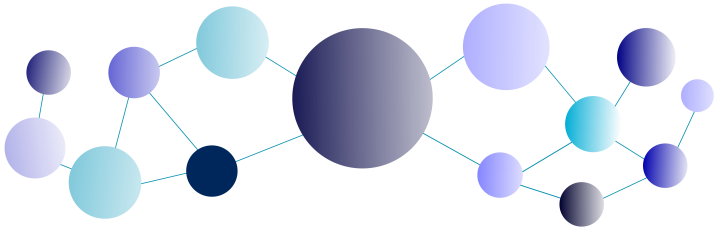
Giuliano Armano, Alessandro Bozzon, Alessandro Giuliani (Eds.)

Proceedings of the 1st International Workshop on Knowledge Discovery on the WEB



KDWEB '15

September 3-5, 2015
Cagliari, Italy
http://www.iascgroup.it/kdweb2015.html

Preface

Nowadays data are continuously created, even if we never notice it is happening. Whenever we sign up for a shopping card, place a purchase using a credit card, or surf the Web, data are created and stored in large sets on powerful computers owned by the companies we deal with every day. With the increasing availability of data, novel tools and systems able to provide effective means of searching and retrieving information are required. Knowledge Discovery is an interdisciplinary area focusing upon methodologies for identifying valid, novel, potentially useful and meaningful patterns from data, often based on underlying large data sets. A major aspect of Knowledge Discovery is Data Mining, the process for discovering valuable knowledge and information from data, is widespread in numerous fields, including science, engineering, healthcare, business, and medicine. In this scenario, Information Retrieval enables the reduction of the so-called "information overload". Information Retrieval tasks are aimed at gathering only relevant information from digital data (e.g., text documents, multimedia files, or webpages), by searching for information within documents and for metadata about documents, as well as searching relational databases and the Web.

Recently, the rapid growth of social networks and online services entailed that Knowledge Discovery approaches focused on the World Wide Web, whose popular use as global information system led to a huge amount of digital data. Hence, there is the need of new techniques and systems able to easily extract information and knowledge from the Web.

Challenges imposed by the large scale of Web Data, Semantic Web, and Linked Data are leading to the adoption of useful tools based on semantic nets, ontologies, or taxonomies. In particular, taxonomies are becoming indispensable to support the mining and retrieval systems, as organizing digital items into hierarchies can help to better understand the information being extracted from data.

KDWeb 2015 is aimed at providing a venue to researchers, scientists, students, and practitioners involved in the fields of Knowledge Discovery on Data Mining, Information Retrieval, and Semantic Web, for presenting and discussing novel and emerging ideas. KDWeb will contribute to discuss and compare suitable novel solutions based on intelligent techniques and applied in real-world applications.

Submitted proposals received three review reports from Program Committee members. Based on the recommendations of the reviewers, 10 full papers and 1 poster paper have been selected for publication and presentation at KDWEB 2015.

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 KDWEB 2015. We hope that you find these proceedings a valuable source of information on intelligent information filtering and retrieval tools, technologies, and applications.

October 2015

GiulianoArmano
(General Chair)

Alessandro Bozzon
(General Chair)

Alessandro Giuliani
(Program Chair)

Organization

General Chairs

- Giuliano Armano (*Department of Electrical and Electronic Engineering, University of Cagliari, Italy)*
- Alessandro Bozzon (*Software and Computer Technology Department, Delft University of Technology, The Netherlands)*

Program Chair

- Alessandro Giuliani (*Department of Electrical and Electronic Engineering, University of Cagliari, Italy)*

Program Committee

- Agapito Ledezma (Universidad Carlos III de Madrid, Spain)
- Antonio Moreno (Universitat Rovira i Virgili, Spain)
- Cataldo Musto (University of Bari, Italy)
- Claudia Hauff (Delft University of Technology, The Netherlands)
- David Sanchez (University Rovira i Virgili, Spain)
- Emanuele Tamponi (University of Cagliari, Italy)
- Flavius Frasincar (Erasmus University Rotterdam, The Netherlands)
- Florian Daniel (University of Trento, Italy)
- Giovanni Semeraro (University of Bari, Italy)
- Gustavo Rossi (LIFIA-F. Informatica. UNLP, Argentina)
- Lorenza Saitta (Universita del Piemonte Orientale, Italy)
- Manuel Wimmer (Vienna University of Technology, Austria)
- Marco Brambilla (Politecnico di Milano, Italy)
- Maria Bielikova Slovak (University of Technology in Bratislava, Slovakia)
- Maristella Matera (Politecnico di Milano, Italy)
- Michal Wozniak (Wroclaw University of Technology, Poland)
- Peter Dolog (Aalborg University, Denmark)
- Schahram Dustdar (TU Wien, Austria)
- Sven Casteleyn (Universitat Jaume I, Spain)

Table of Contents

Harvesting All Matching Information To A Given Query From a Deep Website

Mohammadreza Khelghati[1], Djoerd Hiemstra[1], and Maurice van Keulen[1]
{s.m.khelghati, d.hiemstra, m.vankeulen}@utwente.nl

[1]Databases Group, University of Twente
Enschede, Netherlands

Abstract. In this paper, the goal is harvesting all documents matching a given (entity) query from a deep web source. The objective is to retrieve all information about for instance "Denzel Washington", "Iran Nuclear Deal", or "FC Barcelona" from data hidden behind web forms. Policies of web search engines usually do not allow accessing all of the matching query search results for a given query. They limit the number of returned documents and the number of user requests. In this work, we propose a new approach which automatically collects information related to a given query from a search engine, given the search engine's limitations. The approach minimizes the number of queries that need to be sent by applying information from a large external corpus. The new approach outperforms existing approaches when tested on Google, measuring the total number of unique documents found per query.

1 Introduction

The goal of this research is to harvest all documents matching a given (entity) query from a deep web source. For instance, we aim at retrieving all information about "Denzil Washington", "Iran Nuclear Deal", or "FC Barcelona" from data hidden behind web forms. However, policies of search engines usually do not allow accessing all of the matching query search results for a given query. They limit the number of returned documents (#*ResultsLimited*) and the number of user requests (#*RequestsLimited*).

Given these search engines limitations, we propose a new approach which automatically collects information related to a given query from a search engine. To do so, we rely on search refinement techniques to uncover results beyond what a search engine allows a user to directly access due to #`ResultsLimited` and #`RequestsLimited` limitations. These techniques are typically based on adding extra terms to the initial query to obtain refined search results. We propose an approach which refines search results for the purpose of achieving full data coverage.

In this approach, reformulating queries should be carried out with the aim of obtaining as many new results as possible for each query. Maximizing the number of new results means submitting queries which return as many documents as #`ResultsLimited` limitation allows while minimizing the number of duplicates.

Minimizing duplicates becomes complicated with the presence of *ranking bias* and *query bias* [3]. Search engine's ranking algorithms (e.g. Google Page-rank) and selection of the initial query favour some documents more than others to be returned by the search engine.

To meet this challenge, techniques from *Deep Web harvesting* [1, 12, 10, 4, 2], *Query-Based Sampling* [5, 3], *Topical Crawling*, [14, 16], and *Query Expansion* [6, 13, 7–9] are studied. Based on these studies, several approaches are suggested, implemented, and compared in this paper. We test our approaches on Google, which claims to search 100 PB of Web data (60 trillion URLs)[1]. Google imposes both #`ResultsLimited` and #`RequestsLimited`, and ranking bias through its Page-Rank algorithm.

2 Suggested Approach

To reach this data coverage, we send automatically generated queries to a search engine's API with the goal of retrieving all documents that contain a given entity with a minimum amount of query submissions. We compare the approaches by their capabilities to deal with #`ResultsLimited` and #`RequestsLimited`. The comparison is based on the average number of queries submitted to retrieve all documents for a given query.

We distinguish two kinds of approaches. Section 2.1 describes ideal approaches, for which we estimate the number of queries needed in ideal (simulated) conditions. Section 2.2 describes approaches in which queries are reformulated by using an external corpus.

2.1 Ideal Approaches

The approach mentioned in this section is desirable or perfect but not easily realized. This is investigated with the sole purpose of improving the comparison of the introduced approaches.

Oracle Perfect Approach To achieve a full data coverage on a given entity in a search system with the #`ResultsLimited` and #`RequestsLimited` limitations, the perfect approach is the one which returns not only the maximum possible number of documents but also only unique ones for each request. To have a complete coverage in this situation, it is adequate to send only the $\frac{|CollectionSizeForQuery|}{allowedDocsToBeVisited}$ number of requests. In reality, this is not easily reachable. To do so, you need to know the exact mechanism of search engine ranking algorithm. Then, you might be able to divide the collection into exactly $\frac{||CollectionSizeForQuery||}{allowedDocsToBeVisited}$ sub-collections. In addition to the knowledge of ranking algorithm, you might need additional information. For instance, if a ranking algorithm is based on terms frequencies, you need to know all the term frequencies

[1] Official Google Blog: http://googleblog.blogspot.nl/2008/07/we-knew-web-was-big.html

beforehand. This kind of information is only accessible when you have full index access.

2.2 List-Based Query Generation Approach

In these approaches, the terms to be added to the seed query are selected from a list of words. This list is generated from an external corpus and includes the frequencies in that corpus. In this paper, this list is extracted from the ClueWeb09 dataset, which is a web crawl containing nearly 500 million English pages [15]. Selecting terms from the list of terms and their corresponding document frequencies can be performed in different methods. In the following, these methods are further explained and studied.

List-Based Most/Least Frequent Approach Although primitive, choosing the most or least frequent words from a list are possible options in selecting terms. As the ClueWeb dataset is not a topic-specific corpus, the most frequent words from this corpus are highly probable to be also general in all other not topic-specific corpora.

Pre-determined Frequency Based Approach While submitting the most frequent terms increases the chance of reaching the maximum number of returned results and the least frequent ones increases the probabilities of generating fewer duplicates, it is of a great interest to investigate the likelihood of finding a term frequency which creates a trade-off between these two. To do so, statistical formulas are applicable. If events A and B are independent, then the probability of them both occurring is the product of the probabilities of each occurring ($P(A\&B) = P(A) * P(B)$). With samples smaller than 10 percent of the collection, we can assume two posing query processes as statistically independent events ("The 10% Condition"). Then, the probability of having an overlap between two queries equals with the multiplication of the probability of each query. This is shown in Formula 1.

$$\frac{|MatchingDocs \cap ReturnedDocs|}{|SearchEngine|} = \frac{|MatchingDocs|}{|SearchEngine|} * \frac{|ReturnedDocs|}{|SearchEngine|}$$
$$|ReturnedDocs| = \frac{l * |SearchEngine|}{|MatchingDocs|} \mid (|MatchingDocs \cap ReturnedDocs| = l) \tag{1}$$

With the knowledge of targeted search engine's index size, and also the number of documents matching seed query, through Formula 1, one can determine the frequency of another query for which the overlap of this query and the seed query equals the number of documents allowed to be visited. This means with information on the seed query, returned results and search engine size, a term can be found to formulate a new query returning at least the same number of results that are allowed to be visited. This enables avoiding the permanent

presence of the same highly ranked documents among the results and creates a higher chance in collecting more new documents in each trial. If the size of search engine is unknown, as discussed in [11], the size can be estimated by only using a few number of generated samples from search engine.

As pointed out, applying this formula to our case requires information on terms document frequencies. To access this information from the targeted search system, we should download all its content and count all the terms document frequencies. If this was possible, there was no need for introducing these approaches. Instead, we can use pre-computed terms document frequencies from an external corpus. In this paper, as we test our approaches on Google, we use the ClueWeb dataset. However, the size difference between the ClueWeb and Google should be considered to be able to apply the formula. The easiest solution is to include different sizes in the calculations. For example, assuming $Size^{SearchEngine} = 10^9$, the number of English documents in ClueWeb as 5×10^8, $limitedResults = 100$, and $|MatchingDocuments|$ for a given query to be 4×10^5, the following calculation could provide us with a term document frequency that has higher chance to result in samples of our desired size: $\frac{100}{10^9} = \frac{4 \times 10^5}{10^9} * \frac{x}{5 \times 10^8} \Longrightarrow x = 125000$. In this paper, we refer to this approach as *LB-FixedFreq.* approach.

3 Experiments and Results

3.1 Experiments Settings

Test Search Engine In these experiments, Google as the biggest web search engine with one of the most complicated ranking algorithms is considered as our test search engine. As the only necessary feature for applying any of these suggested approaches is the support of keyword-search interface, targeting Google does not limit our findings only to Google.

Entities Test Set In our experiments, we used four different entities ("Vitol", "Ed Brinksma", "PhD Comics", and "Fireworks Disaster") to test and compare the suggested approaches. We tried to include entities representing different types of entities; Company, Person, Topic, and Event. In addition to difference in type, we tried to cover queries with different estimated results sets sizes.

3.2 Results

In this section, the results of applying the introduced approaches in Section 2 to the test entities (Section 3.1) are presented. The Figure 1 compares the performances of all the approaches for one of the test entities in the test set. This is a straight forward task as it is only required to compare the number of retrieved documents by each approach. However, to compare the approaches' performances on all the test cases, we calculate their average distances from the Oracle Perfect approach. In Figure 3, the performances of all the approaches for all the entities in the test set are compared with the Oracle Perfect approach.

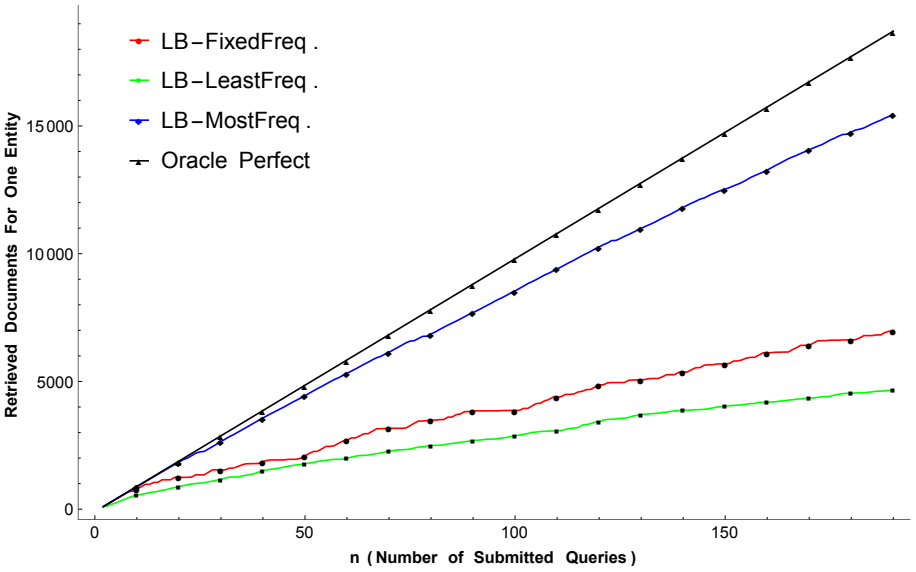


Fig. 1. Average Performance For All One Entity

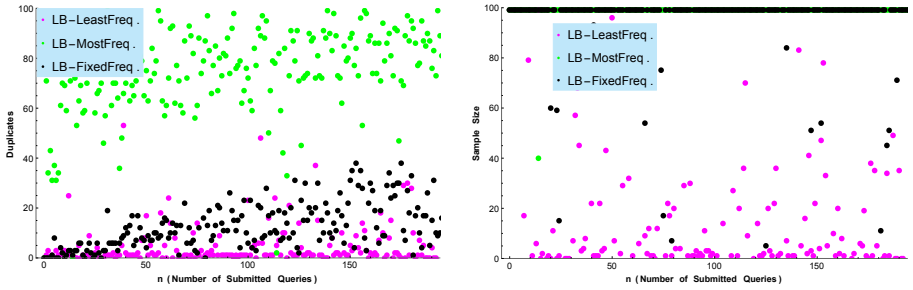


Fig. 2. Sample Sizes and Duplicates For Approaches For One Entity

As it is shown in Fig 3, the LB-FixedFreq. approach performs better than Most-freq. and Least-Freq. approaches. This approach submits queries which result in fewer duplicates than LB-MostFreq. approach while having bigger sample sizes in regards to the Least-Freq approach. This is observable from Figure 2. The right image in this figure shows the number of duplicates resulted from submitting all the queries formulated by adding a term to the initial query (given entity). The left picture shows the corresponding sample size for each of these queries. From comparing these two images, we can conclude that a trade-off between the big sample sizes and number of duplicates is the key to the LB-MostFreq. approach's better performance. In this approach, finding a specific frequency leads to a trade-off between sample sizes and number of duplicates.

