

Cristian Lai, Giovanni Semeraro, Alessandro Giuliani (Eds.)

Proceedings of the 8th International Workshop on
Information Filtering and Retrieval

Workshop of the XIII AI*IA Symposium on Artificial Intelligence



December 10, 2014

Pisa, Italy

<http://aiia2014.di.unipi.it/dart/>

Preface

With the increasing availability of data, it becomes more important to have automatic methods to manage data and retrieve information. Data processing, especially in the era of Social Media, is changing users behaviours. Users are ever more interested in information rather than in mere raw data. Considering that the large amount of accessible data sources is growing, novel systems providing effective means of searching and retrieving information are required. Therefore the fundamental goal is making information exploitable by humans and machines.

DART 2014 intends to provide a more interactive and focused platform for researchers and practitioners for presenting and discussing new and emerging ideas. It is focused on researching and studying new challenges in Intelligent Information Filtering and Retrieval. In particular, DART aims to investigate novel systems and tools to web scenarios and semantic computing. In so doing, DART will contribute to discuss and compare suitable novel solutions based on intelligent techniques and applied in real-world applications. Information Retrieval attempts to address similar filtering and ranking problems for pieces of information such as links, pages, and documents. Information Retrieval systems generally focus on the development of global retrieval techniques, often neglecting individual user needs and preferences. Information Filtering has drastically changed the way information seekers find what they are searching for. In fact, they effectively prune large information spaces and help users in selecting items that best meet their needs, interests, preferences, and tastes. These systems rely strongly on the use of various machine learning tools and algorithms for learning how to rank items and predict user evaluation.

Submitted proposals received three review reports from Program Committee members. Based on the recommendations of the reviewers, 5 full papers have been selected for publication and presentation at DART 2014. In addition, Fabrizio Sebastiani, who is a Principal Scientist at Qatar Computing Research Institute in Doha (Qatar), gave a plenary talk on Explicit Loss Minimization in Quantification Applications.

When organizing a scientific conference, one always has to count on the efforts of many volunteers. We are grateful to the members of the Program Committee, who devoted a considerable amount of their time in reviewing the submissions to DART 2014.

We were glad and happy to work together with highly motivated people to arrange the conference and to publish these proceedings. We appreciate the work of the Publicity Chair Fedelucio Narducci from University of Bari Aldo Moro for announcing the workshop on various lists. Special thanks to Salvatore Ruggieri for the support and help in managing the workshop organization.

We hope that you find these proceedings a valuable source of information on intelligent information filtering and retrieval tools, technologies, and applications.

December 8, 2014
Cagliari

Cristian Lai,
Giovanni Semeraro,
Alessandro Giuliani

Table of Contents

| | |
|--|----|
| Explicit Loss Minimization in Quantification Applications (Preliminary Draft) | 1 |
| <i>Andrea Esuli and Fabrizio Sebastiani</i> | |
| A scalable approach to near real-time sentiment analysis on social networks | 12 |
| <i>Giambattista Amati, Simone Angelini, Marco Bianchi, Luca Costantini and Giuseppe Marcone</i> | |
| Telemonitoring and Home Support in BackHome | 24 |
| <i>Felip Miralles, Eloisa Vargiu, Stefan Dauwalder, Marc Solà, Juan Manuel Fernández, Eloi Casals and José Alejandro Cordero</i> | |
| Extending an Information Retrieval System through Time Event Extraction | 36 |
| <i>Pierpaolo Basile, Annalina Caputo, Giovanni Semeraro and Lucia Siciliani</i> | |
| Measuring Discriminant and Characteristic Capability for Building and Assessing Classifiers | 48 |
| <i>Giuliano Armano, Francesca Fanni and Alessandro Giuliani</i> | |
| A comparison of Lexicon-based approaches for Sentiment Analysis of microblog posts | 59 |
| <i>Cataldo Musto, Giovanni Semeraro and Marco Polignano</i> | |

Program Committee

| | |
|-----------------------|---|
| Marie-Helene Abel | University of Compiègne |
| Giambattista Amati | Fondazione Ugo Bordoni |
| Liliana Ardissono | University of Torino |
| Giuliano Armano | Department of Electrical and Electronic Engineering, University of Cagliari |
| Agnese Augello | ICAR CNR Palermo |
| Pierpaolo Basile | University of Bari Aldo Moro |
| Roberto Basili | University of Rome "Tor Vergata" |
| Federico Bergenti | University of Parma |
| Ludovico Boratto | University of Cagliari |
| Annalina Caputo | University of Bari Aldo Moro |
| Pierluigi Casale | Eindhoven University of Technology |
| Jose Cunha | University Nova of Lisbon |
| Marco De Gemmis | University of Bari Aldo Moro |
| Emanuele Di Buccio | University of Padua |
| Francesca Fanni | Department of Electrical and Electronic Engineering, University of Cagliari |
| Juan Manuel Fernández | Barcelona Digital Technology Center |
| Alessandro Giuliani | Department of Electrical and Electronic Engineering, University of Cagliari |
| Nima Hatami | University of San Diego |
| Leo Iaquina | University of Bari Aldo Moro |
| Jose Antonio Iglesias | University of Madrid |
| Cristian Lai | CRS4, Center of Advanced Studies, Research and Development in Sardinia |
| Pasquale Lops | University of Bari Aldo Moro |
| Massimo Melucci | University of Padua |
| Maurizio Montagnuolo | RAI Centre for Research and Technological Innovation |
| Claude Moulin | University of Compiègne |
| Vincenzo Pallotta | University of Business and International Studies at Geneva |
| Marcin Paprzycki | Polish Academy of Sciences |
| Gabriella Pasi | University of Milan Bicocca |
| Agostino Poggi | University of Parma |
| Sebastian Rodriguez | Universidad Tecnologica Nacional |
| Paolo Rosso | Polytechnic University of Valencia, |
| Giovanni Semeraro | Dipartimento di Informatica - University of Bari Aldo Moro |
| Eloisa Vargiu | Barcelona Digital Technology Center |

Additional Reviewers

H

Hernández Farias, Delia Irazú

Explicit Loss Minimization in Quantification Applications (Preliminary Draft)

Andrea Esuli[†] and Fabrizio Sebastiani[‡]

[†] Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
56124 Pisa, Italy
E-mail: andrea.esuli@isti.cnr.it

[‡] Qatar Computing Research Institute
Qatar Foundation
PO Box 5825, Doha, Qatar
E-mail: fsebastiani@qf.org.qa

Abstract. In recent years there has been a growing interest in *quantification*, a variant of classification in which the final goal is not accurately classifying each unlabelled document but accurately estimating the prevalence (or “relative frequency”) of each class c in the unlabelled set. Quantification has several applications in information retrieval, data mining, machine learning, and natural language processing, and is a dominant concern in fields such as market research, epidemiology, and the social sciences. This paper describes recent research in addressing quantification via explicit loss minimization, discussing works that have adopted this approach and some open questions that they raise.

1 Introduction

In recent years there has been a growing interest in *quantification* (see e.g., [2, 3, 9, 12, 19]), which we may define as the task of estimating the prevalence (or “relative frequency”) $p_S(c)$ of a class c in a set S of objects whose membership in c is unknown. Technically, quantification is a *regression* task, since it consists in estimating a function $h : \mathcal{S} \times \mathcal{C} \rightarrow [0, 1]$, where $\mathcal{S} = \{s_1, s_2, \dots\}$ is a domain of sets of objects, $\mathcal{C} = \{c_1, \dots, c_{|\mathcal{C}|}\}$ is a set of classes, and $h(s, c)$ is the estimated prevalence of class c in set s . However, quantification is more usually seen as a variant of *classification*, a variant in which the final goal is not (as in classification) predicting the class(es) to which an unlabelled object belongs, but accurately

⁰ The order in which the authors are listed is purely alphabetical; each author has given an equally important contribution to this work. Fabrizio Sebastiani is on leave from Consiglio Nazionale delle Ricerche.

estimating the percentages of unlabelled objects that belong to each class $c \in \mathcal{C}$. Quantification is usually tackled via supervised learning; it has several applications in information retrieval [8, 9], data mining [11–13], machine learning [1, 20], and natural language processing [4], and is a dominant concern in fields such as market research [7], epidemiology [18], and the social / political sciences [15].

Classification comes in many variants, including *binary* classification (where $|\mathcal{C}| = 2$ and exactly one class per item is assigned), *single-label multi-class* (SLMC) classification (where $|\mathcal{C}| > 2$ and exactly one class per item is assigned), and *multi-label multi-class* (MLMC) classification (where $|\mathcal{C}| \geq 2$ and zero, one or several classes per item may be assigned). To each such classification task there corresponds a quantification task, which is concerned with evaluating at the aggregate level (i.e., in terms of class prevalence) the results of the corresponding classification task. In this paper we will mostly be concerned with binary quantification, although we will also occasionally hint at how and whether the solutions we discuss extend to the SLMC and MLMC cases.

Quantification is not a mere byproduct of classification, and is tackled as a task on its own. The reason is that the naive quantification approach consisting of (a) classifying all the test items and (b) counting the number of items assigned to each class (the “classify and count” method – CC) is suboptimal. In fact, a classifier may have good classification accuracy but bad quantification accuracy; for instance, if the binary classifier generates many more false negatives (FN) than false positives (FP), the prevalence of the positive class will be severely underestimated.

As a result, several quantification methods that deviate from mere “classify and count” have been proposed. Most such methods fall in two classes. In the first approach a generic classifier is trained and applied to the test data, and the computed prevalences are then corrected according to the estimated bias of the classifier, which is estimated via k -fold cross-validation on the training set; “adjusted classify and count” (ACC) [14], “probabilistic classify and count” (PCC) [3], and “adjusted probabilistic classify and count” (PACC) [3] fall in this category. In the second approach, a “classify and count” method is used on a classifier in which the acceptance threshold has been tuned so as to deliver a different proportion of predicted positives and predicted negatives; example methods falling in this category are the “Threshold@50” (T50), “MAX”, “X”, and “median sweep” (MS) methods proposed in [12]. See also [9] for a more detailed explanation of all these methods.

In this paper we review an emerging class of methods, based on *explicit loss minimization* (ELM). Essentially, their underlying idea is to use (unlike the first approach mentioned above) simple “classify and count” without (unlike the second approach mentioned above) any heuristic threshold tuning, but using a classifier trained via a learning method explicitly optimized for quantification accuracy. This idea was first proposed, but not implemented, in a position paper by Esuli and Sebastiani [8], and was taken up by three very recent works [2, 9, 19] that we will discuss here.

The rest of this paper is organized as follows. Section 2 discusses evaluation measures for quantification used in the literature. Section 3 discusses the reason why approaching quantification via ELM is impossible with standard learning algorithms, and discusses three ELM approaches to quantification that have made use of nonstandard such algorithms. Section 4 discusses experimental results, while Section 5 concludes discussing questions that existing research has left open.

2 Loss Measures for Evaluating Quantification Error

ELM requires the loss measure used for evaluating prediction error to be directly minimized within the learning process. Let us thus look at the measures which are currently being used for evaluating SLMC quantification error. Note that a measure for SLMC quantification is also a measure for binary quantification, since the latter task is a special case of the former. Note also that a measure for binary quantification is also a measure for MLMC quantification, since the latter task can be solved by separately solving $|\mathcal{C}|$ instances of the former task, one for each $c \in \mathcal{C}$.

Notation-wise, by $A(\hat{p}, p, S, \mathcal{C})$ we will indicate a *quantification loss*, i.e., a measure A of the error made in estimating a distribution p defined on set S and classes \mathcal{C} by another distribution \hat{p} ; we will often simply write $A(\hat{p}, p)$ when S and \mathcal{C} are clear from the context.

The simplest measure for SLMC quantification is *absolute error* (AE), which corresponds to the sum (across the classes in \mathcal{C}) of the absolute differences between the predicted class prevalences and the true class prevalences; i.e.,

$$AE(\hat{p}, p) = \sum_{c_j \in \mathcal{C}} |\hat{p}(c_j) - p(c_j)| \quad (1)$$

AE ranges between 0 (best) and $2(1 - \min_{c_j \in \mathcal{C}} p(c_j))$ (worst); a normalized version of AE that always ranges between 0 (best) and 1 (worst) can thus be obtained as

$$NAE(\hat{p}, p) = \frac{\sum_{c_j \in \mathcal{C}} |\hat{p}(c_j) - p(c_j)|}{2(1 - \min_{c_j \in \mathcal{C}} p(c_j))} \quad (2)$$

The main advantage of AE and NAE is that they are intuitive, and easy to understand to non-initiates too.

However, AE and NAE do not address the fact that the same absolute difference between predicted class prevalence and true class prevalence should count as a more serious mistake when the true class prevalence is small. For instance, predicting $\hat{p}(c) = 0.10$ when $p(c) = 0.01$ and predicting $\hat{p}(c) = 0.50$ when $p(c) = 0.41$ are equivalent errors according to AE , but the former is intuitively a more serious error than the latter. *Relative absolute error* (RAE) addresses this problem by relativizing the value $|\hat{p}(c_j) - p(c_j)|$ in Equation 1 to the true class prevalence, i.e.,

$$RAE(\hat{p}, p) = \sum_{c_j \in \mathcal{C}} \frac{|\hat{p}(c_j) - p(c_j)|}{p(c_j)} \quad (3)$$

RAE may be undefined in some cases, due to the presence of zero denominators. To solve this problem, in computing *RAE* we can smooth both $p(c_j)$ and $\hat{p}(c_j)$ via additive smoothing, i.e.,

$$p_s(c_j) = \frac{p(c_j) + \epsilon}{\left(\sum_{c_j \in \mathcal{C}} p(c_j)\right) + \epsilon \cdot |\mathcal{C}|} \quad (4)$$

where $p_s(c_j)$ denotes the smoothed version of $p(c_j)$ and the denominator is just a normalizing factor (same for the $\hat{p}_s(c_j)$'s); the quantity $\epsilon = \frac{1}{2 \cdot |\mathcal{S}|}$ is often used as a smoothing factor. The smoothed versions of $p(c_j)$ and $\hat{p}(c_j)$ are then used in place of their original versions in Equation 3; as a result, *RAE* is always defined and still returns a value of 0 when p and \hat{p} coincide.

RAE ranges between 0 (best) and $\left(\frac{1 - \min_{c_j \in \mathcal{C}} p(c_j)}{\min_{c_j \in \mathcal{C}} p(c_j)} + |\mathcal{C}| - 1\right)$ (worst); a normalized version of *RAE* that always ranges between 0 (best) and 1 (worst) can thus be obtained as

$$NRAE(\hat{p}, p) = \frac{\sum_{c_j \in \mathcal{C}} \frac{|\hat{p}(c_j) - p(c_j)|}{p(c_j)}}{1 - \min_{c_j \in \mathcal{C}} p(c_j) + \frac{\min_{c_j \in \mathcal{C}} p(c_j)}{|\mathcal{C}|} + |\mathcal{C}| - 1} \quad (5)$$

A third measure, and the one that has become somehow standard in the evaluation of SLMC quantification, is *normalized cross-entropy*, better known as *Kullback-Leibler Divergence* (KLD – see e.g., [5]). KLD was proposed as a SLMC quantification measure in [10], and is defined as

$$KLD(\hat{p}, p) = \sum_{c_j \in \mathcal{C}} p(c_j) \log \frac{p(c_j)}{\hat{p}(c_j)} \quad (6)$$

KLD is a measure of the error made in estimating a true distribution p over a set \mathcal{C} of classes by means of a predicted distribution \hat{p} . *KLD* is thus suitable for evaluating quantification, since quantifying exactly means predicting how the items in set s are distributed across the classes in \mathcal{C} .

KLD ranges between 0 (best) and $+\infty$ (worst). Note that, unlike *AE* and *RAE*, the upper bound of *KLD* is not finite since Equation 6 has predicted probabilities, and not true probabilities, at the denominator: that is, by making a predicted probability $\hat{p}(c_j)$ infinitely small we can make *KLD* be infinitely large. A normalized version of *KLD* yielding values between 0 (best) and 1 (worst) may be defined by applying a logistic function, e.g.,

$$NKLD(\hat{p}, p) = \frac{e^{KLD(\hat{p}, p)} - 1}{e^{KLD(\hat{p}, p)}} \quad (7)$$

Also *KLD* may be undefined in some cases. While the case in which $p(c_j) = 0$ is not problematic (since continuity arguments indicate that $0 \log \frac{0}{a}$ should be

taken to be 0 for any $a \geq 0$), the case in which $\hat{p}(c_j) = 0$ and $p(c_j) > 0$ is indeed problematic, since $a \log \frac{a}{0}$ is undefined for $a > 0$. To solve this problem, also in computing *KLD* we use the smoothed probabilities of Equation 4; as a result, *KLD* is always defined and still returns a value of zero when p and \hat{p} coincide.

The main advantage of *KLD* is that it is a very well-known measure, having been the subject of intense study within information theory [6] and, although from a more applicative angle, within the language modelling approach to information retrieval [23]. Its main disadvantage is that it is less easy to understand to non-initiates than *AE* or *RAE*.

Overall, while no measure is advantageous under all respects, *KLD* (or *NKLD*) wins over the other measures on several accounts; as a consequence, it has emerged as the *de facto* standard in the SLMC quantification literature. We will hereafter consider it as such.

3 Quantification Methods Based on Explicit Loss Minimization

A problem with the quantification methods hinted at in Section 1 is that most of them are fairly heuristic in nature. A further problem is that some of these methods rest on assumptions that seem problematic. For instance, one problem with the ACC method is that it seems to implicitly rely on the hypothesis that estimating the bias of the classifier via k -fold cross-validation on *Tr* can be done reliably. However, since the very motivation of doing quantification is that the training set and the test set may have quite different characteristics, this hypothesis seems adventurous. In sum, the very same arguments that are used to deem the CC method unsuitable for quantification seem to undermine the previously mentioned attempts at improving on CC.

Note that all of the methods discussed in Section 1 employ *general-purpose* supervised learning methods, i.e., address quantification by leveraging a classifier trained via a general-purpose learning method. In particular, most of the supervised learning methods adopted in the literature on quantification optimize zero-one loss or variants thereof, and not a quantification-specific evaluation function. When the dataset is imbalanced (typically: when the positives are by far outnumbered by the negatives), as is frequently the case in text classification, this is suboptimal, since a supervised learning method that minimizes zero-one loss will generate classifiers with a tendency to make negative predictions. This means that *FN* will be much higher than *FP*, to the detriment of quantification accuracy¹.

In this paper we look at new, theoretically better-founded quantification methods based upon the use of classifiers explicitly optimized for the evaluation function used for assessing quantification accuracy. The idea of using learning

¹ To witness, in the experiments reported in [9] the 5148 test sets exhibit, when classified by the classifiers generated by the linear SVM used for implementing the CC method, an average *FP/FN* ratio of 0.109; by contrast, for an optimal quantifier this ratio is always 1.

algorithms capable of directly optimizing the measure (a.k.a. “loss”) used for evaluating effectiveness is well-established in supervised learning. However, in our case following this route is non-trivial; let us see why.

As usual, we assume that our training data $Tr = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_{|Tr|}, y_{|Tr|})\}$ and our test data $Te = \{(\mathbf{x}'_1, y'_1), \dots, (\mathbf{x}'_{|Te|}, y'_{|Te|})\}$ are independently generated from an unknown joint distribution $P(\mathcal{X}, \mathcal{Y})$, where \mathcal{X} and \mathcal{Y} are the input and output spaces, respectively. In this paper we will assume \mathcal{Y} to be $\{-1, +1\}$.

Our task is to learn from Tr a hypothesis $h \in \mathcal{H}$ (where $h : \mathcal{X} \rightarrow \mathcal{Y}$ and \mathcal{H} is the hypothesis space) that minimizes the expected risk $R^A(h)$ on sets Te of previously unseen inputs. Here A is our chosen loss measure; note that it is a set-based (rather than an instance-based) measure, i.e., it measures the error incurred by making an entire set of predictions, rather than (as instance-based measures λ do) the error incurred by making a single prediction. Our task consists thus of finding

$$\arg \min_{h \in \mathcal{H}} R^A(h) = \int A((h(\mathbf{x}'_1), y'_1), \dots, (h(\mathbf{x}'_{|Te|}, y'_{|Te|}))) dP(Te) \quad (8)$$

If the loss function A over sets Te can be linearly decomposed into the sum of the individual losses λ generated by the members of Te , i.e., if

$$A((h(\mathbf{x}'_1), y'_1), \dots, (h(\mathbf{x}'_{|Te|}, y'_{|Te|}))) = \sum_{i=1}^{|Te|} \lambda(h(\mathbf{x}'_i), y'_i) \quad (9)$$

then Equation 8 comes down to

$$\arg \min_{h \in \mathcal{H}} R^A(h) = \arg \min_{h \in \mathcal{H}} R^\lambda(h) = \int \lambda(h(\mathbf{x}', y')) dP(\mathbf{x}', y') \quad (10)$$

Discriminative learning algorithms estimate the expected risk $R^A(h)$ via the empirical risk (or “training error”) $R^A_{Tr}(h)$, which by virtue of Equation 9 becomes

$$\hat{R}^A(h) = R^A_{Tr}(h) = R^\lambda_{Tr}(h) = \sum_{i=1}^{|Tr|} \lambda(h(\mathbf{x}_i, y_i)) \quad (11)$$

and pick the hypothesis h which minimizes $R^\lambda_{Tr}(h)$.

The problem with adopting this approach in learning to quantify is that all quantification loss measures A are such that Equation 9 does not hold. In other words, such loss measures are *nonlinear* (since they do not linearly decompose into the individual losses brought about by the members in the set) and *multivariate* (since A is a function of all the instances, and does not break down into univariate loss functions). For instance, we cannot evaluate the contribution to *KLD* of a classification decision for a single item \mathbf{x}'_i , since this contribution will depend on how the other items have been classified; if the other items have given rise, say, to more false negatives than false positives, then misclassifying a negative example (thus bringing about an additional false positive) is even beneficial!, since false

positives and false negatives cancel each other out when it comes to quantification accuracy. This latter fact shows that *any* quantification loss (and not just the ones discussed in Section 2) is *inherently* nonlinear and multivariate. This means that, since Equations 9–11 do not hold for quantification loss measures \mathcal{A} , we need to seek learning methods that can explicitly minimize $R_{Tr}^{\mathcal{A}}(h)$ holistically, i.e., without making the “reductionistic” assumption that $R^{\mathcal{A}}(h) = R^{\lambda}(h)$.

