University of Trento
Center for Mind/Brain Sciences

Adaptive Personality Recogntion from Text

PhD:
Fabio Celli

Advisor:
Massimo Poesio

Cycle
XXIV

Adaptive Personality Recognition from Text

Fabio Celli

LaTeX

Contents

Abstract

We address the issue of domain adaptation for automatic Personality Recognition from Text (PRT). The PRT task consists in the classification of the personality traits of some authors, given some pieces of text they wrote. The purpose of our work is to improve current approaches to PRT in order to extract personality information from social network sites, which is a really challenging task. We argue that current approaches, based on supervised learning, have several limitations for the adaptation to social network domain, mainly due to 1) difficulties in data annotation, 2) overfitting, 3) lack of domain adaptability and 4) multilinguality issues. We propose and test a new approach to PRT, that we will call Adaptive Personality Recognition (APR). We argue that this new approach solves domain adaptability problems and it is suitable for the application in Social Network Sites.

We start from an introduction that covers all the background knowledge required for understanding PRT. It includes arguments

like personality, the the Big5 factor model, the sets of correlations between language features and personality traits and a brief survey on learning approaches, that includes also feature selection and domain adaptation. We also provide an overview of the state-of-the-art in PRT and we outline the problems we see in the application of PRT to social network domain.

Basically, our APR approach is based on 1) an external model: a set of features/correlations between language and Big5 personality traits (taken from literature); 2) an adaptive strategy, that makes the model fit the distribution of the features in the dataset at hand, before generating personality hypotheses; 3) an evaluation strategy, that compares all the hypotheses generated for each single text of each author, computing confidence scores. This allows domain adaptation, semi-supervised learning and the automatic extraction of patterns associated to personality traits, that can be added to the initial correlation set, thus combining top-down and bottom-up approaches.

The main contributions of our approach to the research in the field of PRT are: 1) the possibility to run top-down PRT from models taken from literature, adapting them to new datasets; 2) the definition of a small, language-independent and resource-free feature/correlation set, tested on Italian and English; 3) the possibility to integrate top-down and bottom-up PRT strategies, allowing

the enrichment of the initial feature/correlation from the dataset at hand; 4) the development of a system for APR, that does not require large labeled datasets for training, but just a small one for testing, minimizing the data annotation problem.

Finally, we describe some applications of APR to the analysis of personality in online social network sites, reporting results and findings. We argue that the APR approach is very useful for Social Network Analysis, social marketing, opinion mining, sentiment analysis, mood detection and related fields.

Acknowledgments

We wish to thank François Mairesse and James Pennebaker for giving us access to their data annotated with personality; Luca Polonio for the Big5 test that allowed us to produce the gold standard for social network data; Massimo Poesio for the very useful advices; Luca Rossi for the contribution in the analysis of social network data and, last but not least, Valentina Perazzini for developing the website for the online experiment with subjects. We also wish to give her a special thanks for the patience and the support.

Chapter 1

Introduction

Personality Recognition from Text (PRT henceforth) consists in the automatic classification of authors' personality traits from pieces of text they wrote. This task, that is partially connected to authorship attribution, requires skills and techniques from several different disciplines, like Linguistics, Psychology, Data Mining and Communication Sciences. For instance, PRT requires some correlations between language features and personality traits, a solid background in Data Mining for feature selection and classification, a good knowledge of communication practices for experiment design and, most important, a formalized personality schema in order to define classes.

Most scholars, with some isolated exceptions, use the so called

"Big5" factor model, that describes personality along five traits formalized as bipolar scales. They are:

1) **Extraversion** (x) (sociable vs shy)

2) **Emotional stability** (e) (calm vs neurotic)

3) **Agreeableness** (a) (friendly vs uncooperative)

4) **Conscientiousness** (c) (organized vs careless)

5) **Openness** (o) (insightful vs unimaginative).

The bipolar scales are suitable for computational processing, because they can be turned into continuous (-1, 0, 1) or nominal (y, o, n) variables, as shown in figure 1.1. From a theoretical point

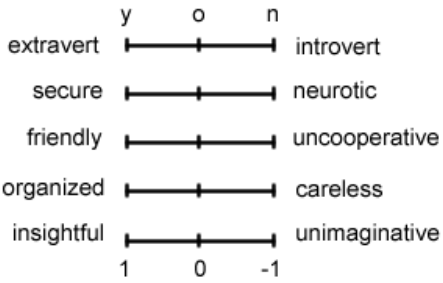

Figure 1.1: Formalization of Personality for computational purposes.

of view, it is very interesting to note that this way of formalizing the Big5 is a non-symbolic model that stands for a quality dimension, like representations in Conceptual Spaces (Gärdenfors 2004 [31], Gärdenfors & Williams 2001 [32]). This allows us the integration of formalized personality into a theoretical framework where it

can be linked to other dimensions, such as sentiment (see Cambria et Al 2010 [15]). From a more practical point of view, the extraction of personality only from text, without considering for example the prosodic or facial dimensions, is surely a limitation, but it can bring out important issues that have been so far underestimated, like the extraction of personality in communicative processes.

In recent years the interest of the scientific community towards automatic PRT has focused mainly 1) on the application of PRT to languages different from English (see Kermanidis 2012 [45] and Bai et Al 2012 [6]), and 2) on learning personality of users in social networks (see for example Quercia et Al. 2011 [64] and Golbeck et Al. 2011 [35]). This interest is due to the fact that PRT is very useful in Social Network Analysis and Opinion Mining, that are large and developing fields of research. Although online social networks are huge repositories of written data, suitable for PRT, there are some serious problems in sampling and using them. For instance, when it is not protected by privacy, social network data is 1) often not publicly available, 2) unlabeled, 3) very difficult to annotate with personality judgements and 4) in a lot of different languages.

In this work we address the issue of domain adaptation for automatic PRT. We provide an overview of what has been done in PRT, we outline the problems and the limitations of current approaches,

that are based on supervised learning, and we develop a new approach to PRT, that we will call Adaptive Personality Recognition (APR). The main contributions of our adaptive approach to PRT are:

1) the possibility to run top-down PRT from models taken from literature, adapting them to new datasets;

2) the definition of a small, language-independent and resource-free feature/correlation set, tested on Italian and English;

3) the possibility to integrate top-down and bottom-up PRT strategies, allowing the enrichment of the initial feature/correlation from the dataset at hand;

4) the development of a system for APR, that does not require large labeled datasets for training, but just a small one for testing, minimizing the data annotation problem.

In this chapter we will cover everything is needed in order to understand how personality recognition from text works. We will cover arguments from different disciplines, including machine learning, feature selection, domain adaptation and the psychological studies on personality. If the reader is familiar with these arguments can safely skip the corresponding sections. We included in the introduction all the background knowledge required to understand things presented in this work, replacing, where possible, complex formulas with plain explanations.

1.1 Personality

According to psychologists (DeYoung 2010 [25], Block 2002 [9]) and neuroscientists (Adelstein et Al. 2011 [2]), personality is an affect processing system that describes persistent human behavioural responses to broad classes of environmental stimuli, characterising a unique individual (Mairesse et Al 2007 [50]). It is involved in communication processes and connected to how people interact one another.

The Big5 factor model, introduced in psychology by Norman 1963 [57], emerged from empirical analyses of rating scales, and has become a standard over the years. The five bipolar personality traits, namely extraversion, Emotional Stability, Agreeableness, Conscientiousness and Openness, have been proposed by Costa & MacCrae 1985 [24]. Extraversion is bound to energy, positive emotions, surgency, assertiveness, sociability and talkativeness. Emotional stability is bound to impulse control, and is sometimes referred by its low pole: neuroticism that is the tendency to experience unpleasant emotions easily, such as anger, anxiety, depression, or vulnerability. Agreeableness refers to the tendency to be compassionate and cooperative rather than suspicious and antagonistic towards others. Conscientiousness is the tendency to show self-discipline, act dutifully, and aim for achievement; planned rather than spontaneous behaviour, organized, and dependable. Open-

ness to experience is bound to the appreciation for unusual ideas, to curiosity, and variety of experience. It often reflects the degree of intellectual curiosity, creativity and a preference for novelty and variety.

According to Digman 1990 [26], there has been a lot of studies in psychology that independently came to the conclusion that five are the right dimensions to describe personality. Despite there is a general agreement on the number of traits, there is no full agreement on their meaning, since some traits are vague. For example there is some disagreement about how to interpret the openness factor, which is sometimes called "intellect" rather than openness to experience.

The Big5 has been replicated in a variety of different languages and cultures, such as Chinese (Trull & Geary 1997 [72]) and Indian (Lodhi et Al. 2002 [47]). Some researchers, such as Bond et Al. 1975 [12] and Cheung et Al. 2011 [20] suggest that the Openness trait is particularly unsupported in asian cultures such as Chinese and Japanese, and that a different fifth factor is sometimes identified. Also the relationship between language and personality has been investigated (see Gill 2004 [33] for a survey), although yet there are few applications in PRT in languages different from English.

Detractors of the Big5, argue that the theoretical background

behind the five traits is weak, due to the fact that the research that brought to its development has been mostly empirical. Nevertheless there are recent developments in psychology that proposed higher order personality traits (see Digman 1997 [27]) and efforts toward a theory of personality that could better explain personality traits. For example Block 2002 and DeYoung 2010 argue that emotional stability and conscientiousness are related to "ego-control", the ability of maintain goals and decision-making, and openness and extraversion are related to "ego-resiliency", the ability to find new goals.

Despite all the problems and criticisms, the Big5 is nevertheless a formalization of personality suitable for computational and learning approaches. It is useful also for the fact that can be applied to many languages, which is the normal condition in social network sites. The only caution is to keep in mind that openness to experience is unsupported in eastern cultures.

Of course personality is also something that changes over time and adapts to the environment. For example, as DeYoung 2010 pointed out, goals, motivations and context influence the way people display their personality. People may also pretend to have different personality traits, and this is an aspect that has not been studied in detail and it is beyond the scope of our research. The general position of psychologists about these problems (that is also

at the basis of Adelstein et Al's work) is that individuals have some rather fixed core personality traits and other more variable peripheral traits.

Chapter 2

Machine Learning Techniques

Learning is the act of grouping together things that are similar and divide things that are not. This action can be turned into a function and formalized as a problem that a machine can compute and solve. Figure 2.1, adapted from Kotsiantis 2007 [46], shows a typical flow chart for a learning problem. In the preprocessing phase the data has to be defined in terms of instances, each one characterised by its own features. Classification algorithms can be exploited to generalize features of instances, producing models from data. These models then can be used for predictions on new

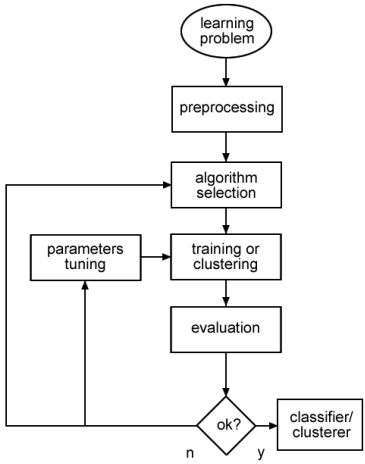

Figure 2.1: Flow chart of a learning problem.

data of the same type. Clustering algorithm instead group together similar instances according to their features without producing any model. In the case of a classifier, the model is evaluated on a labeled test set, while in the case of clustering, the evaluation has to be run post-hoc, manually or in other ways. If the classifier's or clusterer's performance is good (in other words is above the state-of-the -art or some baseline) the learning problem is solved, otherwhise we have to modify some parameters, like features or algorithm selection, until we achieve the desired performance.

Elements of a learning problem, as we have seen, are instances and features. Instances are objects characterized by some attributes,

called features, that can be used to distinguish them. A dataset is a set of instances defined by the same features. The goal is to group together similar instances using their features, raising the amount of knowledge we have about them. A learning problem can be formalized as a Cartesian coordinate system where the features are two dimensions (x and y axes), the instances are points in the space described by the coordinates and the classifiers or clusterers are functions dividing instances with a certain degree of correctness. The function can be computed using different type of algorithms (see section 2.1). The degree of error can be estimated or computed, and becomes the evaluation of how well the system solved the learning problem or, in other words, how well it learned to distinguish and classify instances.

There are many conditions under which one can try to solve a learning problem. Those conditions are for example data collection (few instances or many instances); feature types (numerical, nominal, boolean); amount of information about data (labeled or unlabeled data), type of variables to be learned (nominal, like classes, or real valued, like scores). The type of variable to be learned for example affects the learning technique that can be used (classification and clustering can be used for predicting/grouping nominal data, regression and density estimation for numerical data). The size of the collected data affects the predictive power of the model

learned and the feature types affect the choice of the algorithm (for example probabilistic algorithms usually work better with numerical values rather then nominal ones). Labeled or unlabeled data affect heavily the way we can evaluate the performance of the systems.

There are four main learning approaches in Computational Linguistics and Information Extraction in general, in the following paragraphs we will give a theoretical overview of them, followed by some examples, useful to understand how to select algoritms and approaches to some learning problems under different conditions.

Supervised learning. The supervised approach is the common way to solve a learning problem when there are labeled datasets available. In the supervised approach some models are learned from labeled data using learning algorithms and tested against gold standard labeled data (see Kotsiantis 2007). The learned models are functions that can be used to make predictions on new data with the same features. This approach usually yields good results and it has been widely exploited in Personality recognition as well as in many other learning tasks. The drawbacks in the supervised approach are i) issues related to overfitting the dataset, which come out if the model is too detailed or the dataset is too small; ii) the fact that classes must be decided a-priori, before extracting the models and iii) the fact that producing labeled datasets is often

expensive and time-consuming, and sometimes it is very hard or even infeasible.

Unsupervised learning. The unsupervised approach is useful in case there is no labeled data available or there are no predefined classes (see Grira et Al. 2005 [37]): basically a clustering algorithm is applied to unlabeled data to group similar instances together without the need to extract a model from it. Unsupervised learning makes use of clustering and density estimation techniques. The former can be used for nominal data and the latter for numerical data. Common problems in unsupervised methods have to do with i) deciding the number of clusters to work with, ii) selecting the similarity measure to use and iii) the nature of clusters (fuzzy vs crisp, 1-leveled vs hierarchical). An unsupervised learning procedure is usually more difficult to evaluate than a supervised one because there is no labeled data available. Validation procedures can be the measure of variation inside clusters (enthropy and purity) or against data labeled a-posteriori.

Semi-supervised learning. The semi-supervised approach is very useful for those learning problems where there are classes, a lot of unlabeled data and labeled data is difficult to obtain. Under this approach a small number of seed labeled examples are exploited to label a large number of unlabeled data. According to Abney 2008 [1], it is really important to understand and match data struc-

ture in order to select good seed labeled examples and improve the performance of a classifier or a clusterer with unlabeled data. In literature (see Zhu 2005 [79]) there are many ways to perform semi-supervised learning, depending on the learning problem conditions: i) self-training (see for example Yarowsky 1995 [77]) can be a good choice if there are supervised models that achieve high accuracy on the learning problem. It consists in selecting the best labeled instances using a confidence score in order to iteratively re-train a classifier. ii) Co-training (see Blum & Mitchell 1998 [11]) can be a good choice if the feature set naturally splits in two parts. It consists in using different parts of the feature set to train two independent classifiers on the labeled data. The instances on which the classifiers' predictions agree can be exploited to re-train new classifiers. iii) Label propagation can be used when clustering has a good performance on the dataset. This method consists in clustering labeled and unlabeled instances, then exploiting the labeled ones in order to assign lables to cluster, turning them into classes. iv) Graph based methods (see Blum & Chawla 2001 [10]) can be useful when instances with similar features are mainly put in the same class. Graph based methods consist in propagating class labels from the labeled instances to the unlabeled ones according to similarity and distance between instances. v) Self Taugth Learning, proposed by Raina et Al. 2007 [65], is based on the idea of

transforming basic features into more informative ones using unsupervised techniques, and then solving the learning problem exploiting the new features to train a supervised classifier. Which one is the best learning method depends on the type of task at hand.