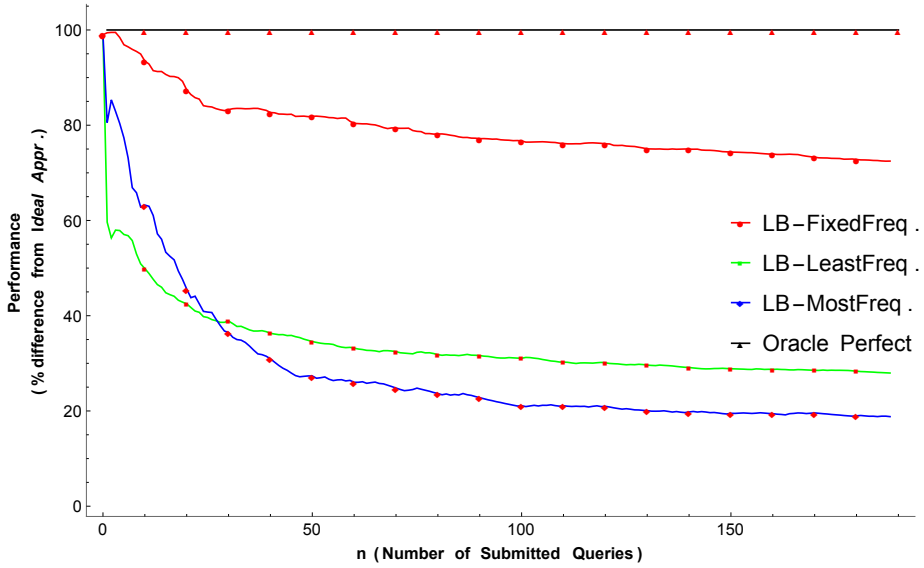


Fig. 3. Average Performance For All The Entities

4 Conclusion and Future Work

In this work, we assessed different query generation mechanisms for harvesting a web data source to move forward towards achieving a full data coverage on a given (entity) query. From the experiments, we found that the key to success in these approaches is to send queries which result in the maximum possible number of results with the minimum possible number of documents downloaded in previous query submissions. To have this success factor, we suggested three different approaches based on different frequencies. Among these approaches, the LB-FixedFreq. performed better than the others.

Future Work In addition to the frequency of terms extracted from an external corpus, we can include terms present in the previously retrieved documents to select the best next query to submit. The frequency of these terms could also be applied for a more efficient query expansion technique.

References

1. Manuel Álvarez, Juan Raposo, Alberto Pan, Fidel Cacheda, Fernando Bellas, and Víctor Carneiro. Deepbot: a focused crawler for accessing hidden web content. In *Proceedings of the 3rd international workshop on Data enginering issues in E-commerce and services: In conjunction with ACM Conference on Electronic Commerce (EC '07)*, DEECS '07, pages 18–25, New York, NY, USA, 2007. ACM.
2. Luciano Barbosa and Juliana Freire. Siphoning hidden-web data through keyword-based interfaces. In *SBBD*, pages 309–321, 2004.

3. Krishna Bharat and Andrei Broder. A technique for measuring the relative size and overlap of public web search engines. *Comput. Netw. ISDN Syst.*, 30:379–388, April 1998.
4. Michael Cafarella. Extracting and Querying a Comprehensive Web Database. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, 2009.
5. James P. Callan and Margaret E. Connell. Query-based sampling of text databases. *ACM Trans. Inf. Syst.*, 19(2):97–130, 2001.
6. Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 243–250, New York, NY, USA, 2008. ACM.
7. Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50, January 2012.
8. Kevyn Collins-Thompson and Jamie Callan. Estimation and use of uncertainty in pseudo-relevance feedback. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 303–310, New York, NY, USA, 2007. ACM.
9. Ben He and Iadh Ounis. Combining fields for query expansion and adaptive query expansion. *Inf. Process. Manage.*, 43(5):1294–1307, September 2007.
10. Yeye He, Dong Xin, Venkatesh Ganti, Sriram Rajaraman, and Nirav Shah. Crawling deep web entity pages. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 355–364, New York, NY, USA, 2013. ACM.
11. Mohammadreza Khelghati, Djoerd Hiemstra, and Maurice van Keulen. Size estimation of non-cooperative data collections. IIWAS '12, pages 239–246, New York, NY, USA, 2012. ACM.
12. Jayant Madhavan, David Ko, Lucja Kot, Vignesh Ganapathy, Alex Rasmussen, and Alon Halevy. Google's Deep Web crawl. *Proc. VLDB Endow.*, 1(2):1241–1252, August 2008.
13. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.
14. Filippo Menczer, Gautam Pant, and Padmini Srinivasan. Topical web crawlers: Evaluating adaptive algorithms. *ACM Transactions on Internet Technology*, 4:http://dollar.biz.ui, 2004.
15. The Lemur Project. A dataset to support research on information retrieval and related human language technologies. http://lemurproject.org/clueweb09.php, 2014.
16. Sergej Sizov, Martin Theobald, Stefan Siersdorfer, Gerhard Weikum, Jens Graupmann, Michael Biwer, and Patrick Zimmer. The bingo! system for information portal generation and expert web search. In *CIDR*, 2003.

Design Criteria to Model Groups in Big Data Scenarios: Algorithms and Best Practices*

Ludovico Boratto, Gianni Fenu, and Pier Luigi Pau

Dipartimento di Matematica e Informatica,
Università di Cagliari, Via Ospedale 72 - 09124 Cagliari, Italy
{ludovico.boratto,fenu,pierluigipau}@unica.it

Abstract. There are different types of information systems, such as those that perform *group recommendations* and *market segmentations*, which operate with groups of users. In order to combine the individual preferences and properly address suggestions to users, *group modeling* strategies are employed. Nowadays, data is characterized by large amounts in terms of volume, speed, and variety (the so-called *big data* issue). In this paper, we are going to tackle the problem of modeling group preferences in big data scenarios. This study will present the existing strategies, and we are going to present criteria to design the algorithms that implement them when big amounts of data have to be combined. Moreover, a set of best practices discusses under which conditions the presented strategies can be adopted in big data scenarios.

Keywords: Group Modeling, Big Data, Algorithms, Design.

1 Introduction

Combining the preferences of individual users is a central problem for the information systems that operate with groups. The most challenging and widely studied, both by the industry and the academia, are the *group recommender* [1, 2] and *market segmentation* [3, 4] systems, which aggregate information about large groups of users and tens of items in order to filter the data and produce suggestions for the users in terms of items or ads. Therefore, nowadays these systems have to deal with big data and to be able to filter large amounts of information.

The task of aggregating the individual preferences into a single model is known as *group modeling*, and several strategies have been studied in the literature [5]. It is known that no strategy is better than another and that the

* This work is partially funded by Regione Sardegna under project SocialGlue, through PIA - Pacchetti Integrati di Agevolazione "Industria Artigianato e Servizi" (annualità 2010), and by MIUR PRIN 2010-11 under project "Security Horizons". Pier Luigi Pau gratefully acknowledges Sardinia Regional Government for the financial support of his PhD scholarship (P.O.R. Sardegna F.S.E. Operational Programme of the Autonomous Region of Sardinia, European Social Fund 2007-2013 - Axis IV Human Resources, Objective l.3, Line of Activity l.3.1.).

group modeling strategy adopted by an information system should be chosen after a deep analysis of the application domain in which the groups have to be modeled [6].

In this paper, we tackle the novel problem of *studying criteria for applicable and efficient design of group modeling algorithms in big data scenarios*. More specifically, we are going to answer the following research question: *which group modeling strategies can actually be employed in real-world contexts characterized by big data?* In order to answer this question, we first present the existing group modeling strategies (Section 2), then we propose design guidelines to efficiently implement these strategies in big data scenarios, and discuss with a set of best practices which strategies are applicable in real-world big data contexts (Section 3). Our aim is to guide future research in this area towards the development of approaches that are efficient and effective at the same time. This study is concluded with a summary of the proposed criteria and with perspectives for future work in this research area (Section 4).

2 Background and Related Work

Group modeling [5] is the process adopted to combine multiple user models into a single model. In this section, we are going to present the modeling strategies that have been employed in the literature. In order to facilitate their understanding, an example of the results produced by the strategies is given as a reference, then we present each of them.

2.1 Group Modeling: Working Examples.

Here, we present an example of how each group modeling strategy operates. We consider three users (denoted as u_1, u_2, and u_3), who rate ten items (i_1, ..., i_{10}) with a rating from 1 to 10. Table 1 reports the output of the strategies that combine individual ratings, while tables 2, 3, and 4, show how the *Borda Count*, *Copeland Rule*, and *Plurality Voting* strategies respectively work (these tables are based on the ratings in Table 1).

2.2 Additive Utilitarian [AU]

The individual ratings for each item are summed and a list of items ranked by sum is created. The list produced by each strategy is the same that would be generated when averaging the individual ratings, so it is also called 'Average strategy'. An example of how the strategy works is given in Table 1 (*AU* line).

The strategy has proven to be effective in different contexts [7], like the combination of preferences on different types of features (e.g., location, cost, cuisine) when recommending restaurants to a group [8].

2.3 Multiplicative Utilitarian [MU]

The ratings given by the users for each item are multiplied and a ranked list of items is produced. An example of how the strategy works is given in Table 1 (MU line).

This strategy was employed in the music recommendation domain by [9].

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	8	10	7	10	9	8	10	6	3	6
u_2	7	10	6	9	8	10	9	4	4	7
u_3	5	1	8	6	9	10	3	5	7	10
$-AU$	20	21	21	25	26	28	22	15	14	23
$-MU$	280	100	336	540	648	800	270	120	84	420
$-AV$	2	2	3	3	3	3	2	1	1	3
$-LM$	5	1	6	6	8	8	3	4	3	6
$-MP$	8	10	8	10	9	10	10	6	7	10
$-AWM$	20	-	21	25	26	28	-	15	-	23
$-MRP$	8	10	7	10	9	8	10	6	3	6

Table 1. Output of the strategies that combine the original ratings

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	4.5	8	3	8	6	4.5	8	1.5	0	1.5
u_2	3.5	7.5	2	6.5	5	7.5	6.5	0.5	0.5	3.5
u_3	2.5	0	5	3	6	7.5	1	2.5	4	7.5
$-BC$	10.5	15.5	10	17	17	19.5	15.5	4.5	4.5	12.5

Table 2. Example of how the *Borda Count* strategy works, based on the ratings in Table 1

2.4 Borda Count [BC]

The strategy assigns to an item a number of points, according to the position in the list of each user. The least favorite one gets 0 points and a point is added each time the next item in the list is considered. If a user gave the same rating to more than one item, the points are distributed. Considering the example in Table 2, items i_8 and i_9 were rated by user u_2 with the lowest rating and share the lowest positions with 0 and 1 points, by getting (0+1)/2=0.5 points. A group preference is obtained by adding the individual points of an item.

This strategy was implemented in [10].

2.5 Copeland Rule [CR]

It is a form of majority voting that sorts the items according to their *Copeland index*, which is calculated as the number of times in which an alternative beats

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
i_1	0	+	-	+	+	+	+	-	-	0
i_2	-	0	-	0	-	0	0	-	-	-
i_3	+	+	0	+	+	+	+	-	-	+
i_4	-	0	-	0	-	+	-	-	-	-
i_5	-	+	-	+	0	+	+	-	-	-
i_6	-	0	-	-	-	0	-	-	-	-
i_7	-	0	-	+	-	+	0	-	-	-
i_8	+	+	+	+	+	+	+	0	0	+
i_9	+	+	+	+	+	+	+	0	0	+
i_{10}	0	+	+	+	+	+	+	-	-	0
Index	-2	+6	-3	+6	+1	+8	+4	-8	-8	-2

Table 3. Example of how the *Copeland Rule* strategy works, based on the ratings in Table 1

	1	2	3	4	5	6
u_1	i_2, i_4, i_7	i_4, i_7	i_5	i_1	i_3	i_8
u_2	i_2, i_6	i_4, i_7	i_5	i_1, i_{10}	i_3	i_8, i_9
u_3	i_6, i_{10}	i_{10}	i_{10}	i_{10}	i_3	i_9
Group	i_2, i_6	i_4, i_7	i_5	i_1, i_{10}	i_3	i_8, i_9

Table 4. Example of how the *Plurality Voting* strategy works, based on the ratings in Table 1

the others, minus the number of times it loses against the other alternatives. In the example in Table 3, item i_2 beats item i_1, since it received a higher rating by both users u_1 and u_2.

The approach proposed in [11] proved that a form of majority voting is the most successful in a *requirements negotiation* context.

2.6 Plurality Voting [PV]

Each user votes for her/his favorite option. The one that receives the highest number of votes wins. If more than one alternative needs to be selected, the options that received the highest number of votes are selected. An example of how the strategy works is given in Table 4.

This strategy was implemented and tested by [12, 13] in the TV domain.

2.7 Approval Voting [AV]

Each user votes for as many items as she/he wants, and a point is assigned to all the ones a user likes. To show how the strategy works, in the example in Table 1 (*AV* line) we suppose that each user votes for all the items with a rating above a threshold (for example, 5). A group preference is obtained by adding the individual points of an item.

To choose the pages to recommend to a group, *Let's Browse* [14] evaluates if the page currently considered by the system matches with the user profile above a certain threshold and recommends the one with the highest score. This strategy also proved to be successful in contexts in which the similarity between the users in a group is high [15].

2.8 Least Misery [LM]

The group rating produced for an item is the lowest rating expressed for that item by any of the users in the group. This strategy usually models small groups, to make sure that every member is satisfied. A drawback is that if the majority of the group really likes something, but one person does not, the item will not to be recommended to the group. This is what happens in Table 1 for the items i_2 and i_7. An example of how the strategy works is given in Table 1 (LM line).

This strategy is used by [16], to recommend movies to small groups.

2.9 Most Pleasure [MP]

The rating assigned to an item for a group is the highest one expressed for that item by a member of a group. An example of how the strategy works is given in Table 1 (MP line).

This strategy is used by [17] in a system that faces the cold start problem.

2.10 Average Without Misery [AWM]

The rating assigned to an item for a group is the average of the ratings given by each user. All the items that received a rating under a certain threshold by a user are not included in the group model (in the example in Table 1 - AWM line, the threshold rating is 4).

In order to model a group to decide the music to play in a gym, in [18] the individual ratings are summed, discarding the ones under a minimum degree.

2.11 Fairness [F]

This strategy is based on the idea that users can be recommended something they do not like, as long as they also get recommended something they like. This is done by allowing each user to choose her/his favorite item. If two items have the same rating, the choice is based on the other users' preferences. This is done until everyone made a choice. Next, everyone chooses a second item, starting from the person who chose last the first time.

If in the example in Table 1, we suppose that user u_1 chose first, she/he would consider i_2, i_4, and i_7, and would choose i_4, because it has the highest average considering the other users' ratings. Next, u_2 would choose between i_2 and i_6 and would select i_6 for the same reason. Then, u_3 would choose item i_{10}. Since everyone chose an item, it would be u_3's turn again and i_5 would be chosen. User u_2 would choose i_2, which has the highest rating along with i_6 (which was already chosen). Then, u_1 would choose i_7, which is the one with the highest rating and was not chosen yet. The final sequence of items that models the group would be: i_4, i_6, i_{10}, i_5, i_2, i_7, i_1, i_3, i_9, i_8.

This strategy is adopted by [9] in the music recommendation context.

2.12 Most Respected Person (Dictatorship) [MRP]

This strategy selects the items according to the preferences of the most respected person, using the preferences of the others just in case more than one item received the same evaluation. The idea is that there are scenarios is which a group is guided/dominated by a person. In the example in Table 1, u_1 is the most respected person.

This strategy is adopted to select tourist attractions advantaging the users with particular needs [19], or when experts are recognized in a group [20]. Moreover, there are studies that highlight that when people interact, a user or a small portion of the group influences the choices of the others [21].

3 Criteria for Applicable Design of Group Modeling Algorithms in Big Data Scenarios

This section presents design criteria to implement the strategies presented in the previous section in scenarios characterized by big data. In Section 3.1, we are going to present design criteria from an algorithmic point of view, while in Section 3.2 we are going to study the nature of each strategy to evaluate their applicability in real-world scenarios characterized by big data.

3.1 Algorithms Design

Each strategy is fairly trivial to implement in an efficient way, by adopting data structures that can be quickly accessed. A possible way to implement a group model might be a *hash table* that stores the item ids as keys and the group rating as values. Each time a new individual rating arrives, the hash table can be efficiently updated with average complexity $O(1)$.