As mentioned in the introduction, the idea to use ELM in quantification applications was first proposed, but not implemented, in [8]. In this section we will look at three works [2, 9, 19] that have indeed exploited this idea, although in three different directions.

3.1 Quantification via Structured Prediction I: SVM(KLD) [9]

In [8] Esuli and Sebastiani also suggested using, as a “holistic” algorithm of the type discussed in the previous paragraph, the *SVM for Multivariate Performance Measures* (SVM_{perf}) learning algorithm proposed by Joachims [16]².

SVM_{perf} is a learner of the Support Vector Machine family that can generate classifiers optimized for any non-linear, multivariate loss function that can be computed from a contingency table (as all the measures presented in Section 2 are). SVM_{perf} is an algorithm for *multivariate prediction*: instead of handling hypotheses $h : \mathcal{X} \rightarrow \mathcal{Y}$ mapping an individual item \mathbf{x}_i into an individual label y_i , it considers hypotheses $\bar{h} : \bar{\mathcal{X}} \rightarrow \bar{\mathcal{Y}}$ which map entire tuples of items $\bar{\mathbf{x}} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ into tuples of labels $\bar{\mathbf{y}} = (y_1, \dots, y_n)$, and instead of learning hypotheses of type

$$h(\mathbf{x}) : \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) \tag{12}$$

it learns hypotheses of type

$$\bar{h}(\bar{\mathbf{x}}) : \arg \max_{\bar{\mathbf{y}} \in \bar{\mathcal{Y}}} (\mathbf{w} \cdot \Psi(\bar{\mathbf{x}}, \bar{\mathbf{y}})) \tag{13}$$

where \mathbf{w} is the vector of parameters to be learnt during training and

$$\Psi(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = \sum_{i=1}^n \mathbf{x}_i y'_i \tag{14}$$

(the *joint feature map*) is a function that scores the pair of tuples $(\bar{\mathbf{x}}, \bar{\mathbf{y}}')$ according to how “compatible” the tuple of labels $\bar{\mathbf{y}}'$ is with the tuple of inputs $\bar{\mathbf{x}}$.

While the optimization problem of classic soft-margin SVMs consists in finding

$$\begin{aligned} \arg \min_{\mathbf{w}, \xi_i \geq 0} & \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{|Tr|} \xi_i \\ \text{such that} & y_i [\mathbf{w} \cdot \mathbf{x}_i + b] \geq (1 - \xi_i) \text{ for all } i \in \{1, \dots, |Tr|\} \end{aligned} \tag{15}$$

² In [16] SVM_{perf} is actually called *SVM \mathcal{A}_{multi}* , but the author has released its implementation under the name SVM_{perf} ; we will indeed use this latter name.

the corresponding problem of SVM_{perf} consists instead of finding

$$\begin{aligned} & \arg \min_{\mathbf{w}, \xi \geq 0} \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C\xi \\ & \text{such that } \mathbf{w} \cdot [\Psi(\bar{\mathbf{x}}, \bar{y}) - \Psi(\bar{\mathbf{x}}, \bar{y}') + b] \geq \Lambda(\bar{y}, \bar{y}') - \xi \text{ for all } \bar{y}' \in \bar{\mathcal{Y}}/\bar{y} \end{aligned} \quad (16)$$

Here, the relevant thing to observe is that the sample-based loss Λ explicitly appears in the optimization problem.

We refer the interested reader to [16, 17, 21] for more details on SVM_{perf} and on SVMs for structured output prediction in general. From the point of view of the user interested in applying it to a certain task, the implementation of SVM_{perf} made available by its author is essentially an off-the-shelf package, since for customizing it to a specific loss Λ one only needs to write a small module that describes how to compute Λ from a contingency table.

While [8] only went as far as *suggesting* the use of SVM_{perf} to optimize a quantification loss, its authors later went on to actually implement the idea, using *KLD* as the quantification loss and naming the resulting system SVM(KLD) [9]. In Section 4 we will describe some of the insights that they obtained from experimenting it.

3.2 Quantification Trees and Quantification Forests [19]

Rather than working in the framework of SVMs, the work of Milli and colleagues [19] perform explicit loss minimization in the context of a decision tree framework. Essentially, their idea is to use a quantification loss as the splitting criterion in generating a decision tree, thereby generating a *quantification tree* (i.e., a decision tree specialized for quantification). The authors experiment with three different quantification loss measures: (a) (a proxy of) absolute error, i.e., $D(\hat{p}, p) = \sum_{c_j \in \mathcal{C}} |FP - FN|$; (b) KLD; (c) $MOM(\hat{p}, p) = \sum_{c_j \in \mathcal{C}} |FP_j^2 - FN_j^2|$. Measure (c) is of particular significance since it is not a “pure” quantification loss. In fact, notice that $MOM(\hat{p}, p)$ is equivalent to $\sum_{c_j \in \mathcal{C}} (FN_j + FP_j) \cdot |FN_j - FP_j|$, and that while the second factor ($|FN_j - FP_j|$) may indeed be seen as representing quantification error, the first factor ($FN_j + FP_j$) is a measure of *classification* error. The motivation behind the authors’ choice is to minimize at the same time classification and quantification error, based on the notion that a quantifier that has good quantification accuracy but low classification accuracy is somehow unreliable, and should be avoided.

The authors go on to propose the use of *quantification forests*, i.e., random forests of quantification trees, where these latter are defined as above. For more details we refer the interested reader to [19].

It should be remarked that [19] is the only one, among the three works we review in this section, that directly addresses SLMC quantification. The other two works that we have discussed [2, 9] instead address binary quantification only; in order to extend their approach to SLMC quantification, binary quantification has to be performed independently for each class and the resulting class prevalences have to be rescaled so that they sum up to 1. This is certainly suboptimal, but

better solutions are not known since a SLMC equivalent of SVM_{perf} , which is binary in nature, is not known.

3.3 Quantification via Structured Prediction II: SVM(Q) [2]

Barranquero et al.’s forthcoming work [2] proposes an approach to binary quantification that combines elements of the works carried out in [9] and [19]. As suggested in [8], and as later implemented in [9], Barranquero et al. also use SVM_{perf} to directly optimize quantification accuracy. However, similarly to [19], they optimize a measure (which they call *Q-measure*) that combines classification accuracy and quantification accuracy. Their Q-measure is shaped upon the famous F_β measure [22, Chapter 7], leading to a loss defined as

$$A = (1 - Q_\beta(\hat{p}, p)) = \left(1 - \frac{(\beta^2 + 1)\Gamma_c(\hat{p}, p) \cdot \Gamma_q(\hat{p}, p)}{\beta^2\Gamma_c(\hat{p}, p) + \Gamma_q(\hat{p}, p)}\right)$$

where Γ_c and Γ_q are a measure of classification “gain” (the opposite of loss) and a measure of quantification gain, respectively, and $0 \leq \beta \leq +\infty$ is a parameter that controls the relative importance of the two; for $\beta = 0$ the Q_β measure coincides with Γ_c , while when β tends to infinity Q_β asymptotically tends to Γ_q .

As a measure of classification gain Barranquero et al. use recall, while as a measure of quantification gain they use $(1 - NAE)$, where NAE is as defined in Equation 2. The authors motivate the (apparently strange) decision to use recall as a measure of classification gain with the fact that, while recall by itself is not a suitable measure of classification gain (since it is always possible to arbitrarily increase recall at the expense of precision or specificity), to include precision or specificity in Q_β is unnecessary, since the presence of Γ_q in Q_β has the effect of ruling out anyway those hypotheses characterized by high recall and low precision / specificity (since these hypotheses are indeed penalized by Γ_q). The experiments presented in the paper test values for β in $\{0.5, 1, 2\}$.

4 Experiments

The approaches that the three papers mentioned in this section have proposed have never been compared experimentally, since the experimentation they report use different datasets. The only paper among the three where the experimentation is carried out on high-dimensional datasets is [9], where tests are conducted on text classification datasets, while [19] and [2] only report test on low-dimensional ones; in the case of [19], this is due to the fact that the underlying technology (decision trees) does not scale well to high dimensionalities.

We are currently carrying out experiments aimed to compare the approaches of [2] and [9] on the above high-dimensional datasets, testing a range of different experimental conditions (different class prevalence, different distribution drift, etc.) similarly to what done in [9]. We hope to have the results ready in time for them to be presented at the workshop.

5 Discussion

The ELM approach to quantification combines solid theoretical foundations with state-of-the-art performance, and promises to provide a superior alternative to the mostly empirical approaches that have been standard in the quantification literature. The key question that the (few) past works along this line leave open is: should one (a) optimize a combination of a quantification and a classification measure, or rather (b) optimize a pure quantification measure? In other words: how fundamental is classification accuracy to a quantifier? Approach (a) has indeed intuitive appeal, since we intuitively tend to trust a quantifier if it is also a good classifier; a quantifier that derives its good quantification accuracy from a high, albeit balanced, number of false positives and false negatives makes us a bit uneasy. On the other hand, approach (b) seems more in line with accepted machine learning wisdom (“optimize the measure you are going to evaluate the results with”), and one might argue that being serious about the fact that classification and quantification are fundamentally different means that, if a quantifier delivers consistently good quantification accuracy at the expense of classification accuracy, this latter fact should not be our concern. Further research is needed to answer these questions and to determine which among these contrasting intuitions is more correct.

Acknowledgements

We thank Thorsten Joachims for making SVM_{perf} available, and Jose Barranquero for providing us with the module that implements the Q-measure within SVM_{perf} .

References

1. Rocío Alaíz-Rodríguez, Alicia Guerrero-Curieses, and Jesús Cid-Sueiro. Class and subclass probability re-estimation to adapt a classifier in the presence of concept drift. *Neurocomputing*, 74(16):2614–2623, 2011.
2. Jose Barranquero, Jorge Díez, and Juan José del Coz. Quantification-oriented learning based on reliable classifiers. *Pattern Recognition*, 48(2):591–604, 2015.
3. Antonio Bella, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana. Quantification via probability estimators. In *Proceedings of the 11th IEEE International Conference on Data Mining (ICDM 2010)*, pages 737–742, Sydney, AU, 2010.
4. Yee Seng Chan and Hwee Tou Ng. Estimating class priors in domain adaptation for word sense disambiguation. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL 2006)*, pages 89–96, Sydney, AU, 2006.
5. Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. John Wiley & Sons, New York, US, 1991.
6. Imre Csiszár and Paul C. Shields. Information theory and statistics: A tutorial. *Foundations and Trends in Communications and Information Theory*, 1(4):417–528, 2004.

7. Andrea Esuli and Fabrizio Sebastiani. Machines that learn how to code open-ended survey data. *International Journal of Market Research*, 52(6):775–800, 2010.
8. Andrea Esuli and Fabrizio Sebastiani. Sentiment quantification. *IEEE Intelligent Systems*, 25(4):72–75, 2010.
9. Andrea Esuli and Fabrizio Sebastiani. Optimizing text quantifiers for multivariate loss functions. *ACM Transactions on Knowledge Discovery and Data*, 2014. Forthcoming.
10. George Forman. Counting positives accurately despite inaccurate classification. In *Proceedings of the 16th European Conference on Machine Learning (ECML 2005)*, pages 564–575, Porto, PT, 2005.
11. George Forman. Quantifying trends accurately despite classifier error and class imbalance. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*, pages 157–166, Philadelphia, US, 2006.
12. George Forman. Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2):164–206, 2008.
13. George Forman, Evan Kirshenbaum, and Jaap Suermondt. Pragmatic text mining: Minimizing human effort to quantify many issues in call logs. In *Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data Mining (KDD 2006)*, pages 852–861, Philadelphia, US, 2006.
14. John J. Gart and Alfred A. Buck. Comparison of a screening test and a reference test in epidemiologic studies: II. A probabilistic model for the comparison of diagnostic tests. *American Journal of Epidemiology*, 83(3):593–602, 1966.
15. Daniel J. Hopkins and Gary King. A method of automated nonparametric content analysis for social science. *American Journal of Political Science*, 54(1):229–247, 2010.
16. Thorsten Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, pages 377–384, Bonn, DE, 2005.
17. Thorsten Joachims, Thomas Hofmann, Yisong Yue, and Chun-Nam Yu. Predicting structured objects with support vector machines. *Communications of the ACM*, 52(11):97–104, 2009.
18. Gary King and Ying Lu. Verbal autopsy methods with multiple causes of death. *Statistical Science*, 23(1):78–91, 2008.
19. Letizia Milli, Anna Monreale, Giulio Rossetti, Fosca Giannotti, Dino Pedreschi, and Fabrizio Sebastiani. Quantification trees. In *Proceedings of the 13th IEEE International Conference on Data Mining (ICDM 2013)*, pages 528–536, Dallas, US, 2013.
20. Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14(1):21–41, 2002.
21. Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.
22. Cornelis J. van Rijsbergen. *Information Retrieval*. Butterworths, London, UK, second edition, 1979.
23. ChengXiang Zhai. Statistical language models for information retrieval: A critical review. *Foundations and Trends in Information Retrieval*, 2(3):137–213, 2008.

A scalable approach to near real-time sentiment analysis on social networks

G. Amati, S. Angelini, M. Bianchi, L. Costantini, G. Marcone

Fondazione Ugo Bordoni, Viale del Policlinico 147, 00161 Roma, Italy
{gba, sangelini, mbianchi, lcostantini, gmarcone}@fub.it

Abstract. This paper reports about results collected during the development of a scalable Information Retrieval system for near real-time analytics on social networks. More precisely, we present the end-user functionalities provided by the system, we introduce the main architectural components, and we report about performances of our multi-threaded implementation. Since sentiment analysis functionalities are based on techniques for estimating document category proportions, we report about a comparative experimentation aimed to analyse the effectiveness of such techniques.

1 Introduction

The development of platforms for near real-time analytics on social networks poses very challenging research problems to the Artificial Intelligence and Information Retrieval communities. In this context, sentiment analysis is a tricky task. In fact, sentiment analysis for social networks can be defined as a *search-and-classify* task, that is a pipeline of two processes: retrieval and classification [12], [14]. The accuracy of a search-and-classify task thus suffers of the multiplicative effects of independent errors produced by both the retrieval and the classification. The search-and-classify task however is just an example of a most general problem of near real-time analytics. Near real-time analytics is actually based on five main tasks: the retrieval of a preliminary set (the posting lists of the query terms), the assignment of a retrieval score to these documents, the application of binary filters (for example, by selecting documents by period of time and opinion polarity), the mining of hidden entities, and, finally, the final sort to display statistical outcomes and to decorate document pages of results.

All these functionalities must be finally thought and designed to handle *big-data*, as that of Twitter, that generates unbounded streams of data. Moreover, near real-time sentiment analysis for social networks includes end-user functionalities that are typical of either *data-warehouses* or *real-time big data analytics* platforms. For example, the topic of interest is often represented as a large query to be processed in batch mode, and several search tools must support the query specification phase. On the other hand, systems need to continuously index a huge flow of data generated by multiple data-sources, to make new data available as soon as possible, and to prompt reactive detection of incoming events of interest.

In this scenario we report the experience acquired in the development of a system specialized on near-realtime analytics for the Twitter platform.

In Section 2 we describe our system. More precisely, we present end-users functionalities allowing end-users to search, classify and estimate category proportions for real-time analytics. The implementation of these functionalities relies

on some architectural components defined downline of the analysis of a typical retrieval process performed by a search engine. As a consequence, we show how all functionalities can be implemented according to a single retrieval process and how to *scale-up* by a multithreaded parallelization, or *scale-out* by mean of distribution of processes on different computational nodes. We conclude the section reporting the results of an experimentation aimed to assess the performance of our multi-thread implementation. The assessment of the distributed version of the system is still in progress. Even if the system is not yet optimized, the experimentation validates the viability of our solution.

Among all implemented functionalities, in Section 3 we focus on the proportion estimation of categories for sentiment analysis, since quantification for sentiment analysis is particularly complex to be accomplished in near real-time analysis. It is indeed an example of a complex task that requires many steps of Information Retrieval and Machine Learning processing to be performed. Because of this, we describe several techniques for category proportion estimation and we provide their comparison. Section 4 concludes the paper.

2 A scalable system for near real-time sentiment analysis

In order to identify requirements for a system enabling near real-time analysis of phenomena occurring on social networks, we took into consideration two kinds of end-users: *social scientists* and *data scientists*.

Broadly speaking, a social scientist is a user interested in finding answers to questions such as: what are the most relevant/recent tweets, how many tweets convey a positive/negative opinion, what are concepts related to a given topic, how is the trend of a given topic, what are the most important topics, and so on. In general, social scientists interact with the system by submitting several queries formalizing their information needs, they empirically evaluate the quality of the answer provided by the system. The role of social scientist can be played by any user interested in studying or reporting phenomena of social networks that can be connected to scientific discipline such as sociology, psychology, economics, political science, and so on. On the contrast, a data scientist is interested in developing and improving functionalities for social scientists. More precisely, data scientists implement machine learning processes and they take under control the quality of answers provided by the system by means of statistical analyses. Furthermore, they take in charge of define and develop new functionalities for reporting, charting, summarizing, etc.

The following Section presents the end-user functionalities provided by the system. They are the result of a user-requirement analysis activity, jointly conducted by social scientists, data scientist and software engineers.

2.1 End-user functionalities for analytics and sentiment analysis

From the end-user perspective, a system for near real-time analytics and sentiment analysis should provide three main classes of functions: *search*, *count* and *mining* functionalities.

Given a query, *search functionalities* consist in a suite of operations useful to find: the most relevant tweets (*topic retrieval*); the most recent tweets in any interval

of time (*topical timeline*); a representative sample of tweets conveying opinions about the topic (*topical opinion retrieval*); a representative sample of tweets conveying *positive* or *negative* opinions about the topic (*polarity driven topical opinion retrieval*); any mixture of tweets resulting from the combination of relevance, time and opinion search dimensions. Search functionalities are used by social scientist in order to explore tweets indexed by the system, to detect emerging topics, to discover new keywords or accounts to be tracked on Twitter; on other hands, they are used by data scientists to empirically assess the effectiveness of the system.

Count functionalities quantify the result-set size of a given query. As a consequence, they are useful to quantify, for example, the number of positive tweets related to a given topic. The system offers two main methods for counting: the *exact count*, that is a database-like function returning the exact number of tweets matching the query, and the *estimated count*, that statistically estimates the number of tweets belonging to a given results-set. As described in Section 3.1 there are some different strategies to perform the estimation count: for sake of exposition we anticipate that the two main approaches are *classify-and-count* and *category size estimation*.

Finally, a suite of *mining functionalities* is available: trending topics, query-related concept mining, geographic distribution of tweets, most representative users for a topic, and so on.

Both count and mining functionalities are mainly used by social scientists for their studying and reporting aims.

In the next Section we show how the above mentioned functionalities can be implemented adopting an Information Retrieval approach.

2.2 A search engine based system architecture

Functionalities presented in the previous Section can be implemented by a system based on a search engine, specifically extended for this purpose. In fact, classic index structures have to be properly configured to host some additional information about tweets. Among the others, an opinion score, a positive opinion score and a negative opinion score, computed at indexing-time and stored in the index, enable the implementation of sentiment analysis functionalities. These scores can be computed by using a dictionary-based approach, as proposed in [1], or by means of an automatic-classifier, such as SVM or Bayesian classifiers. As described in Section 3, these scores can be used at querying-time for implementing functionalities as *exact* and *estimated counting*.

Furthermore, due to the scalability system requirement, index data structures have to support mechanisms for document or term partitioning [13]. In the first case, documents are partitioned into several sub-collections and are separately indexed; in the second case, all documents are indexed as a single collection, and then some data structures (i.e. the lexicon and the posting lists) are partitioned. Even if the term partitioning approach has some advantages in query processing (e.g. making the routing of queries easier and thus resulting in a lower utilization of resources [13]), it does not scale well: because of this we adopt a document partitioning approach.

Once the partitioning approach has been selected, it becomes crucial to define a proper document partition strategy. We opt for partitioning tweets just on the basis of their timestamps: this implies each index contains all tweets generated

during a certain period of time. In our case this strategy is more convenient than others [2],[4],[9],[11],[15], since it is suitable in presence of an unbounded stream of tweets delivered in chronological order; moreover, it enables the optimization of the query process when a time-based constraint is specified for the query.