Distant learning. Distant learning is the exploitation of lexical resources like WordNet (Miller 1995 [53], Fellbaum 1998 [29]), CYC (Reed & Lenat 2002 [66]), YAGO (Suchanek et Al. 2007 [70]), CONCEPTNET (Havasi et Al. 2007 [38]), or other knowledge bases for the annotation of raw text (see Mintz et Al. 2009 [54]). It usually yields results with very high precision, but low coverage. Its application is bound to the existance of resources in the desired language, but also freely available resources like Wikipedia or Wikitionary can be exploited for distant learning (see for example Zesch et Al. 2008 [78]). For PRT there are psycholinguistic machine readable dictionaries, such as MRC2 (See Colthearth 1981 [23]), and LIWC (see Pennebaker et Al. 2001 [60]), that maps words to scores like familiarity and imageability, or to personality traits directly.

Summing up: the supervised approach is a good choice if there is labeled data available. If labeled data is not available, then the unsupervised approach is the only available choice. If we have labeled data but we want to measure how classes fit the data, then an unsupervised approach can be used to compare clusters with

classes. The semi-supervised approach gives a chance to understand data structure deeply, and can be used if we have at least a small labeled dataset. Distant learning is a good approach if there are resources available, but it does not produce a classifier or a clusterer. In real world, learning approaches are often mixed. For example distant learning can be used to label unlabeled data or to work in conjunction with learning algorithms, like in Girju et Al. 2006 [34], who used WordNet structure as a feature for a supervised system.

2.1 Algorithms

In the previous section we have introduced two kind of algorithms: classifiers and clusterers, and we have seen them respectively in relation to supervised and unsupervised learning. Understanding algorithms is important in order to have a deep knowledge of their strength points and weaknesses. We do not want to go too much into the technical details of learning algorithms, but just introduce some notions useful to understand algorithms that are mentioned repeatedly in the cited literature and in this work.

Naive Bayes classification. It is a classification algorithm based on Probability. Given a labeled dataset, the classifier learns probability of each class and conditional probability of each fea-

ture per class. Following Bayes' theorem (see for example John

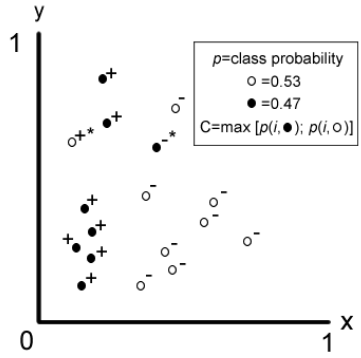

Figure 2.2: Leaning algorithms: Naive Bayes (NB).

& Langley 1995 [43]), it is possible to compute the probability of each instance to fall in each class, given its features and the classes' probabilities. Instances are classified in the highest probability class (max function), as in figure 2.2. The strong assumption underlying this algorithm is the fact that features' probabilities should be independent, and this is not true in many cases. In spite of their naive design and apparently over-simplified assumptions, naive Bayes classifiers have proved to work quite well in many complex real-world tasks.

Decision Trees and M5' classification. Decision trees, like for example the famous C4.5 algorithm (see Quinlan 1993 [62]), modelize classification into fixed rule-based graphs called trees. An example is depicted in figure 2.3. Each node in a tree represents

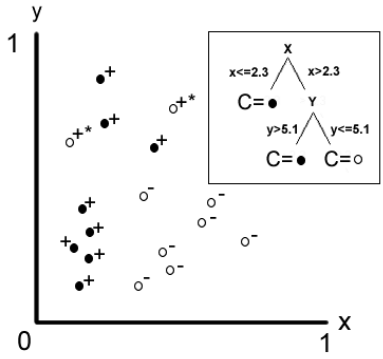

Figure 2.3: Leaning algorithms: decision trees and M5'.

a feature in an instance to be classified following one of the paths of the tree. The feature that best divides the dataset is computed using different confidence scores, like information gain or conditional accuracy, and it is placed at the root node of the tree. A tree with too much embedded nodes has a high risk of overfitting, because it uses features with low confidence score. Decision trees avoid this with pruning, that eliminates branches below a threshold confidence score. While decision trees are good for the classification of nominal data, M5' are trees suitable for the prediction of numerical values. Like conventional decision trees, the M5' algorithm (see Holmes et Al. 1999 [39] for details) builds a tree by splitting the data and placing the most predictive feature at the root node. Instead of selecting attributes using a confidence score, M5' computes a linear regression model for each node in place of

binary rules. The tree is then pruned back from the leaves to the root, so long as the expected error of the linear models at each node decreases. For example in figure 2.3 in place of the rules C=0 and C=1 we have formulas to compute linear models.

Suppor Vector Machine classification. Support Vector Machines (SVMs) are a supervised machine learning technique introduced by Vapnik 1995 [74] and optimized, among others, by Platt 1998 [61]. Given a representation of instances in a n-dimensional space (see figure 2.4), SVMs can find the maximum margin that

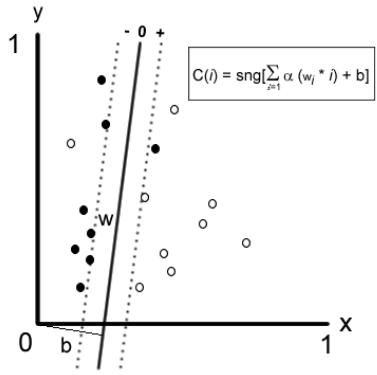

Figure 2.4: Leaning algorithms: Support vector Machines (SVM).

separates binary classes, thereby creating the largest possible distance between the separating hyperplane and the instances on both sides. The distance is computed using the support vectors (the dotted lines). The classification is performed by means of the sign function *sgn* (being greater or smaller than 0) of the summation

of each instance's coordinates in the feature space (the x and y axes in figure 2.4), multiplied by its weight w (that is the distance from the hyperplane, the black continuous line in figure 2.4) and compared to the model α, plus the slope b of the hyperplane. The maximum margin hyperplane is the one that minimizes the probability of error among all possible hyperplanes. Note that the sign function can only separate binary classes. For multi-class tasks one has to train several binary classifiers.

Simple K Mean clustering and kNN classification. Simple K Means is a clustering algorithm. It consists in randomly sample a small number of seed instances, usually one per cluster, and turn them into cluster centroids to compute the distance of other instances, as can be seen in figure 2.5. The nearest instances are grouped in the same cluster. The algorithm is iterated to recompute the position of the centroids until clusters remain the same. The corresponding classification algorithm is called K Nearest Neighbour (KNN, see Wang & Zucker 2000 [75] for details) and it is based on the same principle of simple K Means: instances in the same feature space that share similar properties are near and are likely to be classified in the same class. The kNN locates the k nearest instances to the seed instance and determines its class by identifying the single most frequent class label. The power of kNN has been demonstrated in a number of real domains, but there are

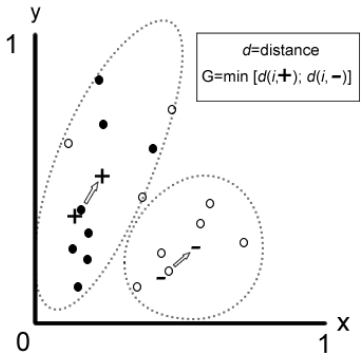

Figure 2.5: Leaning algorithms: Simple k means (SKM).

some reservations about the usefulness of kNN, such as: i) they have large storage requirements, ii) they are sensitive to the choice of the similarity function that is used to compare instances, iii) they lack a principled way to choose k, except through cross-validation or similar, computationally-expensive technique.

The choice of an algorithm always depend on the task at hand. In general, following Kotsiantis 2007, SVMs tend to perform much better when dealing with multidimensional and continuous features. On the other hand decision trees and rule based algorithms tend to perform better when dealing with discrete or nominal data. For SVMs usually a large dataset is required in order to achieve the maximum prediction accuracy whereas Naive Bayes may need a relatively small dataset. Table 2.1, reported and adapted from Kotsiantis 2007, compares pros and cons of the mentioned algorithms.

Overall, SVM is the most accurate algorithm for classification,

feature	DT/M5'	NB	kNN	SVM
accuracy	**	*	**	****
learning speed	***	****	****	*
classification speed	****	****	*	****
tolerance to irrelevant features	***	**	**	****
tolerance to noise	**	***	*	**
prevent overfitting	**	***	***	**
model parameter handling	***	****	***	*

Table 2.1: Comparing learning algorithms. (**** stars represent the best performance and * star the worst). Adapted from Kotsiantis 2007.

but it does not prevent the risk to overfit the dataset as well as other algorithms, like Naive Bayes. In the task of PRT the risk of overfitting, due to the unavailability of large labeled datasets and the scarcity of general predictive rules, is really high. We will go deeper into this problem in section 3.2.

2.2 Evaluation Metrics

Usually the performance of a system is evaluated comparing the outcomes predicted by the system itself to the gold standard labeled data. The result of that comparison is a confuzion matrix with the counts of true positives (positive prediction matches a positive label *tp*), true negatives (negative prediction matches a negative label *tn*), false positives (positive prediction matches a negative label *fp*) and false negatives (negative prediction matches a positive label *fn*), like in table 2.2. From the matrix in table 2.2 one can compute **error** (amount of wrong predicitons *fp*+*fn*),

general error (the expected error), **loss** (difference between predicted and actual values) and **risk** (the expected loss).

From that confusion matrix can be computed also other met-

	label: +	label: -
prediction: +	*tp*	*fp*
prediciton: -	*fn*	*tn*

Table 2.2: Confusion matrix of predicted outcomes and labeled, gold standard data. *tp*=true positives, *fp*=false positives, *fn*=false negatives, *tn*=true negatives.

rics that are widely used for performance evaluation: precision (p), recall (r) and f-measure (f), defined as

$$p = \frac{tp}{tp + fp} \quad r = \frac{tp}{tp + fn} \quad f = 2 * \frac{p * r}{p + r}$$

Another more intuitive metric is accuracy, defined as:

$$a = \frac{tp + tn}{tp + fp + tn + fn}$$

Accuracy gives a measure of the degree of closeness of predicted values to actual values, precision measures is the degree to which repeated measurements under unchanged conditions show the same results. Precision can be seen as a measure of exactness or quality, whereas recall is a measure of completeness, coverage or quantity. F-measure is the weighted harmonic mean of precision and recall, and can give a quantitative and qualitative evaluation in one measure. Since in PRT there are bipolar classes, as we have

seen in chapter 1, we will consider as *tp* instances correctly classified in both poles, as *fp* instances classified incorrectly and as *fn* instances for which the classifier abstains. We lack *tn*, and this brings to choose f-measure as evaluation metric.

In case there is no labeled data available it is not possible to run proper evaluation, but still there are some ways to run some kind of evaluation. For example some learning algorithms can also implement confidence, that is a measure of the probability of how much a prediction is correct. For example in Naive Bayes classification, the probability of an instance to fall in a certain class is the confidence score for that class. In SVMs the vector w, that is the distance of each instance from the separating hyperplane, can be used a confidence measure. In distance based algorithms such as kNN and SKM, the distance of each instance from the centroids is a confidence score. Decision trees instead implement information gain to put the most distinguishing features at the root of the tree. Information gain is the total entropy for a feature, it is not a confidence score but rather it can be used for feature selection.

2.3 Feature Selection

The feature selection problem is defined by Molina et Al. 2002 [55], as the selection of a subset from a set of features in order to opti-

mize system's performance, according to some objective. Usually target objectives are: 1) optimize evaluation measure; 2) fit some constraints 3) find the best balance between feature set size and performance. We will see in the next chapters that for APR we selected the features set in order to fit the multilinguality constraint.

Feature Selection Algorithms (FSAs) typically fall into two categories, based on the output they give: feature ranking and subset selection. Feature ranking outputs a list of weights for the feature set and eliminates all features that do not achieve an adequate score. Subset selection provides the optimal feature subset from an initial feature set. If we consider instead FSAs from the point of view of the interaction with learning algorithms, they can be grouped into three categories: embedded FSAs, filters and wrappers. Embedded FSAs work in parallel with learning algorithms, decision trees, that put the most informative feature at the root of the tree, are an example of this. Filters take place before the learning process and their role is to clean the feature space from unuseful information. Wrappers take place after the learning process and the models learned can be used to evaluate feature selection, with the drawback that much computational power is required. According to Molina et Al. 2002, all FSAs can be characterized by three dimensions: search organization, generation of successor and evaluation measure.

Search organization is the general strategy with which the feature space is explored. From the search organization depends most of the computational power required by the FSA and its speed. Search methods can be

1) exponential (the FSA tries many or all the combinations of features and evaluates them),

2) sequential (worst performing features are substituted)

3) random (combinations are generated randomly, this prevents the FSA to select the first best combination).

Generation of successors is the mechanism by which possible variants of features are selected from the feature set in order to generate new combinations. There are five operators that allow a FSA to do it:

1) forward (select features not yet selected and stop when all features have been tried),

2) backward (remove features from the combination under evaluation and stop when the result does not increase)

3) compound (use the forward and backward strategies iteratively and stop when both return no increment)

4) weighting (change weight of best or worst performing features iteratively according to the evaluation measure)

5) random (random change of bad performing features).

Evaluation measure is, as should be clear at this point, the

function by which successors candidate features are evaluated and different performances of feature subsets are compared. There are several evaluation measures focused on diverse characterizations of feature relevance:

1) consistency (find the feature subset that reduces the error)

2) dependence (compute correlations between predictions and features, high correlations indicate good features)

3) probability (estimate or compute the distribution of instances per class and then which feature combination approximates that distribution)

4) divergence (compute or estimate the difference between class conditional probabilities: significant differences indicate good class separability)

5) distance (similar to divergence: find class centroids and compute the distance between them, greater distances indicate good separability)

6) information (compute or estimate class probability and weight features that keep it balanced or not).

Some evaluation measures, like consistency and dependence, are suitable for supervised or semisupervised learning, because require labeled data, others, like divergence and distance, can be used also in unsupervised learning, since they allow probability estimation. According to Dy & Brodley 2004 [28], feature selection based on

separability, like divergence and distance, outperforms feature selection based on likelihood, such as dependence, in unsupervised learning tasks.

2.4 Domain Adaptation

Domain adaptation problems arise when the distribution of the data on which we are applying a learned model (target domain) is different from the distribution of the data from which we extracted the model (source domain). The learning theory community has only recently started to analyse domain adaptation problems (the first formulation is in Ben-David et Al. 2006 [7]), but it is constantly attracting attention, because it is a very significant challenge for many tasks based on real-world data, like document classification, sentiment analysis and image processing among others.

According to Mansour 2009 [51], domain adaptation is a learning problem where, given labeled data from one or more source domains, we have to learn a hypothesis performing well on different, yet related, domains for which no labeled data is available. This hypothesis is a generalization across domains and it is successful when it minimizes the difference in classifier's performance between the source and the target domains (Ben-David et Al. 2006).

According to Jiang 2008 [42], there are at least 4 approaches

to domain adaptation, three exploiting labeled source domain and unlabeled target domain, and three with labeled source and labeled target domains:

1) Instance weighting, that assigns a weight to instances in order to minimize the expected loss on the target domain. This approach includes also class imbalance techniques (changing the model class probability from the distribution of the target dataset), covariate shift (re-weight the model parameters at each instance comparing it to the general distribution) and change of functional relation (use heuristic methods to remove misleading instances from the source domain training set, based on the target domain, then retrain a classifier).

2) Semi-supervised learning, that treats unlabeled data as a resource to retrain a classifier previously trained on labeled data.