In order to suggest the items to the users, the items in the group model have to be sorted by group rating. An efficient sorting algorithm for big data, known as *two-way replacement selection*, has been proposed in [22]. The algorithm presents a variant of the Merge Sort algorithm, specifically designed for big data scenarios, and it currently represents the state of the art.

3.2 Applicability in Big Data Scenarios: Best Practices

Here, we present a set of best practices related to the applicability of the previously presented group modeling strategies in big data scenarios. Most of these best practices are derived from a case-study conducted in the group recommendation domain and presented in [23], and from considerations on the aspects that characterize big data scenarios.

The main argument against the deployment of a strategy in a big data scenario will be represented by a high computational cost of performing inserts and updates of ratings in large sets of data. More precisely, it is assumed that group ratings, resulting from the application of a specific strategy on a set of

user ratings, are stored for later use in order to save computation power, and that a recalculation of group ratings is required following the insertion or update of user ratings. This evaluation takes into account the computational cost of updating group ratings. Furthermore, for the sake of completeness, it will also be noted when a strategy is simply inadequate for modeling large groups of users, regardless of any difficulties in handling large amounts of data.

The *Additive Utilitarian* strategy can be easily employed in big data scenarios, as the only operation required to update the group model is to add the rating expressed by a user for an item to the existing group rating for that item.

Likewise, the *Multiplicative Utilitarian* would be very simple to implement. However, it would not be advisable to employ it in presence of large groups, as an overflow would most certainly occur when the original user ratings are multiplied[1]. Moreover, given the large amounts of operations that the strategy would perform for such a group, rating normalization to avoid the problem would lead to a loss in precision and to a drop in the accuracy of the system.

Borda Count, *Copeland Rule*, *Plurality Voting*, and *Fairness* require to update the individual model of each user each time she/he assigns a new rating. After the individual model is updated, the group model can be updated accordingly. Therefore, these strategies would not be efficiently applicable in big data scenarios.

The *Average Without Misery*, *Approval Voting*, *Least Misery*, and *Most Pleasure* strategies, can apparently be easily and efficiently adapted in big data scenarios, as they only require to calculate an average, the maximum, or the minimum of the individual ratings given to each item. However, they should not be adopted in big data due to their nature. Indeed, *Average Without Misery* discards a group rating if at least a user has given to an item a rating lower than the considered threshold. Therefore, even with a small threshold value, like 2, the vast majority of the items would not be modeled by the strategy in a context in which groups are large (the larger is the group, the higher is the number of times an item is rated, and higher is the probability that at least one user did not like the item). Considering the *Approval Voting* strategy with high threshold values (for example, 5), too many ratings would be discarded by the model because only the items with a high rating (i.e., with a rating above 5), would be considered. The other two strategies (i.e., *Least Misery* and *Most Pleasure*) are usually employed to model small groups; indeed, if a group is large, the group model would contain respectively only low or high ratings, which would not represent the preferences of the group as a whole.

Lastly, in case a person that guides the group or whose preferences align with most of the group exists, the *Most Respected Person* strategy would be at the same time effective and efficient to employ.

[1] Given 60 users who expressed a very low rating (like 2) for an item, a 64 bit machine would not be able to handle the group rating, since it cannot process numbers higher that 2^{52}.

4 Conclusions

In this paper, we analyzed the existing group modeling strategies and presented criteria for applicable design in real-world scenarios characterized by big data. As a result of this study, we can say that the vast majority of the strategies do not present limitations from an algorithmic point of view and could be efficiently implemented. However, due to how the strategies operate, their effectiveness is affected in big data scenarios. Indeed, some of them do not consider a group rating if a user or a part of the group has given a low rating to an item, or some others would lead the group model to be composed just with low or high ratings, blurring the knowledge on what the users in the group like or do not like. In conclusion, in presence of big data, a simple but very effective strategy, like *Additive Utilitarian*, which considers all the users and all the ratings given by them, should be preferred. Future work will be devoted at experimenting these strategies in the real-world scenarios characterized by big data to analyze their applicability and validate these design criteria.

References

1. Boratto, L., Carta, S.: State-of-the-art in group recommendation and new approaches for automatic identification of groups. In: Information Retrieval and Mining in Distributed Environments. Volume 324 of Studies in Computational Intelligence. Springer Berlin Heidelberg (2011) 1–20
2. Jameson, A., Smyth, B.: Recommendation to groups. In: The Adaptive Web, Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer, Berlin (2007) 596–627
3. Yankelovich, D., Meer, D.: Rediscovering market segmentation. Harvard Business Review **84**(2) (2006) 1–10
4. Liu, Y., Kiang, M., Brusco, M.: A unified framework for market segmentation and its applications. Expert Syst. Appl. **39**(11) (September 2012) 10292–10302
5. Masthoff, J.: Group recommender systems: Combining individual models. In: Recommender Systems Handbook. Springer (2011) 677–702
6. Pizzutilo, S., Carolis, B.D., Cozzolongo, G., Ambruoso, F.: Group modeling in a public space: Methods, techniques and experiences. In: Proceedings of WSEAS AIC 05, Malta, ACM (2005)
7. Pessemier, T., Dooms, S., Martens, L.: Comparison of group recommendation algorithms. Multimedia Tools and Applications (2013) 1–45
8. McCarthy, J.F.: Pocket restaurantfinder: A situated recommender system for groups. In: Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems, Minneapolis (2002)
9. Christensen, I.A., Schiaffino, S.N.: Entertainment recommender systems for group of users. Expert Systems with Applications **38**(11) (2011) 14127–14135
10. Baltrunas, L., Makcinskas, T., Ricci, F.: Group recommendations with rank aggregation and collaborative filtering. In: Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, New York, NY, USA, ACM (2010) 119–126
11. Felfernig, A., Zehentner, C., Ninaus, G., Grabner, H., Maalej, W., Pagano, D., Weninger, L., Reinfrank, F.: Group decision support for requirements negotiation. In: Advances in User Modeling - UMAP 2011 Workshops, Revised Selected Papers. Volume 7138 of Lecture Notes in Computer Science., Springer (2012) 105–116

12. Senot, C., Kostadinov, D., Bouzid, M., Picault, J., Aghasaryan, A.: Evaluation of group profiling strategies. In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI/AAAI (2011) 2728–2733
13. Senot, C., Kostadinov, D., Bouzid, M., Picault, J., Aghasaryan, A., Bernier, C.: Analysis of strategies for building group profiles. In: User Modeling, Adaptation, and Personalization, 18th International Conference, UMAP 2010. Proceedings. Volume 6075 of Lecture Notes in Computer Science., Springer (2010) 40–51
14. Lieberman, H., Dyke, N.W.V., Vivacqua, A.S.: Let's browse: A collaborative web browsing agent. In: IUI. (1999) 65–68
15. Bourke, S., McCarthy, K., Smyth, B.: Using social ties in group recommendation. In: AICS 2011: Proceedings of the 22nd Irish Conference on Artificial Intelligence and Cognitive Science: 31 August-2 September, 2011: University of Ulster-Magee, Intelligent Systems Research Centre (2011)
16. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: Polylens: A recommender system for groups of users. In: Proceedings of the Seventh European Conference on Computer Supported Cooperative Work, Kluwer (2001) 199–218
17. Sánchez, L.Q., Bridge, D.G., Díaz-Agudo, B., Recio-García, J.A.: A case-based solution to the cold-start problem in group recommenders. In: Case-Based Reasoning Research and Development - 20th International Conference, ICCBR 2012. Proceedings. Volume 7466 of Lecture Notes in Computer Science., Springer (2012) 342–356
18. McCarthy, J.F., Anagnost, T.D.: Musicfx: An arbiter of group preferences for computer supported collaborative workouts. In: CSCW '98, Proceedings of the ACM 1998 Conference on Computer Supported Cooperative Work, ACM (1998) 363–372
19. Ardissono, L., Goy, A., Petrone, G., Segnan, M., Torasso, P.: Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices. Applied Artificial Intelligence **17**(8-9) (2003) 687–714
20. Jung, J.J.: Attribute selection-based recommendation framework for short-head user group: An empirical study by movielens and imdb. Expert Systems with Applications **39**(4) (March 2012) 4049–4054
21. Carolis, B.D., Pizzutilo, S.: Providing relevant background information in smart environments. In: E-Commerce and Web Technologies, 10th International Conference, EC-Web 2009. Proceedings. Volume 5692 of Lecture Notes in Computer Science., Springer (2009) 360–371
22. Martinez-Palau, X., Dominguez-Sal, D., Larriba-Pey, J.L.: Two-way replacement selection. Proc. VLDB Endow. **3**(1-2) (September 2010) 871–881
23. Boratto, L., Carta, S.: Modeling the preferences of a group of users detected by clustering: A group recommendation case-study. In: Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14). WIMS '14, New York, NY, USA, ACM (2014) 16:1–16:7

Elaboration of an artificial model for filtering of spam based on Human Renal Function

Reda Mohamed HAMOU [1], Mohamed Amine BOUDIA [3] , Abdelmalek AMINE [3]

Dr. Moulay Tahar University SAÏDA
Department of Computer Saida, Algeria
Laboratory Knowledge Management and Complex Data (GeCoDe Lab)

{ mamiamounti 1, hamoureda 2, abd_amine1 3}@yahoo.fr

Abstract. In today's world of globalization and borderless technology, the explosion in communication has revolutionized the field of electronic communication. The e-mail is therefore one of the most used methods for its efficiency and profitability. In the last few years, the undesirables emails (SPAM) are widely spread as they play an important part in the inbox. so that such emails must be filtered and separated from those non-SPAMS (HAMS). Consequently, several recent studies have provided evidence of the importance of detection and filtering of SPAM as a major interest for the Internet community.
In the present paper, we propose a meta-heuristic based on the renal system for detection and filtering spam. The natural model of the renal system is taken as an inspiration for its purification of blood, the filtering of toxins as well as the regularization of the blood pressure.The message are represented by both a bag words and N-Gram method method which is independent of languages because an email can be received in any language

Keywords: SPAM Detection, SPAM filtering, F-Measure, Recall, Precision, Renal System, Nephron, Bowman's capsule, Glomerulus, afferent artery, efferent artery, blood flow , primitive urine , kidney, reabsorption, secretion, Malpighian pyramids , Loop of Henle, proximal tubule, distal tubule, ADH stimulus, ADH inhibition.

1 Introduction and problematic

The appearance of the Internet and the incredibly rapid development of telecommunication technology have made the world a global village. The Internet has become a major channel for communication. Email is one among the tools for communication that Internet users take advantage of as it is available free of charge and supplies the transfer of files.

According to the most recent report of the Radicati Group (2013), who supplies quantitative and qualitative researches with details on the e-mail, the security, the Instant messaging (IM), the social networks, the archiving of the data, the regulatory

compliance, the wireless technologies, the Web's technologies and the unified communications, there was exactlly:

- 2.9 trillion of active emails accounts in the world.
- 2.4 Billion people who use e-mails regularly and they will be 3 % more by year from 2013 till 2017 to be exceeding 2.7 billion people.
- 67 trillion is the number of e-mails that are sent to by the year 2013,that is to say, 182.9 billion every day in the world on average. This number will increase to 206.6 billion in 2017.
- 1,6 is the number of accounts detained by each person and which should increase to 1,8 in four years.

According to the same reports of the Radicati Group,unsolicited mail, or SPAM, can reach more than 89,1 %; 262 million SPAMS a day. In 2009, about 81 % of the sent emails were SPAM. Consequently, spamming became a global phenomenon. For the CNIL(the National Commission for Computing and Liberties), " the "SPAMMING" or" SPAM " is to send massive and sometimes repeated electronic mail, not requested, to people with whom the sender has had no contact and whose he has captured the email address in an irregular way. "

From the above statistics, the detection and filtering of spam is a major stake to the Internet community making the detection and filtering of spam a crucial task.

The literature gives two broad approaches for the filtering and the detection of SPAM: The approach based on the machine learning and the approach not based on the machine learning. The first approach is based on feature selection which is an important stage in the systems of classification. It aims to reduce the number of features while trying to preserve or improve the performance of the used classifier. On the other hand, the second approach (not based on the machine learning) is based on many existing techniques and algorithms: content analysis, the block lists, black lists and white lists, the authentication of mailbox and the heuristics and finally meta-heuristics.

Even though it is usually easy to decide whether it is a spam / non-spam" by human, we can't tackle SPAMS by manual sorting of email because the number of emails in circulation which we have just quoted is extremely large.

In the human body, an important process for the survival occurs automatically, which is the purification of the blood by the renal system. The human can die if the rate of toxins and unwanted substances found in the blood exceeds some threshold; the renal system purifies and filters the blood in automatic manner and a delicate and precise way. The blood pressure regulation is another role of the renal system.

We propose a method inspired from the renal system for the detection and the filtering of the SPAM with a hybridization of both approaches (based and not based on the machine learning). Further, several techniques in the same system of filtering of SPAM are used including: content analysis, the blacklists, the white lists. Another part of our approach controls the flow of the emails which represents one of the roles of the renal system (the blood pressure regulation) to minimize the risk of DDoS attacks (denial of service attack).

Our approach is a combination of different positive properties of these techniques of filtering at various levels by deploying them in a hybrid approach. This study thus seeks to answer the following research questions:

- Does the meta-heuristic based on the renal system assure more results and protection?
- Does the hybridization of the approaches and the use of several techniques improve the quality of the result of the system of filtering of SPAM?

2 Our proposed approach

Our work aims at modelling a method bio-inspired which is the renal System to problems in computer science, in this case the detection and the filtering of the SPAM. Before explaining and detailing our approach we must describe at first the natural model of the functioning of the renal system and shed light on the aspects which directed us to choose this metaheuristics for our problem which is the detection and the filtering of the SPAM. Then we draw up a table of modelling (the natural model vs the artificial model). Finally we shall explain the artificial model which is the pulp of our approach.

2.1 Natural Model

Why the renal system for filtering of SPAM?.

The role of the renal system is to cleanse, purify the blood and adjust the tension. We have chosen modeling the renal system in order to filter SPAM after the overlap which is explained in details in the following table:

	The renal system	The filtering of SPAM
INPUT	Blood	Text (Messages or email)
Result Or OUTPUT	two outputs: • Purified blood which comes back to the bloodstream • Urine which moves towards the bladder and then outside the body The regularization of the blood pressure is another aspect of the renal system.	two classes: • HAM • SPAM
Type of process	Automatic and continuous in time	Automatic and continuous in time
Fault tolerance	A human subject can live with only one kidney.	The filtering of SPAM must tolerate the break-downs and bugs.

Table 1. Overlapping between the renal system and the filtering of SPAM

Obviously, this overlapping gives a preliminary idea onto the feasibility of this modelling. Another argument that we have taken into consideration is that the functioning of the renal system is automatic (independent of the brain) and very precise as any error in the functioning of the renal system can be fatal or seriously pathological for the human subject.

Functioning of the Renal System.

The Nephrons are supplied by two capillary systems:

- The glomerulus where a glomerular filtration occurs to produce the primitive urine.
- The peritubular capillary network where the processes of reabsorption and of secretion it produce, once these two processes finished one will have the definitive urine.

The functioning of renal System is divided into two stages:

Step1 : the glomerular filtration.

The blood comes from the afferent arteriole and enters the glomerular room (glomerulus + Bowman capsule) to undergo the glomerular filtration.

Glomerular filtration is a nonselective, passive mechanical process: it does not consume energy; the blood pressure in the capillary the glomerular represents the dynamic element of the filtration. It is not selective because any molecule which has a size smaller than the Bowman's capsule hole will be filtered. The glomerular filtration stops when the blood pressure falls below 60 mm Hg.

Once the step was finished, the primitive urine follows the path of renal tubule where its composition will be modified during the second step; filtered blood primitively joined the efferent arteriole, the efferent arteriole will bypass the renal tubule forming the peritubular capillary network.

Step 2: renal tubular transfer.

The composition of the primitive urine produced by glomerular filtration will be modified in the renal tubule by two processes which they happen in parallel:

- Reabsorption: which consists of the transfer of certain constituents of the primitive urine to the peritubular capillary i.e. to the blood (e.g. water, mineral salts, glucose ...)
- Secretion: the toxic or exogenous substances that escaped glomerular filtration are added to the tubular urine

At the connecting tubule (CNT) where will be made the control the volume and acidity of the urine by the ADH stimulus (to make the second reabsorption of water and acidified urine) or ADH inhibition (to dilute the urine and adjustment of the balance of fluid in the blood).