Finally, we have to decide if to implement a solution to *scale-up* or to *scale-out* in terms of the number of indexed tweets. In the first case, a *multi-index* composed by several *shards* can be created, updated and used on a single machine: as a consequence, the time needed to resolve a query depends on the calculating capacity and the main memory availability on the machine. In the second case, each machine of a computer cluster has to be responsible for a sub-collection and to act as an *indexing* and *query server*: with respect to the time needed to resolve a query, this solution (referred as *distributed index* in the following) exploits the calculating capacity of the entire computer cluster, but introduces some latency due to network communications. Interestingly, in both of the cases, it is possible to define a common set of software components that allow to efficiently implement functionalities presented in Section 2.1. These components, here briefly described, can be implemented to develop an application based on either a multi-index, or a distributed index:

1. *Global Statistics Manager (GSM)*. As soon as new incoming tweets are indexed, the GSM has to update some global statistics, such as the total number of tweets and tokens. Both for multi-index and distributed index solution, the update operation can be simply performed either at query-time, or when the collection changes.
2. *Global Lexicon Manager (GLM)*. The *lexicon* data structure contains the list and statistics of all terms in the collection. Both multi-indexes and distributed indexes require a manager providing information about terms with respect of the entire collection. The GLM can rely on a serialized data structure to be updated every time the collection changes (i.e. a global lexicon), or it can compute at query-time just global information needed to resolve the submitted query.
3. *Score Assigner (SA)*. Any document containing at least one query-term is candidate to be added in the final result-set. SA assigns a ranking score to each document to quantify a relevance degree with respect to the query. Using information provided by GSM and GLM, the scores of document indexed in different shards, or by different query servers, are comparable because computed using global statistics. It is worth noting that opinion scores, needed to sentiment analysis functionalities, are computed once for all at indexing-time, and that they have just to be read in the indexes. In fact, we assume that the classifier model for sentiment analysis does not change over time: as a consequence, any change to global statistics of the collections does not affect already computed sentiment scores, and thus their sentiment classifications.
4. *Global Sorter (S)*. Top-N results are sorted in descending order of score.
5. *Post Processing Retriever (PPR)*: a second pass retrieval can follow the retrieval phase, such as query expansion, or a document score modifier can be applied, such as mixture of relevance, time and sentiment models.
6. *Post Processing Filter and Entity Miner (EM)*: some post-processing operations can be performed in order to filter the final result set by time, country etc. or by sentiment category membership constraints. If the *direct index*, i.e. the posting list of the terms occurring in each document, or other additional

Table 1. Mapping examples of user functionalities over information retrieval processes.

| Functionalities | GSM | GLM | SA | S | PPR | EM | D |
|------------------------------|-----|-----|----|---|-----|----|---|
| Query result set count | | | | | | | |
| Classify and count | | X | | | | X | |
| Category estimation | X | X | | | | X | |
| Ranking | X | X | X | X | | | X |
| Trending topics | X | X | | | X | X | |
| Query-related concept mining | X | X | X | X | X | X | |

data structures are available, text mining operations can be also applied to the result set, for example: extraction of relevant and trendy concepts, or mentions, or entities related to the query.

7. *Decorator (D)*: once the result set is determined and ordered, some efficient DB-like operations can be eventually performed in order to make results ready for presentation to the final user (e.g. posting records are decorated with metadata such title, timestamp, author, text, etc.).

Table 1 shows which components are involved in the implementation of some exemplifying end-user functionalities. To obtain an efficient implementation of these functionalities it is crucial to design and implement the listed components as more decoupled as possible. It is worth noting the *Query result set count* functionality does not depend on any listed component since it only needs local postings retrieval operations.

2.3 Assessing the performance of a multi-index implementation

We have developed a multi-index based implementation of the system adding new data structure to the Terrier framework¹. The current version takes advantage of the multi-threading paradigm to parallelize, as much as possible, reading operations from shards.

In order to assess the efficiency of our solution, we use a collection containing more than 153M of tweets, written in English, concerning the FIFA 2014 World Cup (up to half July 2014), and football news (up to half September 2014). Since June 14 to September 14, a new shard has been daily created and added to the multi-index, independently from the number of tweets downloaded in the last 24 hours. The final index contains 76 shards unbalanced in terms of number of contained tweets, as shown in Figure 1 (each shard contains an average of about 2M tweets). We have focused our assessment on the ranking functionality: more precisely, we have used 2127 queries, retrieving an average of about 44,361 tweets each. Table 2 reports the processing time for each component involved in the functionality under testing. In general, observed performances fit our expectation: anyway, we identify a potential bottleneck in the decoration phase. The *decorator* component will have to be carefully developed in the new version of the system based on a distributed index.

¹<http://www.terrier.org>

Table 2. Processing time the processing time for each components involved in the functionality. GSM is not reported since it has a negligible processing time. Times are milliseconds averaged on queries. We have run the system on a machine having a quad-core i3 CPU clocked at 3.07 GHz with 8GB RAM. Being the system written in the Java programming language, we have allocated about 4GB to the Java Virtual Machine.

| # shards | # docs | GLM | SA | S | D |
|----------|-------------|---------|---------|----------|-------------|
| 25 | 56,304,653 | 3.76 ms | 0.07 ms | 7.23 ms | 0.05 ms/doc |
| 50 | 99,912,639 | 6.86 ms | 0.13 ms | 9.84 ms | 0.06 ms/doc |
| 76 | 153,137,302 | 8.20 ms | 0.16 ms | 10.87 ms | 0.07 ms/doc |

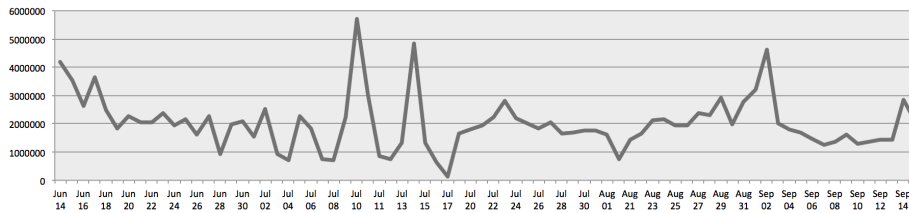


Fig. 1. Number of documents in daily shards used for assessing the performances of the multi-index implementation.

3 Comparing techniques for category proportion estimation

On Twitter, time and sentiment polarity can be important as relevance is for ranking documents. Since sentiment polarity is a classification task, the IR system needs to perform both classification and search tasks in one single shot. In order to obtain a near-real time classification for large data streams, we need to make some computational approximations and to recover the approximation error by introducing a supplementary model able to correct the results, for example by re-sizing the proportions by estimates of such classification errors [5, 6]. Finally, we correct the number of misclassified items by a linear regression model previously learned on a set of training queries, as presented in [1], or using an adjusted classify & count approach (Section 3.1). At the query time we just combine scores either to aggregate for estimates of retrieval category sizes or to select and sort documents by time, relevance and sentiment polarities.

In this Section we report results of a experimental comparison we conducted on different techniques for category proportion estimation.

3.1 Category proportion estimation

Let $D = \{D_1, \dots, D_n\}$ be a set of mutually exclusive sentiment categories over the set of tweets Ω , and let q be a topic (story). The problem of size or proportion estimation of sentiment categories for a story consists in specifying the distribution of the categories $P(D_i|q)$ over the result set of the story q .

Such an estimation is similar to that conducted within a typical statistical problem of social sciences, macroeconomics or epidemiological studies. In general if an unbiased sample of the population can be selected, then it can be used to estimate the population categories together with a statistical error due to the size of the sample. For example, Levy & Kass [10] use the *Theorem of Total Probability* on the observed event A to decompose this event over a set of predefined categories. In Information Retrieval, the observed event A can be, for example, the set of the posting lists of a story. We also assume that $P(A|D_i)$ is obtained by a sample A' of A , that is by $P(A'|D_i)$. The problem of estimating the category proportions $P(D_i)$ is determining these probabilities on a sample of observations $A' \subset A$:

$$P(A') = \sum_{i=1}^n P(A'|D_i)P(D_i).$$

If we monitor the event A' as aggregated outcome of all observable items in the sample, then we may easily rewrite the Theorem of Total Probability in matrix form as a set of linear equations:

$$P(A') = \underset{1 \times |D|}{P(A'|D)} \cdot \underset{|D| \times 1}{P(D)}.$$

We simply derive the category proportions $P(D_i)$ by resolving a system of $|D|$ linear equations into $|D|$ variables. From now on we denote all probabilities by $P(\cdot|q)$ to recall the dependence of observables to the result set of the current query q .

When the assignment of documents of A , or more generally of observables for A , to categories is not performed manually, but automatically, then it is not only the size of the selected sample A that matters, but also both type I and II errors (false positives and false negatives) produced by misclassification that becomes equally significant. In other words, the accuracy of the classifier need also to be known for a correct estimation of all $P(D_i|q)$. If the two types of errors comes out to be similar in size, then the final counting outcomes for category proportions may produce a correct answer. More generally, if the observations is given by a set X of observable variables for the document sample A , then the observables, and their proportions $P(X|D)$, may be used as a set of training data for a linear classifier to derive $P(D|q)$:

$$\underset{|X| \times 1}{P(X|q)} = \underset{|X| \times |D|}{P(X|D, q)} \cdot \underset{|D| \times 1}{P(D|q)}.$$

These equations can be thus resolved, for example, by linear regression. The set of observable variables X can be defined according several approaches.

- *The classify and count methodology*: X is the set of predicted categories \hat{D}_j of a classifier \hat{D} . Misclassification errors are given by the conditional probabilities $P(\hat{D}_k|D_j)$ when $k \neq j$. Counting the errors of the classifier in the training data set, and using these measures to correct the category proportions, is at the basis of the adjusted classify and count approach [10, 5, 6].
- *The profile sampling approach*: X is a random subset of word profiles S_j , where a profile is a subset of words occurring in the collection. This approach is at the basis of Hopkins & King's method [7].
- *The cumulative approach*: X is a set of weighted features f_j of a trained classifier (a weighted sentiment dictionary) [1]. The classifier model then can

be used to score each document in the collection. Differently from Hopkins & King’s method, that counts occurrences of an unbiased covering set of profiles for a topic, the classifier approach correlates a cumulative category score with category proportions for a topic.

Adjusted-Classify and Count The observations A are obtained by a classifier \hat{D} for the categories D

$$P(\hat{D}_j|q) = \sum_{i=1}^n P(\hat{D}_j|D_i, q)P(D_i|q) \quad j=1, \dots, n.$$

We pool the queries results, that is $P(\hat{D}'_j|D'_i, q) = P(\hat{D}'_j|D'_i)$ on a training data set D' and a set of queries. The estimates derive from this pooling set, (i.e. $P(D_i) = P(D'_i)$) solving a simple linear system of $|D|$ equations with $|D|$ variables:

$$\begin{matrix} P(A|q) & = & P(A'|D) \cdot P(D|q). \\ |D| \times 1 & & |D| \times |D| \quad |D| \times 1 \end{matrix}$$

The methodology is automatic and supervised, and therefore does not need to start over at each query. The accuracy of the classifier does not matter, since the misclassification errors are used for the estimation of category sizes. On the other hand, being not based on a query-by-query learning model, it does not achieve as high precision as with the manual evaluation of Hopkins & King’s method.

Hopkins & King’s method Let S' be a sample of profiles of words of the vocabulary \mathbf{V} , that is $S' \subset S = 2^{\mathbf{V}}$, able to cover *well enough* the space of events, and let A be the set of relevant documents for a topic q . Let us assess the sentiment polarities of a sample A' of A . About 500 evaluated documents will suffice for a statistically significant test. The partition of A' over the categories D will yield the statistics for the occurrences of S' in each category, and these proportions are used to estimate $P(A|D, q)$. $P(A)$ instead will be estimated by $P(S')$, that is the total number of occurrences of the word profiles of S' in the sample A' with respect to all word profiles occurring in A' .

The category proportions $P(D|q)$ are estimated as the coefficients of the linear regression model

$$\begin{matrix} P(A|q) & = & P(A|D, q) \cdot P(D|q). \\ |A| \times 1 & & |A| \times |D| \quad |D| \times 1 \end{matrix}$$

This is not a supervised methodology, as it would be with an automated classifier. It is based on counting word profiles from a covering sample. The advantage is a statistically significantly high accuracy (almost 99%, see Table 3). However, there are many drawbacks. The methodology needs to start over at each query, and to achieve such a high accuracy, a long and costly activity of human evaluation of documents is required. The word profile counting is anyway complex since profiles are arbitrary subsets of a very large dictionary, and data are very sparse in Information Retrieval. Moreover, the query-by-query linear regression learning model is also time consuming. In conclusion, this method is not based on a supervised learning model, but it is essentially driven by a manual process, and linear regression and word profiles counting are just used to smooth the maximum likelihood category estimators.

Cumulative approach The cumulative approach is a supervised learning technique that consists in the use of a linear regression to predict and smooth a sentiment category size on the basis of a cumulative score of documents [1]. The approach is information theoretic: for each category, the set F of features for a category are made up of the most informative terms, or equivalently, the highest coding code in that category. Differently from Levy & Kass-Forman’s misclassification recovery model, there is not a pipeline of computational processes to perform, namely classifying, then counting, and finally adjusting the category sizes with the number of estimated misclassified items. The technique of the cumulative approach simply correlate the category size with the total number of bits used to code the occurring category features. Since information is additive, the linear regression model is the natural choice that sets up such a correlation over a set of features spanning over a set of training queries. Similarly to the adjusted classify and count approach the precision of this methodology is high and is reported on Section 3.2.

3.2 Experimentation

To assess the effectiveness of the classifier-based quantification, we have build an annotated corpus composed by 6305 tweets manually classified on the basis of the contained opinion. More precisely: 1358 tweets was classified as *positive* (i.e. containing a positive opinion), 2293 as *negative* (i.e. containing a negative opinion), 382 as *mixed* (i.e. containing both positive and negative opinions); 1959 as *neutral* (i.e. not containing opinions), 313 as *not classifiable*.

We have run two sets of experiments. We have first statistical technique to smooth the proportions from a manual document sample assessment. This experiment is essentially manual because requires a training set for each query. For each query instead of the word profiles as used in the proposed by Hopkins & King we have used two standard classifiers (Multinomial Naive Bayes, MNB, and SVM with a linear kernel), and the adjusted classify & count (ACC) as maximum likelihood estimate smoothing technique. However, Hopkins & King’s results are hardly reproducible since the set of admissible profiles are generated by a complex feature selection, and also a portion of negative examples are removed from the training set of the query. Indeed, these profiles are generated by an adaptation of the technique by King and Lu [8], that randomly chooses subsets of between approximately 5 and 25 words as admissible profiles. This number of words is determined empirically through cross-validation within the labeled set. Therefore, we show our results in comparison to their method on Table 3 as only reported in their paper [7].

Table 4 shows that the supervised methods with the adjusted classify & count (ACC) technique achieves a very high precision (96.63%-97.86%), i.e. a Mean Absolute Proportion error similar to that of Hopkins & King, with a supervised learning process that is not tailored on a single query only, but trained over a set of about 30 queries and with a 5-fold cross validation. The difference of Mean Absolute Proportion error for 30 queries produced by a search like classification process with respect to Hopkins & King method with a single query, is minimal and not statistically significant.

This first outcomes on Table 4 show that standard supervised classification methods can be effectively applied, and fast implemented, for quantification of sentiment analysis of new queries. The second experiment on Table 5 indeed shows

Table 3. Performance of Hopkins & King Approach (HKA) [7] and Support Vector Machine with linear and polynomial kernels.

| Percent of Blog Posts Correctly Classified | | | | |
|--|---------------|------------------------------|----------------------------------|--------------------------------|
| | In-Sample Fit | In-Sample 2-Cross-Validation | Out-of-Sample 2-Cross-Validation | Mean Absolute Proportion Error |
| HKA | - | - | - | 1.2 |
| Linear | 67.6 | 55.2 | 49.3 | 7.7 |
| Polynomial | 99.7 | 48.9 | 47.8 | 5.3 |

Table 4. Performance of Classify & Count and Adjusted Classify & Count (ACC) for MNB and SVM Classifiers. Each query has its training data set.

| Percent of Tweets Correctly Classified | | | |
|--|-----------|-------------------------------------|--------------------------------|
| | # queries | Out-of-Data-Sample Cross-Validation | Mean Absolute Proportion Error |
| SVM | 30 | 78.82 | 2.01 |
| MNB | 30 | 81.12 | 4.25 |
| ACC-SVM | 30 | 78.82 | 3.37 |
| ACC-MNB | 30 | 81.12 | 2.14 |
| HKA | 1 | - | 1.2 |

that the ACC smoothing with the use of classifiers is a fully automated supervised method that performs highly with new queries as the manual classification of HKA on a single query. The classifiers were trained using a set of about 30 queries and 6-fold cross validation, where each test set has new documents coming from the result sets of the new queries (Out-Query-Sample Cross Validation). We also report the sample fit for each fold (In-Query-Sample Fit Cross-Validation) that shows that an almost perfect category counting with the SVM classifier.

Notice that, the Classify & Count process (CC) is much less prone to error than the individual classification accuracy, because of possible error type balancing effect (see Table 5). However, there is not a correlation between individual classification accuracy and Mean Absolute Error Rate of the CC process, so that the CC approach cannot ever be considered reliable estimation or statistically significant. Finally, the cumulative approach achieves high effectiveness (Multiple R-squared is 0.9781 for the negative category with 5-fold cross validation on the same set of queries) [1].

4 Conclusion

This paper reported some experiences gained during the development of a scalable system for real-time analytics on social networks.

We have presented how some architectural components resulting from the analysis of a typical querying process that can be used to implement several func-

Table 5. Performance of Classify & Count and Adjusted Classify & Count (ACC) for MNB and SVM Classifiers. Data set is made up of 30 queries, divided in test set and training set of queries.

| Percent of Tweets Correctly Classified | | | | |
|--|--|-----------------------------------|---|--------------------------------------|
| | In-Queries-Sample Fit Cross-Validation | Mean Absolute Proportion Error | Out-of-Queries- Sample Cross-Validation | Mean Absolute Proportion Error |
| SVM | 99.85 | 0.05 | 74.76 | 5.57 |
| MNB | 94.26 | 3.00 | 78.46 | 3.95 |
| ACC-SVM | 99.85 | 0.03 | 74.76 | 6.23 |
| ACC-MNB | 94.26 | 1.99 | 78.46 | 8.91 |

tionalities of the system. These components can be adopted both for developing a multi-index and a distributed index implementation of the system. We also identified a potential bottleneck in the decoration phase: the related component has to be carefully developed in the distributed version of the system.

Furthermore, we have shown how to estimate real-time document category proportions for topical opinion retrieval for big data. Outcomes are produced either by a direct count or by estimation of category sizes based on a supervised automated classification with a smoothing technique to recover the number of misclassified documents. The use of MNB and SVM classifiers or information-based dictionaries to estimate category proportions are highly effective and achieves almost perfect accuracy if a training phase on the query is also performed.

Search, classify and quantification for analytics can be thus effectively conducted in real-time.

Acknowledgement: Work carried out under the Research Agreement between Almwave and Fondazione Ugo Bordoni.

References

- [1] Giambattista Amati, Marco Bianchi, and Giuseppe Marcone. Sentiment estimation on twitter. In *IIR*, pages 39–50, 2014.
- [2] C. S. Badue, R. Baeza-Yates, B. Ribeiro-Neto, A. Ziviani, and N. Ziviani. Analyzing imbalance among homogeneous index servers in a web search system. *Inf. Process. Manage.*, 43(3):592–608, 2007.
- [3] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern Information Retrieval*, volume 463. ACM press New York, 1999.
- [4] Jamie Callan. Distributed Information Retrieval. In *In: Advances in Information Retrieval*, pages 127–150. Kluwer Academic Publishers, 2000.
- [5] George Forman. Counting positives accurately despite inaccurate classification. In João Gama, Rui Camacho, Pavel Brazdil, Alípio Jorge, and Luís Torgo, editors, *ECML*, volume 3720 of *Lecture Notes in Computer Science*, pages 564–575. Springer, 2005.
- [6] George Forman. Quantifying counts and costs via classification. *Data Min. Knowl. Discov.*, 17(2):164–206, 2008.

- [7] Daniel Hopkins and Gary King. A method of automated nonparametric content analysis for social science. *American Journal of Political Science*, 54(1):229–247, 01/2010 2010.
- [8] Gary King, Ying Lu, and Kenji Shibuya. Designing verbal autopsy studies. *Population Health Metrics*, 8(1), 2010.
- [9] Leah S. Larkey, Margaret E. Connell, and Jamie Callan. Collection Selection and Results Merging with Topically Organized U.S. Patents and TREC Data. In *CIKM 2000*, pages 282–289. ACM Press, 2000.
- [10] P S Levy and E H Kass. A three-population model for sequential screening for bacteriuria. *American J. of Epidemiology*, 91(2):148–54, 1970.
- [11] Xiaoyong Liu and Bruce W. Croft. Cluster-based retrieval using language models. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval*, pages 186–193, New York, NY, USA, 2004. ACM Press.
- [12] Craig Macdonald, Iadh Ounis, and Ian Soboroff. Overview of the TREC 2007 blog track. In Ellen M. Voorhees and Lori P. Buckland, editors, *TREC*, volume Special Publication 500-274. National Institute of Standards and Technology (NIST), 2007.
- [13] Alistair Moffat, William Webber, Justin Zobel, and Ricardo B. Yates. A pipelined architecture for distributed text query evaluation. *Inf. Retr.*, 10(3):205–231, June 2007.
- [14] Iadh Ounis, Maarten de Rijke, Craig Macdonald, Gilad Mishne, and Ian Soboroff. Overview of the trec-2006 blog track. In *Text Retrieval Conference, 2006*.
- [15] Diego Puppini, Fabrizio Silvestri, and Domenico Laforenza. Query-driven document partitioning and collection selection. In *InfoScale '06: Proceedings of the 1st international conference on Scalable information systems*. ACM Press, 2006.

Telemonitoring and Home Support in BackHome

Felip Miralles, Eloisa Vargiu, Stefan Dauwalder, Marc Solà,
Juan Manuel Fernández, Eloi Casals and José Alejandro Cordero

Barcelona Digital Technology Center,
{fmiralles, evargiu, sdauwalder, msola,
jmfernandez, ecasals, jacordero}@bdigital.org

Abstract. People going back to home after a discharge needs to come back to their normal life. Unfortunately, it becomes very difficult for people with severe disabilities, such as a traumatic brain injury. Thus, this kind of users needs, from the one hand, a telemonitoring system that allows therapists and caregivers to be aware about their status and, from the other hand, home support to be helped in performing their daily activities. In this paper, we present the telemonitoring and home support system developed within the BackHome project. The system relies on sensors to gather all the information coming from user's home. This information is used to keep informed the therapist through a suitable web application, namely Therapist Station, and to automatically assess quality of life as well as to provide context-awareness. Preliminary results in recognizing activities and in assessing quality of life are presented.