3) Change of representation, based on the idea that a transformation of the feature space, like a feature subset selection, can solve domain adaptation problems. In order to do that we have to evaluate features, for example by means of a minimization function of the approximated distance between the distributions of the two domains.

4) Bayesian priors, that is based on the idea of changing the probability of model parameters from labeled data in the target domain. Clearly both labeled datasets are required to use this kind of ap-

proach.

5) Multi-learning, that consists in learning models on many different source domains in order to enlarge its coverage. It can be performed by generating copies of features adapted to different dataset distributions or running multi-training.

6) Ensemble methods, that are based on mixed classifiers, able to adapt to different data distributions.

We think that domain adaptation is very useful also in PRT, where distribution of features often changes, depending on the type of data, on the purpose of the text, on the recipient of the message. We will see in the next section that no attempts to implement domain adaptation to personality recognition has been done yet, APR is a first step in this direction.

Chapter 3

Personality Recogntion

There are two main disciplines that are interested in personality recognition: one is computational linguistics, that extracts personality from text, and the other one is the community of social network analysts, that extract information about personality from network configuration (see for example Staiano et Al 2012 [69]) as well as from other extralinguistic cues (see Bai et Al. 2012 [6]).

The computational linguistics community became interested in PRT first. In 2005 a pioneering work by Argamon et Al. [3] (Ar05) classified neuroticism and extraversion using linguistic features such as function words, deictics, appraisal expressions and modal verbs. Oberlander & Nowson 2006 [58] (Ob06) classified extraversion, stability, agreeableness and conscientiousness of blog

authors' using n-grams as features and Naive Bayes as learning algorithm. Mairesse et Al. 2007 (Ma07) reported a long list of correlations between Big5 personality traits and two feature sets: LIWC (see Pennebaker et Al. 2001 for details) and RMC (see Coltheart 1981 for details). The former includes word classification, like "positive emotions" or "anger" and the latter includes scores like word age of acquisition or word imageability. They obtained those correlations from psychological factor analysis on a corpus of Essays (see Pennebaker & king 1999 [59] for details) and developed a supervised system for personality recognition[1]. Luyckx & Daelemans 2008 [48] built a corpus for stylometry and personality prediction from text in Dutch using n-grams of Part-Of-Speech and chunks as features. They used the Myers-Briggs Type Indicator schema in place of the Big5 (it includes 4 binary personality traits, see Briggs & Myers 1980 [14]). Unfortunately their results are not comparable to any other because of the different language and schema used. In a recent work, Iacobelli et Al. 2011 [41] (Ia11) used as features word n-grams extracted from a large corpus of blogs, testing different extraction settings, such as the presence/absence of stop words or inverse document frequency. They found that bigrams, treated as boolean features and keeping stop words, yield very good results using Support Vector Machines (SVM) as learn-

[1]demo available online at `http://people.csail.mit.edu/francois/research/personality/demo.html`

ing algorithm. As is stated by the authors themselves, their model (that is obtained with a bottom-up approach) may overfit the data, since the bigrams extracted are very few in a very large corpus. Kermanidis 2012 [45] (Ke12) followed Mairesse et Al. and developed a supervised system for PRT in modern Greek, based on low level linguistic features, such as Part-of-Speech tags, and psychological features, like words associated to psychological states like in LIWC. She trained a SVM classifier and obtained good results, demonstrating that correlations between personality and language can be successfully ported from English to other languages.

In Social Network Analysis (SNA), personality recognition has

Author	Alg.	Measure	Traits	lang.	Results (avg).
Ar05	NB	acc	xe	en	0.576*
Ob06	NB	acc	xeac	en	0.539*
Ma07	SVM	acc	xeaco	en	0.57
Ia11	SVM	acc	xeaco	en	0.767
Ke12	SVM	f	xeaco	gr	0.687‘
Go11	M5	mae	xeaco	en	0.115
Qu11	M5	rmse	xeaco	en	0.794
Ba12	c4.5	f	xeaco	ch	0.783

Table 3.1: Overview of Personality Recognition from Text and Personality Recognition for Social Networks. *=Results reported in Luyckx & Daelemans 2008. ‘=average computed by the author. =lower scores are best.

even a shorter history. Golbeck et Al. 2011 [35] predicted the personality of 279 users from Facebook, using either linguistic (such as word count) and social network features (such as friend count). Quercia et Al. 2011 [64] used network features to predict the personality of 335 Twitter users, using M5 rules as learning algorithm. In Computational Linguistics there is a tendency to predict classes

of personality traits, and the evaluation measure is often accuracy (acc). In SNA the tendency is to predict personality trait scores rather than classes, and there are measures like mean absolute error (mae) and root mean squared error (rmse). The work of Bai et Al. 2012 is an exception from this point of view: they predicted classes by means of features based on social network site usage, such as friend count, self comments and recent statuses count. They did it on a dataset of 335 users, annotated with an online survey of a reduced version of the Big5 personality test. They used f-measure (f) as evaluation metric and obtained very good results using a decision trees algorithm (c4.5).

An overview of previous work in personality recognition is reported in table 3.1. We can see recent tendencies towards the application of personality recognition to languages different from English, as well as a progressive improvement in the results, that highlights how this is a developing research fields.

3.1 State of the Art in PRT

It is not easy to determine the state of the art in PRT, because each scholar (except Mairesse et Al. 2007 and Argamon et Al 2005) used their own corpora, sampled from different domains and in addition there are several different evaluation metrics that prevent from the

comparison of the results.

Since in this work we are using only linguistic features, we will compare our results to the ones of the computational linguistics community. Here the best results have been obtained by Kermanidis 2012 and Iacobelli et Al 2011. While Iacobelli et Al. reports accuracy, that is not directly comparable to f-measure, Kermanidis 2012 instead reports f-measure, and it is on a language different from English. We believe that Iacobelli's model is overfitted, (we will see more about this in the experiments in chapter 4), since we tested their bigrams, correlated to personality traits, in a different domain. Results confirmed a good precision, but a really poor recall, so we decided to keep Kermanidis 2012's result as the state of the art in computational linguistics.

We are going to use the same dataset used by Mairesse et Al. 2007. This makes possible the comparison of our results to the ones reported in their papers. The only problem is the evaluation metric: Mairesse et Al. reported accuracy, even if they say they used Weka (Witten & Frank 2005 [76]), which provides precision, recall and f-measure, not accuracy. In order to compare our results to the ones in Mairesse et Al. 2007, we are going to replicate their experiment with Weka, using the same settings, and retrieve the average f-measure. We will see it in section 4.3.

3.2 Problems of PRT

All the approaches to PRT we have seen so far are supervised. This means that they are based on the collection of a corpus annotated with personality judgements about text authors, obtained from the Big5 personality test. Scholars trained one (usually binary) classifier per trait and apply the models retrieved on larger dataset of the same type or domain. Regarding feature extraction there are 2 approaches: bottom-up and top-down. The Bottom-up approach (Oberlander & Nowson 2006 for example) starts from the data and seeks for linguistic cues associated to personality traits while the top-down approach (for example Mairesse et Al. 2007) selects a feature set and test the correlations between those features and personality traits. The most common problems with all these approaches are:

1) Limitations in data annotation. Data labeled with personality types is not easy to obtain, because it requires that human subjects take the Big5 personality test, and it is costly and time consuming to do it on a large scale. Also the annotation of data by means of crowdsourcing services, like Mechanical Turk[2], or other social game applications, is difficult or even infeasible, because personality recognition is a frustrating task (as we will see in section 6.1) and labelers tend to cheat a lot. Bai et Al. 2012 used a re-

[2]https://www.mturk.com

duced version of the Big5 to label data from online surveys, but they obtained a small labeled set, like Golbeck et Al. 2011 and Quercia et Al. 2011. These sets are more suitable for testing than for learning.

2) Data overfitting. It refers to the problem that the learned models lose their predictive power when applied on different data and domains. This is a general problem for supervised approaches, exacerbated by the limitations in data annotation and by the type of task. Small labeled datasets suffer of this problem, as well as large datasets with sparse features. It is the case of the bottom-up strategies, used for example by Oberlander & Nowson 2006 and by Iacobelli et Al. 2011, who extracted few linguistic patterns associated to personality traits from large datasets. Although there are some techniques to reduce the impact of overfitting, like pruning for example, the models retrieved in this way are usually poor or domain dependent, even if the performances seem to be pretty good.

3) Evaluation metrics. This is a double problem. From the one hand, there is the choice to predict trait classes or personality scores, this brings to use measures like accuracy and f-measure rather than mean absolute error. From the other hand, accuracy alone is not the best metric to measure the performance of a personality recognition system, because it does not tell anything about

sensitivity and reproducibility of the results. Precision, recall and f-measure might be more appropriate than accuracy, especially when the data distribution is unbalanced.

4) Experiment Design. This is another problem that affects evaluation. Since each personality trait is bipolar one can run a two-tailed experiment, as we did, considering as true positives the correct predictions for both poles, as false positives the wrong predictions and as false negatives the missing predictions. The alternative solution is to run the experiment as a one-tailed test and consider as true positives the correct predictions for one of the two poles, as true negatives the predictions for the other trait pole and treat the wrong predictions for each pole as false positives and false negatives respectively. There is no better solution: the first one is suitable for the evaluation with precision, recall and f-measure, the second one is suitable to compute accuracy. The latter is the most commonly used just because it is supported by processing tools like Weka.

5) Domain and Language Portability. When models are trained on a specific domain or language, they might not be effective when used on different domains, for example shifting from blogs to essays or to social networks, or even to different languages. The language problem is also very present in the use of resources, such as LIWC and MRC, that are language dependent (MRC exists only in En-

glish, LIWC has been adapted in a few languages).

The Adaptive personality Recognition (APR) approach tries to solve the data limitation problem by using small labeled datasets as test sets, rather than for training. If we want to train some model, we prefer to do it on large unlabeled datasets, in a unsupervised or semi-supervised way. We will come back to this point in chapter 5. In the APR approach, data overfitting can be solved by adapting the feature space to the data at hand. APR is able to do this by using the distribution of the features in the dataset in order to compute scores and filters. We will see how in the next chapter. The adaptability of APR is suitable also to solve the language and domain portability problem. We select a cross-linguistic feature subset from LIWC, together with its correlations to personality traits, and we use it for generating hypotheses on data in any language. It is also possible to extract either linguistic and extralinguistic features in order to enrich the initial feature set (see chapters 5 and 7). About the problem of experimental design, we selected to run two-tailed experiments, and we tried to predict either scores and nominal classes, as we will see in the next chapter. As evaluation metric we selected f-measure, for the reasons stated above. In the next chapter we will introduce our approach for Adaptive personality Recognition from Text.

Chapter 4

Adaptive Personality Recognition

Adaptive Personality Recognition (APR) is an approach to Personality Recognition that tries to solve the problems listed in section 3.2, especially the limitations in data annotation and the language portability problems. APR can be implemented on raw text data with authors. It requires a set of correlations (we are going to use sets taken from previous literature) between textual features and personality traits, but extralinguistic correlations can be used as well. Note that feature set and correlation set are two sides of the same coin in APR, because each feature must be associated to one

or more correlations to personality traits.

The APR approach includes the following steps: i) check the distribution of the features/correlations in the dataset (or part of it) for domain adaptation purposes; ii) exploit the correlations, after applying some correction or filtering based on distribution of features in the data, in order to compute scores for each personality trait of each unit of text in the dataset; iii) generate hypotheses on personality traits by turning scores into nominal classes, that can be binary (positive/negative) or ternary (positive/negative/omitted); iv) generalize the hypotheses by comparing all the texts of each single author, and computing a confidence score for the generalized hypothesis of personality or even for each trait; v) test the performance of the generalized personality hypotheses on a labeled dataset (even a very small one) or, if it is impossible, predict accuracy from confidence scores.

With this in mind, we describe the development of a system that performs APR automatically.

4.1 System Developement

The APR system takes as input 1) unlabeled text data with authors; 2) some set of correlations between personality traits and linguistic or extralinguistic correlations. As stated before, the out-

put is one hypothesis of personality for each author. Personality hypotheses are formalized as 5-characters strings, each one representing one trait of the Big5, as depicted in figure 4.1. Each

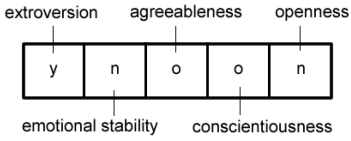

Figure 4.1: Formalization of personality hypotheses.

character in the string can take 3 possible values: positive pole (y), negative pole (n) and omitted/balanced (o). For example "ynoon" stands for an extrovert neurotic and not open mindend person.

Figure 4.2 represents the pipeline of the system. In the preprocessing phase, the system samples a portion of unlabeled data (usually 10-20%, but the amount can be defined when running the system) and extracts average distribution of each feature in the correlation set. This is a strategy introduced by Mairesse et Al. 2007 for performance improvement, that we exploited for domain adaptation, and also with the purpose to prevent overfitting.

In the processing phase the system generates one hypothesis for each written text, checking for matches of linguistic features provided in the correlation set. If it finds a feature value above the average the system increments or decrements a score associated to the personality trait, depending on a positive or negative

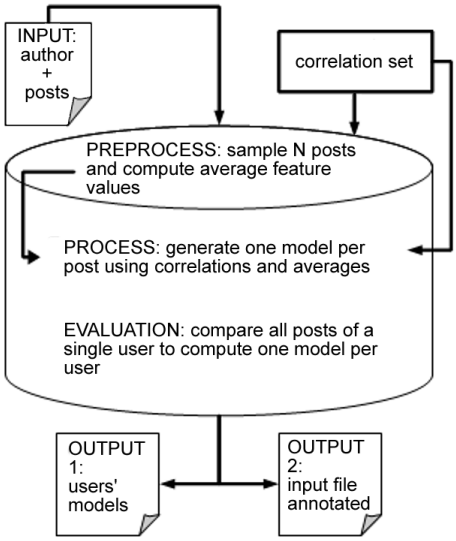

Figure 4.2: APR System pipeline.

correlation. From positive and negative trait scores the system can compute trait confidence scores (tc) for the predictions of each trait, defined as

$$tc = (y - n) \qquad y = \frac{ym}{P} \qquad n = \frac{nm}{P}$$

where ym is the count of matches for the positive pole of personality trait and nm is the count of matches for the negative pole of the trait. P is the count of texts.

In the evaluation phase the system compares all the hypotheses

generated for each single text of each author and retrieves one single hypothesis per author, turning the personality scores into classes (if below 0 predicts a negative pole, if above 0 the positive one if is equal to 0 predicts a "o"). In the evaluation phase the system computes average confidence and variability. Confidence can be computed for the whole hypothesis (average confidence) or for each single trait (trait confidence). Average confidence (c) gives a measure of the robustness of the personality hypothesis. It is defined as

$$c = \frac{mh}{H}$$

where mh is the count of personality hypotheses matching within the same author (for example "y" and"y", "n" and "n", "o" and "o") and H is the total of the hypotheses generated for that author. Variability gives information about how much one author tends to write expressing the same personality traits in all the texts. It is defined as

$$v = \frac{c}{P}$$

where c is the confidence score and P is the count of all author's texts. Note that the system can evaluate personality only for authors that have more than one text, the other users are discarded.

The main problem with the APR approach is that confidence is not a valid testing metric. The best thing would be to test the

performance of the system on a small labeled set. when this is not possible, it is always possible to predict accuracy with some learning algorithm, like linear regression or SVM.

4.2 Data, Features and Settings

Datasets In the experiments we are going to make use of two different datasets, Essays (Pennebaker & King 1999) and PersonalityFB (Celli & Polonio, to appear [18]).

Essays is a collection of reports written in English and collected since 1997 to 2004. It provides text and Big5 personality scores for 2500 authors. We formatted the corpus by splitting each line of the texts, in order to have more information for the evaluation phase of the system. We split the dataset into a dev set(0.5%), training set (98.5%) and test set (1%). We extracted the 2-class (y/n) gold standard for the dev set and the test set from the personality scores, turning values greater than 0 into "y" and less than 0 into "n". We also extracted a gold standard for the whole dataset, but not for the training set, that we want to use as unlabeled data for semisupervised learning.