At the end of the process we shall have: some definitive urine which is going to be driven by ureters towards the bladder then towards the outside of the body; and the

cleansed Blood which is going to join the general blood circulation (the tip of the peritubular capillary network joined the interlobular vein). Noting that the blood pressure will be stabilized in the normal upstream.

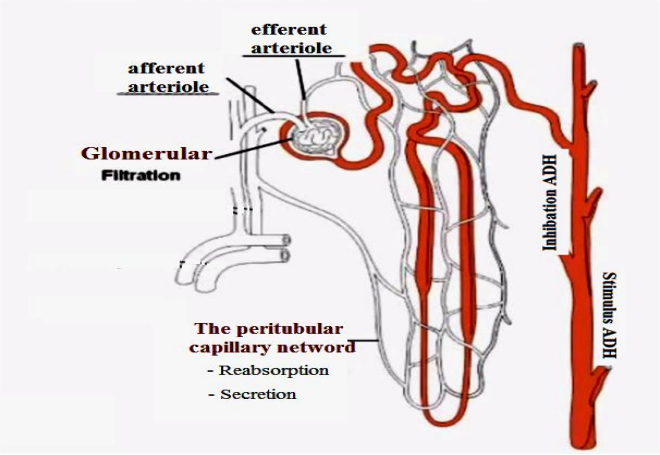


Fig. 1. Functioning of the nephron

Some specific property of the Renal System.

Before proceeding to the artificial model, we must shed light on some property of the functioning of the renal system:

- The glomerular filtration gives a quantity of the primitive urine from 150 to 180 liters per day
- The renal tubular transfer gives a quantity of the final urine from 1.5 to 2.4 liter per day
- The process of the reabsorption is more important qualitatively than that of the secretion
- Renal blood flow is equal to 20% of cardiac output i.e. 1.1 to 1.2 liters of blood per minute
- Kidney consumes 10 to 15% O^2 that it was brought by the Juxtamedullary Nephrons
- the total length of Renal tubule is equal to 115 Km
- The diameter of an afferent artery is greater than that of the efferent arteriole: this serves to:
  - Maintain blood pressure in the glomerulus (for continuity of functioning)
  - Regularization of the blood pressure

2.2 Modeling table: the transition from natural model of renal system to artificial model of renal system for filtering of spam

	natural model of renal system	artificial model of renal system for filtering of spam
New meta-heuristic based on the renal system for the filtering of	Renal consumption = **10 to 15%** of O^2	Training set = **10 to 15%** of SpamBase
	Nephron (several nephrons)	Filtering agent
	Bowman's capsule (**filter diameter**)	artificial Bowman's capsule (**the artificial filtering glomerular threshold**)
	the blood coming in Afferent arteriole is **around 20%** of the cardiac output divised by number of nephron **every minute**	Message entering to be processed is **equal to 20 %** of the total number of messages divised by the number of filtering agent, **Every iteration**
	The glomerular filtration	The artificial glomerular filtration
	Blood in efferent arteriole (before the tubular transfer)	Primitive HAM message (before the optimization)
	Primitive urine	Primitive SPAM message (before the optimization)
	The quantity of the primitive urine from 150 to 180 liters per day	Filtering before optimization must be harsh
New meta-heuristic based on the renal system for the filtering of the SPAM	The renal circulation receives around **20%** of the cardiac output **every minute**	Processing flow rate (number of message to be processed in each iteration) is equal to 20 % of total number of message **by iteration**
	Renal tubular transfer : ✓ Reabsorption ✓ Secretion	Artificial Renal tubular transfer (Optimization by K-Means) ✓ (less than False Negative) and (more of True Positive) ✓ (less than False Positive) and (more of True Negative)
	The length of the Renal tubule is equal to 115 Km	The optimization must not be limited by a number of iterations (stopping criteria must be the stagnation of the classes)
	the final result : ✓ definitive urine the outside of the body ✓ The cleansed Blood : join the blood circulation	the final result : ✓ SPAM messages definitive (after optimization) ✓ HAM messages definitive (after optimization)
	Regularization of blood pressure, and **maintain the functioning of the glomerular filtration**	**Anti DDoS attack and activate (start) and stop the process of filtering**
	types of Nephrons ✓ Cortical Nephrons ✓ Juxtamedullary Nephrons	✓ No update the training set ✓ Update the training set
	Stimulus ADH = concentrated urine	Block list and blacklist
	ADH inhibition = diluted urine	Whitelist

Table 2. Modeling table :natural model system to artificial model

In the table above we explain in general correspondence between the natural model of the renal system and the artificial model which is renal system for the filtering of

the SPAM, In other words we showed the way that we interpreted and used the notions and concepts of the natural model in our artificial model which we propose. These interpretations and modeling will be explained and justified in detail in the development of artificial model in the next section.

2.3 The Artificial model of renal system for filtering of spam

In our approach dedicated to the detection and the filtering of the SPAM we suggest modelling the renal system such that the artificial model will have three states:

Initial state .

To solve the problem of filtering and detection of SPAM, we need a training set and a test set. In parallel, the training set represents the consumption of oxygen by the kidney which is between 10 to 15% of the blood supply (See table 2)

On the other hand, the test set is all the messages (SPAM and HAM) that must be treated. In equivalent with the natural model, the test set is represented by the coming blood from the afferent arteriole.

Initially, the kidneys must be active enough and well-fed to do their role appropriately. In the artificial model, we begin to fill the training set by 15 % of the number of messages.

We propose to launch two threads of the application on the server; each thread will represent a kidney. The goal of this parallelism is to accelerate the process of filtering and to give a fault tolerance so that even if a thread stops accidentally or intentionally, the other thread ensures the minimum services during the time of maintenance in addition to the re-launching of the thread which was broken down.

Moreover, each kidney contains a predefined large number of nephrons which is sufficient in our artificial model to not consume too much from the virtual memory and CPU. Nephrons are identical, i.e., the same parameters are generalized for the whole of nephrons. Obviously, the parameters of the nephrons are identical. If an email is considered SPAM by the first nephrons then it will be also considered SPAM by other nephrons and vice versa. We propose that

Each message (email) passes by only nephron randomly in each iteration

In our artificial model, 15% of the number of artificial nephrons (filtering agent) of each kidney are used to update the training set by messages after the application of the filtering process and optimization. It is the same for the juxtamedullary nephrons which feeds kidney.

State of activity .

The arrival of blood (message) by the afferent arteriole increases the blood pressure within the glomerulus and activate the process of filtration of blood (filtering of SPAM); this process is divided into two steps

Step 1: Artificial glomerular filtration:.

Before beginning this step we have to allocate to each message a score such that the HAM messages will have a high score then they not be filtered by the artificial Bowman's capsule (the sieve) and the SPAM message will have a weak score then they will filtered by artificial Bowman's capsule (the sieve).

There are several techniques and method of scoring, we have chosen to use the Bayesian classifier.

- Question 1: Why the Bayesian classifier?
- Answer 1 : "Our choice is justified by the fact that the performances of the bayesian classifier are strongly and negatively correlated with the number of class (As long as the number of class decreases, the performance of Bayesian classifier increases and vice versa). In addition to that, it allows to make a Hybridization between the approach based on Machine Learning by using the Bayesian classifier and the approach non-based on Machine Learning with our metaheuristics"

We propose to use for scoring a notion of Naive Bayes classifier, the score of each message will be equal to its probability to be HAM : In this way, each message having a big probability to be a HAM will not be filtered by the artificial Bowman's capsule and will join the efferent arteriole (classified as primitive-HAM) Whereas those who will have a weak probability to be a HAM will be filtered and will join the renal tubule (classified as primitive-SPAM)

The only change that will occur on the Bayesian classifier is in the step of class assignments. We will not assign the class according to the biggest probability, we will not even use both of two probability (SPAM and HAM).

We will define a hole diameter of the artificial Bowman's capsule (the sieve), it is **the artificial filtering glomerular threshold**; each message having a score bigger or equal to this threshold will not be filtered by the artificial Bowman's capsule and will join the efferent arteriole (classified as primitive-HAM) whilst those who will have a score smaller to the threshold will be filtered and will join the renal tubule (classified as primitive-SPAM); Let us remind that the score represents the probability that the message is a HAM by using the Bayesian classifier.

In the natural model, the glomerular filtration is very important with producing up to 180 liters of primitive urine per day, this must be taken up in the artificial model by choosing a big **artificial filtering glomerular threshold** so as to have a very harsh filtering of SPAM.

At the end of this first step we will have:

- a set of primitive SPAM that represents the primitive urine and which they will join the renal tubule.
- a set of primitive HAM that represents the filtered blood primitively and which they will join the efferent arteriole.

Noting that there are two different types of nephron, the messages which are in the juxtamedullary nephrons will be used in the next steps to update the training set.

Step 2: artificial renal tubular transfer(optimization).

The renal tubular transfer is made by two processes: Reabsorption and Secretion in the Proximal tubule, loop of Henle and distal convoluted tubule; Let us remind that these two processes will be performed in a automatic way (without intervention of brain).

At the level of the connecting tubule (CNT) two operations will be done : Stimulus ADH to get back the water from urine and Inhibition ADH to reject the water in urine. These two operations (Stimulus ADH and Inhibition ADH) are controlled by a hormone ordered by brain, therefore they are semi automatic operations.

In our work, and in parallel to the natural model; we propose to use a clustering algorithm which his initial state will be the result of the first step, since the renal tubular transfer takes in input the results of glomerular filtration.

This algorithm must represent at the same time the reabsorption process and secretion process, we propose to use the K-Means algorithm with k = 2, initially the centroids will be calculated by the classification resulting from the first step (glomerular filtration) as follows : The centroid of the HAM class will be calculated by the set of primitive-HAM (result of step 1 : glomerular filtration) likewise The centroid of the SPAM class will be calculated by the set of primitive-SPAM (result of step 1 : glomerular filtration)

- Question 2 : Why have we chosen the K-Means algorithm for modeling the renal tubular transfer?
- Answer 2 : " K-Means is the algorithm which well reflects the renal tubular transfer because :

    - The renal tubular transfer = Reabsorption + Secretion (at same times)
        - Reabsorption : transfer from the primitive urine to the primitive blood (from the SPAM-primitive class to the HAM-primitive class)
        - Secretion : transfer from the primitive blood to the primitive urine (from the HAM-primitive class to the SPAM-primitive class)

    The philosophy of the K-Means algorithm is that in every iteration: the documents changing class to join the class of which they are close to its centroid (class), in our case K = 2 (SPAM and HAM) : in each iteration, we have some messages classified as HAM who change class : from HAM class to the SPAM class (this is the secretion process) and other messages classified as SPAM who change class: from SPAM class to the HAM class (this is the reabsorption process)."

In the natural model; at the end of both processes of renal tubular transfer (reabsorption + secretion) two operations follow: the Stimulus ADH and / or the inhibition ADH.

In artificial model, we have represented the Stimulus ADH by the technique of whitelist (which will be generated by the user or the service provider) and we have represented the Inhibition ADH by the technique of blacklist (which will be also generated by the user or the service provider)

Updating of the training set.

In the artificial model 15% of nephrons are juxtamedullary nephrons, the messages which are treated by those nephrons will be used to update the training set. Let us note that messages are randomly distributed on nephrons and Let us also remind that SPAM messages (of test set) are in the renal tubule and that HAM messages (of test set) are in the peritubular capillary network which surrounds the renal tubule.

The training set must not grow up to avoid Overfitting, so we must make a crushing of the most repeated message in it to create diversity in this training set so that a maximum number of cases will be included by it (training set).

Choose a outbound HAM and a outbound SPAM from training set.

We calculate the similarity between the HAM messages in the training set and also for SPAM messages and then we calculate the centroid of each class (SPAM and HAM of training set).

We choose the most similar two HAM (SPAM respectively) (the biggest similarity for each class). We suppose that HAM1 and HAM2 (SPAM1 and SPAM2 respectively) are most similar among all HAM messages in the training set (SPAM respectively), the HAM message to crush (outbound) is the closest to the centroid of HAM class of the training set (SPAM respectively) because it is more similar to the other HAM in the training set (respectively SPAM) and we shall call it the HAM-outbound (SPAM respectively).

The HAM-outbound message from the training set will be crushed and replaced with a copy of the HAM which is in the peritubular capillary network (test set) respectively The SPAM-outbound message from the training set will be crushed and replaced with a copy of the SPAM which is in the renal tubule (test set)

The final state.

After a finite and sufficient number of iterations; All messages classified as spam, which is the final urine will be conducted outside the body (system) so that they do not return to the blood stream. All messages classified as HAM which represents the purified blood and they will join the interlobular vein.

The training set will be updated and the recommendations of the user or service provider will be taken into consideration(Blacklist and Whitelist) as a part of the training set (by the updating), let us note that the training set is quite reduced (15% of SpamBase) so this recommendations (Blacklist and Whitelist) will have a weight in decision-making by the Bayesian classifier (in probability calculation).

In the artificial model, The regularization of the blood pressure serves to limit the number of message to be treated per iteration and it can be a solution to the paralysis of the system which is caused by the spammers by a DDoS attack (denial of service) to stop the security system completely or at least make a bug in the application of filtering of SPAM. At the end of the flow of message the glomerular pressure will be equal to zero what will put the application of filtering of SPAM in paused state; If a bulk of message arrives by the afferent arteriole then the glomerular pressure increases, and will be different from zero what which will reactivate the application of filtering of SPAM.

In summary, at the end of our process that we propose, the results are:

- The SPAM class (classified as SPAM) representing the final urine.
- The HAM class (classified as HAM) representing the purified blood.
- A learning base updated to the next use (a fed kidneys).

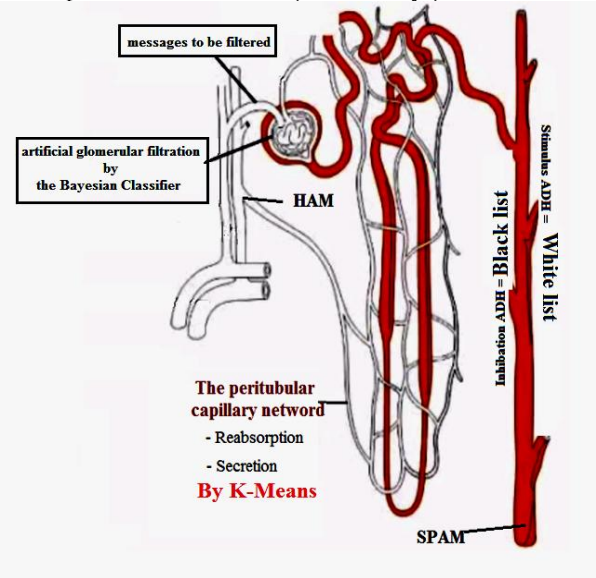


Fig. 2. artificial model vs. natural model

3 Conclusion and perspective

In this paper, we have proposed a new bio-inspired method which is the renal system. We have developed an artificial model for the detection of SPAM based on Renal Function. It is proved that this model based on the renal system is able to detect and filter the spams.

The new approach which we propose is effective and strong. Firstly,we have used the concept of Bayesian classifier which behaves well when it is applied to a classification with limited number of class (in our case the number of class is equal to two (2)); the only difference is that the class assignment is done by **an artificial filtering glomerular threshold** and not by the most majority probability. Secondly the initial state of the optimization algorithm by K-Means (the centroids) is not random, but is based on the result of the first step.

Another strong point of our approach is that the training set is not static. In fact, the training set is updated at each iteration to ensure better representativeness of possible cases and eliminate similar cases. This update is designed to take into consideration the messages we recovered / removed by technical whitelist / blacklist; So , after

sufficient number of iterations that we will experiment in our future work, the training set will consider them.

Actually, our approach ensures fault tolerance by launching two threads of the application on the server, even if a thread stops accidentally or intentionally, Additionally, our approach can resist against the attack of DDoS by the regularization of pressure of treatment, even in the case of overloading of the server by an exponential number of messages our approach treats only a part of this messages at each iteration.

In the future, we plan to experiment our artificial model for the filtering of the SPAM based on human renal function and to compare its performance with other algorithms as well as other techniques of SPAM's filtering. After several experiments, we have tried to present some useful recommendations for further studies and improvement according to the results; the strengths are retained while the weaknesses are rectified and mended rather than ended. Last but not least, we plan to, hopefully, create an appropriate model of functioning parallel to our approach.