1 Introduction

Telemonitoring makes possible to remotely assess health status and Quality of Life (QoL) of individuals. In particular, telemonitoring users' activities allows therapists and caregivers to become aware of user context by acquiring heterogeneous data coming from sensors and other sources. Moreover, Telemonitoring and Home Support Systems (TMHSSs) provide elaborated and smart knowledge to clinicians, therapists, carers, families, and the patients themselves by inferring user behavior. Thus, there are a number of advantages in telemonitoring and home support for both the person living with a disability and the health care provider. In fact, TMHSSs enable the health care provider to get feedback on monitored people and their health status parameters. Hence, a measure of QoL and the level of disability and dependence is provided. TMHSSs provide a wide range of services which enable patients to transition more smoothly into the home environment and be maintained for longer at home [5]. TMHSSs, as an integrated care technology, facilitate services which are convenient for patients, avoiding travel whilst supporting participation in basic healthcare, TMHSS can be a cost effective intervention which promotes personal empowerment [14].

In this paper, we present a sensor-based TMHSS, currently under development in the EU project BackHome¹. The proposed system is aimed at supporting end users which employ Brain Computer Interface (BCI) as an Assistive Technology (AT) and relies on intelligent techniques to provide both physical and social support in order to improve QoL of people with disabilities. In particular, we are

¹<http://www.backhome-fp7.eu/backhome/index.php>

interested in monitoring mobility activities; the main goal being to automatically assess QoL of people. The implemented system is aimed at automatically assessing QoL as well as providing context-awareness. Moreover, the system gives a support to therapist through a suitable Therapist Station. In this way, therapists are constantly aware about the progress of users, their status and the activities they have been performing. Although we are interested in assisting disabling people, by now we only performed preliminary experiments with a healthy user. We are now in the process to install the system in disabled people's homes under the umbrella of the BackHome project².

The rest of the paper is organized as follows: Section 2 briefly recall relevant work in the field of telemonitoring and home support. In Section 3 the BackHome project and its main goals are summarized. Section 4 presents the implemented sensors-based approach whereas Section 5 illustrates the Therapist Station. In Section 6 preliminary experiments aimed at monitoring daily activities and assessing QoL are presented. Finally, Section 7 ends the papers with conclusions and future work.

2 Telemonitoring and Home Support

Telemonitoring systems have been successful adopted in cardiovascular, hematologic, respiratory, neurologic, metabolic, and urologic domains [14]. In fact, some of the more common features that telemonitoring devices keep track of include blood pressure, heart rate, weight, blood glucose, and hemoglobin. Telemonitoring is capable of providing information about any vital signs, as long as the patient has the necessary monitoring equipment at her/his location. In principle, a patient could have several monitoring devices at home. Clinical-care patients' physiologic data can be accessed remotely through the Internet and handled computers [18]. Depending on the severity of the patient's condition, the health care provider may check these statistics on a daily or weekly basis to determine the best course of treatment. In addition to objective technological monitoring, most telemonitoring systems include subjective questioning regarding the patient's health and comfort [13]. This questioning can take place automatically over the phone, or telemonitoring software can help keep the patient in touch with the health care provider. The health care provider can then make decisions about the patient's treatment based on a combination of subjective and objective information similar to what would be revealed during an on-site appointment.

Home sensor technology may create a new opportunity to reduce costs. In fact, it may help people stay healthy and in their homes longer. An interest has therefore emerged in using home sensors for health promotion [11]. One way to do this is by TMHSSs, which are aimed at remotely monitoring patients who are not located in the same place of the health care provider. Those supports allow patients to be maintained in their home [5]. Better follow-up of patients is a convenient way for patients to avoid travel and to perform some of the more basic work of healthcare for themselves, thus reducing the corresponding overall costs [1] [23]. Summarizing, a TMHSS allows: to improve the quality of clinical services, by facilitating the access to them, helping to break geographical barriers; to keep the objective in the assistance centred in the patient, facilitating the communication

²<http://www.backhome-fp7.eu/>

between different clinical levels; to extend the therapeutic processes beyond the hospital, like patient's home; and a saving for unnecessary costs and a better costs/benefits ratio.

In the literature, several TMHSSs have been proposed. Among others, let us recall here the works proposed in [2], [4], and [16]. The system proposed in [2] provides users personalized health care services through ambient intelligence. That system is responsible of collecting relevant information about the environment. An enhancement of the monitoring capabilities is achieved by adding portable measurement devices worn by the user thus vital data is also collected out of the house. Similarly, the TMHSS presented in this paper uses ambient intelligence to personalize the system according to the specific context [3]. Corchado et al. [4] propose a TMHSS aimed at improving healthcare and assistance to dependent people at their homes. That system is based on a SOA model for integrating heterogeneous wearable sensor networks into ambient intelligence systems. The adopted model provides a flexible distribution of resources and facilitates the inclusion of new functionalities in highly dynamic environments. Sensor networks provide an infrastructure capable of supporting the distributed communication needed in the dependency scenario, increasing mobility, flexibility, and efficiency, since resources can be accessed regardless their physical location. Biomedical sensors allow the system to acquire continuously data about the vital signs of the patient. Apart from the BCI system, the TMHSS presented in this paper, does not rely on biomedical sensors. All physiological information is, in fact, provided by the BCI system (i.e., EEG, ECG and EOG signals). Mitchell et al. [16] propose ContextProvider, a framework that offers a unified, query-able interface to contextual data on the device. In particular, it offers interactive user feedback, self-adaptive sensor polling, and minimal reliance on third-party infrastructure.

As for BCI users, some work has been presented to provide smart home control [10] [19] [7] [8]. To our best knowledge, telemonitoring has not been integrated yet with BCI systems apart as a way to allow remote communication between therapists and users [17].

3 BackHome at a Glance

BackHome focuses on restoring independence to people that are affected by motor impairment due to acquired brain injury or disease, with the overall aim of preventing exclusion [6]. In fact, BackHome aims to provide brain-controlled assistive technology, which can be used in the context of social reintegration, rehabilitation and maintenance of remaining capabilities of people with disabilities. Thus, BackHome aims to implement easy-to-set-up-and-use software which requires minimal equipment based on a new generation of practical electrodes. On one hand, the produced software is aimed at making BCI usable for disabled people, with a potentially flexible and extensible inclusion schedule. On the other hand, thanks to the telemonitoring and home support features, the objective system should benefit of detection of user's activity and behaviour to adapt interfaces and trigger support actions. In order to keep the user engaged, BackHome continuously provides feedback to therapist for the follow-up and for personalization and adaptation of rehabilitation plans, for instance.

The BackHome system relies on two stations: (i) the therapist station and (ii) the user station. The former is focused on offering information and services to the

therapists via a usable and intuitive user interface. It is a Web application that allows the therapist to access the information of the user independently of the platform and the device. This flexibility is important in order to get the maximum potential out of the telemonitoring because the therapist can be informed at any moment with any device that is connected to the Internet (PC, a smart phone or a tablet). The latter is the main component that the user interacts with. It contains the modules responsible for the user interface, the intelligence of the system, as well as to provide all the services and functionalities of BackHome [12]. The user station is completely integrated into the home of the user together with the assistive technology to enable execution and control of these functionalities.

4 The Sensor-based Approach

To monitor users at home, we develop a sensor-based TMHSS able to monitor the evolution of the user's daily life activity [22]. The implemented TMHSS is able to monitor indoor activities by relying on a set of home automation sensors and outdoor activities by relying on Moves³.

As for indoor activities, we use presence sensors, to identify the room where the user is located (one sensor for each monitored room) as well as temperature, luminosity, humidity of the corresponding room; a door sensor, to detect when the user enters or exits the premises; electrical power meters and switches, to control leisure activities (e.g., television and pc); and pressure sensors (i.e., bed and seat sensors) to measure the time spent in bed (wheelchair). Figure 1 shows an example of a home with the proposed sensor-based system.

From a technological point of view, we use wireless z-wave sensors that send the retrieved data to a central unit located at user's home. That central unit collects all the retrieved data and sends them to the cloud where they will be processed, mined, and analyzed. Besides real sensors, the system also comprises "virtual devices". Virtual devices are software elements that mash together information from two or more sensors in order to make some inference and provide new information. For instance, sleep hours may be inferred by a virtual device that meshes the information from the bed sensors together with that from the presence sensor located in the bedroom. Let us consider the case in which the user is in bed reading. In that case, the luminosity level measured by the presence sensor assesses that the user is not sleeping, yet, even if the bed sensor is activated. In so doing, the TMHSS is able to perform more actions and to be more adaptable to the context and the user's habits. Furthermore, the mesh of information coming from different sensors can provide useful information to the therapist (e.g., the number of sleeping- or inactivity-hours). In other words, the aim of a virtual device is to provide useful information to track the activities and habits of the user, to send them back to the therapist through the therapist station, and to adapt the user station, with particular reference to its user interface, accordingly.

As for outdoor activities, we are currently using the user's smartphone as a sensor by relying on Moves, an app for smartphones able to recognize physical activities (such as walking, running, and cycling) and movements by transportation. Moves is also able to store information about the location in which the user is, as well as the corresponding performed route(s). Moves provides an API through which is possible to access all the collected data.

³<http://www.moves-app.com/>

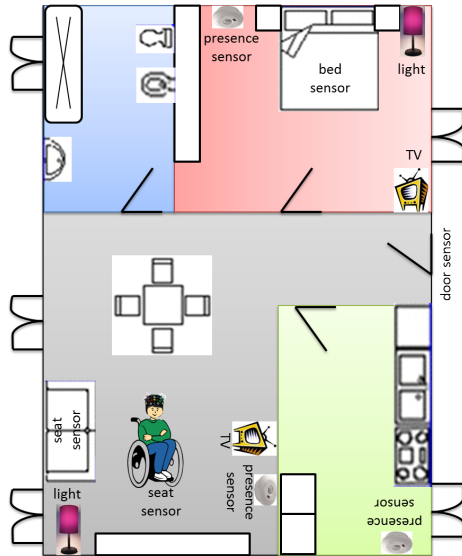


Fig. 1. An example of a home with the sensor-based system installed.

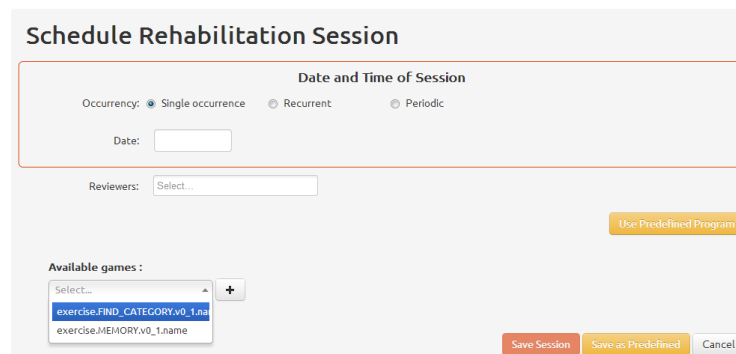
Information gathered by the TMHSS is also used to provide context-awareness by relying on ambient intelligence [3]. In fact, ambient intelligence is essential since people with severe disabilities could benefit very much from the inclusion of pervasive and context-aware technologies. In particular, thanks to the adopted sensors we provide adaptation, personalization, alarm triggering, and control over environment through a rule-based approach that relies on a suitable language [9].

Finally, monitoring users' activities through the TMHSS gives us also the possibility to automatically assess QoL of people [21]. In fact, information gathered by the sensors is used as classification features to build a multi-class supervised classifier; one for each user and for each item of the questionnaire we are interested answer to. In particular, the following features are considered: (i) time spent on bed and (ii) maximum number of continuous hours in bed, extracted from the bed sensor; (iii) time spent on the wheelchair and (iv) maximum number of continuous hours on the wheelchair, extracted from the seat sensor; (v) time spent in each room and (vi) percentage of time in each room, extracted from the presence sensor; (vii) room in which the user spent most of the time, inferred by the virtual device; (viii) total time spent at home, extracted from the door sensor; (ix) total time spent watching the TV and (x) total time spent using the PC, extracted from the corresponding power meters and switches; (xi) number of kilometres covered by transportation, (xii) number of kilometres covered by moving outdoors on the wheelchair and (xiii) number of visited places, provided by Moves. Let us note that more features can be considered depending on the adopted sensors.

5 The Therapist Station

The therapist station is a web application that provides functionality for clinicians/therapists regarding user management, cognitive rehabilitation task management, quality-of-life assessment, as well as communication between therapist and user.

Therapists are able to interact with users remotely in real time or asynchronously and monitor the use and outcomes of the cognitive rehabilitation tasks, quality-of-life assessment as well as performed activities and BCI usage. In fact, the ability for the therapist to plan, schedule, telemonitor and personalize the prescription of cognitive rehabilitation tasks and quality-of-life questionnaires using the therapist station facilitates that the user performs those tasks inside his therapeutic range (i.e. motivating and supporting her progress), in order to help to attain beneficial therapeutic results.



The screenshot shows a web application interface titled "Schedule Rehabilitation Session". It features a "Date and Time of Session" section with three radio buttons for "Occurrence": "Single occurrence" (selected), "Recurrent", and "Periodic". Below this is a "Date:" input field. A "Reviewers:" dropdown menu is set to "Select...". A yellow button labeled "Use Predefined Program" is positioned to the right. Underneath, an "Available games:" section includes a dropdown menu with a plus sign, showing a list of options: "Select...", "exercise.FIND_CATEGORY.v0_1.na", and "exercise.MEMORY.v0_1.name". At the bottom right, there are three buttons: "Save Session" (red), "Save as Predefined" (yellow), and "Cancel" (grey).

Fig. 2. Scheduling cognitive rehabilitation tasks.

As for the cognitive rehabilitation sessions, using the therapist station, healthcare professionals can remotely manage a caseload of people recently discharged from acute sector care. They can prescribe and review rehabilitation sessions (see Figure 2) [20]. Through the therapist station, rehabilitation sessions can be configured, setting the type of tasks that the user will execute, their order in the session and the difficulty level and specific parameters for each one of them. Additionally, the therapist station allows healthcare professionals to establish an occurrence pattern for the session along the time. If the same session must be executed several times, professionals can set the type of occurrence and its pattern to make the session occur at programmed times in the future. Once the session is scheduled, users will see their BCI matrix updated on the user station the day the session is scheduled. Through that icon, the user will start the session. The user can then execute all the tasks contained in it in consecutive order. Upon completion of the session execution on user station, results are sent back to the therapist station for review. At this point, those healthcare professionals involved in the session -the prescriber and the specified reviewers- will be notified with an alert in the therapist station dashboard indicating that the user has completed the session. Healthcare professionals with the right credentials can browse user session

results once they are received. The Therapist Station provides a session results view and an overview of completed sessions to map progress, which shows session parameters and statistics along with the specific results (see Figure 3).

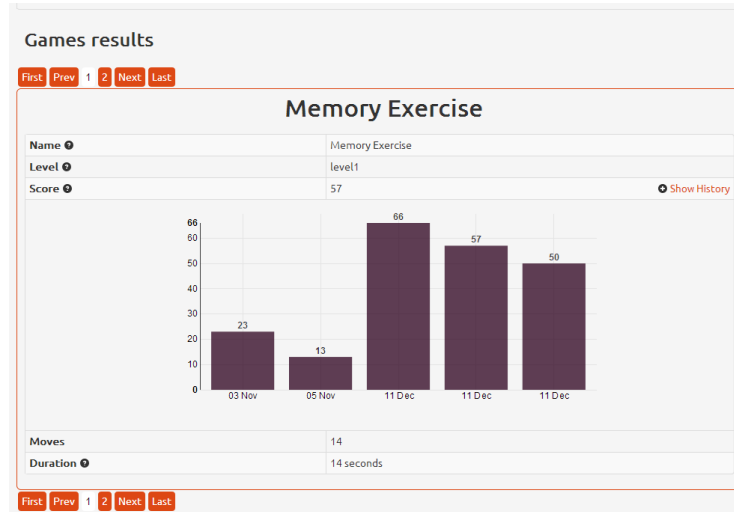


Fig. 3. Task results of the memory-cards task.

As for the quality-of-life assessment, as described in the previous session, one of the goals of the TMHSS is to automatically assess QoL of the users. Accordingly, results and statistics are sent to the therapist station in order to inform the therapist about improvement/worsening of user's QoL. Moreover, the therapist may directly ask the user to fill a questionnaire (Figure 4). Seemly than cognitive rehabilitation sessions, the therapist can decide the occurrence of quality-of-life questionnaire filling and, once scheduled, the user receives an update in the BCI matrix. Once the user, with the help of the caregiver, has filled the questionnaire, results are sent to the therapist that may revise them.

Finally, through the Therapist Station, therapists may consult a summary of activities performed at home by the user; e.g., visited rooms, sleeping hours and time elapsed at home. Moreover, also the BCI usage is monitored and high-level statistics provided. This information includes BCI session duration, setup time and training time as well as the number of selections, the average elapsed time per selection and a breakdown of the status of the session selections. Therapists have also the ability to browse the full list of selections executed by a user, such as context information as application running, selected value, grid size and selected position.

6 Experiments and Results

The system is currently running in a healthy user's home in Barcelona. The corresponding user is a 40-year-old woman who lives alone. This installation is cur-

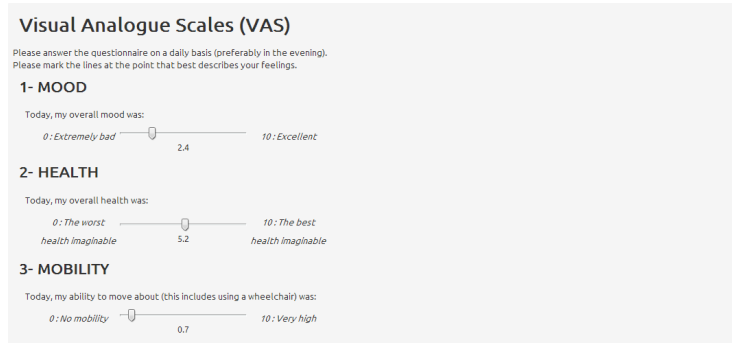


Fig. 4. The first three questions of the adopted quality-of-life questionnaire.

rently available and data continuously collected. According to the home plan, the following sensors have been installed: 1 door sensor; 3 presence sensors (1 living room, 1 bedroom, 1 kitchen); 3 switch and power meters (1 PC, 1 Nintendo WII, 1 kettle); and 1 bed sensor. Moreover, the user has installed in her iPhone the Moves app.

A useful interface allows technicians to remotely view, manage and/or change the configuration of the system and to have a view of the collected data, when needed (see Figure 5).

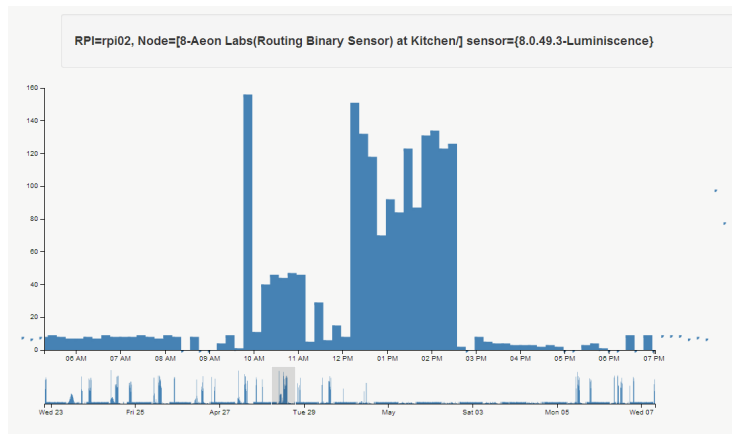


Fig. 5. Status of luminescence of a given sensor.

Collected data have been used to recognize habits as well as to a preliminary study aimed at assessing QoL.

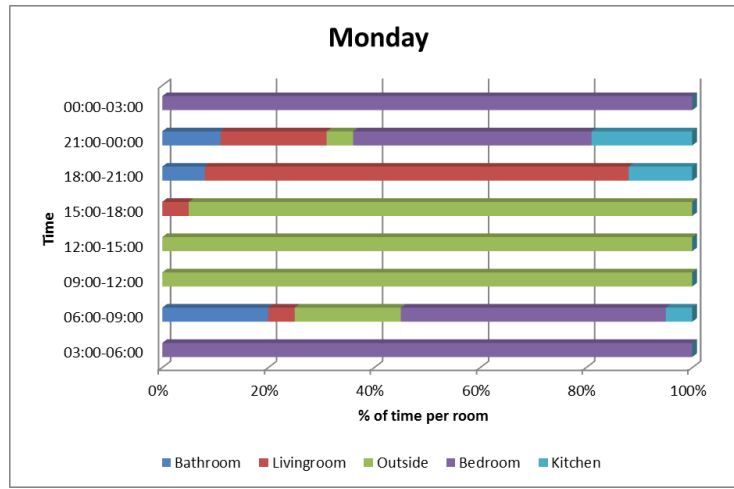


Fig. 6. User's habits: full-time workday.

6.1 Activity Recognition

To recognize user's habits, we performed a preliminary experiment considering indoor habits and relying on presence sensors (one for each monitored room) and the main door sensor (to know when the user enters or leaves the premises). We collected data from one month (November '13 – December '13) and we considered time slot of 3 hours. Our preliminary results show that we can note three different habits depending on the kind of the day: workday, part-time workday and weekend. Results show that it is possible to note changes in the habits of the user depending on the day of the week. In particular, it could be noted the hours in which the user is at home and the room(s) in which passes the majority of the time. Figure 6 and Figure 7 show an example of recognized habits for a full-time (i.e., Monday) and a part-time workday (i.e., Friday), respectively.

6.2 Quality of Life Assessment

As already said, data collected by the TMHSS will be also used to automatically assess QoL of people. Let us summarize here our preliminary results obtained to assess the movement ability of the given user. The interested reader may refer to [15] for a more deep explanation of the approach.