PersonalityFB is a collection of Facebook data and short Reports in Italian that contains a training and a test set. We collected data for the test set from 23 subjects that took the Big5 personal-

ity test. We asked the participants the consent to leave the URL of their Facebook personal page for sampling, and also to write a short essay, minimum 15 lines and maximum 30, on any argument they like. We splitted the lines of the essays in order to use them with the system and to compare them to social network posts. We produced the gold standard personality labels for the users from the results of the Big5 test. We converted the scores of the Big5 into a 2-class format used by the system. To do so we turned the scores above 50 into "y" and all the scores below or equal to 50 into "n". We collected the training set by means of a crowler that exploits Facebook's graph API[1] in order to sample users' statuses. The resulting dataset contains 1100 egonetworks of Italian users and their statuses or comments related to the users who had interactions with them.

Features The system takes as feature some sets of correlations between language and personality traits. We tested four different sets, taken or adapted from literature. They are: 1) Psychological (**M** set, reported in table 4.1), provided by Mairesse et Al. 2007 and based on MRC. Includes: age of acquisition of word (aa); characters count (ch); syllables count (sy); Kucera-Francis word Frequency (Kf); Kucera-Francis category (Kc); Kucera-Francis sample (Ks); Brown frequency (bf); Thorndike-Lorge frequency (Tf); con-

[1] http://developers.facebook.com/tools/explorer

creteness (cc); familiarity (fy); word imageability (wi) and word meaningfulness following Colorado norms (mc).

f.	x	e	a	c	o
ch	-.09**	.09**	-.03	.00	.15**
sy	-.07**	.07**	-.02	.04	.13**
Kf	-.01	.10**	.00	.05*	.07**
Kc	.06**	-.04*	.08**	.07**	-.12**
Ks	.06**	-.01	.03	.05**	-.07**
bf	.05*	-.06**	.03	.06**	-.07**
Tf	.01	.10**	.01	.06**	.05**
cc	.02	-.06**	.03	-.01	-.10**
fy	.08**	-.05*	.08**	.05**	-.17**
wi	.05*	-.04*	.05*	.00	-.08**
mc	.06**	-.10**	.05**	-.01	-.11**
aa	-.01	.05*	-.04*	.06**	.11**

Table 4.1: Correlations for the M set, reported in Mairesse et Al. 2007. * = p smaller than .05 (weak correlation), ** = p smaller than .01 (strong correlaion).

2) Linguistic (**I** set), taken from Iacobelli et Al 2011 and reported in table 4.2. Includes English words and n-grams associated to high and low trait personality scores.

f.	x	e	a	c	o
x+	.01*	.01	.01	.01	.01
x-	-.01*	.01	.01	.01	.01
e+	.01	.01*	.01	.01	.01
e-	.01	-.01*	.01	.01	.01
a+	.01	.01	.01*	.01	.01
a-	.01	.01	-.01*	.01	.01
c+	.01	.01	.01	.01*	.01
c-	.01	.01	.01	-.01*	.01
o+	.01	.01	.01	.01	.01*
o-	.01	.01	.01	.01	-.01*

Table 4.2: Correlations in the I set, assigned by the author on the basis of results of Iacobelli et Al 2011. * = p smaller than .05 (weak correlation), ** = p smaller than .01 (strong correlation).

3) Cross-Language(**C** set), reported in table 4.3. We selected this set by picking up all the language independent features from LIWC and MRC, namely: punctuation (ap); question marks (qm); quotes (qt); exclamation marks (em); numbers (nb); parentheses

(pa); repetition ratio (tt), word frequency (wf, computed on the dataset in the preprocessing phase, whithout require an external resource).

f.	x	e	a	c	o
ap	-.08**	-.04	-.01	-.04	-10**
em	-.00	-.05*	.06**	.00	-.03
nb	-.03	.05*	-.03	-.02	-.06**
pa	-.06**	.03	-.04*	-.01	.10**
qm	-.06**	-.05*	-.04	-.06**	.08**
qt	-.05*	-.02	-.01	-.03	.09**
tt	-.05**	.10**	-.04*	-.05*	.09**
wf	.05*	-.06**	.03*	.06**	.05**

Table 4.3: Correlations for C set, adapted from Mairesse et Al. 2007. * = p smaller than .05 (weak correlation), ** = p smaller than .01 (strong correlation).

2) Psycholinguistic (**L** set, reported in table 4.4), based on LIWC and provided by Mairesse et Al. 2007. Includes: words associated to affects (af), cognitive mechanisms (co), anxiety (ax), anger (an), sadness (sd), sight (se), hear (hr) feel (fe), insights (is), cause (ca), tentativeness (te), certainty (ce), inhibition (ih), inclusion (in), exclusion (ex); words about society (sc), family (fm), friends (fr), humans (hu), home (hm), body (bd), motion (mo), achieve (av), leisure (le), sex (sx), religion (re), death (dt), space (sp), time (tm), positive (pe) and negative (ne) emotions; grammatical indicators like pronouns (pr), such as I (1s), we (1p), you (2p), negative particles (np), fillers (fi), numbers (nb), present (ps) and future (fu) tense and other linguistic indicators, such as swears (sw) and nonfluencies (nf).

f.	x	e	a	c	o
1s	.05*	-.15**	.05*	.04	-.14**
1p	.06**	.07**	.04*	.01	.04
2s	-.01	.03	-.06**	-.04*	.11**
af	.03	-.07**	-.04	-.06**	.04*
an	-.03	-.08**	-.16**	-.14**	.06**
ar	-.08**	.11**	-.03	.02	.11**
as	.01	.02	.00	-.04	.04*
av	.03	.01	-.01	.02	-.07**
ax	-.01	-.14**	.03	.05*	-.04
bd	-.05**	-.04	-.04*	-.04*	.02
ca	.01	-.03	.00	-.04	-.05*
ce	.05*	-.01	.03	.04*	.04
co	-.03	-.02	-.02	-.06**	.02
dt	-.02	-.04	-.02	-.06**	.05*
ex	-.01	.02	-.02	-.01	.07**
fe	-.01	-.09**	.04	.02	-.04*
fi	-.04*	.01	-.01	-.03	-.01
fm	.05*	-.05*	.09**	.04*	-.07**
fr	.06**	-.04*	.02	.01	-.12**
fu	-.02	.01	.02	.07**	-.04
hm	-.01	-.02	.04*	.06**	-.15**
hr	-.03	.00	-.01	-.04*	.04*
hu	.04	-.02	-.03	-.08**	.04
ih	-.03	.02	-.02	-.02	.04*
in	.04*	-.01	.03	.04*	-.03
is	-.01	-.01	.00	-.03	.05*
le	-.03	.07**	.03	-.01	-.05**
mo	.03	-.01	.05*	.03	-.13**
ne	-.03	-.18**	-.11**	-.11**	.04
nf	-.03	.01	.01	-.05*	.02
np	-.08**	-.12**	-.11**	-.07**	.01
pe	.07**	.07**	.05*	.02	.02
pp	.00	.06**	.04	.08**	-.04
pr	.07**	-.12**	.04*	.02	-.06**
ps	.00	-.12**	-.01	-.03	-.09**
re	.00	.03	.00	-.06**	.07**
sc	.08**	.00	.02	-.02	.02
sd	.00	-.12**	.00	.01	-.01
se	.00	.09**	.00	-.03	.05**
sp	-.02	.05*	.03	.01	-.04
sw	-.01	.00	-.14**	-.11**	.08**
sx	.07**	-.02	.00	-.04	.09**
tm	-.02	.02	.07**	.09**	-.15**
te	-.06**	-.01	-.03	-.06**	.05*

Table 4.4: Correlations for L set, reported in Mairesse et Al. 2007. * = p smaller than .05 (weak correlation), ** = p smaller than .01 (strong correlation).

System Options, Parameters and Settings First of all we tested system's parameters in order to find the effect of different settings. We run some experiments on the development set. We tested 3 parameters that we expect to affect system's performance: 1) average feature values threshold (k), 2) preprocessing sample size (s); 3) hypothesis generation approach (v).

The k parameter is a multiplier of the average feature/correlation values, extracted during the preprocessing phase. It rises or decreases the threshold for correlation firing. We expect that as k increases, precision rises and recall decreases. Results, reported in table 4.5, confirm the fact that recall decreases, but precision rises

param.	p	r	f1
rbl	.488	.492	.49
k0 s10% -	.497	.882	.636
k0 s10% v	.503	.851	.632
k1 s5% -	.508	.845	.635
k1 s10% -	.506	.842	.632
k1 s25% -	.505	.842	.631
k1 s50% -	.506	.842	.632
k2 s10% -	.504	.63	.56
k3 s10% -	.496	.563	.527
k4 s10% -	.498	.561	.528
k8 s10% -	.505	.547	.525

Table 4.5: Results of parameters testing on C correlation set on the dev set. Scores are averages over the 5 personality traits.

just a little bit, with a peak at k=2 (precision=.504), that falls with k=3 and then rises regularly with k above 4. The peak at k=2 is good because we have minimum loss in recall.

There are two approaches for hypothesis generation in APR (v parameter): one is constant, the other one is variable. The former

is the simpler one, it generates classes following the rule: if a confidence value is greater than 0 the system generates a label "y", if it is below 0 generates "n" and if it is equal to 0, generates a "o". The variable approach is more complex: the system keeps track of the average confidence value for each trait and generates labels by replacing the 0 with the average confidence values. For instance, if a confidence value is greater than the average for that trait, the system generates label "y", if it is below the average it generates "n" and if it is equal to the average it generates a "o". Results show that the variable approach helps rising precision, decreasing a little bit the recall.

The s parameter (preprocessing sample size) apparently seem to make no difference to the result, but it is related to the speed of the system and to the robustness of the results. The largest is the preprocessing set, the more time the system takes to run and the more stable is the result.

4.3 Experiments with APR System

Replicating Mairesse's Experiment Since we have the same dataset of Mairesse et Al. 2007 and the same feature sets they used, we are able to replicate their experiment in order to test what is the f-measure of the state-of-the-art.

We extracted from the text of each author a feature vector containing all the counts of matches of features in M and L feature sets, then we trained a SMO classifier (Platt 1998), using the default settings in Weka (Witten & Frank 2005). Unlike the original experiment in Mairesse et Al. 2007, we do not have the ranking algorithm they used (RankBoost, see Freund et Al. 1998 [30]), and we set a percentage split in place of the 10-fold cross validation. In other words we test the classifier on different instances with respect to the ones we used to train it. This usually yields slightly lower results than cross-validation, but is the setting we are going to use for our experiments. Results, averaged over the five traits, are p=.557, r=.558 and f=.557. The average f-measure is very close to what is reported in Mairesse et Al. 2007 as accuracy, and we suspect that they called it accuracy but it was f-measure.

Predicting Accuracy We run the system on the whole dataset, generating personality hypotheses and computing the accuracy using the gold standard. We used Weka (Witten & Frank 2005) for predicting accuracy, splitting the whole dataset into 66% training set and 33% test set and using hypothesis confidence, variability and post count as features. We found that average accuracy can be predicted using a linear regression with a Mean Absolute Error of 0.18 and that texts count and estimated average confidence score are good predictors of accuracy.

APR system's Performance: 2-tailed tests and baselines Since each personality trait has two poles, we decided to run the classification as a two-tailed experiment. In a one-tailed test, the majority baseline (mbl henceforth) should be calculated by labeling all instances first with the positive and then with the negative class, and then computing the mean between the two. By doing this way in a two tailed test, we obtain a perfect recall, due to the fact that there are no missing values with the majority class. Where this kind of baseline is not appropriate, we alternatively provide a random baseline (rbl henceforth), computed generating "y" "n" and "o" labels randomly.

We run experiments on the test set, using all the feature sets separately. Results are reported in table 4.6. In general we have

feature set	p	r	f
rbl	.478	.481	.479
C	**.544**	.791	.645
M	.468	.91	.618
L	.525	**.969**	.681
I	.499	.08	.138

Table 4.6: Average precision, recall and f-measure for different feature sets. Averages is computed over the five personality traits.

low precision and high recall, apart for the pattern feature set (I), whose recall is really poor, and precision is pretty high. This suggests that Iacobelli et Al's patters are overfitted on their dataset. The best performance is obtained using the L set, that is the largest one, and yields the best recall because produces few "o" values. It is interesting to note that the C set has the best precision.

Inspired by co-training and multi-training, we tested whether different feature sets are able to improve each other's predictions, for example minimizing the "o" values maintaining good precision, when working together. Following Nigam & Ghani 1998 [56], co-training helps improving the performance of supervised and unsupervised algorithms when there is a natural splitting in the feature set, hence we expect an improvement.

We run the experiments on the test set, using 2 as k threshold and trying all possible combinations of feature sets. Results, reported in table 4.7, show that there is a general improvement, as

feature set	p	r	f
rbl-essays	.478	.481	.479
M+L	.543	.938	.688
M+C	.467	.892	.613
M+I	.463	.814	.59
L+C	.531	.909	.67
L+I	.52	.914	.663
C+I	.541	.664	.596
C+M+L	.552	.905	.686
C+I+L	.515	.925	.662
I+L+M	.554	.929	**.694**
C+M+L+I	.546	.904	.681

Table 4.7: Average precision, recall and f-measure with co-traing and multi-training.

expected, in particular using multi-training. We note that the M feature set, unless used with the L set, generates noisy predictions, decreasing the precision of the I and C sets.

Predicting Personality Scores: 1-Tailed Test We run an experiment to predict personality scores in place of classes. We are

using Weka, with a 10-fold cross validation as evaluation setting, per-trait confidence as features and M5' rules as algorithm. Majority base line (mbl) is computed using the Zero Rule algorithm in Weka. Lowest scores are best. Results, reported in table 4.8, reveal

feature set	mae	rmse
mbl	.831	1.028
C	**.789**	**.992**
M	.829	1.039
L	.842	1.042
I	.831	1.028

Table 4.8: Average Mean Absolute Error and average Root Mean Squared Error for different feature sets.

that the C feature set has the best performance in personality score prediction. Looking closer to the predictions we found that trait confidence rates, using the C feature set, tend to less variation in values. We suggest that this is connected to the good prediction performance of the C dataset. The reasons why the C dataset have less variation can be many, but we suggest that C feature set has a more balanced firing rate of the features/correlations with respect to other feature/correlation sets, and this brings less noise in the evaluation of the generated hypotheses.

Prediction of Personality in Social Network Domain Until now we have seen that the L feature set achieves the best performance and that the C feature set achieve the best precision in PRT on the Essays dataset in English. We also demonstrated that the C feature set has the best performance in the prediction of scores.

Still we have to test what happens if we run the system in a Social Network domain. We run an experiment on PersonalityFB, comparing the essays written offline (persoff) and the Facebook statuses written online (persfb) of the same Italian users. We tested the C and L correlation sets. We used the Italian version of LIWC for the L set. We set 30 as preprocessing instances and 1 as feature threshold. Results, reported in table 4.9, confirm previous findings:

set	p	r	f
rbl (persfb)	.445	.464	.454
rbl (persoff)	.426	.429	.425
L-persoff	.467	.936	**.623**
C-persoff	.474	.808	.597
L-persfb	.436	.93	.594
C-persfb	.555	.765	**.643**

Table 4.9: Comparison of the performance of C and L sets on essays and social network Domain.

the C set yields the best precision and the L set the best recall. But there are two more important results: the first one is that, while the L set achieve the best performance on essays, the C set surprisingly outperforms the L set on Facebook data, achieving a good precision. We suggest that short texts, like Facebook posts, and the kind of language found in a social network domain decrease the predictive power of linguistic features in the L set, while the C set is more suitable for domain adaptation. The second important result is that we applied APR to a language different from English (using correlations extracted from English data) and the performance decreased very little (avg. -0.062, cfr table 4.6)

passing from English to Italian. We would need more data in other languages to confirm this finding, but nevertheless this is a proof that the language portability problem can be solved, and domain adaptation surely helps.