4 Reference

1. Sebastiani, F. (2002). Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1), 1-47.
2. Lokbani, A. C., Lehireche, A., & Hamou, R. M. (2013). Experimentation of Data Mining Technique for System's Security: A Comparative Study. In Advances in Swarm Intelligence (pp. 248-257). Springer Berlin Heidelberg.
3. Zhang, L., Zhu, J., & Yao, T. (2004). An evaluation of statistical spam filtering techniques. ACM Transactions on Asian Language Information Processing (TALIP), 3(4), 243-269.
4. Van Staden, F., & Venter, H. S. (2009). The State of the Art of Spam and Anti-Spam Strategies and a Possible Solution using Digital Forensics. In ISSA (pp. 437-454).
5. Sanz, E. P., Gómez Hidalgo, J. M., & Cortizo Pérez, J. C. (2008). Email spam filtering. Advances in Computers, 74, 45-114.
6. Gupta, G., Mazumdar, C., & Rao, M. S. (2004). Digital Forensic Analysis of E-mails: A trusted E-mail Protocol. International Journal of Digital Evidance, 2(4).
7. Murphy, K., Travers, P., & Walport, M. (2008). Janeway's immunology. Garland science.
8. Oda, T., & White, T. (2005). Immunity from spam: An analysis of an artificial immune system for junk email detection. In Artificial Immune Systems (pp. 276-289). Springer Berlin Heidelberg.
9. Tan, Y., Deng, C., & Ruan, G. (2009, June). Concentration based feature construction approach for spam detection. In Neural Networks, 2009. IJCNN 2009. International Joint Conference on (pp. 3088-3093). IEEE.
10. Ruan, G., & Tan, Y. (2010). A three-layer back-propagation neural network for spam detection using artificial immune concentration. Soft Computing, 14(2), 139-150.
11. Hamou, R. M., Amine, A., & Boudia, A. (2013). A New Meta-Heuristic Based on Social Bees for Detection and Filtering of Spam. International Journal of Applied Metaheuristic Computing (IJAMC), 4(3), 15-33.
12. Guzella, T. S., & Caminhas, W. M. (2009). A review of machine learning approaches to spam filtering. Expert Systems with Applications, 36(7), 10206-10222.

Assessing online media content trustworthiness, relevance and influence: an introductory survey

Eleonora Ciceri, Roman Fedorov, Eric Umuhoza, Marco Brambilla, and Piero Fraternali

Politecnico di Milano. Dipartimento di Elettronica, Informazione e Bioingegneria
Piazza L. Da Vinci 32. I-20133 Milan, Italy
first.last@polimi.it

Abstract. The increasing popularity of social media articles and microblogging systems is changing the way online information is produced: users are both content publishers and content consumers. Since information is produced and shared by common users, who usually have a limited domain knowledge, and due to an exponential growth of the available information, assessing online content trustworthiness is vital. Several works in the state of the art approach this issue and propose different models to estimate online content trustworthiness, content relevance and user influence. In this paper we investigate the most relevant research works in this domain, highlighting their common characteristics and peculiarities in terms of content source type, trust-related content features, trust evaluation methods and performance assessment techniques.

1 Introduction

With the increasing popularity of social media, user-generated content [36] (i.e., content created by users and publicly available on the Web) is reaching an unprecedented mass. The overload of user-generated content makes hard to identify relevant content and to extract trustworthy and high quality information. Assessing trust [24, 31], content relevance [16] and user influence [11] is a critical issue in everyday social activities, where it is vital to filter non-authoritative, low quality and non-verified content to provide users with trusted information and content produced by experts.

As a motivating example, Motutu and Liu [45] report the "Restless Leg Syndrome" case: in 2008, when looking for information about the syndrome on Google, a wrong (and possibly dangerous) treatment promoted by the website WikiHow[1] was returned as top-1 result. This could obviously characterize a serious risk for patients; nevertheless, its rank wrongly suggested it could be trusted as verified and high quality information. Other applications that a misevaluation of user-generated content trustworthiness can affect are: disaster

This work is supported by POR-FESR 2007-2013 PROACTIVE Project and EU FP7-ICT-619172 SmartH2O Project.

[1] http://www.wikihow.com

management [41] (e.g., via false rumors on social networks during emergencies), environmental monitoring [18] (e.g., false reports of environmental phenomena), trend analysis [40] (e.g., via polluting the available information about what users like), detection of news [25] (e.g., via the diffusion of wrong news over the network).

Users often rely on their own knowledge, intuition and analytic capabilities to assess content relevance and trust. However, this becomes unfeasible with the current massive consumption of user-generated content: large volumes of low-quality, non-significant information are produced every day, and valuable content drowns in the large ocean of irrelevant content with little probability of being found by users. Consequently, it is vital to identify an automatic content trust estimation procedure which helps users in discarding unworthy information and focusing on significant content. Three ingredients are necessary to perform trust estimation: *i)* the evaluation of content relevance [15]; *ii)* the identification of influential users and experts [9], which are often focused on a specific topic, and produce mostly valuable content; *iii)* the evaluation of the level of *trust* [26] one can put on the content and people producing it. These ingredients are usually obtained by applying knowledge extraction algorithms and building appropriate trust models on user-generated content.

In recent years, several works in this field have emerged. In particular, several sub-fields significantly overlap between one another [60]: online content quality and relevance estimation [22], user reputation estimation [13] and influencers detection [42] all take part in assessing the quality of information one can find on the Web. This survey overviews the main state-of-the-art methods used in the automatic estimation of content quality, based on either content characteristics (i.e., content trustworthiness, relevance and credibility) or user characteristics (i.e., user trustworthiness and influence), which are strongly intertwined: high quality content often derives from highly experienced and influential users. Specifically, while other survey works go deeper in the details of trust estimation methods and applications [48, 34, 60], we deem our work merges together concepts from all the listed sub-fields and holds a practical relevance for practitioners and researchers who approach these themes for the first time.

The rest of this document is structured as follows: Section 2 introduces the concepts of trust, content relevance and user influence; Section 3 lists content and user profile features used as ingredients to assess content/user trustworthiness; Section 4 discusses methods to aggregate those features and provide a final trust score; Section 5 surveys the different approaches for performance assessment and output validation; finally, Section 6 concludes the work with final considerations and possible future directions in this field.

2 Trust, content relevance and user influence

In this section we introduce the definitions of trust, content relevance and influence, and list the research questions associated with these themes discussed in the state of the art.

2.1 Definitions

The concept of *trust* [39, 44] has been largely studied in the literature, both from a sociological [21] and philosophical [6] point of view. However, with the advent of social media [32, 17], studies on trust have recently shifted towards the construction of a trustworthiness model for digital content [45]. Siegrist and Cvetkovich [56] define trust as a tool that reduces social complexity: users that trust other users believe in their opinions, without making rational judgments. Sztompka [59] defines trust as "*the gambling of the belief of other people's possible future behavior*".

The concept of *relevance* (or *pertinence*) is crucial in the ability of an information retrieval system to find *relevant* content [43]. Many research works study the definition of relevance and its subjectivity in terms of system-oriented relevance [54], user relevance judgment [14], situation relevance [65], etc. Content relevance and popularity [10, 38, 19] are often connected: topic-related high quality content becomes often viral.

Social influence [61, 62] is defined as the power exerted by a minority of people, called *opinion leaders*, who act as intermediaries between the society and the mass media [33]. An opinion leader is a subject which is very informed about a topic, well-connected with other people in the society and well-respected.

The concepts of trust, content relevance and social influence are strongly intertwined: *i)* influential users (i.e., opinion leaders) are often experts in a specific field; *ii)* domain experts produce trustworthy content; *iii)* trustworthy, topic-related content has high relevance to the selected field. Moreover, popularity plays its role too: viral content is transmitted through the network in the same way a disease spreads among the population, and the more influential are the users sharing it, the larger is its popularity [47].

In this work we talk indistinctly about *trust*, *relevance* and *influence*, since they all represent quality measures for the object in question (i.e., either users or content). For the ease of the reader, henceforth, *trust* refers also to other discussed qualities, namely *relevance* and *influence*.

2.2 Research questions

A model of trust is defined as a function that extracts a set of *features* from a content object and aggregates them into a trustworthiness index. The construction of such model raises three research questions:

1. Which features better define the concept of trust and content quality?
2. How do we aggregate such features into a trustworthiness index?
3. How do we assess the quality of the trustworthiness index?

In the next sections these questions are addressed separately.

3 Trust model: features selection

In this section we describe features frequently used in the literature to assess the trustworthiness of Web content.

3.1 Source-based features

User-generated content is retrieved from a Web publishing source. Thus, the features one can extract from content to assess its quality depend on what can be extracted from the Web source. Although each source has its own characteristics and differences, we can classify them into two main categories:

- *Article-based sources* focus on the content itself, published in the form of articles. Content is usually long, and sometimes authors are encouraged to review, edit, rate and discuss it, thus creating high quality, multi-authored information. The author of the content may be thus unknown. Examples of these kind of sources are blogs, online encyclopedias (e.g., Wikipedia[2]) and question-answering communities (e.g., Stackoverflow[3]). Several works apply trust estimation techniques on these sources (e.g., [1, 3, 46]).
- *Social media* promote users as content authors: common people produce content which could become viral in short time. Users' authority becomes a key factor in the evaluation of content trustworthiness: non-expert authors often generate low quality, untrusted content. Examples of these kind of sources include Facebook[4], Twitter[5] and LinkedIn[6]. Several works apply trust estimation techniques on these sources: some examples can be found in [9, 41, 63].

Trust assessment studies performed on article-based sources tend to use content-based features (e.g., article length), since often the author is unknown, while works performed on social media focus both on author properties (e.g., number of connection with others) and content characteristics.

3.2 Content and author-based features

Moturu and Liu [45] propose a classification of features which takes inspiration from what people use to assess the trustworthiness of a person or a content in the real world. To evaluate user and content trustworthiness, we base our analysis on user's past actions (i.e., *reputation*), user/content present status (i.e., *performance*) and user/content perceived qualities (i.e., *appearance*). In the following, we describe each category separately. For a more complete overview see [7, 66, 49].

Reputation User reputation suggests how much one should trust their content [59]. The reputation depends on which actions users perform on social media, such as: *i)* content creation or consumption, *ii)* answers to others' content, *iii)* interactions with others, and *iv)* social networking. Reputation can be further split in the following feature categories:

[2] `http://en.wikipedia.org`
[3] `http://stackoverflow.com`
[4] `http://www.facebook.com`
[5] `http://twitter.com`
[6] `http://www.linkedin.com`

- *Connectedness.* The more a user is connected with others, the higher is his reputation in the network. Connectedness features are related to connections between users, and comprise simple features such as author registration status [45], number of followers/friends [51, 63, 4], number of accounts in different social media [29]. Furthermore, more complex features can be defined in this context, such as author centrality in graph of co-author network [50], social connectedness [45], number of reading lists the author is listed in [51], H-index [52] and IP-influence (i.e., influence vs. passivity) [52]. The identification of highly connected people is vital in case the objective is to spread content virally [55, 58].
- *Actions on the content.* The more acknowledged is the content one produces, the higher is his reputation on the network. Features in this category include the quantity/frequency of contributions to articles [45, 29], the amount of content sharing on social media [45, 3], the number of upvotes/likes [29], the number of answers to others' content [29], the number of retweets and retweeting rate [29, 52] and the Klout influence score [29].

Performance User performance describes the behavior of that user and his actions [45], and can be used to estimate his trustworthiness [59]. On the other hand, content performance can be determined from user's actions towards it and from the interest it generates. Performance-related features vary significantly depending on which social media platform we consider in our analysis. Example of such features include:

- Number of content edits [45].
- Direct actions on the content (e.g., number of responses/comments to a blog post [2] and retweets [29]).
- Characteristics of content update procedures (e.g., edit longevity [50], median time between edits, median edit length, proportion of reverted edits [45]).
- References to content by external sources (e.g., number of internal links [45, 2], incoming links [2], references by other posts [2], weighted reference score [45], publication date and place [29], variance on received ratings [29]).

Appearance External characteristics that represent the individual's appearance, personality, status and identity can be used to assess his trustworthiness. Similarly, the characteristics of content, such as style, size and structure, are useful in judging its quality. The most used features of this category include:

- Measure of the author reliability based on the structure of the content (e.g., length of blog posts, number of sections and paragraphs [45]).
- Language style (e.g., punctuation and typos [8], syntactic and semantic complexity and grammatical quality [3], frequency of terms belonging to a specific category [51], keywords in a tweet [8]).
- Originality of the content (e.g., presence of reused content [29], patterns of content replication over the network [8]).

4 Trust model: features aggregation

In Section 3 we present various feature categories used to assess the trustworthiness of online media content and users. Those features are transformed in a *trust/quality* index (usually scalar) through trustworthiness estimation algorithms.

Although it is common to find naive feature aggregation methods [66, 29, 50], the literature proposes a variety of more complex methods used to compute the final trust score. The categorization of such methods is not trivial, due to a fuzzy separation between *feature definition* and *feature aggregation* methods.

- *Statistical approaches.* It is common for features to be aggregated through cluster rank scores [45, 29, 12] or maximum feature values [2]. Several works use more refined approaches, such as cumulative distribution-based ranking, [66], K-nearest neighbors and Naive-Bayes classification [8], regression trees [4], mixture models [5], Gaussian Mixture Model and Gaussian ranking [49].
- *Graph-based algorithms.* Social connections play an important role in assessing the level of trust of an user and his content: the more connected is the user, the more others are interested in what he produces. Thus, several algorithms use graph-based methods, e.g., PageRank [52, 53, 27, 37, 64, 50, 63] and its variants [28], HITS [35], impact of a user on the social connections graph entropy [55], graph centrality measures [63, 27], indegree vs. outdegree [64] and other custom metrics based on information exchange over graphs [58, 8]. In some cases, trust is computed based on characteristics of a specific content source, e.g., number of followers vs. friends in the Twitter graph [28, 37].
- *Feature correlation.* Several works do not define an aggregation method, and simply study the correlation between features [52, 51].
- *Correlation between user influence and content relevance.* Some works use influencers retrieval techniques to identify influential users from a social network, and then navigate through the content they produce to collect the most relevant one [57].

Generally, the lack of uniformity in the proposed evaluation metrics and the heavy dependence on the type of content source (see Section 3.1) make it difficult to compare such metrics and state which one is most suited for a specific context. We believe that a further standardization of features would encourage the development of more sophisticated aggregation methods, e.g., based on supervised machine learning regressors and classifiers, as already proposed by Agichtein et al. [3] and by Castillo et al. [7].

5 Trust model: evaluation techniques

In this section we describe the experimental evaluation techniques that are used to assess the performance of the proposed trustworthiness estimation methods.

We state that the discussed research fields suffer from the absence of standardized requirements for the expected output. Thus, it is often difficult for the authors to compare their methods with respect to other state-of-the-art approaches.

5.1 Datasets

Due to the high variance of the type of data one can retrieve from each content source type, there exists a large collection of datasets in the state of the art, rarely made publicly available.

- *Custom datasets.* Almost all works create their own dataset by crawling data from the selected content publishing platforms. Several works (e.g., [7, 51, 52, 63]) base their analysis on Twitter, for several reasons: *i)* high volume of publicly available user-generated content; *ii)* presence of both textual and multimedia data; *iii)* access to public user profiles and their connections with other users; *iv)* easy storage of content (for further analysis), due to the limited length of posts. However, sometimes also article based platforms are taken into account (e.g., Wikipedia in Qin et al. [50], or question-answer platforms in Agichtein et al. [3]).
- *Use of standard datasets.* Sometimes, more standard datasets are used, e.g., the Enron Email Database[7] analyzed by Shetty and Adibi [55] or the WikiProject History[8] in [50], in which articles have been assigned class labels according to the Wikipedia Editorial Teams quality grading scheme.
- *Building a gold standard.* To assess the performance of a trust computation technique, it is often necessary to build a *gold standard* (or *ground truth*), i.e., a set of manually annotated data in which annotators are asked to state whether the content can be trusted, and labels are supposed to be error-free. In several contexts, labeling content is usually performed by a group of people (either part of an internal crowd or workers in some crowdsourcing platform [20, 67]), which manually annotate content. Then, the output of the proposed algorithm is compared with the ground truth, to assess the precision and recall of the retrieved set of trusted content/users [45, 66, 49]. However, trustworthiness, content quality and relevance are highly subjective characteristics, and thus the ground truth one builds is based on each annotator's perception of what being trustworthy means, which makes it biased and not reliable.

5.2 Performance assessment

State-of-the-art trust and influence metrics are all different and sometimes difficult to compare. Several works, thus, evaluate their performance with respect to similar algorithms applied to the same content sources. For this reason, the range of the metrics considered in this document is wide.