To assess movement ability, we considered a window of three months (February '14 – April '14) and made comparisons of results for three classifiers: decision tree, k-nn with k=1, and k-nn with k=3. During all the period, the user answered to the question "Today, how was your ability to move about?", daily at 7 PM. Answers have been then used to label the item of the dataset to train and test the classifiers built to verify the feasibility of the proposed QoL approach. Given a category, we consider as true positive (true negative), any entry evaluated as positive (negative) by the classifier that corresponds to an entry labeled by the user as belonging (not belonging) to that class. Seemly, we consider as false positive (false negative), any entry evaluated as positive (negative) by the classifier

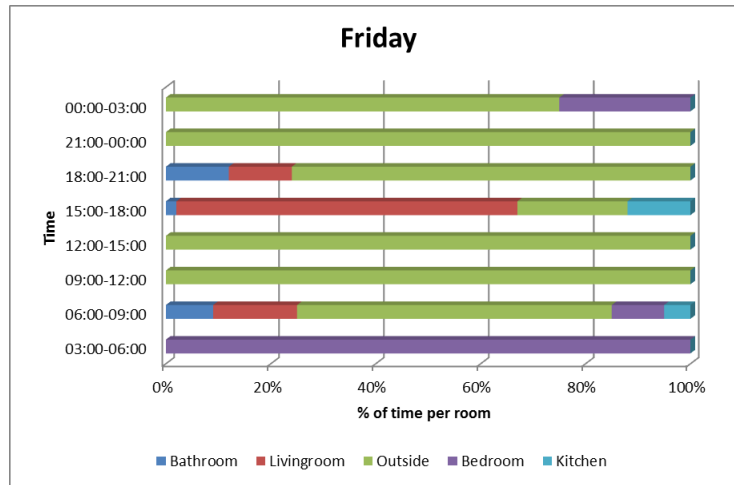


Fig. 7. User's habits: part-time workday.

that corresponds to an entry labeled by the user as not belonging (belonging) to that class. Results have been then calculated in terms of precision, recall, and F_1 measure.

Let us stress the fact that in this preliminary experimental phase, we are considering data coming from a healthy-user. Thus, while analyzing data, the following issues must be considered: tests have been performed with only one user; the user is healthy; and a window of less than 4 months of data has been considered. As a consequence, results can be used and analyzed only as a proof of concept of the feasibility of the approach.

The best results have been obtained using the decision tree. In fact, in that case, on average we calculated a precision of 0.64, a recall of 0.69 and a F_1 of 0.66. It is worth noting that, as expected (the user is healthy and not have difficulty in movements), the best results are given in recognizing "Normal" mobility. In fact, in this case we obtained a precision of 0.80, a recall of 0.89 and an F_1 measure of 0.84.

7 Conclusions and Future Work

Telemonitoring and home support systems help people with severe disabilities as well as their therapists and caregivers. In fact, users may take advantage of telemonitoring and home support to easily come back to their normal life. Moreover, therapists and caregivers can be aware of users' activities providing them support in case of emergencies. For all those reasons, in BackHome a telemonitoring and home support system has been developed. The system consists of a set of sensors installed at user' home as well as of a web application that allows therapist to monitor user' status and activities. Currently, the system is installed in a healthy user's home in Barcelona. Preliminary results show that the system is able to collect and analyse data useful to learn user's habits and it looks promising to assess quality of life.

The next step consists of installing the overall system under the umbrella of BackHome project. In fact, we are currently setting up the proposed telemonitoring and home support system at BackHome real end-users' homes at the facilities of Cedar Foundation⁴ in Belfast. Such installation is scheduled in November 2014. As for future work, starting from data coming from the real end-users, users' daily activities will be deeply monitored, alarms sent back to therapists, and further actions performed to provide home support and context-awareness. Moreover, experiments will be performed to assess quality of life of people, not only "Mobility" but other ambitious items such as "Mood".

Acknowledgments

The research leading to these results has received funding from the European Communitys, Seventh Framework Programme FP7/2007-2013, BackHome project grant agreement n. 288566.

References

1. Artinian, N.: Effects of home telemonitoring and community-based monitoring on blood pressure control in urban African Americans: A pilot study. *Heart Lung* 30, 191–199 (2001)
2. Carneiro, D., Costa, R., Novais, P., Machado, J., Neves, J.: Simulating and monitoring ambient assisted living. In: *Proc. ESM* (2008)
3. Casals, E., Cordero, J.A., Dauwalder, S., Fernández, J.M., Solà, M., Vargiu, E., Miralles, F.: Ambient intelligence by atml: Rules in backhome. In: *Emerging ideas on Information Filtering and Retrieval. DART 2013: Revised and Invited Papers*; C. Lai, A. Giuliani and G. Semeraro (eds.) (2014)
4. Corchado, J., Bajo, J., Tapia, D., Abraham, A.: Using heterogeneous wireless sensor networks in a telemonitoring system for healthcare. *IEEE Transactions on Information Technology in Biomedicine* 14(2), 234–240 (2010)
5. Cordisco, M., Benjaminovitz, A., Hammond, K., Mancini, D.: Use of telemonitoring to decrease the rate of hospitalization in patients with severe congestive heart failure. *Am J Cardiol* 84(7), 860–862 (1999)
6. Daly, J., Armstrong, E., Miralles, F., Vargiu, E., Müller-Putz, G., Hintermller, C., Guger, C., Kuebler, A., Martin, S.: Backhome: Brain-neural-computer interfaces on track to home. In: *RAatE 2012 - Recent Advances in Assistive Technology & Engineering* (2012)
7. Edlinger, G., Holzner, C., Guger, C.: A hybrid brain-computer interface for smart home control. In: *Proceedings of the 14th international conference on Human-computer interaction: interaction techniques and environments - Volume Part II*. pp. 417–425. *HCI'11*, Springer-Verlag, Berlin, Heidelberg (2011)
8. Fernández, J.M., Dauwalder, S., Torrellas, S., Faller, J., Scherer, R., Omedas, P., Verschure, P., Espinosa, A., Guger, C., Carmichael, C., Costa, U., Opisso, E., Tormos, J., Miralles, F.: Connecting the disabled to their physical and social world: The BrainAble experience. In: *TOBI Workshop IV Practical Brain-Computer Interfaces for End-Users: Progress and Challenges* (2013)

⁴<http://www.cedar-foundation.org/>

9. Fernández, J.M., Torrellas, S., Dauwalder, S., Solà, M., Vargui, E., Miralles, F.: Ambient-intelligence trigger markup language: A new approach to ambient intelligence rule definition. In: 13th Conference of the Italian Association for Artificial Intelligence (AI*IA 2013). CEUR Workshop Proceedings, Vol. 1109 (2013)
10. Holzner, C., Schaffelhofer, S., Guger, C., Groenegress, C., Edlinger, G., Slater, M.: Using a p300 brain-computer interface for smart home control. In: World Congress 2009 (2009)
11. Intille, S.S., Kaushik, P., Rockinson, R.: Deploying Context-Aware Health Technology at Home: Human-Centric Challenges. Human-Centric Interfaces for Ambient Intelligence (2009)
12. Käthner, I., Daly, J., Halder, S., Räderscheidt, J., Armstrong, E., Dauwalder, S., Hintermüller, C., Espinosa, A., Vargiu, E., Pinegger, A., Faller, J., Wriessnegger, S., Miralles, F., Lowish, H., Markey, D., Müller-Putz, G., Martin, S., Kübler, A.: A p300 bci for e-inclusion, cognitive rehabilitation and smart home control. In: Graz BCI Conference 2014 (2014)
13. Martí-Lesende, I., Orruño, E., Cairo, C., Bilbao, A., Asua, J., Romo, M., Vergara, I., Bayn, J., Abad, R., Reviriego, E., Larrañaga, J.: Assessment of a primary care-based telemonitoring intervention for home care patients with heart failure and chronic lung disease. The TELBIL study. BMC Health Services Research 11(56) (2011)
14. Meystre, S.: The current state of telemonitoring: a comment on the literature. Telemed J E Health 11(1), 63–69 (2005)
15. Miralles, F., Vargiu, E., Casals, E., Cordero, J., Dauwalder, S.: Today, how was your ability to move about? In: 3rd International Workshop on Artificial Intelligence and Assistive Medicine, ECAI 2014 (2014)
16. Mitchell, M., Meyers, C., Wang, A., Tyson, G.: Contextprovider: Context awareness for medical monitoring applications. In: Conf Proc IEEE Eng Med Biol Soc. (2011)
17. Müller, G., Neuper, C., Pfurtscheller, G.: Implementation of a telemonitoring system for the control of an EEG-based brain-computer interface. IEEE Trans. Neural Syst Rehabil Eng. 11(1), 54–59 (2003)
18. S.Barro, D.Castro, M.Fernndez-Delgado, S.Fraga, M.Lama, J.M.Rodriguez, J.A.Vila: Intelligent telemonitoring of critical-care patients. IEEE ENGINEERING IN MEDICINE AND BIOLOGY MAGAZINE 18, 80–88 (1999)
19. Tonin, L., Leeb, R., Tavella, M., Perdakis, S., Millán, J.: A bci-driven telepresence robot. International Journal of Bioelectromagnetism 13(3), 125 – 126 (2011)
20. Vargiu, E., Dauwalder, S., Daly, J., Armstrong, E., Martin, S., Miralles, F.: Cognitive rehabilitation through bnci: Serious games in backhome. In: Graz BCI Conference 2014 (2014)
21. Vargiu, E., Fernández, J.M., Miralles, F.: Context-aware based quality of life telemonitoring. In: Distributed Systems and Applications of Information Filtering and Retrieval. DART 2012: Revised and Invited Papers. C. Lai, A. Giuliani and G. Semeraro (eds.) (2014)
22. Vargiu, E., Fernández, J.M., Torrellas, S., Dauwalder, S., Solà, M., Miralles, F.: A sensor-based telemonitoring and home support system to improve quality of life through bnci. In: 12th European AAATE Conference (2013)
23. Vincent, J., Cavitt, D., Karpawich, P.: Diagnostic and cost effectiveness of telemonitoring the pediatric pacemaker patient. Pediatr Cardiol. 18(2), 86–90 (1997)

Extending an Information Retrieval System through Time Event Extraction

Pierpaolo Basile, Annalina Caputo, Giovanni Semeraro, and Lucia Siciliani

Department of Computer Science - University of Bari Aldo Moro
Via E. Orabona, 4 - 70125 Bari (ITALY)
e-mail: {pierpaolo.basile@uniba.it, annalina.caputo@uniba.it,
giovanni.semeraro@uniba.it, siciliani.lu@gmail.com}

Abstract. In this paper we propose an innovative Information Retrieval system able to manage temporal information. The system allows temporal constraints in a classical keyword-based search. Information about temporal events is automatically extracted from text at indexing time and stored in an ad-hoc data structure exploited by the retrieval module for searching relevant documents. Our system can search textual information that refers to specific period of times. We perform an exploratory case study indexing all Italian Wikipedia articles.

1 Introduction

Identifying specific pieces of information related to a particular time period is a key task for searching past events. Although this task seems to be marginal for Web users [18], many search domains, like enterprise search, or lately developed information access tasks, such as Question Answering [20] and Entity Search, would benefit from techniques able to handle temporal information.

The capability of extracting and representing temporal events mentioned in a text can enable the retrieval of documents relevant for a given topic pertaining to a specific time. Nonetheless, the notion of *temporal* in the retrieval context has often being associated with the dynamic dimension of a piece of information, i.e. how it changes over time, in order to promote freshness in results. Such kind of approaches focus on when the document was published (*timestamp*) rather than the temporal event mentioned in its content (*focus time*). While traditional search engines take into account temporal information related to a document as a whole, our search engine aims to extract and index single events occurring in the texts, and to enable the retrieval of topics related to specific temporal events mentioned in the documents. In particular, we are interested in retrieving documents that are relevant for the user query, and also match some temporal constraints. For example, the user could be interested in a particular topic —*strumenti musicali (musical instrument)*— related to a specific time period —*inventati tra il 1300 ed il 1500 (invented between 1300 and 1500)*—.

However, looking for happenings in a specific time span requires further, and more advanced, techniques able to treat temporal information. Therefore, our goal is to merge features of both Information Retrieval (IRS) and Temporal Extraction Systems (TES). While an IRS allows us to handle and access the information included in texts, TES locate temporal expressions. We define this kind of system “Time-Aware IR” (TAIR).

In the past, several attempts have been made to exploit temporal information in IR systems [2], with an up-to-date literature review and categorization provided in [7]. Most of these approaches exploit time information related to the document in order to improve the ranking (recent documents are more relevant) [9], cluster documents using temporal attributes [1,3], or exploit temporal information for effectively present documents to the user [16]. However, just a handful of work have focused on temporal queries, that is the capability of querying a collection with both free text and temporal expression [4]. Alonso et al. pointed out as this kind of tasks needs the combination of results from both the traditional keyword-based and the temporal retrieval that can give rise to two different result sets. Vandembussche and Teissèdre [23] dealt with temporal search in the context of both the Web of Content and the Web of Data, but differently from our system, they relied on an ontology of time for temporal queries [11]. Kanhabua and Nørnvåg [13] defined semantic- and temporal-based features for a learning to rank approach by extracting named entities and temporal events from the text. Similarly to our approach, Arikian et al. [5] considered the query as composed by a keyword and a temporal part. Then, the two queries were addressed by computing two different language model-based weights. Exploiting a similar model, Berberich et al. [6] developed a framework for dealing with uncertainty in temporal queries. However, both approaches drawn the probability of the temporal query out of the whole document, thus neglecting the pertinence of temporal events at a sentence level. In order to overcome such a limitation, Matthews et al. [17] introduced two different types of indexes, at a document and a sentence level, with the latter associated with content date.

Preliminary to indexing and retrieval, the Information Extraction phase aims to extract temporal information, and its associated events, from text. In this area [15], several approaches aim at building structured knowledge sources of temporal events. In [12] the authors describe an extension of the YAGO knowledge base, in which entities, facts, and events are anchored in both time and space. Other work exploit Wikipedia to extract temporal events, such as those reported in [10, 14, 25]. Temporal extraction systems can locate temporal expressions and normalize them making this information available for further processing. Currently, there are different tools that can make this kind of analysis on documents, like SUTime [8] or HeidelTime [21] and other systems which took part in TempEval evaluation campaigns. Temporal extraction is not the main focus of this paper, then we remand the interested reader to the TempEval description task papers [22, 24] for a wider overview of the latest state-of-the-art temporal extraction systems.

The paper is organized as follows: Section 2 provides details about the model behind our TAIR system, while Section 3 describes the implementation of our model. Section 4 reports some use cases of the TAIR system which show the potential of our approach, while Section 5 closes the paper.

2 Time-Aware IR Model

A TAIR model should be able to tackle some problems that emerge from temporal search [23], that is: 1) the extraction and normalization of temporal references, 2) the representation of the temporal expressions associated to documents, and 3) the ranking under the constraint of keyword- and temporal-queries.

Our TAIR model consists of three main components responsible to deal with these issues, as sketched in Figure 1:

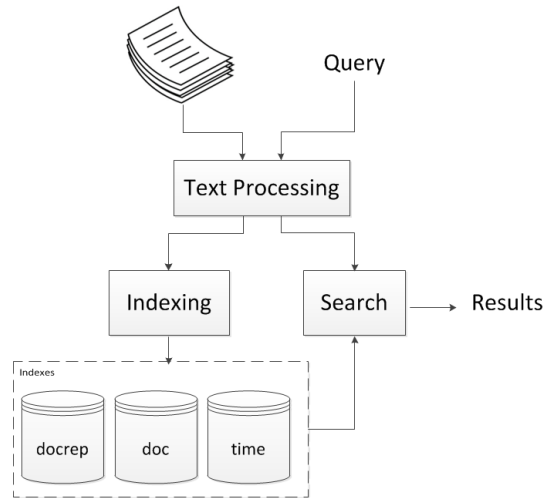


Fig. 1: The IR time-aware Model

Text processing It automatically extracts time expressions from text. The extracted expressions are normalized in a standard format and sent to the indexing component;

Indexing This component is dedicated to index both textual and temporal information. During the indexing, text fragments are linked to time expressions. The idea behind this approach is that the context of a temporal expression is relevant;

Search It analyzes the user query composed by both keywords and temporal constraints, and performs the search over the index in order to retrieve relevant information.

2.1 Text Processing Component

Given a document as input, the text processing component provides as output the *normalized* temporal expressions extracted from the text, along with information about positions in which the temporal expressions are found. For this purpose we adopt a standard annotation language for temporal expressions called TimeML [19]. We are interested in expressions tagged with the TIMEX3 tag that is used to mark up explicit temporal expressions, such as times, dates and durations. In TIMEX3 the value of the temporal expression is normalized according to 2002 TIDES guideline, an extension of the ISO-8601 standard, and is stored in an attribute called *value*. An example of TIMEX3 annotation for the sentence “before the 23th May 1980” is reported below:

```

<TimeML>
  before the
  <TIMEX3 tid="t3" type="DATE" value="1980-05-23">
    23th May 1980
  </TIMEX3>
</TimeML>

```

Where `tid` is a unique identifier, `type` can assume one of the types between: DATE, TIME, DURATION, and SET, while the `value` attribute contains the temporal information that varies accordingly to the type.

ISO-8601 normalizes temporal expressions in several formats. For example, “May 1980” is normalized as “1980-05”, while “23th May 1980” as “1980-05-23”. We choose to normalize all dates using the pattern `yyyy-mm-dd`. All temporal expressions not compliant to the pattern, such as “1980”, must be normalized retaining the lexicographic order between dates. Our solution consists in normalizing all temporal expressions in the form of `yyyy` or `yyyy-mm` to the last day of the previous year or month, respectively. In our previous example, the expression “1980” is normalized as 19791231. Similarly, the expression “1980-05” is normalized as “1980-04-30”. Moreover, the text processing component applies several normalization rules to correctly identify seasons, for example the TimeML tag for Spring “`yyyy-SP`” is normalized as “`yyyy-03-20`”.

Using the correct normalization, the order between periods is respected. In conclusion the text processing component extracts temporal information and correctly normalized them to make different time periods comparable.

2.2 The Indexing Component

After the text processing step, we need to store and index data. In our model we propose to store both documents and temporal expressions in three separated data indexes, as reported in Figure 1.

The first index (*docrep*) stores the text of each document (without processing) with an id, a numeric value that unequivocally identifies the document. This index is used to store the document content only for the presentation purpose. The second index (*doc*) is a traditional inverted index in which the text of each document is indexed and used for keyword-based search. Finally, the last index (*time*) stores temporal expressions found in each document. For each temporal expression, we store the following information:

- The document id;
- The normalized value of the time expression according to the normalization procedure described in Section 2.1;
- The start and end offset of the expression in the document, useful for highlighting;
- The context of the expression: the context is defined by taking all the words that can be found within n characters before and after the time expression. The context is indexed and used by the search component during the retrieval step. The idea is to keep trace of the context where the time expression occurred. The context is tokenized and indexed and exploited in conjunction with the keyword-based search, as we explained in Section 2.3.

It is important to note that a document could have many temporal expressions, for each of these an entry in the *time* index is created. For example, given the

Clavicembalo

Da Wikipedia, l'enciclopedia libera.

Con il termine **clavicembalo** (altrimenti detto **gravicembalo**, arpicordo, cimbalo, cembalo) si indica una famiglia di **strumenti musicali a corde**, dotati di **tastiera**: tra questi, anzitutto lo strumento di grandi dimensioni attualmente chiamato clavicembalo, ma anche i più piccoli **virginale** e **spinetta**.

Questi strumenti generano il suono pizzicando la corda, anziché colpirla come avviene nel pianoforte o nel **clavicordo**. La famiglia del clavicembalo ha probabilmente avuto origine quando una tastiera è stata adattata ad un **salterio**, fornendo così un mezzo per pizzicare le corde. Il termine stesso, che compare per la prima volta in un documento del 1397^[1], deriva dal latino *clavis*, chiave (intesa come il meccanismo che utilizza il movimento del tasto per azionare il leveraggio retrostante), e *cymbalum*, termine che designava nel medioevo gli strumenti musicali con corde parallele tese su una cassa poligonale e senza manico, come i **salteri** e le **cetre**. In ogni caso, la più antica descrizione nota del clavicembalo risale al 1440 circa^[2]. I costruttori di clavicembali e strumenti simili sono detti **cembalari** o **cembalai**^[3].

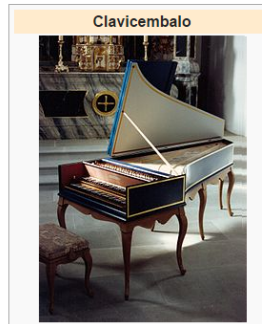


Fig. 2: Wikipedia page example.

Wikipedia page in Figure 2, we store its whole content as reported in Table 1a, while we tokenize and index the page as shown in Table 1b. The most interesting part of the indexing step is the storage of temporal expressions. As depicted in Table 1c, for each temporal expression we store the normalized time value, in this case “13961231”, and the start and end offset of the expression in the text. Finally, we tokenize and index the context in which the expression occurs. In Table 1c, in italics is reported the left context, while the right context is reported in bold. Examples are reported according to the Italian version of Wikipedia, but the indexing step is language independent.

2.3 The Search Component

The search component retrieves relevant documents according to the user query q containing temporal constraints. For this reason we need to make temporal expressions in the query compliant with the expressions stored in the index. The query is processed by the Text Component in order to extract and normalize the time expressions.

The query q is represented by two parts: q_k contains keywords, while q_t only the normalized time expressions. q_k is used to retrieve from the doc index a first results set RS_{doc} . Thus, both q_k and q_t are used to query the $time$ index producing the results set RS_{time} . The search in $time$ index is limited to those documents belonging to RS_{doc} . In RS_{time} , text fragments have to match the time constraints expressed in q_t , while the matching with the keyword-based query q_k is optional. The optional matching with q_k has the effect of promoting those contexts that satisfy both the temporal constraints and the query topics, while not completely removing poorly matching results. The motivation behind this approach is twofold: through RS_{doc} we retrieve those documents relevant for the query topic, while RS_{time} contains the text fragments that match the time query q_t and are related to the query topic.