Error Analysis We have seen that the C feature set is the most suitable for APR and the one that achieves better precision. Still we want to understand whether it is possible to improve its performance, for example raising recall. To this purpose We run error analysis on the entire Essays dataset with the C feature set, and we found (see figure 4.3) two major problems, one due to the intrinsic difficulty of the PRT task and one to the the APR approach.

The first one is that separability is limited to the edges, in other words that to the high-confidence values. This is a characteristic of the personality recognition task. The problem lies in the fact that we have many average- and few high-confidence values. This makes sense if we think about the fact that people show very few well defined personality traits (core traits, that in theory should be detected by high-confidence values) and other more variable traits. The issue of separability is really hard to tackle. We are going to exploit the highest confidence-rated instances to try semisupervised learning approaches, (we will see this approach in the next chapter), however the main problem we expect is the noise generated by the confidence scores.

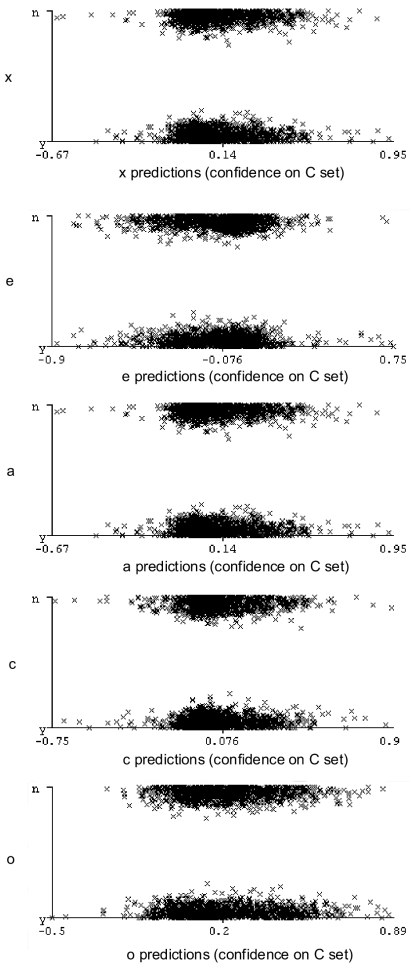

Figure 4.3: Error analysis of results produced by the C feature set.

The second problem is related to the fact that some personality traits' confidence scores are skewed. For example for the emotional stability trait, scores are skewed towards the negative pole and viceversa for the openness trait. This is due to the fact that feature/correlation sets used by the APR approach, are more powerful for the prediciton of one pole of the personality traits with respect to the others. We suggest that this happens because there are differences in features' firing rate. There are many ways to contrast skewness. For example one is to use the variable hypothesis generation option, another one is to weight features in the set in order to balance firing rate. To the purpose of have a better understanding of the skewness problem we propose to use a three-way classification, changing the gold standard to include the "o" class, and test whether the performance increase.

System's performance. Three-way classification, Two-tailed test Until now we tested the system against a gold standard annotated with 2 classes. We did that although the system generates 3 class labels, because the "o" class is considered abstention. Here we will compare the hypotheses generated by the classifier against a three-class gold standard.

The first thing to do before running the three-way classification is to produce a new gold standard. In order to do so, we measured the personality scores minimum and maximum on the entire

dataset. We report them in table 4.10, We tried two ways to set the

trait	min	max
x	-4,051	2.426
e	-2.839	2.923
a	-3.818	2.521
c	-4.075	2.508
o	-3.871	2.487

Table 4.10: min and max values of the scores per personality trait.

threshold for the classes: in the first case we set the "o" class from +1 to -1 and in the second case from +0.5 to -0.5. We produced two gold standard sets, turning values above the threshold into "y", below the threshold into "n" and the rest into "o". A manual survey of the gold standards revealed that the threshold +1 and -1 produced a lot of empty personality strings "ooooo", unuseful for analysis, hence we decided to run the experiment using the gold standard with the thresholds at +0.5 and -0.5. We run the experiment on the test set, setting 300 instances for the preprocessing sample, and 2 as k threshold parameter. We run the experiment again with all the feature sets. Results, reported in table 4.11 show

feature set	p	r	f
rbl	.391	.597	473
C	.532	.558	**.545**
M	.515	.47	.491
L	.446	.517	.479
I	.764	.392	.514

Table 4.11: Average precision, recall and f-measure for different feature sets in a 3-way classification task, 2-tailed test. Average is computed over the five personality traits. mbl is the average of the baselines for the four feature sets.

that the C set achieves the best performance, because it gains a good balance between precision and recall. We suggest that the

L feature set obtained a bad performance because there are many features in it and this raises the the amount of false positives.

The two-way classifcation in general yields better performances with respect to the three-way classification, as it is reported also in Bai et Al. 2012. This is not really surprising, since adding classes adds complexity to the classification task. From the point of view of the interpretation of data, we think that there is no much difference between a three-way and a two-way classification, because both can bring out information about core/peripheral traits discussed in chapter 1 ("o" classes being the peripheral and "y"/"n" the core).

In the next chapter we will introduce some modification to the system in order to improve the performance.

Chapter 5

Improving Adaptive Personality Recogntion

In the previous chapter we described the pipeline of the APR approach, and we tested different settings and feature sets. From now on we will keep only the cross-language (C) feature set, because it proved to be the most versatile and suitable for Adaptive Personality Recognition. In addition, it can be applied cross-language and it is not commercial. We also have seen, in the previous chapter, that we obtained the highest performance (f=.694) with the conjunction of the I+M+L sets, thus we want to improve the performance of the C set to outperform the results obtained with the other feature

sets. We are going to use different strategies and machine learning techniques.

First of all we are going to implement automatic feature weighting, hypothesis correction and heuristics techniques, that we will describe in detail in section 5.1. But the real improvements to the system, shown in figure 5.1, are given by the hybridization of

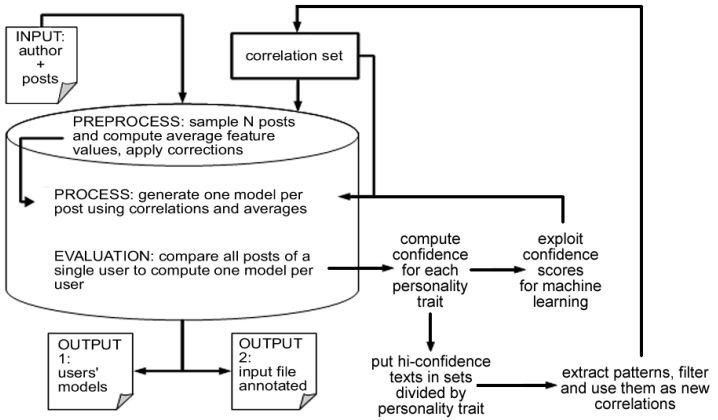

Figure 5.1: Improvements to the APR System pipeline.

APR with machine learning, and by the extraction of new patterns correlated to personality traits, described in sections 5.2 and 5.3 respectively. Both the extensions to the system's pipeline have been implemented starting from personality trait confidence scores, produced during the evaluation phase. The hybridization between APR and machine learning produces new hypotheses that can be

used to replace the ones previously generated by the APR system, or exploited for label correction. The extraction of new patterns is useful in order to enrich the correlation set of new features, derived from the dataset itself.

5.1 Adding new Parameters

First of all we added new parameters to the system and run new experiments in order to test what are the best combinations to improve the classification performance. The parameters we tested are the following: automatic feature weighting (w); confidence scores normalization (n); and correction based on skewness (r). We will see each parameter in detail.

Automatic Feature Weighting. According to Mairesse et Al 2007, feature selection it is a good way to boost automatic personality recognition. We already made a manual feature selection in order to fit the constraint of language applicability: the result is the C feature set. We are going to implemented automatic feature selection in the preprocessing phase. The type of feature selector exploits sequentiality as search organization, weighting as generation of successors, and a metric similar to divergence, but based on firing rate, as evaluation measure. Since features in the C set are few, we preferred to use weights rather than discard them. Ba-

sically, a high firing rate score decreases trait feature weight on the fly, during the processing phase, thus balancing the generation of hypotheses among all features. We run an experiment to test the effect of this kind of feature selection. Results on English and Italian, reported in table 5.1, show that the weighted scheme

dataset	par	p	r	f
rbl (es-test)	-	.478	.481	.479
es-test	w	.522	.851	.647
es-test	-	.544	.848	.663
rbl (fb-test)	-	.445	.464	.454
fb-test	w	.522	.878	**.655**
fb-test	-	.497	.852	.628

Table 5.1: Average precision, recall and f-measure on essays (es) and PersonalityFB (fb). Average is computed over the five personality traits. (w)=weigthed features.

(w) in general helps rising recall, decreasing precision on essays, and increasing it on PersonalityFB. In general the results show, once again, that domain adaptation works and that automatic feature weighting works better on adapted domains. We suggest that this is due to a different, more skewed, distribution of the features/correlations in PersonalityFB with respect to essays, and the weighted scheme is able to catch the information provided by the less frequent features.

Using Skewness for Heuristics. After the findings of the error analysis (see section 4.3), We also decided to implement some heuristics based on personality trait skewness in the preprocessing phase. We modified the system in order to generate hypotheses on the data sampled during the preprocessing phase. Then we com-

puted skewness scores for each personality trait by calculating the difference between "y" and "n" labels on the sample. We calculated the average skewness, and we considered skewed distributions the traits that have a skewness value above the average, be them positive or negative (hence producing "y" and "n" labels respectively). If a distribution of personality trait labels is skewed, it means that the system makes better predictions for one pole with respect to the other. The heuristics consist in the application of a correction method to the worst predictive trait pole. For example if we find that the extraversion trait has a great skewness in the "n" pole, in other words it tend to predict more introverted than extrovert users, we can trigger a correction function, that can be tailored on the task at hand. Here we applied random correction (r), that consists into assign a random label to the worst predicted pole.

Normalization. Normalization is the process of adjusting values measured on different scales in order to make them comparable. We normalized per-trait confidence values dividing them by the number of texts per author. We will refer to this normalization as parameter (n). By normalizing per-trait confidence scores, that are integers, we obtain values between 0 and 1. When we run the system in the variable hypothesis generation mode (v), these values are hardly equal to the average, hence the system is going to reduce

a lot the amount of "o" labels generated in the hypotheses. This way the system prevents the classifier from abstain, and we can rise recall to 1 or close to 1, but still remains the question whether precision rises or decreases. We expect it to decrease a little bit, but we also expect the overall performance, in terms of f-measure, to rise.

We set the preprocessing sample size to 10% and the feature

dataset	par	p	r	f
mbl-persfb	-	.437	1	.608
fb-test	n	.477	.855	.612
fb-test	nv	.472	1	.641
fb-test	nw	.492	.87	.629
fb-test	nr	.478	.86	.614
fb-test	nvr	.472	1	.641
fb-test	nvrk=1	.493	1	**.661**
fb-test	nvrk=2	.483	1	.651
mbl-es-test	-	.487	1	.655
es-test	n	.544	.861	.667
es-test	nv	.537	1	.699
es-test	nw	.525	.855	.651
es-test	nr	.549	.908	.684
es-test	nvr	.535	1	.697
es-test	nvrk=1	.536	1	**.698**
es-test	nvrk=2	.517	1	.682

Table 5.2: Average precision, recall and f-measure for different datasets in a 2-way classification task. 2-tailed test. Average is computed over the five personality traits. (w)=weigthed features, (t)=bigrams extraction, (n)=normalization, (v)=variable hypothesis generation, (r)=random correction based on skewness, (k)=threshold on feature average.

average threshold (k) to 0, except where otherwise indicated. We run the experiments combining the new parameters we have introduced thus far with the ones introduced in chapter 4, like hypothesis generation mode (v) and feature average threshold (k). Results, reported in table 5.2, show that the best performances are obtained combining all the parameters, with threshold=1 (nvrk=1). We sug-

gest that normalization (n) must be used with caution. However, the variable hypothesis generation approach yields the best results when combined with normalization, and feature average threshold (k) set to 1 can be exploited to raise precision. Random correction (r) is suitable to balance per-trait performance. It raises low personality trait scores and decreases high scores, bringing results closer to the average (we discuss the details about the differences of single traits in chapter 6). In conclusion all those parameters can have a positive effect on the performance of the system, but the when to use them depends on the conditions of the data at hand.

Summing up: normalization (n), when paired with variable hypotheses generation (v), reduces a lot the amount of "o" labels, rising recall; feature weighting helps rising performance in general and it is suitable for domain adaptation; average feature threshold (k) generally raises precision and decreases recall; eventually random correction (r) based on skewness can be exploited to balance the performance among traits.

5.2 Learning with APR

We implemented unsupervised and semisupervised learning in the system. They were integrated into APR in two ways: one is the single approach and the other one is the hybrid approach. In the

single approach we exploited APR in order to produce confidence scores (per trait and global), variability and post count, and we used them as higher-order features for learning. In the hybrid approach we run APR and we used learning correction for skewed traits' distributions, thus predicting only the labels of the trait pole where APR is suspected to perform bad.

As unsupervised algorithm we used a simple K-means clusterer, based on euclidean distance. It takes as features the "y" and "n" label counts, generated by the APR system for each trait separately, and clusters on the fly the traits for each user, keeping track of the values of the same personality trait of all the previous users. results are reported in table 5.3.

We adopted a self-training semisupervised approach. For instance we used a small portion of the gold standard to retrieve information about 1) the distribution of confidence scores in relation to personality classes and 2) about the probability distribution of classes per trait. We modified the system on order to train a naive bayes classifier on the fly. We chose to use naive bayes because, according to Kotsiantis 2007, it is very resistent to noisy data and very fast to train. Like the supervised classifiers, also this one exploits hypothesis confidence and per-trait confidence as features. We also developed a probabilistic semisupervised classifier, that exploits just the class probability per trait and assign

classes according to the probability distribution. We repeated all the experiments two times: one with the normal learning approach and one with the hybrid approach. Results are reported in table 5.3.

Results, reported in table 5.3, reveal that unsupervised learn-

dataset	par	p	r	f
mbl-fb-test	-	.436	1	.608
fb-test	unsup	.411	1	.583
fb-test	semi	.523	1	**.687**
fb-test	semi-p.	.575	1	**.73**
fb-test	unsup+g	.351	.849	.497
fb-test	semi+g	.472	.947	.63
fb-test	semi-p+g	.493	.917	.641
mbl-es-test	-	.487	1	.655
es-test	unsup	.46	1	.63
es-test	semi	.448	1	.619
es-test	semi-p	.517	1	.682
es-test	unsup+g	.521	.932	.668
es-test	semi+g	.556	.936	**.698**
es-test	semi-p+g	.528	.913	.669

Table 5.3: Average precision, recall and f-measure for different datasets in a 2-way classification task. 2-tailed test. Average is computed over the five personality traits. p=semisupervised-probabilistic learning; g=hybrid approach (APR+learning).

ing has a bad performance, while the semisupervised approach has a good one. In particular, it achives a good performance on PersonalityFB using a simple probabilistic classifier, while on Essays the hybrid approach perform best. We suggest that it is due to the fact that class distribution in PersonalityFB is much more informative than in Essays. Anyway a very good result is achieved also with simple semisupervised learning.

