLA: A temperament and psychological type prediction framework from Twitter data”. In: *PloS one* 14.3 (2019).

[21] François Mairesse et al. “Using linguistic cues for the automatic recognition of personality in conversation and text”. In: *Journal of artificial intelligence research* 30 (2007), pp. 457–500.

[22] Robert R McCrae and Paul T Costa. “Validation of the five-factor model of personality across instruments and observers.” In: *Journal of personality and social psychology* 52.1 (1987), p. 81.

[23] Robert R McCrae and Oliver P John. “An introduction to the five-factor model and its applications”. In: *Journal of personality* 60.2 (1992), pp. 175–215.

[24] Zachary Miller, Brian Dickinson, and Wei Hu. “Gender prediction on twitter using stream algorithms with n-gram character features”. In: (2012).

[25] Isabel Briggs Myers and Peter B Myers. *Gifts differing: Understanding personality type*. Nicholas Brealey, 2010.

[26] Clifford Nass and Kwan Min Lee. “Does computer-generated speech manifest personality? An experimental test of similarity-attraction”. In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 2000, pp. 329–336.

[27] Mª Ángeles Oviedo-García et al. “Metric proposal for customer engagement in Facebook”. In: *Journal of research in interactive marketing* (2014).

[28] Wendy Patton and Mary McMahon. *Career development and systems theory: Connecting theory and practice*. Vol. 2. Springer, 2014.

[29] Marco Pennacchiotti and Ana-Maria Popescu. “A machine learning approach to twitter user classification”. In: *Fifth international AAAI conference on weblogs and social media*. 2011.

[30] James W Pennebaker and Laura A King. “Linguistic styles: Language use as an individual difference.” In: *Journal of personality and social psychology* 77.6 (1999).

[31] Daniele Quercia et al. “Our twitter profiles, our selves: Predicting personality with twitter”. In: *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*. IEEE. 2011, pp. 180–185.

[32] Daniel M Romero et al. “Influence and passivity in social media”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2011, pp. 18–33.

[33] Xianzhi Ruan, Steven Wilson, and Rada Mihalcea. “Finding optimists and pessimists on twitter”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2016, pp. 320–325.

[34] Arthur L Samuel. “Some studies in machine learning using the game of checkers”. In: *IBM Journal of research and development* 3.3 (1959), pp. 210–229.

[35] H Andrew Schwartz et al. “Extracting human temporal orientation from Facebook language”. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2015, pp. 409–419.

[36] H Andrew Schwartz et al. “Personality, gender, and age in the language of social media: The open-vocabulary approach”. In: *PloS one* 8.9 (2013), e73791.

[37] Chris Sumner et al. “Predicting dark triad personality traits from twitter usage and a linguistic analysis of tweets”. In: *2012 11th International Conference on Machine Learning and Applications*. Vol. 2. IEEE. 2012, pp. 386–393.

[38] Yla R Tausczik and James W Pennebaker. “The psychological meaning of words: LIWC and computerized text analysis methods”. In: *Journal of language and social psychology* 29.1 (2010), pp. 24–54.

[39] Christian Torrero, Carlo Caprini, and Daniele Miorandi. “A Wikipedia-based approach to profiling activities on social media”. In: *arXiv preprint arXiv:1804.02245* (2018).

[40] Reut Tsarfaty. “The Natural Language Programming (NLPRO) Project: Turning Text into Executable Code.” In: *REFSQ Workshops*. 2018.

[41] Paul Voigt and Axel Von dem Bussche. “The eu general data protection regulation (gdpr)”. In: *A Practical Guide, 1st Ed., Cham: Springer International Publishing* (2017).

[42] Michael Wilson. “MRC psycholinguistic database: Machine-usable dictionary, version 2.00”. In: *Behavior research methods, instruments, & computers* 20.1 (1988), pp. 6–10.

[43] Mohammadzaman Zamani, Anneke Buffone, and H Andrew Schwartz. “Predicting Human Trustfulness from Facebook Language”. In: *arXiv preprint arXiv:1808.05668* (2018).