For example given the query $q = \text{“clavicembalo [1300 TO 1400]”}$, we identify the two fields: $q_k = \text{“clavicembalo”}$ and $q_t = [12991231 \text{ TO } 13991231]$. It is

| <i>Field</i> | <i>Value</i> |
|--------------|--|
| ID | 42 |
| Content | Con il termine clavicembalo (altrimenti detto gravicembalo, arpicordo, cimballo, cembalo) si indica una famiglia di strumenti musicali a corde [...] |

(a) *docrep* index.

| <i>Field</i> | <i>Value</i> |
|--------------|---|
| ID | 42 |
| Content | {‘Con’, ‘il’, ‘termine’, ‘clavicembalo’, ‘altrimenti’, ‘detto’, ‘gravicembalo’, ‘arpicordo’, ‘cimballo’, ‘cembalo’, ‘si’, ‘indica’, ‘una’, ‘famiglia’, ‘di’, ‘strumenti’, ‘musicali’, ‘a’, ‘corde’ [...]} |

(b) *doc* index.

| <i>Field</i> | <i>Value</i> |
|--------------|--|
| ID | 42 |
| Time | 13961231 |
| Start Offset | 350 |
| End Offset | 354 |
| Context | {‘ <i>Il</i> ’, ‘ <i>termine</i> ’, ‘ <i>stesso</i> ’, ‘ <i>che</i> ’, ‘ <i>compare</i> ’, ‘ <i>per</i> ’, ‘ <i>la</i> ’, ‘ <i>prima</i> ’, ‘ <i>volta</i> ’, ‘ <i>in</i> ’, ‘ <i>un</i> ’, ‘ <i>documento</i> ’, ‘ <i>del</i> ’, ‘ <i>deriva</i> ’, ‘ <i>dal</i> ’, ‘ <i>latino</i> ’, ‘ <i>clavis</i> ’, ‘ <i>chiave</i> ’ [...]} |

(c) *time* index.

Table 1: The three indices used by the system.

important to underline that in this example we adopted a particular syntax to identify range queries, more details about the system implementation are reported in Section 3.

The retrieval step produces two results sets: RS_{doc} and RS_{time} . Considering the query q in the previous example: RS_{doc} contains the doc 42 with a relevance score s_{doc} . While the results set RS_{time} contains the temporal expression reported in Table 1c with a score s_{time} . The last step is to combine the two results sets. The idea is to promote text fragments in RS_{time} that comes from documents that belong to RS_{doc} . We simply boost the score of each result in RS_{time} multiplying its score by the score assigned to its origin document in RS_{doc} . In our example the temporal expression occurring in RS_{time} obtains a final score computed as: $s_{doc} \times s_{time}$. We have chosen to boost score rather than linearly combine them, in this way we avoid the use of combination parameters.

Finally, we sort the re-ranked RS_{time} and provide it to the user as final result of the search. It is important to underline that our system does not produce a list of document as a classical search engine does, but we provide all the text passages that are both relevant for the query and compliant to temporal constraints.

3 System Implementation

We implemented our TAIR model in a freely available system¹ as an open-source software under the GNU license V.3. The system is developed in JAVA and extends the indexing and search open-source API Apache Lucene².

The text processing component is based on the HeidelbergTime tool³ [21] to extract temporal information. We adopt this tool for two reasons: 1) it obtained good performance in the TempEval-3 task, and 2) it is able to analyze text written in several languages including the Italian. HeidelbergTime is a rule based system that can be extended to support other languages or specific domains.

Our system provides all the expected functionalities: text analysis, indexing and search. The query language supports all operators provided by the Lucene query syntax⁴. Moreover the temporal query q_t can be formulated using natural time expressions, for example “12 May 2014” or “yesterday”. The search component tries to automatically translate the user query in the proper time expressions. However, the user can directly formulate q_t using normalized time expressions and query operators. Table 2 shows some time operators.

| <i>Query</i> | <i>Description</i> |
|------------------------|--|
| 20020101 | match exactly 1st January 2002 |
| [20020101 TO 20030101] | match from 1st January 2002 to 1st January 2003 |
| [* TO 20030101] | before 1st January 2003 |
| [20020101 TO *] | after 1st January 2002 |
| 01??2002 | any first day of the month in 2002, * should be used for multiple character match, for example 01*2002 |
| 20020101 AND 20020131 | the first and last day of January 2002, AND and OR operator can be used to combine exact match and range query |

Table 2: Example of time query operators.

Currently the system does not provide a GUI for searching and visualizing the results, but it is designed as an API. As future works we plan to extend the API with REST Web functionalities.

4 Use case

We decided to set up a case study to show the potentialities of the proposed IR framework. The case study involves the indexing of a large collection of docu-

¹ <https://github.com/pippokill/TAIR>

² <http://lucene.apache.org/>

³ <https://code.google.com/p/heideltime/>

⁴ http://lucene.apache.org/core/4_8_1/queryparser/org/apache/lucene/queryparser/classic/package-summary.html

ments and a set of example queries exploiting specific scenarios in which temporal expressions play a key role. Moreover, another goal is to provide performance information about the system in terms of indexing and query time, and index space.

We propose an exploratory use case indexing all Italian Wikipedia articles. Our choice is based on the fact that Wikipedia is freely available and contains millions of documents with many temporal events. We need to set some parameters: we index only documents with at least 4,000 characters, remove special pages (e.g. category pages), we set the context size in temporal index to 256 characters.

We perform the experiment on a virtual machine with four virtual cores and 32GB of RAM. Table 3 reports some statistics related to the indexing step. The indexing time is very high due to the complexity of the temporal extraction algorithm and the huge number of documents. We speed up the temporal event extraction implementing a multi threads architecture, in particular in this evaluation we enable four threads for the extraction.

| <i>Statistics</i> | <i>Value</i> |
|--------------------------------|--------------|
| Number of documents | 168,845 |
| Number of temporal expressions | 6,615,430 |
| Indexing time | 68 hours |
| Indexing time (doc./min.) | 41,38 |

Table 3: Indexing performance.

One of the most appropriate scenarios consists in finding events that happened in a specific date. For example, one query could be interested in listing all events happened on 29 April 1981. In this case the time query is “19810429” while the keyword query is empty. The first three results are shown in Table 4.

We report in bold the temporal expressions that match the query. It is important to note that in the first result the year “1981” appears distant from both the month and the day, but the Text Processing component is able to correctly recognize and normalize the date.

Another interesting scenario is to find events related to a specific topic in a particular time period. For example, Table 5 reports the first three results for the query: “terremoti tra il 1600 ed il 1700” (*earthquakes between 1600 and 1700*). This query is split in its keyword q_k = “terremoti” (*earthquakes*) and temporal component q_t = [15991231 TO 16991231].

Table 6 shows the usage of time query operators, in particular of wild-cards. We are interested in facts related to *computers* which happened in January 1984 using the time query pattern “198401??”.

As reported in Table 6, the first two results regard events whose time interval encompasses the time expressed in the query, since they took place in 1984, while the third result shows an event that completely fulfil the time requirements expressed in the temporal query.

| Result Rank | Wikipedia page | Time Context |
|-------------|--------------------------------------|---|
| 1 | Paul Breitner | nel 1981 , richiamato da Jupp Derwall, nel frattempo divenuto nuovo commissario tecnico della Germania Ovest, e con il quale aveva comunque avuto accese discussioni a distanza. Il “nuovo debutto” avviene ad Amburgo il 29 aprile contro l’Austria. |
| 2 | ...E tu vivrai nel terrore! L’aldilà | Warbeck e Catriona McColl, presente nei contenuti speciali del DVD edito dalla NoShame. Accoglienza. Il film uscì in Italia il 29 aprile 1981 e incassò in totale 747.615.662 lire. Distribuito per i mercati esteri dalla VIP International, ottenne un ottimo successo |
| 3 | RCS Media Group | L’operazione venne perfezionata il 29 aprile 1981 . Quel giorno una società dell’Ambrosiano (quindi di Calvi), la “Centrale Finanziaria S.p.A.” effettuò l’acquisto del 40% di azioni Rizzoli |

Table 4: Results for the query “19810429”

| Result Rank | Wikipedia page | Time Context |
|-------------|---|--|
| 1 | Terremoto della Calabria dell’8 giugno 1638 | Il terremoto dell’ 8 giugno 1638 fu un disastroso terremoto che colpì la Calabria, in particolare il Crotonese e parte del territorio già colpito nei giorni 27 e 28 marzo del 1638 |
| 2 | Eruzione dell’Etna del 1669 | 1669 10 marzo - M = 4.8 Nicolosi Terremoto con effetti distruttivi nel catanese in particolare a Nicolosi in seguito all’eruzione dell’Etna conosciuta come Eruzione dell’Etna del 1669 . Il 25 febbraio e l’ 8 e 10 marzo del 1669 una serie di violenti terremoti. |
| 3 | Terremoto del Val di Noto del 1693 | l’evento catastrofico di maggiori dimensioni che abbia colpito la Sicilia orientale in tempi storici. Il terremoto del 9 Gennaio 1693 |

Table 5: Results for the query “earthquakes between 1600 and 1700”

| Result Rank | Wikipedia page | Time Context |
|-------------|-----------------|---|
| 1 | Apple III | L'Apple III, detto anche Apple ///, fu un personal computer prodotto e commercializzato da Apple Computer dal 1980 al 1984 come successore dell'Apple II |
| 2 | Home computer | Apple Macintosh (1984), il primo home/personal computer basato su una interfaccia grafica, nonch il primo a 16/32-bit |
| 3 | Apple Macintosh | Apple Computer (oggi Apple Inc.). Commercializzato dal 24 gennaio 1984 al 1 ottobre 1985, il Macintosh il capostipite dell'omonima famiglia |

Table 6: Results for the query “computer” with the temporal pattern “198401??”

5 Conclusions and Future Work

We proposed a “Time-Aware” IR system able to extract, index, and retrieve temporal information. The system expands a classical keyword-based search through temporal constraints. Temporal expressions, automatically extracted from documents, are indexed through a structure that enables both keyword- and time-matching. As a result, TAIR retrieves a list of text fragments that match the temporal constraints, and are relevant for the query topic. We proposed a preliminary case study indexing all the Italian Wikipedia and described some retrieval scenarios which would benefit from the proposed IR model.

As future work we plan to improve both recognition and normalization of time expressions, extending some particular TimeML specifications that in this preliminary work were not taken into account during the normalization process. Moreover, we will perform a deep “in-vitro” evaluation on a standard document collection.

Acknowledgements

This work fulfils the research objectives of the projects PON 01_00850 ASK-Health (Advanced System for the interpretation and sharing of knowledge in health care) and PON 02_00563_3470993 project “VINCENTE - A Virtual collective INtelligenCe ENvironment to develop sustainable Technology Entrepreneurship ecosystems” funded by the Italian Ministry of University and Research (MIUR).

References

1. Alonso, O., Gertz, M.: Clustering of Search Results Using Temporal Attributes. In: Proceedings of the 29th annual international ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 597–598. ACM (2006)

2. Alonso, O., Gertz, M., Baeza-Yates, R.: On the Value of Temporal Information in Information Retrieval. *SIGIR Forum* 41(2), 35–41 (2007)
3. Alonso, O., Gertz, M., Baeza-Yates, R.: Clustering and Exploring Search Results Using Timeline Constructions. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. pp. 97–106. *CIKM '09*, ACM (2009)
4. Alonso, O., Strötgen, J., Baeza-Yates, R.A., Gertz, M.: Temporal Information Retrieval: Challenges and Opportunities. In: *Proceedings of the 1st International Temporal Web Analytics Workshop (TAWW 2011)*. vol. 11, pp. 1–8 (2011)
5. Arikan, I., Bedathur, S.J., Berberich, K.: Time Will Tell: Leveraging Temporal Expressions in IR. In: Baeza-Yates, R.A., Boldi, P., Ribeiro-Neto, B.A., Cambazoglu, B.B. (eds.) *Proceedings of the 2ND International Conference on Web Search and Web Data Mining, WSDM 2009, Barcelona, Spain, February 9-11, 2009*. ACM (2009)
6. Berberich, K., Bedathur, S., Alonso, O., Weikum, G.: A Language Modeling Approach for Temporal Information Needs. In: *Proceedings of the 32Nd European Conference on Advances in Information Retrieval*. pp. 13–25. *ECIR'2010*, Springer-Verlag (2010)
7. Campos, R., Dias, G., Jorge, A.M., Jatowt, A.: Survey of Temporal Information Retrieval and Related Applications. *ACM Computing Surveys* 47(2), 15:1–15:41 (2014)
8. Chang, A.X., Manning, C.D.: SUTime: A library for recognizing and normalizing time expressions. In: *LREC*. pp. 3735–3740 (2012)
9. Elsas, J.L., Dumais, S.T.: Leveraging Temporal Dynamics of Document Content in Relevance Ranking. In: *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*. pp. 1–10. *WSDM '10*, ACM (2010)
10. Hienert, D., Luciano, F.: Extraction of Historical Events from Wikipedia. In: *Proceedings of the First International Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data*. pp. 25–36 (2011)
11. Hobbs, J.R., Pan, F.: An Ontology of Time for the Semantic Web. *ACM Transactions on Asian Language Information Processing (TALIP) - Special Issue on Temporal Information Processing* 3(1), 66–85 (2004)
12. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. *Artificial Intelligence* 194, 28–61 (2013)
13. Kanhabua, N., Nørvåg, K.: Learning to Rank Search Results for Time-sensitive Queries. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. pp. 2463–2466. *CIKM '12*, ACM (2012)
14. Kuzey, E., Weikum, G.: Extraction of temporal facts and events from Wikipedia. In: *Proceedings of the 2nd Temporal Web Analytics Workshop*. pp. 25–32. ACM (2012)
15. Ling, X., Weld, D.S.: Temporal Information Extraction. In: *Proceedings of the 24th Conference on Artificial Intelligence (AAAI 2010)*. Atlanta, GA. (2010)
16. Matthews, M., Tolchinsky, P., Blanco, R., Atserias, J., Mika, P., Zaragoza, H.: Searching through time in the New York Times. In: *Proceedings of the Fourth Workshop on Human-Computer Interaction and Information Retrieval (HCIR 10)*. pp. 41–44 (2010)

17. Matthews, M., Tolchinsky, P., Blanco, R., Atserias, J., Mika, P., Zaragoza, H.: Searching through time in the New York Times. In: Proceedings of the 4th Workshop on Human-Computer Interaction and Information Retrieval, HCIR Challenge 2010. pp. 41–44 (2010)
18. Nunes, S., Ribeiro, C., David, G.: Use of temporal expressions in web search. In: Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval, pp. 580–584. ECIR'08, Springer-Verlag (2008)
19. Pustejovsky, J., Castano, J.M., Ingria, R., Sauri, R., Gaizauskas, R.J., Setzer, A., Katz, G., Radev, D.R.: TimeML: Robust Specification of Event and Temporal Expressions in Text. *New Directions in Question Answering* 3, 28–34 (2003)
20. Sauri, R., Knippen, R., Verhagen, M., Pustejovsky, J.: Evita: A Robust Event Recognizer for QA Systems. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. pp. 700–707. ACL (2005)
21. Strötgen, J., Zell, J., Gertz, M.: HeidelTime: Tuning English and Developing Spanish Resources for TempEval-3. In: 2nd Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the 7th International Workshop on Semantic Evaluation. pp. 15–19. ACL (2013)
22. UzZaman, N., Llorens, H., Derczynski, L., Allen, J., Verhagen, M., Pustejovsky, J.: Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In: 2nd Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the 7th International Workshop on Semantic Evaluation. pp. 1–9. ACL (2013)
23. Vandembussche, P.Y., Teissèdre, C.: Events Retrieval Using Enhanced Semantic Web Knowledge. In: Workshop DeRIVE 2011 (Detection, Representation, and Exploitation of Events in the Semantic Web) in conjunction with 10th International Semantic Web Conference 2011 (ISWC 2011) (2011)
24. Verhagen, M., Sauri, R., Caselli, T., Pustejovsky, J.: SemEval-2010 Task 13: TempEval-2. In: Proceedings of the 5th International Workshop on Semantic Evaluation. pp. 57–62. ACL (July 2010)
25. Whiting, S., Jose, J., Alonso, O.: Wikipedia As a Time Machine. In: Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion. pp. 857–862. International World Wide Web Conferences Steering Committee (2014)

Measuring Discriminant and Characteristic Capability for Building and Assessing Classifiers

Giuliano Armano, Francesca Fanni and Alessandro Giuliani

Dept. of Electrical and Electronic Engineering, University of Cagliari
Piazza d'Armi I09123, Cagliari, Italy

{armano, francesca.fanni, alessandro.giuliani}@diee.unica.it

Abstract. Performance metrics are used in various stages of the process aimed at solving a classification problem. Unfortunately, most of these metrics are in fact *biased*, meaning that they strictly depend on the class ratio –i.e., on the imbalance between negative and positive samples. After pointing to the source of bias for the most acknowledged metrics, novel unbiased metrics are defined, able to capture the concepts of discriminant and characteristic capability. The combined use of these metrics can give important information to researchers involved in machine learning or pattern recognition tasks, such as classifier performance assessment and feature selection.

1 Introduction

Several metrics are used in pattern recognition and machine learning in various tasks concerning classifier building and assessment. An important category of these metrics is related to confusion matrices. Accuracy, precision, sensitivity (also called recall) and specificity are all relevant examples [5] of metrics that belong to this category. As none of the above metrics is able to give information about the process under assessment in isolation, two different strategies have been adopted so far for assessing classifier performance or feature importance: i) devising single metrics on top of other ones and ii) identifying proper pairs of metrics able to capture the wanted information. The former strategy is exemplified by F_1 [6] and MCC (Matthews Correlation Coefficient) [4], which are commonly used in the process of model building and assessment. Typical members of the latter strategy are sensitivity vs. specificity diagrams, which allow to draw relevant information (e.g., ROC curves [1]) in a Cartesian space. Unfortunately, regardless from the strategies discussed above, most of the existing metrics are in fact *biased*, meaning that they strictly depend on the class ratio –i.e., on the imbalance between positive and negative samples. However, the adoption of biased metrics can only be recommended when the statistics of input data is available. In the event one wants to assess the *intrinsic* properties of a classifier, or other relevant aspects in the process of classifier building and evaluation, the adoption of biased metrics does not appear a reliable choice. For this reason, in the literature, some proposals have been made to introduce unbiased metrics –see in particular the work of Flach [2]. In this paper a pair of unbiased metrics is proposed, able to capture the concepts of *discriminant* and *characteristic* capability. The former is expected to measure to which extent positive samples

can be separated from the negative ones, whereas the latter is expected to measure to which extent positive and negative samples can be grouped together. After giving pragmatic definitions of these metrics, their semantics is discussed for binary classifiers and binary features. An analysis focusing on the combined use of the corresponding metrics in form of Cartesian diagrams is also made.

The remainder of the paper is organized as follows: after introducing the concept of normalized confusion matrix, obtained by applying Bayes decomposition to any given confusion matrix, in Section 2 a brief analysis of the most acknowledged metrics is performed, pointing out that most of them are in fact biased. Section 3 introduces novel metrics devised to measure the discriminant and characteristic capability of binary classifiers or binary features. Section 4 reports experiments aimed at pointing out the potential of Cartesian diagrams drawn using the proposed metrics. Section 5 highlights the strengths and weaknesses of this paper and Section 6 draws conclusions.

2 Background

As the concept of confusion matrix is central in this paper, let us preliminarily illustrate the notation adopted for its components (also because the adopted notation slightly differs from the most acknowledged one). When used for classifier assessment, the generic element ξ_{ij} of a confusion matrix Ξ accounts for the number of samples that satisfy the property specified by the subscripts. Limiting our attention to binary problems, in which samples are described by binary features, let us assume that 1 and 0 identify the presence and the absence of a property.

In particular, let us denote with $\Xi_c(P, N)$ the confusion matrix of a run in which a classifier \hat{c} , trained on a category c , is fed with P positive samples and N negative samples (with a total of M samples). With \hat{X}_c and X_c random variables that account for the output of classifier and oracle, the joint probability $p(X_c, \hat{X}_c)$ is proportional, through M , to the expected value of $\Xi_c(P, N)$.

Assuming statistical significance, the confusion matrix obtained from a single test (or, better, averaged over multiple tests in which the values for P and N are left unchanged) gives us reliable information on the performance of the classifier. In symbols:

$$\Xi_c(P, N) \approx M \cdot p(X_c, \hat{X}_c) = M \cdot p(X_c) \cdot p(\hat{X}_c | X_c) \quad (1)$$

In so doing, we assume that the transformation performed by \hat{c} can be isolated from the inputs it processes, at least from a statistical perspective. Hence, the confusion matrix for a given set of inputs can be written as the product between a term that accounts for the number of positive and negative instances, on one hand, and a term that represents the expected recognition / error rate of \hat{c} , on the other hand. In symbols:

$$\Xi_c(P, N) = M \cdot \underbrace{\begin{bmatrix} \omega_{00} & \omega_{01} \\ \omega_{10} & \omega_{11} \end{bmatrix}}_{\Omega(c) \approx p(X_c, \hat{X}_c)} = M \cdot \underbrace{\begin{bmatrix} n & 0 \\ 0 & p \end{bmatrix}}_{\Theta(c) \approx p(X_c)} \cdot \underbrace{\begin{bmatrix} \gamma_{00} & \gamma_{01} \\ \gamma_{10} & \gamma_{11} \end{bmatrix}}_{\Gamma(c) \approx p(\hat{X}_c | X_c)} \quad (2)$$

where:

- $\omega_{ij} \approx p(X_c = i, \hat{X}_c = j)$, $i, j = 0, 1$, denotes the joint occurrence of correct classifications ($i = j$) or misclassifications ($i \neq j$). According to the total probability law: $\sum_{ij} \omega_{ij} = 1$.
- p is the percent of positive samples and n is the percent of negative samples.
- $\gamma_{ij} \approx p(\hat{X}_c = j | X_c = i)$, $i, j = 0, 1$, denotes the percent of inputs that have been correctly classified ($i = j$) or misclassified ($i \neq j$) by \hat{X}_c . $\gamma_{00}, \gamma_{01}, \gamma_{10}$, and γ_{11} respectively denote the *rate* of true negatives, false positives, false negatives, and true positives. According to the total probability law: $\gamma_{00} + \gamma_{01} = \gamma_{10} + \gamma_{11} = 1$. An estimate of the conditional probability $p(\hat{X}_c | X_c)$ for a classifier \hat{c} that accounts for a category c will be called *normalized confusion matrix* hereinafter.