5.3 Extraction of New Patterns

We exploited the Adaptive Personality Recognition system in order to extract automatically new patterns to add to the feature set. Following Iacobelli et Al 2011, we decided to extract n-grams, for instance word bigrams. The n-gram extraction works as follows: i) we add as input a large unlabeled training set, that we will use for pattern search. ii) In the preprocessing phase we exploit correlations in order to lable the training set with labels and confidence scores, and iii) then we put the texts of each author in different sets, one for each pole of each trait, according to the generated label. iv) Finally we extract the 20 most frequent bigrams from each set, selecting the non-overlapping bigrams by running a symmetric difference between sets, paired for each trait, as illustrated in figure 5.2. We end up with ten different sets of bigram patterns (the ones in white in figure 5.2), each one associated to a personality trait pole. We use the bigram patterns as new features, counting bigram matching in the text as weak correlations to their corresponding personality trait pole.

We tested the precision of the confidence-generated labels, obtaining an average of .527 over a random baseline of .496 on the dev set. The results of the impact of the bigrams extracted on the performance of the system are reported in table 5.4. Results show that patterns are good for the emotional stability and agree-

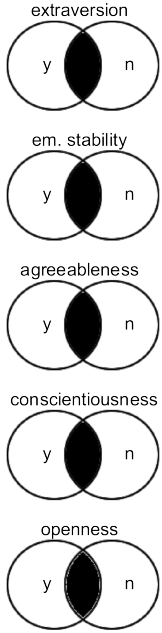

Figure 5.2: Symmetric difference on pairs of n-gram sets.

ableness traits, and noisy for the conscientiousness trait, that has a poor performance on Italian (PersonalityFB). Results are more balanced on English, where the lowest one is extraversion. Overall the average is the same: f=.686. This confirms that pattern extraction work very well for domain adaptation. It is a very good result, especially on PersonalityFB, and confirms the fact that APR is suitable for social networks domain.

Looking at the details of the performance of each single trait, we can see that the results on PersonalityFB outperform the ones

dataset	trait	par	p	r	f
es-test	x	tnvrk=1	.482	1	.65
es-test	e	tnvrk=1	.573	1	.729
es-test	a	tnvrk=1	.533	1	.695
es-test	c	tnvrk=1	.503	1	.669
es-test	o	tnvrk=1	.523	1	.687
es-test	avg	tnvrk=1	.523	1	**.686**
fb-test	x	tnvrk=1	.564	1	.721
fb-test	e	tnvrk=1	.616	1	.762
fb-test	a	tnvrk=1	.667	1	.8
fb-test	c	tnvrk=1	.308	1	.471
fb-test	o	tnvrk=1	.513	1	.678
fb-test	avg	tnvrk=1	.534	1	**.686**

Table 5.4: Precision of confidence generated labels per trait and results of the integration of n-grams in the system. (w)=weigthed features, (t)=bigrams extraction, (n)=normalization, (v)=variable hypothesis generation, (r)=random correction based on skewness, (k)=threshold on feature average

on Essays for almost all traits, except conscientiousness, that performs very bad, and openness, that decreases just a little bit. This fact is hard to explain. We suggest that the main reason of this bad performance for conscientiousness can be found in a particularly skewed distribution for that trait (almost all population in PersonalityFB test set has low conscientiousness scores) or in the fact that patterns extracted for this trait are too much generic. We will return on this point in the next chapter.

Chapter 6

Beyond Adaptive Personality Recogntion

In the previous chapters we have seen how it is possible to extract personality from written text, using cross-language features like punctuation, parentheses and so on. We have seen how it is possible to run automatic domain adaptation to use these features in different domains and languages, we mixed a top-down (correlation set) and a bottom-up approach (bigram patterns), improving the performance of the system. We answered some questions regarding the computational aspects of personality recognition, like the fact that some parameters rise precision and some others rise recall.

Still remain some unanswered questions, like: how do human subjects understand personality of other people from written text? Are there some traits that can be found frequently associated? Can the findings on personality traits tell us something significative from a psychological point of view? In this chapter we try to answer these questions, running some new experiments.

6.1 How Human Subjects predict Personality

We run a psychological experiment with human subjects in order to understand how they make judges about people's personality from written text in a social network domain. We run the experiment online[1], asking the raters to read some portions of text from 10 authors of PersonalityFB, each one written by one single author. Raters were asked to express a judgement about authors' extraversion, stability, agreeableness, conscientiousness and openness to experienc, using the same three classes of the system (yes, no, I do not know). We did that to the purpose of capturing the rate with which subjects decide whether to classify or not personality, and how much they agree on classification. Raters were required to be Italian native speakers and to complete the session in one trial.

[1] http://personality.altervista.org

They were recruited via email or from Facebook. We recruited 35 raters from a different geographical region with respect to the one of the authors, thus preventing the possibility that we have people who know each other.

We computed the inter-rater agreement, compared raters' classification with respect to the Big5, and counted the rate of omitted judgements. There are many inter-coder agreement measures in literature, such as 1) observed agreement (A_o), the percentage of judgements on which two raters agree when coding the same data independently; 2) chance-corrected agreement (like Scott's π and Cohen's *kappa*), based on expected agreement, assuming that if coders were operating by chance alone we would get the same (Scott's π) or a different (Cohen's *kappa*) distribution for each coder; 3) generalized agreement (such as Fleiss's k) which is like chance corrected agreement but it is suitable for many raters; 4) weighted agreement (such as Krippendorff's α), which is applicable to any number of coders and takes into account the differences between types of disagreements. According to Arnstein & Poesio 2008 [5], among all the inter-coder agreement measures used in computational linguistics, weighted agreement is the more informative one, but also the more difficult to interpret. We choose to use Fleiss's k as agreement measure, because it is suitable for many raters, takes into account chance-correction and it is easier

to interpret with respect to Krippendorff's α.

Fleiss's *kappa*, precision, recall, f-measure and omission percentage for each personality trait are reported in table 6.1. On the one hand, kappa measures the agreement among raters and can be interpreted as expressing the extent to which agreement exceeds what would be expected if all raters made their ratings randomly. On the other hand Precision, recall and f-measure can be interpreted here as the agreement between raters and the outcomes of the Big5 test, mediated by written text and measured on the same scale of the APR system, in order to make some comparisons.

According to Sim & Wright 2005 [68], the kappa will be higher

trait	*kappa*	p	r	f	o%
x	.077	.709	.746	.726	21.7%
e	.011	.404	.52	.453	30.6%
a	.079	.293	.372	.327	31.4%
c	.029	.445	.382	.408	44.3%
o	.039	.405	.302	.345	51.7%
avg	.047	.451	.464	.452	35.9%

Table 6.1: Results of the classification test done by human subjects. *kappa*=Fleiss's kappa, p=precision, r=recall, f=f-measure. o%= omission percentage.

when there are fewer categories, here we had three ("y" "n" "o"), but results show that the agreement among raters is poor in general. Only the extraversion and agreeableness traits show a slight agreement. This is an indication that the raters have prediction skills on personality that are slightly above chance, expecially for the emotional stability trait. It is very interesting to note that the extraversion trait has by far the best precision, recall and f-

measure, and the lowest percentage of omissions. This indicates that extraversion is clearly detected by human subjects when reading a text in a social network domain. It seems that they have much more difficulties with the other personality traits. The fact that the agreeableness trait has relatively high kappa, but poor precision, recall and f-measure suggest that subjects do not agree with the Big5. We suggest that the increasing omission percentages (subjects filled in the fields for personality traits in that order: extraversion, emotional stability, agreeableness, conscientiousness, openness) reflects the frustration of subjects doing a task where they have a performance close to chance. In other words we tend to think that people judges personality by chance, unless they detect clear clues of particularly evident traits, that are difficult to judge from textual cues. The fact that they can recognize extrovert people pretty well from text means that extraversion is expressed more by means of linguistic or "semantic" expressions, with respect to other traits. We will analyse this phenomenon more in detail in the next sections.

6.2 Characterise Personality Traits

We found some interesting things that characterise personality traits. For example that confidence scores on some traits, obtained better

performances when paired with other specific traits, rather than when taken separately.

We found this while training supervised classifiers on Essays-dev set. We trained 5 different classifiers, one for each personality trait, and retrieved the models using Weka (Platt's SMO support vector machine algorithm with 10-fold cross-validation [61]). We used hypothesis confidence (m) and confidence per trait (x, e, a, c, o) as features. We found that the best results, reported in table 6.2, are obtained with the feature configurations reported in column 3

trait	run	feat.	p	r	f
X	mbl	m+x+e	.278	.527	.364
X	SMO	m+x+e	.453	.45	.449
E	mbl	m+e	.258	.508	.342
E	SMO	m+e	.553	.55	.548
A	mbl	m+a	.489	.496	.398
A	SMO	m+a	.505	.504	.5
C	mbl	m+c+e	.266	.516	.351
C	SMO	m+c+e	.499	.5	.5
O	mbl	m+c+o	.278	.527	.364
O	SMO	m+c+o	.565	.558	.556

Table 6.2: Average precision, recall and f-measure for supervised models in Weka on essays-dev.

of table 6.2. It is interesting to note that the stability trait helps in learning extraversion and conscientiousness, and that conscientiousness helps in the recognition of openness to experience. Apart for the agreeableness trait, the SMO algorithm improves classifier's performance a lot. We also tried to use other algorithms like decision trees and Naive Bayes, but no algorithm outperformed SMO.

We decided to run an experiment to extract association rules from essays training set (2800 instances) in order to see what per-

sonality traits come together more often. We used Weka Apriori association algorithm (Agrawal & Srikant 1994 [4]; Bing et Al. 1998 [8]), that iteratively reduces the minimum support until it finds the required number of rules with the given minimum confidence. We set minimum confidence to 0.6 and we extracted the best 10 rules, reported in table 6.3. Results show that confidence is not very high,

rank	rule	conf.
1	if e=n & c=y then a=y	0.67
2	if x=n & a=n then e=y	0.67
3	if x=y & e=n then a=y	0.66
4	if e=n & c=y then x=y	0.65
5	if a=y & c=y then e=n	0.64
6	if a=n & c=n then e=y	0.64
7	if x=y & a=y then e=n	0.64
8	if e=n & o=y then a=y	0.64
9	if c=y & o=y then x=y	0.64
10	if e=n & a=y then x=y	0.64

Table 6.3: Best association rules extracted from Essays-training set.

thus indicating that there is variability. Nevertheless, association rules can tell a lot about the relationships between personality and the environment (such as language/culture or a specific social network) when extracted from different domains and compared. This is another reason why Adaptive personality recognition can be useful for research.

The weakness of the theoretical background behind the Big5 does not help much the interpretation of single personality traits, nevertheless the recent efforts in psychology toward a theory of the personality that we introduced in section 1.1, such as Block 2002 and DeYoung 2010, argue that emotional stability and conscien-

tiousness are related to "ego-control", the ability of maintain goals and decision-making, and that openness and extraversion are related to "ego-resiliency", the ability to find new goals. Our finding about conscientiousness and emotional stability can be considered as a hint in supporting this theory.

6.3 Remarks on Extraversion

We have seen that human raters can recognize extraversion from text, but not the other traits, with high f-measure. The contrary is true for the system: on the same datset, extraversion is the worst performing trait, as shown in table 6.4, reporting the details of the best performance on PersonalityFB from table 5.2.

If we compare results in tables 6.4 and 5.4, we can see that

trait	par	set	p	r	f
x	nvrk=1	fb	.359	1	.528
e	nvrk=1	fb	.462	1	.632
a	nvrk=1	fb	.616	1	.762
c	nvrk=1	fb	.462	1	.632
o	nvrk=1	fb	.564	1	.721
avg	nvrk=1	fb	.493	1	**.661**
x	nvrk=1	es	.523	1	.687
e	nvrk=1	es	.533	1	.695
a	nvrk=1	es	.523	1	.687
c	nvrk=1	es	.573	1	.729
o	nvrk=1	es	.523	1	.687
avg	nvrk=1	es	.536	1	**.698**

Table 6.4: Per-trait details of best-performing settings. (n)=normalization, (v)=variable hypothesis generation, (r)=random correction, (k)=threshold on feature average.

bigram patterns improve the precision a lot on extraversion (from

.359 to .564), but it decreases the performance of conscientiousness (from .462 to .308). This indicates that extraversion can be detected mainly by means of a bottom up approach, from words and semantics in general, that is also the approach of human raters judging personality. For the other traits, and in particular conscientiousness, non-semantic features, like the ones in the C correlation set, obtain good performances. We note also that word patterns are important also to detect emotional stability and agreeableness, while they seem less important in detecting conscientiousness and openness to experience. We suggest that this is due to a lack of specific words or patterns associated to traits like conscientiousness or openness to experience. Rather we think that non-semantic cues, like dots for introvert and neurotic are a more or less robust way to express personality in text. The best way to extract personality from text, also for domain adaptation, is to combine bottom-up and top-down approaches.

Chapter 7

Applications: APR for Social Network Analysis

So far we have seen how Adaptive Personality Recognition works; we tested its performance on different languages and domains, and finally we made some considerations about the associations of personality traits. Now we will see some applications of APR to the analysis of text in a social network domain. We will present the results of two analyses: one on the emotional stability trait on Twitter and the other one over all personality traits on Facebook.

These studies show that, although it is not easy to sample and test data from social network sites, the analyses can bring out interesting phenomena that cannot be observed from a qualitative point of view.

7.1 Emotional Stability in Twitter Conversations

In this work, we collected a corpus of about 200000 Twitter posts and we annotated it with our APR system. We modified the system in order to exploit not only linguistic features, such as punctuation, but also network features, such as followers count and retweeted posts. We tested the system on a dataset annotated with personality models produced from human judgements and against the output of onother system. Network analysis shows that neurotic users post more than secure ones and have the tendency to build longer chains of interacting users. Secure users instead have more mutual connections and simpler networks.

Twitter[1] is one of the most popular micro-blogging web services. It was founded in 2006, and allows users to post short messages up to 140 characters of text, called "tweets". According to Boyd et Al. 2010 [13], there are many features that affect practices and conver-

[1] http://twitter.com

sations in Twitter. First of all, connections are directed rather than mutual: users follow other users' feeds and are followed by other users. Public messages can be addressed to specific users with the symbol `@`. According to Honeycutt & Herring 2009 [40] this is used to reply to, to cite or to include someone in a conversation. Messages can be marked and categorized using the "hashtag" symbol `#`, that works as an aggregator of posts having words in common. Another important feature is that posts can be shared and propagated using the "retweet" option. Boyd et Al. 2010 emphasize the fact that retweeting a post is a means of participating in a diffuse conversation. Moreover, posts can be marked as favorites and users can be included into lists. Those practices enhance the visibility of the posts or the users.

Since Tweets are really short, it is very challenging to extract information from them, hence we modified the correlation set, also introducing some new features/correlations based on network structure taken from Quercia et Al. 2011. The list of the features used is reported in table 7.1 (we report only correlations to emotional stability since we are going to extract only that trait).