[7] `https://www.cs.cmu.edu/~./enron/`
[8] `https://en.wikipedia.org/wiki/Wikipedia:WikiProject_History`

- *Manual validation.* Many works tend to evaluate and discuss the results through manual inspection, where an internal crowd [55, 49, 9, 35, 27, 66] or anonymous users via user studies [23, 28, 12] assess the quality of the retrieved set of users/content.
- *Classification performance.* In some works, the authors manage to cast the trust evaluation problem as a classification problem, in which users are classified as influential/non-influential and content is labeled as trusted/non-trusted. These works are likely to present standard classification performance metrics: precision, TP-rate, FP-rate, accuracy [7] and ROC curves [3].
- *Evaluation of rankings.* In other cases, the output of the algorithm is a ranked list of authoritative content/users, and thus ranking correlation indexes (i.e., Pearson correlation [49] or generalized Kendall-Tau metrics [37]) are used to assess the performance of the proposed algorithm. In the same perspective, NDCG [30] (originally designed to test the ability of a document retrieval query to rank documents by relevance) is used to evaluate quality, trustworthiness and influence estimations, both in article-based content sources [45, 50] and microblogging platforms [66].
- *Comparison with known rankings.* Some works compare the output ranking of content/user with some rankings one can found on the Web, e.g., Digg [2], Google Trend and CNN Headlines [37].
- *Characteristics of users.* In some cases, one takes into account some characteristics of the involved users (e.g., activity [64] or validation of profile on Twitter [5]) to assess the performance of the algorithm. A high-performance result, in this sense, is the one maximizing the overlap between the set of active (validated) users and the users retrieved by the proposed algorithm.
- *Custom metrics.* Finally, some works build their own performance metrics, since in such cases it is difficult to compare the proposed algorithm with the ones available in the state of the art [51].

6 Conclusions and open challenges

In this survey we presented an overview of major recent works in the field of automatic estimation of trustworthiness, relevance and influence of online content. As discussed, trust estimation is important in Web search, and can be performed by capturing multiple signals deriving from both user profiles and content characteristics: authoritative (or influential) users produce mainly high quality content, and high quality content is largely trusted on the network of users. We thus reviewed several algorithms, listing their common characteristics and peculiarities in terms of content type, trust evaluation features and algorithms and performance assessment metrics.

We believe that these recent research topics are of great interest and practical importance in several domains such as automatic content retrieval and analysis, viral marketing, trend analysis, sales prediction and personal security. Nevertheless, in our opinion, there is enough space and need for future works that aim at building a concrete base of gold standards common to all discussed topics, and

solidly integrating the proposed techniques to merge the efforts and converge towards a unified approach for user trust and content relevance estimation.

Current research works by the authors include methods for multi-platform and multimedia collective intelligence extraction from user-generated content, e.g., to perform trend analysis on the preference of Twitter users and to estimate environmental characteristics such as the presence of snow on mountains. Extracting relevant information from user-generated content implies: *i)* the identification of the influential users; *ii)* the estimation of content relevance; *iii)* the estimation of content trustworthiness. We believe that a strong cooperation of methods operating on multiple platforms and multiple content types (e.g., text, images, videos) is fundamental to define new standards this field lacks of.

References

1. Adler, B.T., Chatterjee, K., De Alfaro, L., Faella, M., Pye, I., Raman, V.: Assigning trust to wikipedia content. In: Proceedings of the 4th International Symposium on Wikis. p. 26. ACM (2008)
2. Agarwal, N., Liu, H., Tang, L., Yu, P.S.: Identifying the influential bloggers in a community. In: Proceedings of the 2008 international conference on web search and data mining. pp. 207–218. ACM (2008)
3. Agichtein, E., Castillo, C., Donato, D., Gionis, A., Mishne, G.: Finding high-quality content in social media. In: Proceedings of the 2008 International Conference on Web Search and Data Mining. pp. 183–194. ACM (2008)
4. Bakshy, E., Hofman, J.M., Mason, W.A., Watts, D.J.: Everyone's an influencer: quantifying influence on twitter. In: Proceedings of the fourth ACM international conference on Web search and data mining. pp. 65–74. ACM (2011)
5. Bi, B., Tian, Y., Sismanis, Y., Balmin, A., Cho, J.: Scalable topic-specific influence analysis on microblogs. In: Proceedings of the 7th ACM international conference on Web search and data mining. pp. 513–522. ACM (2014)
6. Blomqvist, K.: The many faces of trust. Scandinavian journal of management 13(3), 271–286 (1997)
7. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on twitter. In: Proceedings of the 20th international conference on World wide web. pp. 675–684. ACM (2011)
8. Cataldi, M., Aufaure, M.A.: The 10 million follower fallacy: audience size does not prove domain-influence on twitter. Knowledge and Information Systems pp. 1–22 (2014)
9. Cha, M., Haddadi, H., Benevenuto, F., Gummadi, P.K.: Measuring user influence in twitter: The million follower fallacy. ICWSM 10(10-17), 30 (2010)
10. Cha, M., Kwak, H., Rodriguez, P., Ahn, Y.Y., Moon, S.: Analyzing the video popularity characteristics of large-scale user generated content systems. IEEE/ACM Transactions on Networking (TON) 17(5), 1357–1370 (2009)
11. Chan, K.K., Misra, S.: Characteristics of the opinion leader: A new dimension. Journal of advertising 19(3), 53–60 (1990)
12. Chen, C., Gao, D., Li, W., Hou, Y.: Inferring topic-dependent influence roles of twitter users. In: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. pp. 1203–1206. ACM (2014)

13. Cook, K.S., Yamagishi, T., Cheshire, C., Cooper, R., Matsuda, M., Mashima, R.: Trust building via risk taking: A cross-societal experiment. Social Psychology Quarterly 68(2), 121–142 (2005)
14. Cuadra, C.A., Katter, R.V.: Opening the black box of'relevance'. Journal of Documentation 23(4), 291–303 (1967)
15. De Choudhury, M., Counts, S., Czerwinski, M.: Find me the right content! diversity-based sampling of social media spaces for topic-centric search. In: ICWSM (2011)
16. De Choudhury, M., Counts, S., Czerwinski, M.: Identifying relevant social media content: leveraging information diversity and user cognition. In: Proceedings of the 22nd ACM conference on Hypertext and hypermedia. pp. 161–170. ACM (2011)
17. Ellison, N.B., et al.: Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication 13(1), 210–230 (2007)
18. Fedorov, R., Fraternali, P., Tagliasacchi, M.: Snow phenomena modeling through online public media. In: Image Processing (ICIP), 2014 IEEE International Conference on. pp. 2174–2176. IEEE (2014)
19. Figueiredo, F., Benevenuto, F., Almeida, J.M.: The tube over time: characterizing popularity growth of youtube videos. In: Proceedings of the fourth ACM international conference on Web search and data mining. pp. 745–754. ACM (2011)
20. Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., Dredze, M.: Annotating named entities in twitter data with crowdsourcing. In: Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk. pp. 80–88 (2010)
21. Golbeck, J.: Combining provenance with trust in social networks for semantic web content filtering. In: Provenance and Annotation of Data, pp. 101–108 (2006)
22. Grady, C., Lease, M.: Crowdsourcing document relevance assessment with mechanical turk. In: Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's mechanical turk. pp. 172–179. Association for Computational Linguistics (2010)
23. Hannon, J., Bennett, M., Smyth, B.: Recommending twitter users to follow using content and collaborative filtering approaches. In: Proceedings of the fourth ACM conference on Recommender systems. pp. 199–206. ACM (2010)
24. Hsieh, H.F., Shannon, S.E.: Three approaches to qualitative content analysis. Qualitative health research 15(9), 1277–1288 (2005)
25. Hu, M., Liu, S., Wei, F., Wu, Y., Stasko, J., Ma, K.L.: Breaking news on twitter. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 2751–2754. ACM (2012)
26. Huang, F.: Building social trust: A human-capital approach. Journal of Institutional and Theoretical Economics (JITE)/Zeitschrift für die gesamte Staatswissenschaft pp. 552–573 (2007)
27. Huang, P.Y., Liu, H.Y., Chen, C.H., Cheng, P.J.: The impact of social diversity and dynamic influence propagation for identifying influencers in social networks. In: Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on. vol. 1, pp. 410–416. IEEE (2013)
28. Jabeur, L.B., Tamine, L., Boughanem, M.: Active microbloggers: identifying influencers, leaders and discussers in microblogging networks. In: String Processing and Information Retrieval. pp. 111–117. Springer (2012)
29. Jaho, E., Tzoannos, E., Papadopoulos, A., Sarris, N.: Alethiometer: a framework for assessing trustworthiness and content validity in social media. In: Proceedings

of the 23th International Conference on World Wide Web Companion. pp. 749–752. International World Wide Web Conferences Steering Committee (2014)
30. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Transactions on Information Systems (TOIS) 20(4), 422–446 (2002)
31. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision support systems 43(2), 618–644 (2007)
32. Kaplan, A.M., Haenlein, M.: Users of the world, unite! the challenges and opportunities of social media. Business horizons 53(1), 59–68 (2010)
33. Katz, E., Lazarsfeld, P.F.: Personal Influence, The part played by people in the flow of mass communications. Transaction Publishers (1955)
34. Kelton, K., Fleischmann, K.R., Wallace, W.A.: Trust in digital information. Journal of the American Society for Information Science and Technology 59(3), 363–374 (2008)
35. Kong, S., Feng, L.: A tweet-centric approach for topic-specific author ranking in micro-blog. In: Advanced Data Mining and Applications, pp. 138–151 (2011)
36. Krumm, J., Davies, N., Narayanaswami, C.: User-generated content. IEEE Pervasive Computing (4), 10–11 (2008)
37. Kwak, H., Lee, C., Park, H., Moon, S.: What is twitter, a social network or a news media? In: Proceedings of the 19th international conference on World wide web. pp. 591–600. ACM (2010)
38. Lerman, K., Hogg, T.: Using a model of social dynamics to predict popularity of news. In: Proceedings of the 19th international conference on World wide web. pp. 621–630. ACM (2010)
39. Maheswaran, M., Tang, H.C., Ghunaim, A.: Towards a gravity-based trust model for social networking systems. In: Distributed Computing Systems Workshops, 2007. ICDCSW'07. 27th International Conference on. pp. 24–24. IEEE (2007)
40. Mathioudakis, M., Koudas, N.: Twittermonitor: trend detection over the twitter stream. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. pp. 1155–1158. ACM (2010)
41. Mendoza, M., Poblete, B., Castillo, C.: Twitter under crisis: Can we trust what we rt? In: Proceedings of the first workshop on social media analytics. pp. 71–79. ACM (2010)
42. Merton, R.K.: Patterns of influence: Local and cosmopolitan influentials. Social theory and social structure 2, 387–420 (1957)
43. Mizzaro, S.: How many relevances in information retrieval? Interacting with computers 10(3), 303–320 (1998)
44. Molm, L.D., Takahashi, N., Peterson, G.: Risk and trust in social exchange: An experimental test of a classical proposition. American Journal of Sociology pp. 1396–1427 (2000)
45. Moturu, S.T., Liu, H.: Quantifying the trustworthiness of social media content. Distributed and Parallel Databases 29(3), 239–260 (2011)
46. Nam, K.K., Ackerman, M.S., Adamic, L.A.: Questions in, knowledge in?: a study of naver's question answering community. In: Proceedings of the SIGCHI conference on human factors in computing systems. pp. 779–788. ACM (2009)
47. Ni, M., Chan, B., Leung, G., Lau, E., Pang, H.: Data from: Transmissibility of the ice bucket challenge among globally influential celebrities: retrospective cohort study (2014), `http://dx.doi.org/10.5061/dryad.n4sc4`
48. Nurse, J.R., Rahman, S.S., Creese, S., Goldsmith, M., Lamberts, K.: Information quality and trustworthiness: A topical state-of-the-art review. In: Proceedings of the International Conference on Computer Applications and Network Security (ICCANS) (2011)

49. Pal, A., Counts, S.: Identifying topical authorities in microblogs. In: Proceedings of the fourth ACM international conference on Web search and data mining. pp. 45–54. ACM (2011)
50. Qin, X., Cunningham, P.: Assessing the quality of wikipedia pages using edit longevity and contributor centrality. arXiv preprint arXiv:1206.2517 (2012)
51. Quercia, D., Ellis, J., Capra, L., Crowcroft, J.: In the mood for being influential on twitter. In: Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on. pp. 307–314. IEEE (2011)
52. Romero, D.M., Galuba, W., Asur, S., Huberman, B.A.: Influence and passivity in social media. In: Machine learning and knowledge discovery in databases, pp. 18–33. Springer (2011)
53. Saez-Trumper, D., Comarela, G., Almeida, V., Baeza-Yates, R., Benevenuto, F.: Finding trendsetters in information networks. In: Proceedings of the 18th ACM SIGKDD. pp. 1014–1022. ACM (2012)
54. Schamber, L., Eisenberg, M.B., Nilan, M.S.: A re-examination of relevance: toward a dynamic, situational definition. Information processing & management 26(6), 755–776 (1990)
55. Shetty, J., Adibi, J.: Discovering important nodes through graph entropy the case of enron email database. In: Proceedings of the 3rd international workshop on Link discovery. pp. 74–81. ACM (2005)
56. Siegrist, M., Cvetkovich, G.: Perception of hazards: The role of social trust and knowledge. Risk analysis 20(5), 713–720 (2000)
57. Silva, A., Guimarães, S., Meira Jr, W., Zaki, M.: Profilerank: finding relevant content and influential users based on information diffusion. In: Proceedings of the 7th Workshop on Social Network Mining and Analysis. p. 2. ACM (2013)
58. Sun, B., Ng, V.T.: Identifying influential users by their postings in social networks. Springer (2013)
59. Sztompka, P.: Trust: A sociological theory. Cambridge University Press (1999)
60. Thirunarayan, K., Anantharam, P., Henson, C., Sheth, A.: Comparative trust management with applications: Bayesian approaches emphasis. Future Generation Computer Systems 31, 182–199 (2014)
61. Watts, D.J., Dodds, P.S.: Influentials, networks, and public opinion formation. Journal of consumer research 34(4), 441–458 (2007)
62. Weimann, G.: The influentials: People who influence people. SUNY Press (1994)
63. Weitzel, L., Quaresma, P., de Oliveira, J.P.M.: Measuring node importance on twitter microblogging. In: Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics. p. 11. ACM (2012)
64. Weng, J., Lim, E.P., Jiang, J., He, Q.: Twitterrank: finding topic-sensitive influential twitterers. In: Proceedings of the third ACM international conference on Web search and data mining. pp. 261–270. ACM (2010)
65. Wilson, P.: Situational relevance. Information storage and retrieval 9(8), 457–471 (1973)
66. Zhai, Y., Li, X., Chen, J., Fan, X., Cheung, W.K.: A novel topical authority-based microblog ranking. In: Web Technologies and Applications, pp. 105–116. Springer (2014)
67. Zubiaga, A., Liakata, M., Procter, R., Bontcheva, K., Tolmie, P.: Crowdsourcing the annotation of rumourous conversations in social media. In: Proceedings of the 24th International Conference on World Wide Web Companion. pp. 347–353 (2015)

Linking Accounts across Social Networks: the Case of StackOverflow, Github and Twitter

Giuseppe Silvestri[1,2], Jie Yang[1], Alessandro Bozzon[1], and Andrea Tagarelli[2]

[1] Delft University of Technology, the Netherlands
giuseppe.silvestri.gios@gmail.com, {j.yang-3, a.bozzon}@tudelft.nl
[2] University of Calabria, Italy
andrea.tagarelli@unical.it

Abstract. Social Web accommodates a wide spectrum of user activities, including information sharing via social media networks (e.g., Twitter), question answering in collaborative Q&A systems (e.g., StackOverflow), and more profession-oriented activities such as social coding in code sharing systems (e.g., Github). Social Web enables the distinctive opportunity for understanding the interplay between multiple user activity types. To enable such studies, a basic requirement, and a big challenge, is the ability to link user profiles across multiple social networks.
By exploiting user attributes, platform-specific services, and different matching strategies, this paper contributes a methodology for linking user accounts across StackOverflow, Github and Twitter. We show how tens of thousands of accounts in StackOverflow, Github, and Twitter could be successfully linked. To showcase the type of research enabled by datasets built with our methodology, we conduct a comparative study of user interaction networks in the three platforms, and investigate correlations between users interactions across the different networks.

1 Introduction

Social Web comprises a diversity of social networking platforms, which cover a wide range of user activities. With the fact that a single user has multiple accounts across different social networks, it has now become increasingly important to link distributed user profiles belonging to the same user from multiple sources, which can benefit various applications. For instance, it has been shown that aggregating user profiles could improve personalized Web service such as recommendation systems by solving the cold-start problem [1].