The separation between inputs and the intrinsic behavior of a classifier reported in Equation (2) suggests an interpretation that recalls the concept of transfer function, where a set of inputs is applied to \hat{c} . In fact, Equation (2) highlights the separation of the optimal behavior of a classifier from the deterioration introduced by its actual filtering capabilities. In particular, $\mathcal{O} \approx p(X_c)$ represents the *optimal behavior* obtainable when \hat{c} acts as an *oracle*, whereas $\Gamma \approx p(\hat{X}_c | X_c)$ represents the *expected deterioration* caused by the actual characteristics of the classifier. Hence, under the assumption of statistical significance of experimental results, any confusion matrix can be divided in terms of optimal behavior and expected deterioration using the Bayes theorem.

A different interpretation holds for confusion matrix subscripts when they are used to investigate binary features. In this case i still denotes the actual category, whereas j denotes the truth value of the binary feature (with 0 and 1 made equivalent to *false* and *true*, respectively). However, as a binary feature can always be thought of as a very simple classifier whose classification output reflects the truth value of the feature in the given samples, all definitions and comments concerning classifiers can be applied to binary features as well.

Let us now examine the most acknowledged metrics deemed useful for pattern recognition and machine learning according to the above perspective. The classical definitions for accuracy (a), precision (π), and recall (ρ) can be given in terms of false positives rate (fp), true positives rate (tp) and class ratio (the imbalance between negative and positive samples, σ) as follows:

$$\begin{aligned}
 a &= \frac{\text{trace}(\Omega)}{|\Omega|} = \frac{\omega_{00} + \omega_{11}}{1} = \frac{\sigma \cdot (1 - \gamma_{01}) + \gamma_{11}}{\sigma + 1} = \frac{\sigma \cdot (1 - fp) + tp}{\sigma + 1} \\
 \pi &= \frac{\omega_{01}}{\omega_{01} + \omega_{11}} = \left(1 + \sigma \cdot \frac{\gamma_{01}}{\gamma_{11}}\right)^{-1} = \left(1 + \sigma \cdot \frac{fp}{tp}\right)^{-1} \\
 \rho &= \frac{\omega_{11}}{\omega_{11} + \omega_{10}} = \gamma_{11} = tp
 \end{aligned} \tag{3}$$

Equation (3) highlights the dependence of accuracy and precision from the class ratio, only recall being unbiased. Note that the expression concerning accuracy has been obtained taking into account that $p + n = 1$ implies $p = 1/(\sigma + 1)$ and $n = \sigma/(\sigma + 1)$.

As pointed out, when the goal is to assess the intrinsic properties of a classifier or a feature, biased metrics do not appear a proper choice, leaving room for alternative definitions aimed at dealing with the imbalance between negative and positive samples.

In [2], Flach gave definitions of some unbiased metrics starting from classical ones. In practice, unbiased metrics can be obtained from classical ones by setting the imbalance σ to 1. In the following, if needed, unbiased metrics will be denoted using the subscript u .

3 Definition of Novel Metrics

To our knowledge, no satisfactory definitions have been given so far able to account for the need of capturing the potential of a model according to its discriminant and characteristic capability. With the goal of filling this gap, let us spend few words on the expected behavior of any metrics intended to measure them. Without loss of generality, let us assume the metrics be defined in $[-1, +1]$. As for the discriminant capability, we expect its value be close to $+1$ when a classifier or feature partitions a given set of samples in strong accordance with the corresponding class labels. Conversely, the metric is expected to be close to -1 when the partitioning occurs in strong discordance with the class label. As for the characteristic capability, we expect its value be close to $+1$ when a classifier or feature tend to cluster most of the samples as if they were in fact belonging to the main category. Conversely, the metric is expected to be close to -1 when most of the samples are clustered as belonging to the alternate category.¹ An immediate consequence of the desired behavior is that the above properties are not independent. In other words, regardless from their definition, the metrics devised to measure discriminant and characteristic capability of a classifier or feature (say δ and φ , hereinafter) are expected to show an orthogonal behavior. In particular, when the absolute value of one metric is about 1 the other should be close to 0.

Let us now characterize δ and φ with more details, focusing on classifiers only (similar considerations can also be made for features):

- $fp \approx 0$ and $tp \approx 1$ – We expect $\delta \approx +1$ and $\varphi \approx 0$, meaning that the classifier is able to partition the samples almost in complete accordance with the class labels.
- $fp \approx 1$ and $tp \approx 1$ – We expect $\delta \approx 0$ and $\varphi \approx +1$, meaning that almost all samples are recognized as belonging to the main class label.
- $fp \approx 0$ and $tp \approx 0$ – We expect $\delta \approx 0$ and $\varphi \approx -1$, meaning that almost all samples are recognized as belonging to the alternate class label.
- $fp \approx 1$ and $tp \approx 0$ – We expect $\delta \approx -1$ and $\varphi \approx 0$, meaning that the classifier is able to partition the domain space almost in complete discordance with the class labels (however, this ability can still be used for classification purposes by simply turning the classifier output into its opposite).

The determinant of the normalized confusion matrix is the starting point for giving proper definitions of δ and φ able to satisfy the constraints and boundary conditions

¹It is worth noting that the definition of characteristic capability proposed in this paper is in partial disagreement with the classical concept of “characteristic property” acknowledged by most of the machine learning and pattern recognition researchers. The classical definition only focuses on samples that belong to the main class, whereas the conceptualization adopted in this paper applies to all samples. The motivation of this choice should become clearer later on.

discussed above. It can be rewritten as follows:

$$\begin{aligned}
\Delta &= \gamma_{00} \cdot \gamma_{11} - \gamma_{01} \cdot \gamma_{10} = \gamma_{00} \cdot \gamma_{11} - (1 - \gamma_{00}) \cdot (1 - \gamma_{11}) \\
&= \gamma_{00} \cdot \gamma_{11} - 1 + \gamma_{11} + \gamma_{00} - \gamma_{00} \cdot \gamma_{11} = \gamma_{11} + \gamma_{00} - 1 \\
&= \rho + \bar{\rho} - 1 \equiv tp - fp
\end{aligned} \tag{4}$$

When $\Delta = 0$, the classifier under assessment has no discriminant capability whereas $\Delta = +1$ and $\Delta = -1$ correspond to the highest discriminant capability, from the positive and negative side, respectively. It is clear that the simplest definition of δ is to *make it coincident to Δ* , as the latter has all the desired properties required by the discriminant capability metric.

As for φ , considering the definition of δ and the constraints that must apply to a metric intended to measure the characteristic capability, the following definition appear appropriate, being actually dual with respect to δ also from a syntactic point of view:

$$\varphi = \rho - \bar{\rho} = tp + fp - 1 \tag{5}$$

Figure 1 reports the isometric curves drawn for different values of δ and φ , respectively, with varying tp and fp .

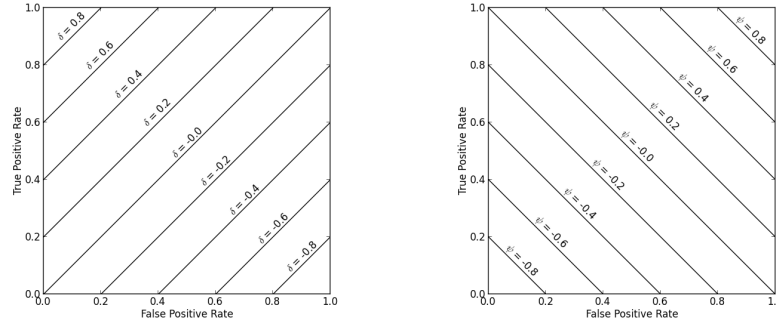


Fig. 1: Isometric plotting of δ and φ with varying false and true positive rate.

The two measures can be taken in combination for investigating properties of classifiers or features. The run of a classifier over a specific test set, different runs of a classifier over multiple test sets, and the statistics about the presence/absence of a feature on a specific dataset are all examples of potential use cases. However, while reporting information about classifier or feature properties in $\varphi - \delta$ diagrams, one should be aware that the $\varphi - \delta$ space is constrained by a rhomboidal shape. This shape depends on the constraints that apply to δ , φ , tp , and fp .

In particular, as $\delta = tp - fp$ and $\varphi = tp + fp - 1$, the following relations hold:

$$\delta = -\varphi + (2 \cdot tp - 1) = +\varphi + (2 \cdot fp + 1) \tag{6}$$

Considering fp and tp as parameters, we can easily draw the corresponding isometric curves in the $\varphi - \delta$ space. Figure 2 shows their behavior for $tp = \{0, 0.5, 1\}$ and for $fp = \{0, 0.5, 1\}$.

As the definitions of δ and φ are given as linear transformations over tp and fp , it is not surprising that the isometric curves of fp and tp drawn in the $\varphi - \delta$ space are again straight lines.

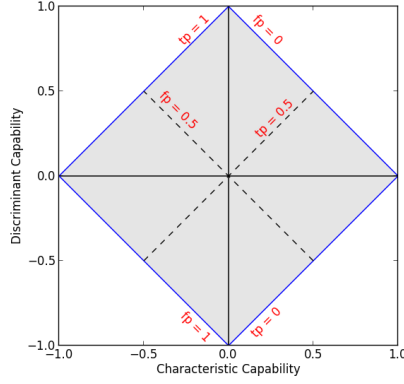


Fig. 2: Shape of the $\varphi - \delta$ space: the rhombus centered in $(0,0)$ delimits the area of admissible value pairs.

Semantics of the $\varphi - \delta$ space for classifiers. As for binary classifiers, their discriminant capability is strictly related to the *unbiased accuracy*, which in turn can be given in terms of *unbiased error* (say e_u). The following equivalences make explicit the relation between a_u , e_u and δ :

$$a_u = \frac{tn + tp}{2} = \frac{1 + \delta}{2} = 1 - \frac{1 - \delta}{2} = 1 - \frac{fp + fn}{2} = 1 - e_u \quad (7)$$

It is worth pointing out that the actual discriminant capability of a classifier is not a redefinition of accuracy (or error), as a classifier may still have high discriminant capability also in presence of high unbiased error. Indeed, as already pointed out, a low-performance classifier can be easily transformed into a high-performance one by simply turning its output into its opposite. Thanks to the “turning-into-opposite” trick, the actual discriminant capability of a classifier could in fact be made coincident with the absolute value of δ . However, for reasons related to the informative content of $\varphi - \delta$ diagrams, we still take apart the discriminant capability observed from the positive side from the one observed on the negative side. As for the characteristic capability, let us

preliminarily note that, in presence of statistical significance, we can write:

$$\begin{aligned} E[X_c] &\approx \frac{1}{M} \cdot (P - N) = (p - n) \\ E[\widehat{X}_c] &\approx \frac{1}{M} \cdot (\widehat{P} - \widehat{N}) = (p - n) + 2 \cdot n \cdot fp - 2 \cdot p \cdot fn \end{aligned} \quad (8)$$

Hence, the difference in terms of expected values between oracle and classifier is:

$$E[X_c - \widehat{X}_c] = E[X_c] - E[\widehat{X}_c] \approx -2 \cdot n \cdot fp + 2 \cdot p \cdot fn \quad (9)$$

According to Friedman [3], it is easy to show that Equation (9) actually represents an estimate of the *bias* of a classifier, measured over the confusion matrix that describes the outcomes of the experiments performed on the test set(s). Summarizing, in a $\varphi - \delta$ diagram used for assessing classifiers, the δ -axis and the φ -axis represent the unbiased accuracy and the unbiased bias, respectively. It is worth pointing out that a high positive value of δ means that the classifier at hand approximates the behavior of an *oracle*, whereas a high negative value approximates the behavior of a classifier that is almost always wrong (say *anti-oracle* when $\delta = -1$). Conversely, a high positive value of φ denotes a *dummy classifier* that almost always consider input items as belonging to the main category, whereas a high negative value denotes a *dummy classifier* that almost always consider input items as belonging to the alternate category.

Semantics of the $\varphi - \delta$ space for features. As for binary features, δ measures to which extent a feature is able to partition the given samples in accordance ($\delta \simeq +1$) or in discordance ($\delta \simeq -1$) with the main class label. In either case, the feature has high discriminant capability. As already pointed out for classifiers, instead of considering the absolute value of δ as a measure of discriminant capability, we take apart the value observed on the positive side from the one observed on the negative side for reasons related to the informative content of $\varphi - \delta$ diagrams. On the other hand, φ measures to which extent the feature at hand is spread over the given dataset. A high positive value of φ indicates that the feature is mainly true along positive and negative samples, whereas a high negative value indicates that the feature is mainly false in the dataset –regardless of the class label of samples.

4 Experiments

Some experiments have been performed with the aim of assessing the potential of $\varphi - \delta$ diagrams. In our experiments we use a collection in which each document is a webpage. The dataset is extracted from the DMOZ taxonomy². Let us recall that DMOZ is the collection of HTML documents referenced in a Web directory developed in the Open Directory Project (ODP). We choose a set of 174 categories containing about 20000 documents, organized in 36 domains.

In this scenario, we expect terms important for categorization appear at the upper or lower corner of the $\varphi - \delta$ rhombus, in correspondence with high values of $|\delta|$. As

²<http://www.dmoz.org>

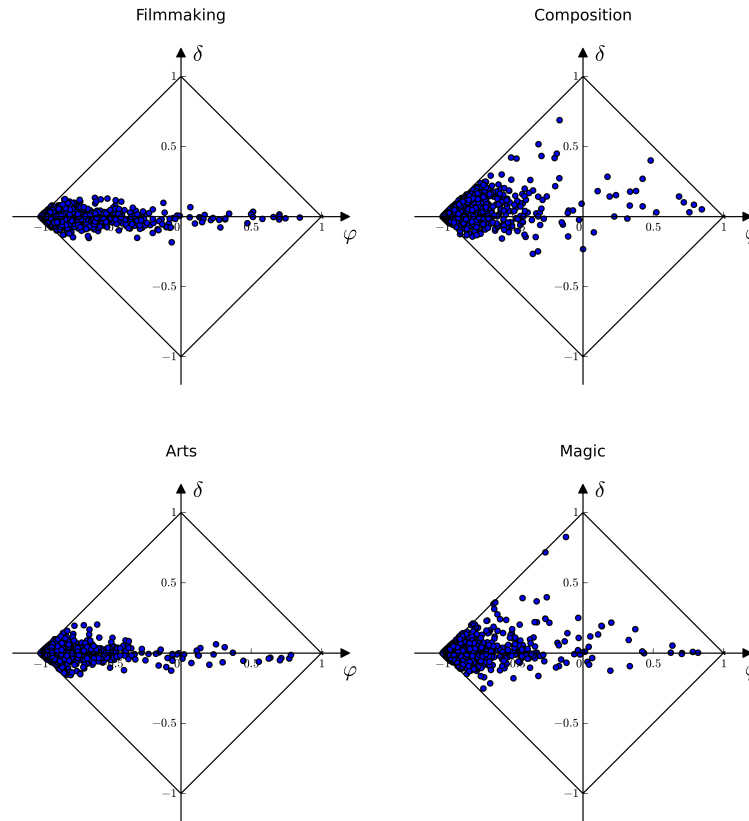


Fig. 3: Position of terms within $\varphi - \delta$ diagrams for the selected DMOZ's categories.

for the characteristic capability, terms that occur barely on documents are expected to appear at the left hand corner (high negative values of φ), while the so-called *stopwords* are expected to appear at the right hand corner (high values of φ).

Experiments have been focusing on the identification of discriminant terms and stopwords. Figure 3 plots the “signatures” obtained for DMOZ's categories *Filmmaking*, *Composition*, *Arts*, and *Magic*. Alternate categories have been derived considering the corresponding siblings. Note that, in accordance with the Zipf's law [7], most of the words are located at the left hand corner of the constraining rhombus. Looking at the drawings, it appears that *Filmmaking* and *Arts* are expected to be the most difficult categories to predict, as no terms with a significant value of $|\delta|$ exist for it. On the contrary, documents of *Composition* and *Magic* appear to be relatively easy to classify, as several terms exist with significant discriminant value. This conjecture is confirmed after training 50 decision trees using only terms t whose characteristic capability satisfies the

constraint $|\varphi(t)| < 0.4$. For each category, test samples have been randomly extracted at each run, whereas the remainder of the samples trained the classifiers.

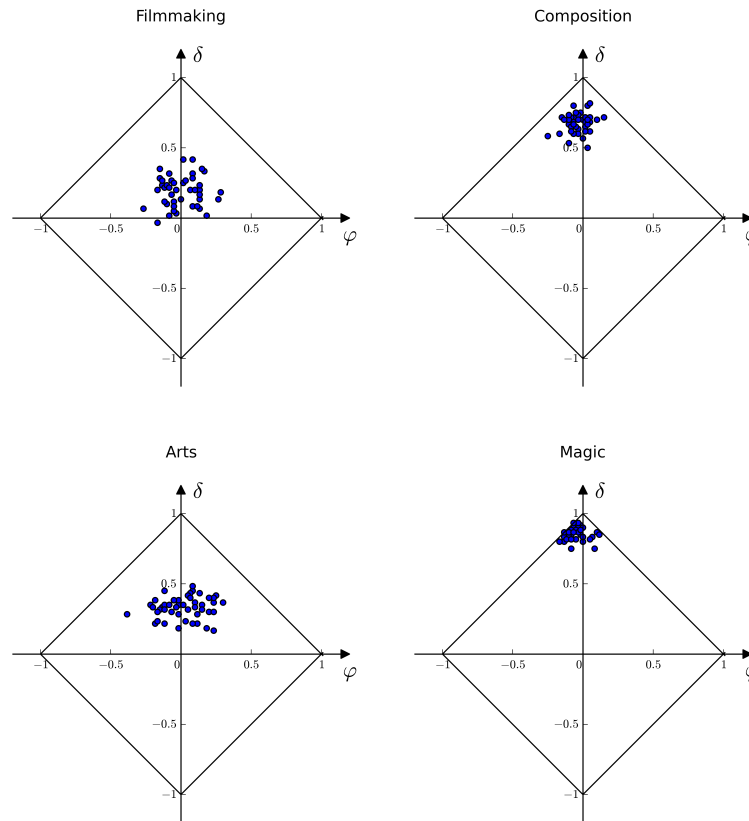


Fig. 4: Four diagrams reporting the classification results.

Figure 4 reports the signatures of classifiers. The figure clearly points out that, as expected, the average (unbiased) accuracies obtained on categories *Composition* and *Magic* are higher than the ones obtained on categories *Filmmaking* and *Arts*. Besides, $\varphi - \delta$ diagrams point out that also variance and bias of classifiers trained for categories *Filmmaking* and *Arts* are apparently worse than those measured on classifiers trained for categories *Composition* and *Magic*.

5 Strengths and Weaknesses of This Proposal

Apart from the analysis of existing metrics, the paper has been mainly concerned with the definition of two novel metrics deemed useful in the task of developing and assessing machine learning and pattern recognition algorithms and systems. All in all, there is no magic in the given definitions. In fact, the $\varphi - \delta$ space is basically obtained by rotating the $fp - tp$ space of $\pi/4$. Although this is not a dramatic change of perspective, it is clear that the $\varphi - \delta$ space allows to analyze *at a glance* the most relevant properties of classifiers or features. In particular, the (unbiased) accuracy and the (unbiased) bias of a classifier are immediately visible on the vertical and horizontal axis of a $\varphi - \delta$ space, respectively. Moreover, an estimate of the variance of a classifier can be easily investigated by just reporting the results of several experiments in the $\varphi - \delta$ space (see, for instance, Figure 4, which clearly points out to which extent the performance of individual classifiers change along experiments). All the above measures are completely independent from the imbalance of data by construction, as the $\varphi - \delta$ space is defined on top of unbiased metrics (i.e., ρ and $\bar{\rho}$). This aspect is very important for classifier assessment, making it easier to compare the performance obtained on different test data, regardless from the imbalance between negative and positive samples. Summarizing, the $\varphi - \delta$ space for classifiers can be actually thought of as a *bias vs. accuracy* (or *error*) space, whose primary uses can be: (i) assessing the accuracy of a classifier over a single or multiple runs, looking at its δ axis; (ii) assessing the bias of a classifier over a single or multiple runs, looking at the φ axis; (iii) assessing the variance of a classifier, looking at the scattering of multiple runs on the $\varphi - \delta$ space. As for binary features, an insight about the potential of $\varphi - \delta$ diagrams in the task of assessing their importance has been given in Section 4. In particular, let us recall that the most important features related to a given domain are expected to have high values of $|\delta|$, whereas not important ones are expected to have high values of $|\varphi|$. Moreover, in the special case of text categorization, stopwords are expected to occur at the right hand corner of the rhombus that constrains the $\varphi - \delta$ space.

It is worth mentioning that alternative definitions could also be given in the $\varphi - \delta$ space for other relevant properties, e.g., ROC curves and AUC (or Gini's coefficient). Although these aspects are beyond the scope of this paper, let us spend few words on ROC curves. It is easy to verify that random guessing for a classifier would constrain the ROC curve to the φ axis, whereas the ROC curve of a classifier acting as an oracle would coincide with the positive border of the surrounding rhombus.

6 Conclusions and Future Work

After discussing and analyzing some issues related to the most acknowledged metrics used in pattern recognition and machine learning, two novel metrics have been proposed, i.e. δ and φ , intended to measure discriminant and characteristic capability for binary classifiers and binary features. They are unbiased and are obtained as linear transformations of false and true positive rates. Moreover, the corresponding isometric curves show that they are orthogonal. The applications of $\varphi - \delta$ diagrams to pattern recognition and machine learning problems are manifold, ranging from feature selection

to classifier performance assessment. Some experiments performed in a text categorization setting confirm the usefulness of the proposal. As for future work, the properties of terms in a scenario of hierarchical text categorization will be investigated using δ and φ diagrams. A generalization of δ and φ to multilabel categorization problems with multivalued features is also under study.