Testing the system: we run two tests, the first one to evaluate the accuracy in predicting human judges on personality, and the second one to evaluate the performance of the system on Twitter data. In the first one, we compared the results of our system

Features	Corr. to Em. Stab.	from
exclam. marks	-.05*	Mai07
neg. emot.	-.18**	Mai07
numbers	.05*	Mai07
pos. emot.	.07**	Mai07
quest. marks	-.05*	Mai07
long words	.06**	Mai07
w/t freq.	.10**	Mai07
following	-.17**	Qu11
followers	-.19**	Qu11
retweeted	-.03*	Qu11

Table 7.1: Features used in the system and their Pearson's correlation coefficients with personality traits as reported in Mairesse et Al. 2007 and Quercia et Al. 2011. * = p smaller than .05 (weak correlation), ** = p smaller than .01 (strong correlation)

on a dataset, called Personage (see Mairesse & Walker 2007 [49]), annotated with personality ratings from human judges. Raters expressed their judgements on a scale from 1 (low) to 7 (high) for each of the Big Five personality traits on English sentences. In order to obtain a gold standard, we converted this scale into our three-values scheme applying the following rules: if value is greater or equal to 5 then we have "s" (secure), if value is 4 we have "o" and if value is smaller or equal to 3 we have "n" (neurotic). We used a balanced set of 8 users (20 sentences per user), we generated personality hypotheses automatically and we compared them to the gold standard. We obtained an accuracy of .625 over a majority baseline of 0.5. In the second test we compared the output of our system to the score of Analyzewords[2], an online tool for

[2]http://www.analyzewords.com/index.php

Twitter analysis based on LIWC features. This tool does not provide Big5 traits but, among others, it returns scores for "worried" and "upbeat", and we used those classes to evaluate "n" and "s" respectively. We randomly extracted 18 users from our dataset (see section 3 for details), 10 neurotics and 8 secure, and we manually checked whether the classes assigned by our system matched the scores of Analyzewords. Results, reported in table 7.2, reveal

	p	**r**	**f1**
n	0.8	0.615	0.695
s	0.375	0.6	0.462
avg	0.587	0.607	0.578

Table 7.2: Results of test 2.

that our system has a good precision in detecting worried/neurotic users. The bad results for upbeat/secure users could be due to the fact that the class "upbeat" do not correspond perfectly to the "secure" class. Overall the performance of our system is good.

Collection of the Dataset: we collected a corpus, called "Personalitwit2", starting from Twitter's public timeline[3]. The sampling procedure is depicted in figure 7.1. We sampled data from December 25th to 28th, 2011 but most of the posts have a previous posting date since we also collected data from user pages, where 20 recent tweets are displayed in reverse chronological order. For each public user, sampled from the public timeline, we col-

[3]http://twitter.com/public timeline

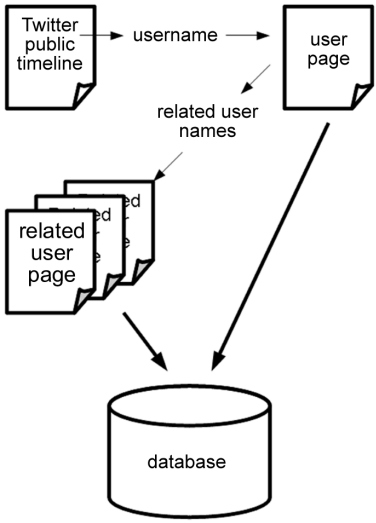

Figure 7.1: Twitter Data sampling pipeline.

lected the nicknames of the related users, who had a conversation with the public users, using the @ symbol. We did this in order to capture users that are included in social relationships with the public users. We excluded from sampling all the retweeted posts because they are not written by the user themselves and could affect linguistic-based personality recognition. The dataset contains all the following information for each post: username; text; post date; user type (public user or related user); user retweet count; user following count; user followers count; user listed count; user favorites count; total tweet count; user page creation year; time zone; related users (users who replied to the sampled user); reply score

(rp), defined as $rp = \frac{page\ reply\ count}{page\ post\ count}$ and retweet score (rt), defined as $rt = \frac{page\ retweet\ count}{page\ post\ count}$. In the corpus there are 200000 posts, more than 13000 different users and about 7800 ego-networks, where public users are the central nodes and related users are the edges. We annotated the corpus with our personality recognition system. The average confidence is 0.601 and the average variability is 0.049. We kept only English users (5392 egonetworks), discarding all the other users.

Analysis: First of all we checked the frequency distribution of emotional stability trait in the corpus is as follows: 56.1% calm users, 39.2% neurotic users and 4.7% balanced users. Then we run

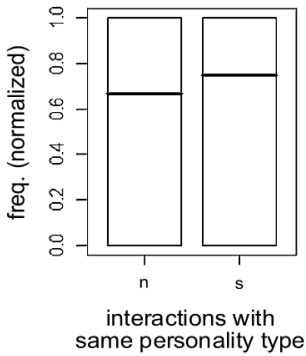

Figure 7.2: Relationships between users with the same personality traits.

a first experiment to check whether neurotic or calm users tend to have conversations with other users with the same personality trait.

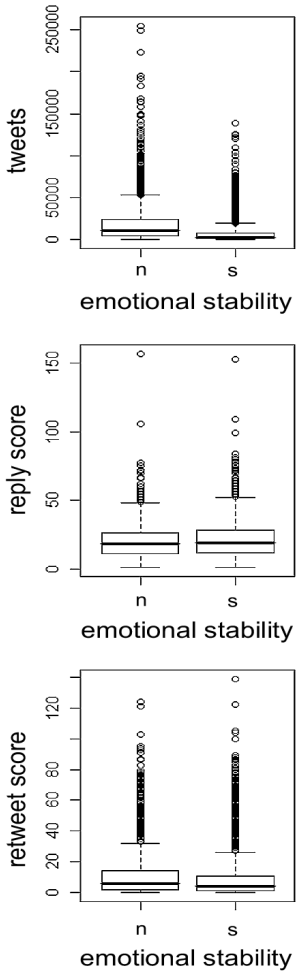

Figure 7.3: Relationships between emotional stability and Twitter activity.

To this purpose we extracted all the ego-networks annotated with personality. We automatically extracted the trait of the personality of the "public-user" (the center of the network) and we counted how many edges of the ego-network have the same personality trait. The frequency is defined as $freq = \frac{trait\ count}{egonetwork\ nodes\ count}$ where the same trait is between the public-user and the related users. The experiment, whose results are reported in figure 7.2, shows that there is a general tendency to have conversations between users that share the same traits.

We run a second experiment to find which personality type is most incline to tweet, to retweet and to reply. Results, reported in figure 7.3, show that neurotic users tend to post and to retweet more than stable users. Stable users are slightly more inclined to reply with respect to neurotic ones. In order to study if conversational practices among users with similar personality traits might generate different social structure, we applied a social network analysis to the collected data through the use of the Gephi software[4]. We analysed separately the network of interactions between neurotic users (n) and calm users (s) to point out any personality related aspect of the emerging social structure. Visualisations are shown in figure 7.4 A.

[4]http://www.gephi.org

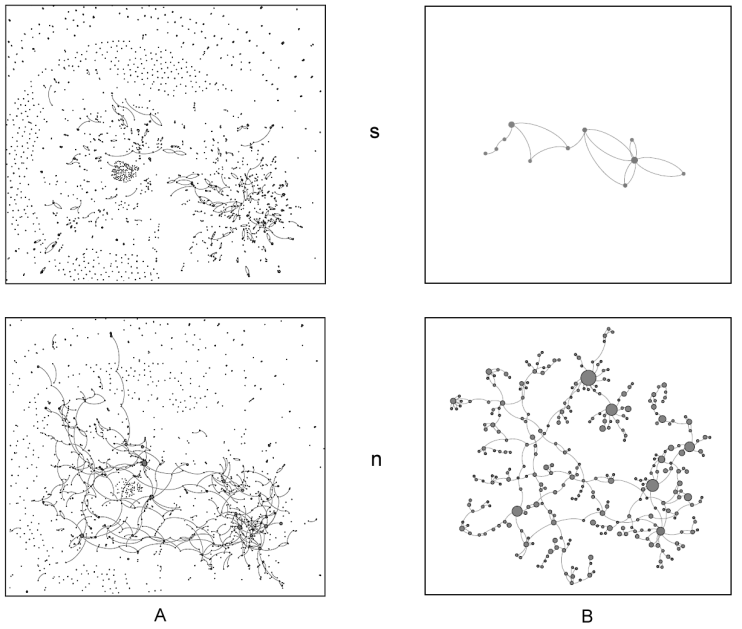

Figure 7.4: Social structures of stable (s) and neurotic (n) users.

The extraction of the ego networks allowed us to detect a rather interesting phenomena: neurotic users seem to have the tendency to build longer chains of interacting users while calm users have the tendency to build mutual connections. This means that a tweet propagated in "neurotic networks" has potentially higher visibility.

The average path length value of neurotic users is 1.551, versus the average path length measured on the calm users of 1.334. This difference results in a network diameter of 6 for the network made of only neurotic users and of 5 for the network made of secure users. A single point of difference in the network diameter produces a neurotic network much more complex than the calm network. While this difference might be overlooked in large visualisations due to the presence of many minor clusters of nodes it becomes evident when we focus only on the giant component of the two networks in figure 7.4 B. The giant components are those counting the major part of nodes and can be used as an example of the most complex structure existing within a network. As it should appear clear neurotic network contains more complex interconnected structures than calm network even if, as we claimed before, have on average smaller social networks.

7.2 Analysis of Facebook Ego-Networks

In this work we addressed the issue of how users' personality affects the way people interact and communicate in Facebook. Due to the strict privacy policy and the lack of a public timeline in Facebook, we automatically sampled data from the timeline of one "access user". Exploiting Facebook's graph APIs, we collected a corpus of about 1100 ego-networks of Italian users (about 5200 posts) and the users that commented their posts. We considered the communicative exchanges, rather than friendships, as a network. We annotated users' personality by means of our personality recognition system and we tested the performance on a small gold standard test set, containing statuses of 23 Facebook users who took the Big5 personality test. Results showed that the system has a average f-measure of .628 (computed over all the five personality traits). The analysis of the network, that has a average path length of 6.635 and a diameter of 14, showed that open-minded users have the highest number of interactions (highest edge weight values) and tend to be influential (they have the highest degree centrality scores), while users with low agreeableness tend to participate in many conversations.

Collection of the Dataset: Sampling data from Facebook is hard. This is due to different factors, like the lack of a public timeline and the strict privacy policy. Both factor prevents from

sampling data from users of which we do not have the friendship. The sampling pipeline can be seen in figure 7.5. We developed a

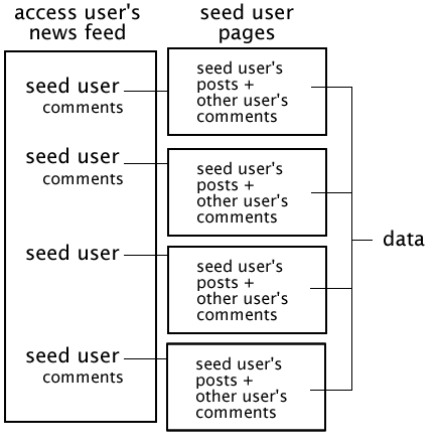

Figure 7.5: Sampling pipeline.

crowler that exploits Facebook's graph API[5] in order to sample users' statuses. The system starts from the news feed of a "access user", who suscribed onto Facebook developer and can take the "access token" key for the API. From the timeline of the access user the system extracts some "seed users" and samples all the statuses and comments written either by the seed users and by the "related users" who interacted with them. The system collects a minimum of 2 posts or comments per user and keeps track of all the users'IDs sampled, in order to avoid duplicates. Finally we filtered

[5]http://developers.facebook.com/tools/explorer

out groups and fanpages and we kept only users. The resulting dataset contains the egonetworks of the seed and related users. Seed users are linked to the related users with weighted "communicative exchanges" relationships. This means that the more a related user commented a seed user, the more the communicative relationship is considered strong. In the dataset there are more than 5000 posts and 1100 users. We annotated the personality of each user by means of our personality recognition system.

Experiments: First of all we retrieved some statistics about the distribution of personality traits in the network and about its topology. The network has a diameter of 14, an average path length of 6.635, average degree centrality of 2.175 and average clustering coefficient of 0.017. This indicates that it is a small network where users have on average a couple of comment-relations each one and with low clustering level. Centrality measures and clustering coefficient have skewed distributions, meaning that a few users have high values and most of them have very low values. The distribution of personality traits, reported in table 7.3, highlights the low

trait	y	o	n
extr.	6.2%	66.4%	27.4%
em. st.	13.7%	49.9%	36.4%
agree	31.9%	65%	3.1%
consc.	13.2%	50.4%	36.4%
open.	27.9%	62.2%	9.9%

Table 7.3: Distribution of personality traits in the network.

number of extroverted, mentally closed and uncooperative people in the network. We suggest that this might be due to the personality of the access user ("noyyy"), that influences the selection of people who are in the network. We will refer to this problem as the "access user bias", that is related to the sampling procedure and does not take place in those networks, like Twitter, where there is a public timeline available.

We analysed the relationship between personality and interactions by computing the association between personality traits and some topology measures, like degree centrality, correlation coefficient and edge weight. In order to do that we measured association scores by computing $as = \frac{bti}{td}$, where bti are the 10 most frequent personality traits associated to each topology measure used, and td is the trait distribution reported in table 7.3.

Results, reported in table 7.4, show several interesting phe-

degree centr.	extr.	em. st.	agree.	consc.	open.
y	0.774	1.387	2.687	2.167	**3.244**
o	0.215	0.381	0.22	0.472	0.077
n	**2.956**	1.701	0	1.308	0.485
edge weight	extr.	em. st.	agree.	consc.	open.
y	0	2.335	2.351	1.923	**3.405**
o	0.15	0.2	0.307	0.198	0.08
n	**3.284**	0.364	1.61	1.785	0
clustering c.	extr.	em. st.	agree.	consc.	open.
y	1.396	0.912	1.567	0.477	1.57
o	0.848	0.501	0.674	0.869	0.905
n	1.016	1.717	**2.032**	1.374	0

Table 7.4: Association scores.

nomena. First of all that introverted and open minded users have the highest degree centrality in the network. In other words they are the ones that are more central and more prone to catch conversations. It is not a surprise that open minded users are in this position, but it is very interesting to note that introvert people have a high degree centrality score too. A closer look to the data reveals that the open minded and introvert traits come often together in the dataset. We suggest this might be due again to the access user bias, because there is a general tendency to have conversations between users that share the same traits (see Celli & Rossi 2012 [17]). The highest edge weight scores are again associated to open minded and introverted users. This means that those users have the strongest links, in other words the highest number of comments. We interpret this as a consequence of the position those users occupy in the topology of the network. Also Agreeable and emotionally stable users have high degree centrality and edge weight scores, indicating that those personality traits play a role in being influential in a conversation network. The distribution of high edge weights is very skewed: there are very few strong links and really a lot of links with low weight. The personality trait associated to high clustering coefficient scores is low agreeableness. If clustering coefficient is related to users' connectedness and links represent comment relationships here, we can interpret this fact as

a hint that uncooperative users tend to participate in many conversations in order to debate in a polemic way. The distribution of clustering coefficient scores is very skewed too.

Final remarks: The outcomes of this experiment show the role that personality traits play in social interactions in a micro network. From the analysis of the most frequent traits associated to topology measures like degree centrality and correlation coefficients, emerged that open minded and introvert users have the highest degree centrality and the strongest links. We interpreted this evidence as introvert and open minded users (those traits come frequently together in the dataset) tend to be very interested to the information that passes through the network, and tend to post interesting (high commented) statuses. Another interesting result is that the users that have high correlation coefficient have low agreeableness. We interpreted this fact as as a hint that uncooperative users tend to participate in many conversations in order to debate in a polemic way. The access user bias, that is due to the restrictions imposed by Facebook and to the lack of a public timeline, prevents from the generalization of those results. Yet it is interesting to observe that a micro network is filtered by the access user according to personality, among other factors. This underlines one more time the importance of personality recognition in the study of social networking.

Chapter 8

Conclusions

In this work we outlined the main problems of PRT (see chapter 3) and we proposed a new approach that tries to overcome these problems (see chapter 4). We developed a system that, given a set of correlations between language and personality traits, and a set of authors and texts, generates personality hypotheses for each author that has more than one text. We experimented a lot with many different parameters and under different conditions.

In particular we compared the performance of our system in two datasets, different for domain and language, finding that the system achieves the same performance on the two (average f=.686). This indicates that our system applies domain adaptation successfully to PRT. The best performances of the system achieved average

f=.698 in essays domain and average f=.73 in social network domain. Our result is in line with the one obtained by Kermanidis 2012 (average f=.687 on modern Greek, essyas domain).