Linking user profiles across multiple social networks also provides an opportunity for better understanding the interplay between different types of people's activities. Let us take as an instance the domain of software programmers: they share software related content in Twitter, seek or provide answers to software engineering related questions in StackOverflow, and collaboratively code in Github. These three different social networks (i.e., Twitter, StackOverflow and Github) are used by programmers differently, in terms of their purposes and correspondingly their activities. By aggregating the data sources from multiple networks, we might explore at large scale the complete spectrum of programmers' on-line professional activities.

Linking users' accounts across multiple social networks is considered a well-known problem, thus attracting multiple techniques and solutions [1, 6–8, 4, 5]. Previous studies addressed the online activities of professional users, but investigated a single type of activities in a single system [6, 7], or between two systems from a single perspective. For instance, [8] analyzes how participation in Q&A systems influences developers' productivity. [2] also considered the influence that each user has within and across two platforms, while exploiting features provided by StackOverflow (Up Votes and Questions) and Github (popular users are engaged more in commits, projects and issues). [3] focused on bridge users, in order to recognize how these users can favor information exchange across networks.

To drive a deeper investigation over users' professional activities, we are motivated to construct a cross-system users' accounts matching dataset from Twitter, StackOverflow, and Github, to enable future studies of professional activities from multiple perspectives. For instance, a dataset as such can help us understand how different types of users (e.g., users with different expertise) are engaged in different professional activities; it can also help in understanding how different types of social interactions among users can influence the evolution of communities of different professional activities. This paper contributes a methodology to link online users' accounts across Twitter, StackOverflow and Github, by exploiting different attributes of user profiles, platform specific API's and services, and a variety of accounts' matching strategies. As a first trail of valuing this dataset, we construct three social networks, including follower-followee networks of Twitter and Github, and helper-helpee networks of StackOverflow. By characterizing the networks features, we present our findings of how users interact with others in different activities, and how different activities of the same user correlate with each other.

The rest of the paper is organised as follows. Section 2 describes our methodology of matching users across StackOverflow, Github and Twitter, together with the corresponding results of user matching. Based on these matched users, Section 3 introduces our comparative study of user interactions between three user interaction networks in StackOverflow, Github ad Twitter, and Section 4 concludes our work.

2 Linking Accounts across Social Networks

This section describes our methodology of matching users across StackOverflow, Github and Twitter. We first discuss the general settings of data retrieval for the three social networking platforms, then present our user matching strategies and workflows.

2.1 Retrieving data from multiple platforms

StackOverflow. We downloaded the most recently released data dump from Internet archive[3]. Due to privacy concern, since the end of 2014[4] StackOverflow

[3] `https://archive.org/details/stackexchange`, accessed at April, 2015

[4] `http://meta.stackexchange.com/questions/221027/where-did-emailhash-go`

data dump no longer contains hashed user emails. While not crucial, hashed emails are a convenient and effective way to unambiguously match accounts. To overcome this limitation, we extended the data from the data dump released on September 2013 (which is the last released dump with hashed email addresses) with the latest data contained in the 2015 data dump.

Github. The GHTorrent project[5] has incrementally released Github data every two months since March 2012. We parsed its data from the first release containing user information (i.e., July, 2012) until the latest one on March 2015, and kept all versions of user information in our database for account matching.

Twitter. Given an existing user name, the related account information (e.g., profile picture, website) and related posts in Twitter can be retrieved via Twitter REST API[6]. The Twitter.com Search[7] functionality, on the other hand, allows for fuzzy retrieval of users accounts, returning a candidate set of accounts having screen names similar to the one provided as input. For our purposes, the latter proved more useful than the former for fuzzy matching.

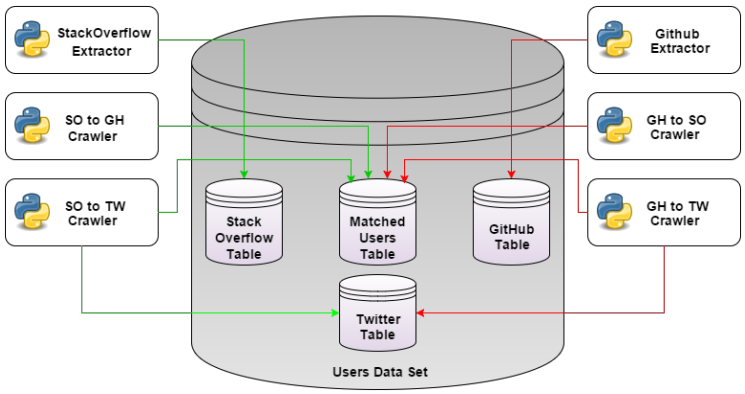


Fig. 1: Data collection workflow. SO, GH, TW are short for StackOverflow, Github, and Twitter, respectively.

The main workflow of accounts' linking across the three platforms is depicted in Figure 1. Accounts from StackOverflow and Github were dumped and processed first. We retrieved 4,132,407 and 4,288,132 accounts in StackOverflow and Github, respectively. These sets of accounts were then matched to each other and the resulting overlap further matched to a set of Twitter accounts. The latter was retrieved using a strategy that we will discuss later in this section.

[5] `http://ghtorrent.org`, accessed at April, 2015
[6] `https://dev.twitter.com/rest/public`
[7] `https://twitter.com/search-home`

2.2 Matching accounts across multiple platforms

We design three account matching strategies to find the same set of users in the platforms under study:

- **explicit matching**, which aims at identifying the links explicitly provided by users in one platform to their accounts in other platforms for user matching.
- **attribute-based matching**, which leverages unique attributes of users' accounts such as email to connect profiles across multiple platforms from the same user.
- **fuzzy matching**, which exploits less accurate user attributes such as login names and profile images to match user profiles.

Explicit matching is performed to link user accounts between StackOverflow and Github, and further link them to Twitter. Attribute-based matching is performed only between StackOverflow and Github, while fuzzy matching aims at linking matched users in StackOverflow and Github to Twitter. We introduce as follows the concrete steps we took for each of the matching strategies.

Explicit matching. Starting from our built dumps of StackOverflow and Github, we perform explicit matching by analyzing user-provided links from the user profiles in each of these platforms to the other platforms. We consider this a very reliable method for account linking: matching information are provided by users themselves, with strong incentives for truthful linking.

From StackOverflow to Github, Twitter. We analyze StackOverflow user profiles to find explicit links to GitHub and Twitter users. For StackOverflow users that provide links to their Github link, we parse the direct links, which are in the form of `https://github.com/GitHubLoginName` and obtain their Github login names, i.e., `GithubLoginName`. For StackOverflow users that provide direct links to Twitter, which is usually in the form of `http://www.twitter.com/TwitterScreenName`, we parse the Twitter screen name, i.e., `TwitterScreenName`. Both GitHub login name and Twitter screen name uniquely identifies one user in GitHub and Twitter, respectively.

From Github to StackOverflow, Twitter. We analyze Github user profiles similarly to match user profiles in StackOverflow and Twitter. For StackOverflow, we adopt an additional strategy to obtain a cross-reference to the same user: since some Github users provide their StackOverflow Careers profile[8], which is a CV-like page of senior StackOverflow users, we parse the HTML code of the corresponding pages in order to retrieve the direct link (in the form `http://stackoverflow.com/users/id`) to their real StackOverflow profile pages.

The result of explicit matching is reported in Table 1. As it can be observed, we are able to match thousands of users between the three platforms.

[8] `http://careers.stackoverflow.com/`, StackOverflow Careers

From	To	#Matched Users
StackOverflow	Github	4,536
	Twitter	10,068
Github	StackOverflow	433
	Twitter	7,012

Table 1: Explicit matching.

Attribute-based matching. StackOverflow and Github provide users with the option of registering their emails, which are encrypted into MD5 hashes in the data dumps. This technique is known from literature [8, 2] to be a reliable way to match users by their email reference.

There are in total 2,185,162 (≈52.9%) StackOverflow users and 510,523 (≈11.9%) Github users with email hash. Note that email hashes were previously considered for matching users between StackOverflow and Github in [8]. Besides using the email hashes explicitly provided by users, we exploit Gravatar[9] to increase the number of available hashes in both platforms. We find that many users use Gravatar to have a unique profile image across StackOverflow and Github. By making HTTP request for a Gravatar profile image, we obtain a user's MD5 email hash [10]. We identified 2,897,175 (≈67.6%) Github users, and 430,860 (≈10.4%) StackOverflow users with Gravatar email hash available.

$$\begin{aligned} query = & ((StackOverflowUsers[emailhash] \cap GithubUsers[emailhash]) \\ & \cup (StackOverflowUsers[gravatarid] \cap GithubUsers[gravatarid]) \\ & \cup (StackOverflowUsers[emailhash] \cap GithubUsers[gravatarid]) \\ & \cup (StackOverflowUsers[gravatarid] \cap GithubUsers[emailhash])) \end{aligned} \quad (1)$$

Combing email hashes explicitly provided by users, and implicitly revealed from their Gravatar Id, we use Query 1 for StackOverflow-Github user matching, which encodes all meaningful joins between MD5 email hash and Gravatar Id attributes across the two platforms. The result of attribute-based matching is shown in Table 2. We finally obtained more than 600k exactly matched users between StackOverflow and Github.

Fuzzy matching. Matching accounts from StackOverflow and Github with Twitter accounts is intrinsically more difficult, since Twitter profiles need to be obtained via Twitter API services.

Lookup and search. Two types of query requests are here considered, namely Twitter REST API and Twitter.com Search, hereinafter referred to as *Lookup*

[9] `https://en.gravatar.com/` Gravatar, a globally recognized avatar.
[10] `https://en.gravatar.com/site/implement/images/` Gravatar: Image Request

Type	#Matched Users
SO emailhash - GH gravatarid	580,979
SO emailhash - GH emailhash	107,572
SO gravatarid - GH emailhash	1,224
SO gravatarid - GH gravatarid	4,752
Union all above types	604,083

Table 2: Attribute-based matching between StackOverflow and Github.

and *Search*, respectively. The former method returns the full profile information of the user corresponding to a given user screen name. Using Twitter REST API, each request can process up to 100 inputs. By contrast, Twitter.com Search permits to process only a single input for each request. While being less efficient, Twitter.com Search is however more flexible in terms of the input — it accepts any textual input.

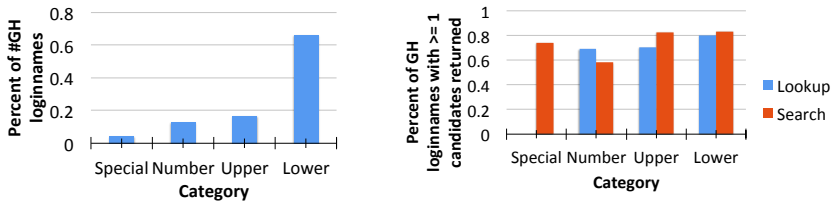
We consider the following options of input for the *Search* method:

- login names, and names of users' StackOverflow and Github accounts;
- URLs of user's StackOverflow and Github accounts;
- users' website URLs identified from their StackOverflow and Github profiles.

To find the best input for the *Search* method, we analyzed how many accounts can be matched by using different user attributes. Matching is performed in two steps: (1) given a user attribute, retrieve candidate users via Twitter.com Search; (2) try matching the website URL of the Twitter candidates and the website URL of the user StackOverflow (Github) profile. Results have shown that using Github login name provides better matching of Twitter profiles than the URLs of their accounts in StackOverflow or Github, as well as their website URLs. We therefore chose to take Github login name as an input for *Search* to retrieve candidate Twitter profiles for matching.

Accuracy of *Lookup* and *Search* methods. To assess the performance of the *Lookup* and *Search* matching methods, we first categorized the Github login names into the following categories: 1) the login contains only lower-case characters, 2) it contains at least one upper-case character, 3) it contains numbers, and 4) it contains special characters. Figure 2a shows the distribution of Github login names according to the categorization above, from which we observe that the majority of them are in the "lower-case" category.

To understand how different categories differ in the probability that at least one candidate can be returned by *Lookup* and *Search*, in Figure 2b we analyzed the percentages of Github login names that have at least one candidate returned by *Lookup* and *Search*. High values indicate higher probability that the user can be matched. We observe that the *Search* method performs better than *Lookup* in all categories except in the "Number" category.



(a) (b)

Fig. 2: Distributions of Github login names in selected categories (a) and number of candidates returned by *Lookup* and *Search* for each category (b).

Category	Lookup	Search	Gain
Lower	.29	.39	+.10
Upper	.28	.33	+.05
Number	.24	.27	+.03
Special	.00	.19	+.19

Table 3: Accuracy of Twitter user matching using *Lookup* and *Search* for different categories of Github login name.

For each category, we randomly selected 100 Github login names, took them as input for both *Lookup* and *Search* methods, then manually checked the matched accounts. A user is considered to be matched with a Twitter account if there is explicit Twitter information (e.g., personal website, profile description) that can identify the user with high confidence. Table 3 shows that *Search* performs better than *Lookup*, especially for Github login name that belong to the "lowercase" and "special characters" categories. The least gain of *Search* over *Lookup* corresponds to the category "Number" (less than 5%). Considering Figure 2b and the higher efficiency of *Lookup* method, we chose to use *Lookup* for Github login names in the "Number" category, and *Search* for the other categories.

Workflow of fuzzy matching. Figure 3 depicts the workflow of *Lookup* and *Search* methods. Given a user Github login name, it first determines whether to use *Lookup* or *Search*, then checks Twitter profiles for account matching. In the step of "Twitter Profile Check", a user is matched to a Twitter account if s/he satisfies the following criteria:

1. the website attribute of the user's Twitter profile is exactly the same as the website of his/her StackOverflow (Github) profile;
2. otherwise, the Twitter profile picture needs to be highly similar (e.g., $\geq 90\%$) to her/his profile picture in StackOverflow (Github).

In criterion 1 we ignored ambiguous websites such as `http://facebook.com`, which can bring to have False Positive for website matching, while for

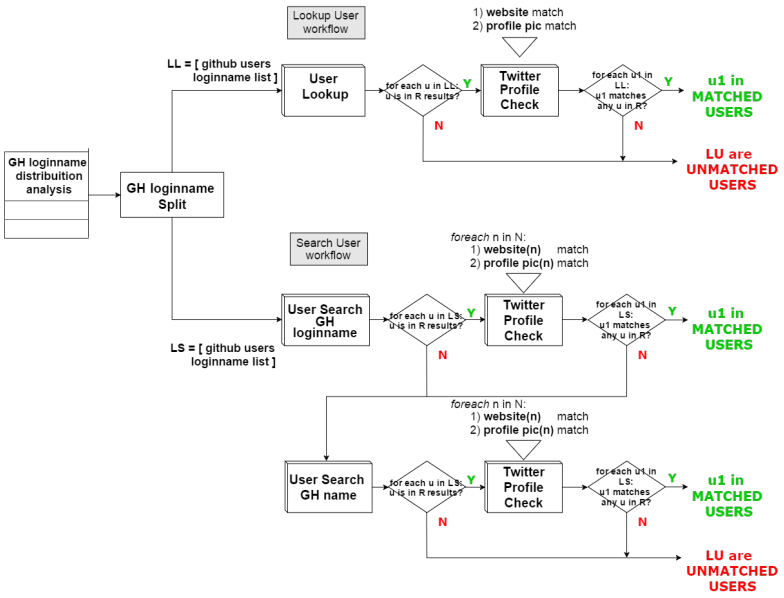


Fig. 3: Twitter *Lookup* and *Search* workflows.

criterion 2 we performed image similarity via *image hashing* [11]. We manually checked 100 users matched by website and profile picture, respectively. As before, user profiles were considered as matched if the provided information gives high confidence that they belong to the same user. The true positive rate of website and profile pictures are 100% and 98%, respectively, which indicate that users can be regarded as exactly matched.

Search method	#Users analyzed	#Users matched	Matching %
Lookup	176,508	9,316	5.28%
Search	240,000	37,449	18.43%
Total	**416,508**	**46,765**	**11.23%**

Table 4: Twitter user matching results.

To account for limitations with the Twitter APIs, at the time of this writing we were able to analysis a subset of linked accounts from StackOverflow and Github. We ordered accounts according to their popularity (measured by #fol-

[11] `http://hzqtc.github.io/2013/04/image-duplication-detection.html`, Image Duplication Detection

Graph	# Nodes	# Edges	Density
G_{SO}	6672	18995	4.267e-04
G_{GH}	13160	106792	6.167e-04
G_{TW}	16070	829846	2.213e-03

Table 5: Characteristics of the user networks in Twitter, StackOverflow, and Github.

lowees) in Github, and matched them to Twitter accordingly. Table 4 reports the user matching results. We analyzed 416k accounts, specifically 240k by using *Search* and 176k by using *Lookup*. The number of accounts matched are 37k and 9k, respectively, with a total of 46k accounts matched to Twitter.