Acknowledgments. This work has been supported by LR7 2009 - Investment funds for basic research (funded by the local government of Sardinia).

References

1. Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recogn.*, 30(7):1145–1159, July 1997.
2. Peter A. Flach. The geometry of roc space: understanding machine learning metrics through roc isometrics. In *in Proceedings of the Twentieth International Conference on Machine Learning*, pages 194–201. AAAI Press, 2003.
3. Jerome H. Friedman and Usama Fayyad. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1:55–77, 1997.
4. B. W. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta*, 405:442–451, 1975.
5. Vijay Raghavan, Peter Bollmann, and Gwang S. Jung. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Trans. Inf. Syst.*, 7(3):205–229, July 1989.
6. C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.
7. George K. Zipf. *Human Behavior and the Principle of Least Effort*. Addison-Wesley (Reading MA), 1949.

A comparison of Lexicon-based approaches for Sentiment Analysis of microblog posts

Cataldo Musto, Giovanni Semeraro, Marco Polignano

Department of Computer Science
University of Bari Aldo Moro, Italy
{cataldo.musto,giovanni.semeraro,marco.polignano}@uniba.it

Abstract. The exponential growth of available online information provides computer scientists with many new challenges and opportunities. A recent trend is to analyze people feelings, opinions and orientation about facts and brands: this is done by exploiting Sentiment Analysis techniques, whose goal is to classify the polarity of a piece of text according to the opinion of the writer.

In this paper we propose a lexicon-based approach for sentiment classification of Twitter posts. Our approach is based on the exploitation of widespread lexical resources such as SentiWordNet, WordNet-Affect, MPQA and SenticNet. In the experimental session the effectiveness of the approach was evaluated against two state-of-the-art datasets. Preliminary results provide interesting outcomes and pave the way for future research in the area.

Keywords: Sentiment Analysis, Opinion Mining, Semantics, Lexicons

1 Background and Related Work

Thanks to the exponential growth of available online information many new challenges and opportunities arise for computer scientists. A recent trend is to analyze people feelings, opinions and orientation about facts and brands: this is done by exploiting Sentiment Analysis [13, 8] techniques, whose goal is to classify the polarity of a piece of text according to the opinion of the writer.

State of the art approaches for sentiment analysis are broadly classified in two categories: *supervised approaches* [6, 12] learn a classification model on the ground of a set of labeled data, while *unsupervised* (or *lexicon-based*) ones [18, 4] infer the sentiment conveyed by a piece of text on the ground of the polarity of the word (or the phrases) which compose it. Even if recent work in the area showed that supervised approaches tend to overcome unsupervised ones (see the recent SemEval 2013 and 2014 challenges [10, 15]), the latter have the advantage of avoiding the hard-working step of labeling training data.

However, these techniques rely on (external) lexical resources which are concerned with mapping words to a categorical (*positive*, *negative*, *neutral*) or numerical sentiment score, which is used by the algorithm to obtain the overall

sentiment conveyed by the text. Clearly, the effectiveness of the whole approach strongly depends on the goodness of the lexical resource it relies on. As a consequence, in this work we investigated the effectiveness of some widespread available lexical resources in the task of sentiment classification of microblog posts.

2 State-of-the-art Resources for Lexicon-based Sentiment Analysis

SentiWordNet: SentiWordNet [1] is a lexical resource devised to support Sentiment Analysis applications. It provides an annotation based on three numerical sentiment scores (positivity, negativity, neutrality) for each WordNet synset [9]. Clearly, given that this lexical resource provides a synset-based sentiment representation, different senses of the same term may have different sentiment scores. As shown in Figure 1, the term *terrible* is provided with two different sentiment associations. In this case, SentiWordNet needs to be coupled with a Word Sense Disambiguation (WSD) algorithm to identify the most promising meaning.

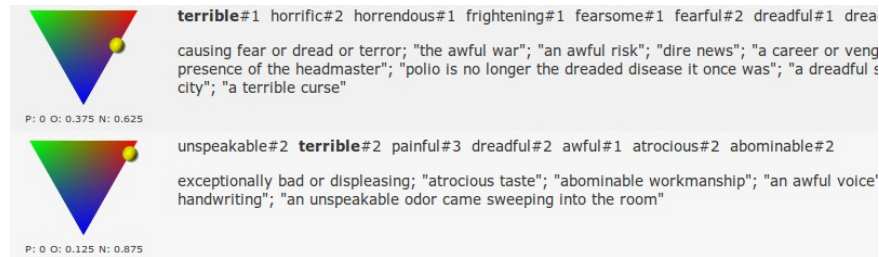


Fig. 1. An example of sentiment association in SentiWordNet

WordNet-Affect: WordNet-Affect [17] is a linguistic resource for a lexical representation of affective knowledge. It is an extension of WordNet which labels affective-related synsets with affective concepts defined as A-LABELS (e.g. the term *euphoria* is labeled with the concept *positive-emotion*, the noun *illness* is labeled with *physical state*, and so on). The mapping is performed on the ground of a domain-independent hierarchy (a fragment is provided in Figure 2) of affective labels automatically built relying on WordNet relationships.

MPQA: MPQA Subjectivity Lexicon [19] provides a lexicon of 8,222 terms (labeled as *subjective expressions*), gathered from several sources. This lexicon contains a list of words, along with their POS-tagging, labeled with polarity (positive, negative, neutral) and intensity (strong, weak).

SenticNet: SenticNet [3] is a lexical resource for *concept-level* sentiment analysis. It relies on the Sentic Computing [2], a novel multi-disciplinary paradigm for Sentiment Analysis. Differently from the previously mentioned resources, SenticNet is able to associate polarity and affective information also to complex

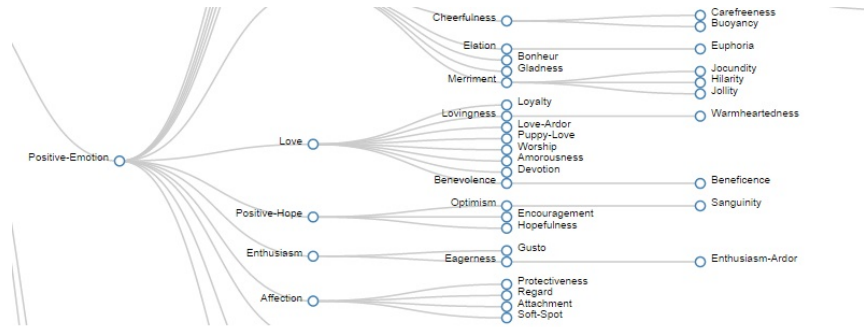


Fig. 2. A fragment of WordNet-Affect hierarchy

concepts such as *accomplishing goal*, *celebrate special occasion* and so on. At present, SenticNet provides sentiment scores (in a range between -1 and 1) for 14,000 common sense concepts. The sentiment conveyed by each term is defined on the ground of the intensity of sixteen *basic* emotions, defined in a model called Hourglass of Emotions (see Figure 3).

3 Methodology

Typically, lexicon-based approaches for sentiment classification are based on the insight that the polarity of a piece of text can be obtained on the ground of the polarity of the words which compose it. However, due to the complexity of natural languages, a so simple approach is likely to fail since many facets of the language (e.g., the presence of the negation) are not taken into account. As a consequence, we propose a more fine-grained approach: given a Tweet T , we split it in several micro-phrases $m_1 \dots m_n$ according to the *splitting cues* occurring in the content. As *splitting cues* we used punctuations, adverbs and conjunctions. Whenever a *splitting cue* is found in the text, a new *micro-phrase* is built.

3.1 Description of the approach

Given such a representation, we define the sentiment S conveyed by a Tweet T as the sum of the polarity conveyed by each of the *micro-phrases* m_i which compose it. In turn, the polarity of each *micro-phrase* depends on the sentimental score of each term in the micro-phrase, labeled as $score(t_j)$, which is obtained from one of the above described lexical resources. In this preliminary formulation of the approach we did not take into account any *valence shifters* [7] except of the negation. When a *negation* is found in the text, the polarity of the whole micro-phrase is inverted. No heuristics have been adopted to deal with neither *language intensifiers* and *downtoners*, or to detect *irony* [14].

We defined four different implementations of such approach: BASIC, NORMALIZED, EMPHASIZED and EMPHASIZED-NORMALIZED. In the BASIC formulation, the



Fig. 3. The Hourglass of Emotions

sentiment of the Tweet is obtained by first summing the polarity of each micro-phrase. Then, the score is normalized through the length of the whole Tweet. In this case the micro-phrases are just exploited to invert the polarity when a negation is found in text.

$$S_{basic}(T) = \sum_{i=1}^n \frac{pol_{basic}(m_i)}{|T|} \quad (1)$$

$$pol_{basic}(m_i) = \sum_{j=1}^k score(t_j) \quad (2)$$

In the NORMALIZED formulation, the *micro-phrase-level* scores are normalized by using the length of the *single* micro-phrase, in order to weigh differently the micro-phrases according to their length.

$$S_{norm}(T) = \sum_{i=1}^n pol_{norm}(m_i) \quad (3)$$

$$pol_{norm}(m_i) = \sum_{j=1}^k \frac{score(t_j)}{|m_i|} \quad (4)$$

The EMPHASIZED version is an extension of the basic formulation which gives a bigger weight to the terms t_j belonging to specific POS categories:

$$S_{emph}(T) = \sum_{i=1}^n \frac{pol_{emph}(m_i)}{|T|} \quad (5)$$

$$pol_{emph}(m_i) = \sum_{j=1}^k score(t_j) * w_{pos(t_j)} \quad (6)$$

where $w_{pos(t_j)}$ is greater than 1 if $pos(t_j) = adverbs, verbs, adjectives$, otherwise 1.

Finally, the EMPHASIZED-NORMALIZED is just a combination of the second and third version of the approach:

$$S_{emphNorm}(T) = \sum_{i=1}^n pol_{emphNorm}(m_i) \quad (7)$$

$$pol_{emphNorm}(m_i) = \sum_{j=1}^k \frac{score(t_j) * w_{pos(t_j)}}{|m_i|} \quad (8)$$

3.2 Lexicon-based Score Determination

Regardless of the variant which is adopted, the effectiveness of the whole approach strictly depends on the way $score(t_j)$ is calculated. For each lexical resource, a different way to determine the sentiment score is adopted.

As regards *SentiWordNet*, t_j is processed through an NLP pipeline to get its POS-tag. Next, all the synsets mapped to that POS of the terms are extracted. Finally, $score(t_j)$ is calculated as the weighted average of all the *sentiment scores* of the synsets.

If *WordNet-Affect* is chosen as lexical resource, the algorithm tries to map the term t_j to one of the nodes of the affective hierarchy. The hierarchy is climbed until a matching is obtained. In that case, the term inherits the sentiment score (extracted from SentiWordNet) of the A-Label it matches. Otherwise, it is ignored.

The determination of the score with *MPQA* and is quite straightforward, since the algorithm first associates the correct POS-tag to the term t_j , then looks for it in the lexicon. If found, the term is assigned with a different score according to its categorical label.

A similar approach is performed for *SenticNet*, since the knowledge-base is queried and the polarity associated to that term is obtained. However, given that SenticNet also models common sense concepts, the algorithm tries to match more complex expressions (as *bigrams* and *trigrams*) before looking for simple unigrams.

4 Experimental Evaluation

In the experimental session we evaluated the effectiveness of the above described lexical resources in the task of sentiment classification of microblog posts. Specifically, we evaluated the accuracy of our lexicon-based approach on varying both the four lexical resources as well as the four versions of the algorithm.

Dataset and Experimental Design: experiments were performed by exploiting SemEval-2013 [10] and Stanford Twitter Sentiment (STS) datasets [5]. SemEval-2013¹ dataset consists of 14,435 Tweets already split in training (8,180 Tweets) and test data (3,255). Tweets have been manually annotated and are classified as *positive*, *neutral* and *negative*. STS dataset contains more than 1,600,000 Tweets, already split in training and test test, but test set is considerably smaller than training (only 359 Tweets). In this case tweets have been collected through Twitter APIs² and automatically labeled according to the emoticons they contained.

Even if our approach can work in a totally unsupervised manner, we used training data to learn positive and negative classification thresholds through a simple Greedy strategy. For SemEval-2013 all the data were used to learn the thresholds, while for STS only 10,000 random tweets were exploited, due to computational issues. As regards the emphasis-based approach, the boosting factor w is set to 1.5 after a rough tuning (the score of adjectives, adverbs and nouns is increased by 50%). As regards the lexical resources, the last versions of MPQA, SentiWordNet and WordNet-Affect were downloaded, while SenticNet

¹ www.cs.york.ac.uk/semeval-2013/task2/

² <https://dev.twitter.com/>

was invoked through the available REST APIs³. Some statistics about the coverage of the lexical resources is provided in Table 1. For POS-tagging of Tweets, we adopted TwitterNLP⁴ [11], a resource specifically developed for POS-tagging of microblog posts. Finally, The effectiveness of the approaches was evaluated by calculating both accuracy and F1-measure [16] on test sets, while stastical significance was assessed through McNemar’s test⁵.

| Lexicon | SemEval-Test | STS-Test |
|------------------------|---------------|--------------|
| <i>Vocabulary Size</i> | <i>18,309</i> | <i>6,711</i> |
| SentiWordNet | 4,314 | 883 |
| WordNet-Affect | 149 | 48 |
| MPQA | 897 | 224 |
| SenticNet | 1,497 | 326 |

Table 1. Statistics about coverage

Discussion of the Results: results of the experiments on SemEval-2013 data are provided in Figure 4. Due to space reasons, we only report *accuracy* scores. Results shows that the best-performing configuration is the one based on *SentiWordNet* which exploits both *emphasis* and *normalization*. By comparing all the variants, it emerges that the introduction of *emphasis* leads to an improvement in 7 out of 8 comparisons (0.4% on average). Differences are statistically significant only by considering the introduction of emphasis on normalized approach with SenticNet ($p < 0.0001$) and SentiWordNet ($p < 0.0008$). On the other side, the introduction of normalization leads to an improvement only in 1 out of 4 comparisons, by using the WordNet-Affect resource ($p < 0.04$). By comparing the effectiveness of the different lexical resources, it emerges that SentiWordNet performs significantly better than both SenticNet and WordNet-Affect ($p < 0.0001$). However, even if the gap with MPQA results quite large (0.7%, from 58.24 to 58.98), the difference is not statistically significant ($p < 0.5$). *To sum up, the analysis performed on SemEval-2013 showed that SentiWordNet and MPQA are the best-performing lexical resources on such data.*

Figure 5 shows the results of the approaches on STS dataset. Due to the small number of Tweets in the test set, results have a smaller statistical significance. In this case, the best-performing lexical resource is SenticNet, which obtained 74.65% of accuracy, greater than those obtained by the other lexical resources. However, the gap is statistically significant only if compared to WordNet-Affect ($p < 0.00001$) and almost significant with respect to MPQA ($p < 0.11$). Finally, even if the gap with SentiWordNet is around 2% (72.42% accuracy), the difference does not seem statistically significant ($p < 0.42$). Differently from SemEval-2013 data, it emerges that the introduction of *emphasis*

³ <http://sentic.net/api/>

⁴ <http://www.ark.cs.cmu.edu/TweetNLP/>

⁵ http://en.wikipedia.org/wiki/McNemar's_test

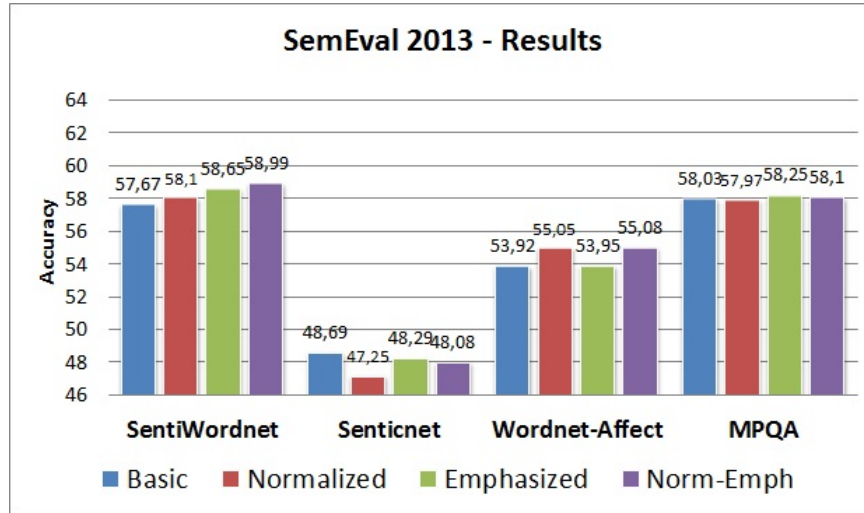


Fig. 4. Results - SemEval 2013 data

leads to an improvement only in 2 comparisons (+0.28% only on MPQA and WordNet-Affect), while in all the other cases no improvement was noted. The introduction of normalization produced a improvement in 3 out of 4 comparisons (average improvement of 0.6%, peak of 1.2% on MPQA). In all these cases, no statistical differences emerged on varying the approaches on the same lexical resource.

5 Conclusions and Future Work

In this paper we provided a thorough comparison of lexicon-based approaches for sentiment classification of microblog posts. Specifically, four widespread lexical resources and four different variants of our algorithm have been evaluated against two state of the art datasets.

Even if the results have been quite controversial, some interesting behavioral patterns were noted: **MPQA** and **SentiWordNet** emerged as the best-performing lexical resources on those data. This is an interesting outcome since even a resource with a smaller coverage as MPQA can produce results which are comparable to a general-purpose lexicon as SentiWordNet. This is probably due to the fact that subjective terms, which MPQA strongly rely on, play a key role for sentiment classification. On the other side, results obtained by **WordNet-Affect** were not good. This is partially due to the very small coverage of the lexicon, but it is likely that the choice of relying sentiment classification only on affective features filters out a lot of relevant terms. Finally, results obtained by **SenticNet** were really interesting since it was the best-performing configuration

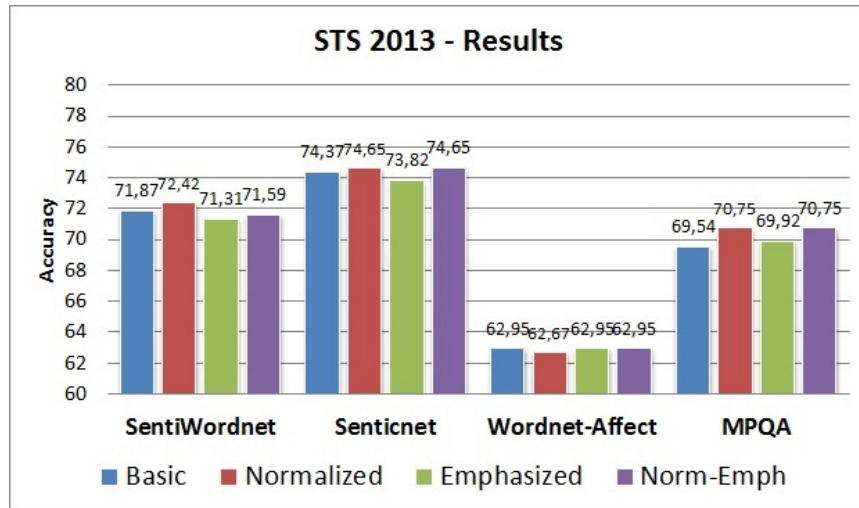


Fig. 5. Results - STS data

on STS and the worst-performing one on SemEval data. Further analysis on the results showed that this behaviour was due to the fact that SenticNet can hardly classify neutral Tweets (only 20% accuracy on that data), and this negatively affected the overall results on a three-class classification task. Further analysis are needed to investigate this behavior.

As future work, we will extend the analysis by evaluating more lexical resources as well as more datasets. Moreover, we will refine our technique for threshold learning and we will try to improve our algorithm by modeling more complex syntactic structures as well as by introducing a word-sense disambiguation strategy to make our approach semantics-aware.

Acknowledgments. This work fulfils the research objectives of the project "VINCENTE - A Virtual collective INTElligenCe ENVIRONMENT to develop sustainable Technology Entrepreneurship ecosystems" funded by the Italian Ministry of University and Research (MIUR)

References

1. Andrea Esuli Baccianella, Stefano and Fabrizio Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of LREC*, volume 10, pages 2200–2204, 2010.
2. Erik Cambria and Amir Hussain. *Sentic computing*. Springer, 2012.
3. Erik Cambria, Daniel Olsher, and Dheeraj Rajagopal. Senticnet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. *AAAI, Quebec City*, pages 1515–1521, 2014.

4. Xiaowen Ding, Bing Liu, and Philip S Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 231–240. ACM, 2008.
5. Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.
6. Xia Hu, Lei Tang, Jiliang Tang, and Huan Liu. Exploiting social relations for sentiment analysis in microblogging. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 537–546. ACM, 2013.
7. Alistair Kennedy and Diana Inkpen. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2):110–125, 2006.
8. Bing Liu and Lei Zhang. A survey of opinion mining and sentiment analysis. In *Mining Text Data*, pages 415–463. Springer, 2012.
9. George A Miller. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
10. Preslav Nakov, Zornitsa Kozareva, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Theresa Wilson. Semeval-2013 task 2: Sentiment analysis in twitter. 2013.
11. Olutobi Owoputi, Brendan O’Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. Improved part-of-speech tagging for online conversational text with word clusters. In *HLT-NAACL*, pages 380–390, 2013.
12. Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*, 2010.
13. Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.
14. Antonio Reyes, Paolo Rosso, and Tony Veale. A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation*, 47(1):239–268, 2013.
15. Sara Rosenthal, Preslav Nakov, Alan Ritter, and Veselin Stoyanov. Semeval-2014 task 9: Sentiment analysis in twitter. *Proc. SemEval*, 2014.
16. Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.
17. Carlo Strapparava and Alessandro Valitutti. Wordnet affect: an affective extension of wordnet. In *LREC*, volume 4, pages 1083–1086, 2004.
18. Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, 2011.
19. Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210, 2005.