From the experiments we run, emerged that PRT is a task where there is a strong class separability problem, due to the lack of powerful features that allow to separate classes clearly. We also suggested that the separability problem is reflected in the fact that, unlike most semantic tasks in computational linguistics, personality recogntion from text is really hard even for human subjects. This is confirmed by raters' predictions, that are close to chance rate and often subjective, such as judgenments about agreeableness.

If we compare the results obtained by human raters (table 6.1) and by the APR system (table 5.3), we can see that a machine can do this task much more better, except for the extraversion trait. Obviously, when doing this comparison, we must keep in mind that personality recognition from text has to be considered a classification task whose goal is to predict, from few textual cues, the same personality classes that a Big5 test would predict.

Human raters might of course disagree with the Big5. Research in Personality Recognition is based on the assumption that the Big5 can provide an objective point of view over personality while human raters have a subjective one. Of course this Big5-centricity

can be questioned, but the real problem is that, as we have seen, the agreement among raters is poor, surely not sufficient to be an alternative base for reasearch in PRT.

About feature selection we found that a small, cross-language correlation set yelds very good performances with the APR system. This makes the APR system suitable for the analysis of social network sites, where authors are found with their texts in many different languages, and for which it is very difficult to obtain data annotated with personality. We also tested the performance of the system, whose correlations derive from experiments done in English, on Italian, and specifically on social network domain.

We found that the best performance is obtained using a combination of random and semisupervised approach, but really good performances can be achieved also combining bottom-up with top-down approaches, in order to enrich the initial correlation set with patterns extracted from the data at hand.

This second solution has the advantage that does not require even the minimum supervision. We found that simple normalization with the APR approach yields very good results too, but we suggest to use it with caution because it produces hypotheses without “o” labels, that are more difficult to interpret. While normalization improves the perfromance by eliminating “o” labels, thus rising recall, the random-semisupervised approach (and random

correction in general) rises also precision, expecially with unbalanced distributions, where class probability is very informative.

We suggest that this result could be related to the fact that human raters often predict lables by chance, when they do not have enough cues to express their judges. This appears to be a good strategy for such a hard task. The difference between a human and a machine in this case lies in the fact that human get frustrated soon by the task, and their omission rate increases very fast, cutting down recall. A machine, on the contrary, can complete the task without omissions, obtaining a perfect recall, while precision usually ranges between .45 and .56, the random baseline being around .47.

A very interesting result, yet to be explained, is that human raters can predict extraversion (but not other traits) with a surprisingly high precision from written text. This might be also interpreted as Italian subjects agree with the Big5 for the extraversion and not for other traits.

For the future we wish that further psychological experiments on how human subjects detect and predict extraversion could bring more light about how to classify personality traits in a better way, and the efforts in the study of personality as an affect processing system could lead to the classification of few high-level personality traits, like “ego-control” and “ego-resiliency”, that could make the

classification task easier with respect to what it is now.

Bibliography

[1] Abney, S. *Semisupervised Learning for Computational Linguistics*. Chapman & Hall/CRC, London, UK. 2008.

[2] Adelstein J.S, Shehzad Z, Mennes M, DeYoung C.G, Zuo X-N, Kelly C, Margulies D.S, Bloomfield A, Gray J.R, Castellanos X.F, Milham M.P. Personality Is Reflected in the Brain's Intrinsic Functional Architecture. In *PLoS ONE* 6(11). 2011.

[3] Argamon, S., Dhawle S., Koppel, M., Pennebaker J. W. . Lexical Predictors of Personality Type. In *Proceedings of Joint Annual Meeting of the Interface and the Classification Society of North America*. . 2005.

[4] Agrawal,R., Srikant, R. Fast Algorithms for Mining Association Rules in Large Databases. In: *20th International Conference on Very Large Data Bases*, 1994.

[5] Artstein R., Poesio M. Intercoder agreement for Computational Linguistics. In *Computational Linguistics*, 34(4). 2008.

[6] Bai, S., Zhu, T., Cheng, L. Big-Five Personality Prediction Based on User Behaviors at Social Network Sites. In *eprint arXiv:1204.4809*. Available at `http://arxiv.org/abs/1204.4809v1`. 2012.

[7] Ben-David, S., Blitzer, J., Crammer, K., Pereira, F. Analysis of Representations for Domain Adaptation. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*. 2006.

[8] Bing L., Wynne H., Yiming M. Integrating Classification and Association Rule Mining. In: *Fourth International Conference on Knowledge Discovery and Data Mining*, 1998.

[9] Block, J. Personality as an affect-processing system: Toward an integrative theory. Mahwah, NJ: Erlbaum. 2002.

[10] Blum, A., Chawla, S. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of 18th International Conf. on Machine Learning*. 2001.

[11] Blum, A., Mitchell, T. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*. 1998,

[12] Bond, M.H. Nakazato, H.S. Shiraishi, D. Universality and distinctiveness in dimensions of Japanese Person Perception. In *Journal of Cross-Cultural Psychology*. 6. . 1975.

[13] Boyd, D. Golder, S., Lotan, G. Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter. In *Proceedings of HICSS-43*. 2010.

[14] Briggs, I. Myers, P.B. *Gifts differing: Understanding personality type*. Mountain View, CA: Davies-Black Publishing. 1980.

[15] Cambria, E., Hussain, A., Havasi, C., Eckl, C.: SenticSpace: Visualizing Opinions and Sentiments in a Multi-dimensional Vector Space. In *Proceedings of Knowledge-Based and Intelligent Information and Engineering Systems*. 2010.

[16] Celli, F., Unsupervised Personality Recognition for Social Network Sites. In *Proceedings of ICDS*, 2012.

[17] Celli, F., Rossi, L. The role of Emotional Stability in Twitter Conversations. In *Proceedings of Workshop on Semantic Analysis in Social Media, in conjunction with EACL*, Avignon. 2012.

[18] Celli, F., Polonio, L. Relationships between Personality and Interactions in Facebook. In Anne T. Heatherton A. T. and Walcott V. A. (Editors). *Handbook of Social Interactions in the 21st Century*. (to appear).

[19] Chandra P., Cambria E., Pradeep A. Enriching Social Communication through Semantics and Sentics. In *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology*. 2011.

[20] Cheung, F. M., van de Vijver, F. J. R., Leong, F. T. L. Toward a new approach to the study of personality in culture. In *American Psychologist*, Advance online publication. 2011.

[21] Cohen, W. W. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*. 1995.

[22] Cohen, J. A coefficient of agreement for nominal scales, In *Educational and Psychological Measurement*. 20(1). 1960.

[23] Coltheart, M. The MRC psycholinguistic database. In *Quarterly Journal of Experimental Psychology*. 33A. 1981.

[24] Costa, P.T., Jr. McCrae, R.R. The NEO Personality Inventory manual. In *Psychological Assessment Resources*. 1985.

[25] DeYoung, C.G. Toward a Theory of the Big Five. In *Psychological Inquiry*. 21. 2010.

[26] Digman, J.M. Personality structure: Emergence of the five-factor model. In *Annual Review of Psychology*. 41. 1990.

[27] Digman, J. M. Higher-order factors of the Big Five. In *Journal of Personality and Social Psychology*. 73. 1997.

[28] Dy,J. G. Brodley, C. E. Feature Selection for Unsupervised Learning. In *Journal of Machine Learning Research.* (5). 2004.

[29] Fellbaum, C. (Ed.) 1998. *WordNet: An Electronic Lexical Database.* Cambridge: MIT Press.

[30] Freund, Y., Iyer, R., Schapire, R. E., Singer, Y. An efficient boosting algorithm for combining preferences. In *Proceedings of the 15th International Conference on Machine Learning.* 1998.

[31] Gärdenfors, P. Conceptual Spaces as a Framework for Knowledge Representation. In *Mind and Matter* 2(2). 2004.

[32] Gärdenfors, P. Williams, M.A. Reasoning about Categories in Conceptual Spaces. In *Proceedings of IJCAI.* 2001.

[33] Gill, A.J. *Personality and Language: The projection and perception of personality in computer-mediated communication.* Ph.D. Thesis, University of Edinburgh. 2004.

[34] Girju, R., Badulescu, A., Moldovan, D. 2006. Automatic Discovery of Part-Whole Relations. In *Computational Linguistics*, 32(1), pp. 83-136.

[35] Golbeck, J. and Robles, C., and Turner, K. Predicting Personality with Social Media. In *Proceedings of the 2011 annual*

conference extended abstracts on Human factors in computing systems. . 2011.

[36] Goldberg, L., R. The Development of Markers for the Big Five factor Structure. In *Psychological Assessment*, 4(1). 1992.

[37] Grira, N. Crucianu, M. Boujemaain, N. Unsupervised and Semi-supervised Clustering: a Brief Survey *A Review of Machine Learning Techniques for Processing Multimedia Content, Report of the MUSCLE European Network of Excellence (6th Framework Programme).* 2005.

[38] Havasi, C., Speer, R., Alonso, J. ConceptNet 3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge. In *Proceedings of Recent Advances in Natural Languges Processing.* 2007.

[39] Holmes, G. Hall, M. Frank, E.: Generating Rule Sets from Model Trees. In: *Twelfth Australian Joint Conference on Artificial Intelligence.* 1999.

[40] Honeycutt, C., Herring, S. C. Beyond microblogging: Conversation and collaboration via Twitter. In *Proceedings of the Forty-Second Hawaii International Conference on System Sciences.* 2009.

[41] Iacobelli, F., Gill, A.J., Nowson, S. Oberlander, J. Large scale personality classification of bloggers. In *Lecture Notes in Computer Science (6975)*. 2011.

[42] Jiang, J. A Literature Survey on Domain Adaptation of Statistical Classifiers, Draft available at `http://sifaka.cs.uiuc.edu/jiang4/domain_adaptation/survey/`. 2008.

[43] John, G. H., Langley P. Estimating Continuous Distributions in Bayesian Classifiers. In: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. 1995.

[44] John, O. P., Donahue, E. M., Kentle, R. L. The Big Five Inventory: Versions 4a and 5b. *Tech. rep.*. 1991.

[45] Kermanidis, K.L. Mining Authors' Personality Traits from Modern Greek Spontaneous Text. In *4th International Workshop on Corpora for Research on Emotion Sentiment & Social Signals, in conjunction with LREC12*. 2012.

[46] Kotsiantis, S. B. Supervised Machine Learning: A Review of Classification Techniques. In *Informatica*. (31). 2007.

[47] Lodhi, P. H., Deo, S., Belhekar, V. M. The Five-Factor model of personality in Indian context: measurement and correlates. In R. R. McCrae J. Allik (Eds.), *The Five-Factor model of personality across cultures* (pp. 227248). N.Y.: Kluwer Academic Publisher. 2002.

[48] Luyckx K. Daelemans, W. Personae: a corpus for author and personality prediction from text. In: *Proceedings of LREC-2008, the Sixth International Language Resources and Evaluation Conference*. 2008.

[49] Mairesse, F., and Walker, M. PERSONAGE: Personality Generation for Dialogue. In: *In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2007.

[50] Mairesse, F. and Walker, M. A. and Mehl, M. R., and Moore, R, K. Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text. In *Journal of Artificial intelligence Research*, 30. 2007.

[51] Mansour, Y. Learning and Domain Adaptation. In *Discovery Science. Lecture Notes in Computer Science* (5808), 2009.

[52] D. Maynard and K. Bontcheva and D. Rout. (2012). Challenges in developing opinion mining tools for social media. In *Proceedings of NLP can u tag usergeneratedcontent?! Workshop at LREC 2012*, 2012.

[53] Miller, G. A., WordNet: a lexical database for English. In *Commun. of ACM*. 38(11). 1995.

[54] Mintz, M., S. Bills, R. Snow, D. Jurafsky. Distant supervision for relation extraction without labelled data. In *Proceedings of ACL-IJCNLP*. 2009.

[55] Molina, L.C., Belanche, L., Nebot, A. Feature Selection Algorithms: A Survey and Experimental Evaluation. In *Proceedings of the 2002 IEEE International Conference on Data Mining*. 2002.

[56] Nigam, K. Ghani, R. Analyzing the Effectiveness and Applicability of Co-training. In *Ninth International Conference on Information and Knowledge Management (CIKM-2000)*. 2000.

[57] Norman, W., T. Toward an adequate taxonomy of personality attributes: Replicated factor structure in peer nomination personality rating. In *Journal of Abnormal and Social Psychology*, 66. 1963.

[58] Oberlander, J., and Nowson, S. Whose thumb is it anyway? classifying author personality from weblog text. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics ACL*. 2006.

[59] Pennebaker, J. W., King, L. A. Linguistic styles: Language use as an individual difference. In *Journal of Personality and Social Psychology*, 77. 1999.

[60] Pennebaker, J. W., Francis, M. E., Booth, R. J. *Inquiry and Word Count: LIWC 2001.* Lawrence Erlbaum, Mahwah, NJ. 2001.

[61] Platt, J. Machines using Sequential Minimal Optimization. In Schoelkopf, B., Burges, C., Smola, A. (ed), *Advances in Kernel Methods, Support Vector Learning.* 1998.

[62] Quinlan, R. C4.5: Programs for Machine Learning. *Morgan Kaufmann Publishers*, 1993.

[63] Quinlan, R. J. : Learning with Continuous Classes. In: *5th Australian Joint Conference on Artificial Intelligence.* 1992.

[64] Quercia, D. and Kosinski, M. and Stillwell, D., and Crowcroft, J. Our Twitter Profiles, Our Selves: Predicting Personality with Twitter. In *Proceedings of SocialCom2011.* 2011. pp. 180–185.

[65] Raina, R., Battle, A., Lee, H., Packer, B., Ng, A. Y. Self-taught learning: Transfer learning from unlabeled data. In *The 24th International Conference on Machine Learning.* 2007.

[66] Reed, S., Lenat, D. Mapping Ontologies into Cyc. In *Proceedings of AAAI 2002 Conference Workshop on Ontologies For The Semantic Web.* 2002.

[67] Scott, J. Social Network Analysis: developements, advances, and prospects. In *Social Network Analysis and Mining*, 1(1). 2011.

[68] Sim, J. and Wright, C. C. The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements. in *Physical Therapy.* 85(3). 2005.

[69] Staiano J, Lepri B, Aharony N, Pianesi F, Sebe N, Pentland A.S. Friends dont Lie - Inferring Personality Traits from Social Network Structure. In *Proceedings of International Conference on Ubiquitous Computing.* 2012.

[70] Fabian M. Suchanek, F., M. Kasneci, G. Weikum, G. Yago: A Core of Semantic Knowledge. In *16th international World Wide Web conference*, 2007.

[71] Tausczik, Y. R., Pennebaker, J. W. The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. In *Journal of Language and Social Psychology.* 29(1). 2010.

[72] Trull, T. J. Geary, D. C. Comparison of the big-five factor structure across samples of Chinese and American adults. *Journal of Personality Assessment.* 69(2). 1997.

[73] Van Zalk, N., Van Zalk, M., Kerr, M. and Stattin, H. Social Anxiety as a Basis for Friendship Selection and Socialization in

Adolescents' Social Networks. In *Journal of Personality.* (79). 2011.

[74] Vapnik, V. *The Nature of Statistical Learning Theory.* Springer Verlag. 1995.

[75] Jun Wang, J., Zucker, J.D. Solving Multiple-Instance Problem: A Lazy Learning Approach. In: *17th International Conference on Machine Learning.* 2000.

[76] Witten, I, H., Frank, E. *Data Mining. Practical Machine Learning Tools and Techniques with Java implementations.* Morgan and Kaufman, 2005.

[77] Yarowsky, D. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics.* 1995.

[78] Zesch, T., Müller C., Gurevych, I. Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In *Proceedings of LREC.* 2008.

[79] Zhu, X. Semi-supervised learning literature survey. *Technical Report 1530, Department of Computer Sciences, University of Wisconsin.* 2005.