3 User Interaction across Networks

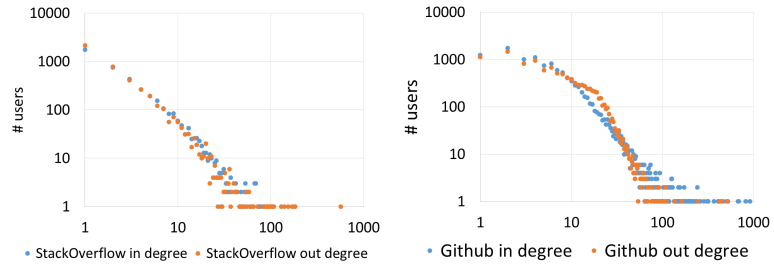
To showcase the type of research that is enabled by a dataset built with our methodology, we designed a study aimed at providing an answer to the following two research questions: *RQ1: how do users connect with each other in different social networks? RQ2: does the relative importance of users vary across social networks?* To this end, we first inferred the interaction networks over the same set of users in the three platforms, then analyzed network features and correlations of user centrality in the three networks.

Building user interaction networks. We constructed two directed graphs G_{TW}, G_{GH} that encode *following* relationships of users in Twitter and Github, respectively, i.e., a directed edge $e = u \rightarrow v$ indicates that user u follows user v. While being absent of explicit following-followee relationship, StackOverflow provides an implicit "help network" among users according to *who answers to whom*. Therefore, we built a directed graph G_{SO} such that an edge $e = u \rightarrow v$ indicates that user u is helped by v, i.e., at least one question of u is answered by v.

Due to the rate limit of Twitter REST API, we built the three user interaction network graphs for the 20k most popular users among the 46k matched users (Table 4). As before, popularity is defined according to #followees in Github.

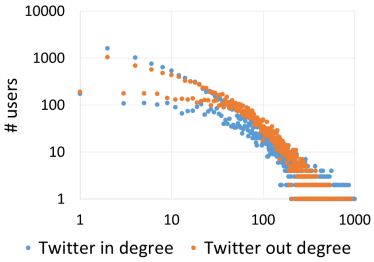
RQ1: How do users connect with each other in different social networks? Table 5 reports basic statistics of the users' networks in the considered social networks. By comparing the #nodes in the three networks, we observe that, in the same set of 20k users, more users are involved in both Github and Twitter interaction networks than those involved in StackOverflow interaction network. This indicates that users are more likely to be active in explicit interaction based on followship than in helping-based interaction.

Comparing the density of these networks, results show that users have similar connection intensity in StackOverflow and Github, both of which are however



(a) StackOverflow in- and out- degree.

(b) Github in- and out-degree.



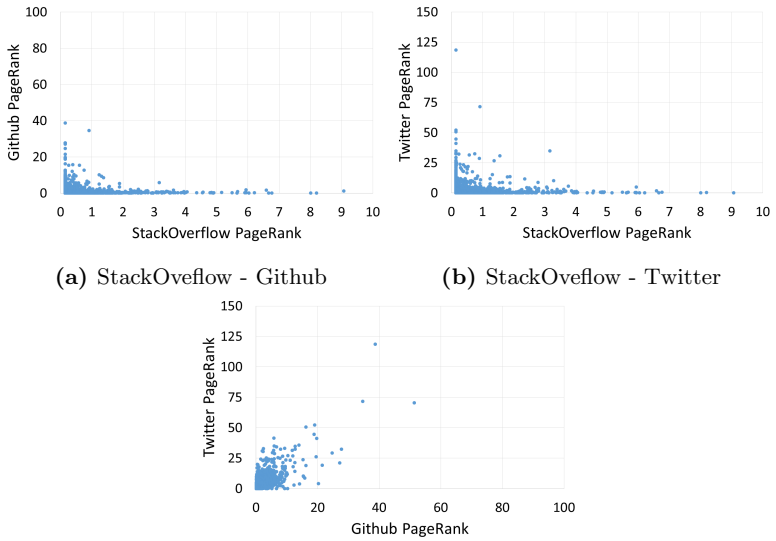
(c) Twitter in- and out-degree.

Fig. 4: Degree distribution for Twitter, StackOverflow and Github networks.

10 times lower than user interaction in Twitter. This would imply that users are more likely to connect with each other in general-purpose social networks like Twitter than in profession-oriented networks like StackOverflow and Github.

Figure 4 shows the in-degree and out-degree distributions over the three networks. In StackOverflow, both distributions conforms to power-law, indicating that most users follow (resp. are followed by) a small number of users, while there is a small number of users that follow (resp. are followed by) many users. In addition, in-degree distribution looks more skewed than out-degree distribution – in other words, users tend to follow the same set of users, who is followed by many users. Similarly in Github and Twitter, in-degree distribution is more skewed than out-degree distribution, indicating that a small number of users are highly popular in the network. Comparing the three networks, StackOverflow is the one that has most similar distributions of in-degree and out-degree. We consider the fact that the StackOverflow helping-helpee network is built implicitly from question-answering activity between users, while the following-followee relations in Github and Twitter are explicitly constructed by users. The result suggests that explicit connection mechanisms result in a more skewed popularity among the users of a platform.

RQ2: does the relative importance of users vary across social networks? To answer this question, we choose to correlate users' centrality scores



(a) StackOveflow - Github **(b)** StackOveflow - Twitter

(c) Github - Twitter

Fig. 5: Pair-wise network centrality correlations.

in the different networks. A high cross-network correlation of user centrality scores would indicate similar user importance in different settings; for instance, a high correlation of user centrality in StackOverflow and Github networks will suggest that a user who is helpful in answering to others' questions in StackOverflow will be followed by many users in Github (and vice versa); on the contrary, a low correlation would indicate that users' activity in one platform is not indicative of their activities in another platform, e.g., an influential user in Github may not likely to answer questions in StackOverflow.

To obtain users' centrality values, we used classic *PageRank* model. We then calculated Pearson correlation of the centrality values for the same set of users in every pair of graphs. Results are shown in Figure 5. For StackOveflow and Github networks, we have a Pearson coefficient of *-0.0185170* that reveals no linear correlation between PageRank values of users on both platforms; this means that, as shown in Figure 5a, most influential users on StackOverflow do not have the same importance on Github and vice versa. Similar remark can be made on StackOveflow versus Twitter, where Pearson correlation is *-0.0014857*. By contrast, in the Github - Twitter case, we observe a Pearson coefficient of *0.7554060*, which implies that user interactions of Github and Twitter networks are correlated.

4 Conclusions and Future Works

We addressed the problem of user matching across StackOverflow, Github and Twitter social networks. We proposed a methodology that combines different matching strategies and makes use of different user attributes and platform-specific services for linking user accounts. Many of the proposed linking strategies can be generalized to other social networking platforms. For instance, most social networking platforms provide REST API's and search, for which the linking techniques *Lookup* and *Search* can be applied. These methods together allow us to obtain much better results than in literature. Our study of interaction networks based on the matched users in the three platforms has provided interesting insights: 1) users in general-purpose social media networks like Twitter are more connected than in profession-oriented social networks like StackOverflow and Github; 2) social networking platforms that enable the functionality of explicit user connection (Github and Twitter) will result in more skewed distribution of user popularity, and more correlated user activities between them, than (with) the one that only provides implicit user connection mechanisms (StackOverflow). As part of future work, we plan to deepen our analysis of the user interaction networks properties such as the formation and evolution of communities, and the topics discussed by the users and communities across the three networks.

References

1. F. Abel, E. Herder, G.-J. Houben, N. Henze, and D. Krause. Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction*, 23(2-3):169–209, 2013.
2. A. S. Badashian, A. Esteki, A. Gholipour, A. Hindle, and E. Stroulia. Involvement, contribution and influence in github and stack overflow. In *CASCON '14 Proceedings of 24th Annual International Conference on Computer Science and Software Engineering*, pages 19–33. ACM, 2014.
3. F. Buccafurri, V. D. Foti, G. Lax, A. Nocera, and D. Ursino. Bridge analysis in a social internetworking scenario. *Inf. Sci.*, 224:1–18, 2013.
4. F. Buccafurri, G. Lax, A. Nocera, and D. Ursino. Discovering missing me edges across social networks. *Inf. Sci.*, 319:18–37, 2015.
5. P. Jain, P. Kumaraguru, and A. Joshi. @i seek 'fb.me': Identifying users across multiple online social networks. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 1259–1268, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.
6. C. Treude, O. Barzilay, and M.-A. Storey. How do programmers ask and answer questions on the web?: Nier track. In *Software Engineering (ICSE), 2011 33rd International Conference on*, pages 804–807. IEEE, 2011.
7. J. Tsay, L. Dabbish, and J. Herbsleb. Influence of social and technical factors for evaluating contribution in github. In *Software Engineering (ICSE), 2014 36rd International Conference on*, pages 356–366. ACM, 2014.
8. B. Vasilescu, V. Filkov, and A. Serebrenik. Stackoverflow and github: associations between software development and crowdsourced knowledge. In *Social Computing (SocialCom), 2013 International Conference on*, pages 188–195. IEEE, 2013.

Social Network and Sentiment Analysis on Twitter: Towards a Combined Approach

Paolo Fornacciari, Monica Mordonini, Michele Tomauiolo

Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Parma
Parma, Italy
e-mail: paolo.fornacciari@studenti.unipr.it,
{monica.mordonini,michele.tomaiuolo@unipr.it}

Abstract. Twitter is a platform which may contain opinions, thoughts, facts and other information. Within it, many and various communities are originated by users with common interests, or with similar ways to feel part of the community. This paper presents a possible combined approach between Social Network Analysis and Sentiment Analysis. In particular, we have tried to associate a sentiment to the nodes of the graphs showing the social connections, and this may highlight the potential correlations. The idea behind it is that, on the one hand, the network topology can contextualize and then, in part, unmask some incorrect results of the Sentiment Analysis; on the other hand, the polarity of the feeling on the network can highlight the role of semantic connections in the hierarchy of the communities that are present in the network. In this work, we illustrate the approach to the issue, together with the system architecture and, then, we discuss our first results.

Keywords: Sentiment Analysis, Social Network, Hierarchical Classification

1 Introduction

An increasing number of people is progressively approaching to the social networking sites, which become more and more popular and complex: within their context many and various communities are originated by users with common interests or with similar ways to feel part of the community. The kinds of analysis as well as information that can be extracted from the social networking sites are varied and increasingly appealing both to the world of marketing and to the social or political one. The classical approach to Social Network Analysis allows to study the topology of a network through the connections that develop within it, giving rise to a hierarchy of communities within the main topic. Furthermore, certain types of social networks, like Twitter, allow to track relationships also in those cases in which knowledge is not mutual: simply a node is a follower of another node. The number of followers defines in part the popularity of a node within the network, but it is not able to point out if this popularity is positive or negative.

On the other hand, the explosion of data on the Web has made the research in automatic cataloging of texts increasingly interesting, as well as the extraction of

information or meta-information and the Sentiment Analysis of a review, an emotion, a tweet. Moreover, in this area the explosion of microblogging, and the use of a simple "like" or a retweet as a form of acceptance or sounding board for information as well as the dynamism and the speed with which everyone reads and writes content make the analysis of these opinions hard, if you use the methods of text mining, while they introduce, or amplify, new issues and problems for sentiment analysis (such as citations, irony, role of emoticons) that are difficult to deal with regardless the context in which they are written.

This paper presents a combined approach between Social Network and Sentiment Analysis. In particular we have tried to introduce some kind of information about sentiments on the graphs showing the results of the Social Network Analysis (SNA): in this way we hope to highlight other potential correlations among the nodes of net under examination. The idea behind it is that, on the one hand, the network topology and the selected topics of the network can contextualize and then, in part, unmask some incorrect results of the Sentiment Analysis (SA), and, the other hand, the polarity of the feeling on the network can highlight the role of semantic connections, as a possible foundation for the organization and the hierarchy of the communities highlighted by the Social Network Analysis.

In the following, after a brief description of the background, the system architecture will be showed, together with the choices which we made, then some results obtained from the initial evaluation of the system will be discussed.

2 Background

SNSs are a collection of web-based services that allow users to build a profile within the system and define a list of other users with whom they have some kind of connection [7]. The architecture of social networking platforms is very differentiated. While the most popular platforms are built as essentially centralized systems, other platforms have a distributed architecture [10][11]. The decentralized systems try to address some of the risks associated with online social networking, which are often perceived as quite serious by many users and have already led to serious incidents [6]. SNA has the objective to model social structures with different properties, starting from the mathematical theory of graphs and the use of matrix algebra, and is often augmented though computer-based simulations [11]. SA is a branch of Opinion Mining, that aims to listen and process the data that users post on social media. Generally SA classifies web comments into positive, neutral, and negative categories. To make these systems more intelligent ad flexible, a deeper analysis of affective knowledge could be incorporated [9][8]. In some case an ontology driven approach is used [5][24][3].

In this research work, we built a system for social network and sentiment analysis, which can operate on Twitter data, one of the most popular social networks. The analysis of large amount of data is an exciting challenge for researchers, but it is also crucial for all those who work at different levels in the current information society: Twitter has been the subject of attention from researchers as early as 2009 [13].

Some recent studies about American candidates are important for understanding how public sentiment is shaped and its polarization [19]. In [2] geo-spatial information related to tweets is used for estimating happiness in the Italian cities. Being Twitter a microblogging service, the techniques used generally in SA and Text Classification must be adapted to the famous 140-character tweet and this opens the way for new issues [1][17][16][26].

Another quite important problem to work on Twitter data is how to automatically collect a corpus for SA and, in general, Opinion Mining purposes: example of how to perform this task is in, for example, [21][14].

3 System Architecture

In this paragraph we describe our system for social network and sentiment analysis, which can operate on Twitter data.

Twitter is a platform which may contain opinions, thoughts, facts, references to images and other media and, recently, stream video filmed live and put online by users. So it is more than just a SNCs in which a user displays and increases their social relationships, it is a real communication channel in which a user can choose its topics and its node of reference according to his interests and culture.

A study of the network topology and the number of interconnections of a node are able to highlight the communities in the network and also in part to how the information is propagated, but they are not able to say anything about the degree of agreement and cohesion of members of a community. To solve this task you need to carry out an investigation into the semantic content of the messages.

Compared to the problems of classic data mining, sentiment analysis shows many difficulties in terms of effectiveness. This is mainly due to the subtle distinction that exists between positive and negative sentiment or between neutral and positive one. Let us suppose for example a sentence containing irony or sarcasm, where the interpretation of the meaning is strictly subjective. In this case, two human beings may be in disagreement about the real feeling that it expresses. Furthermore, not always the opinions are expressed through the use of opinion words, in many cases the special language constructs (such as the figures of speech) come into play.

Difficulties also are due to the use of non-formal expressions and slangs that do not belong to the vocabulary of a language. These terms are often used in an intensive way to express a particular opinion or a certain mood.

Additional problems are due to the domain of the subject: in particular we note that the feelings that are expressed by a word are often dependent on the topic. We look at this sentence as an example: “It's quiet!”. It shall render a positive opinion if we are talking of a car engine, but it reveals a disapproval if the matter of discussion is a phone.

As a microblogging service, Twitter is used to publish short messages counting a maximum of 140 characters (tweets). This characteristic if one side it may seem easier because it forces people to take a position, on the other side the few words not allow

the user to repeat concepts or emotions: he rather uses slangs shared by the community, emoticons and punctuation.

Besides the ease of retweet increases the difficulty in perceiving what is the real feeling of the user who runs it and the intense use of citations can also distort the sentiment enclosed in the tweet.

However, by combining the information of SA with those of the SNA we can hope to disambiguate some actual cases and the opportunity to know the slang of the channel under examination can improve the efficiency of machine learning algorithms for the SA.

3.1 Social Network Analysis: data selection

As a social networking platform, Twitter is structured as a directed graph, in which each user can choose to follow a number of other users (followees), and can be similarly followed by other users (followers). Thus, the "follow" relationship is asymmetrical, it does not require mandatory acknowledgement, and it is essentially used to receive all public messages published by any followee user.

Consequently, in our analysis we collected three types of data (Fig. 1): the User type represents users' profiles; the Tweet type represents posted messages; the Friend type represents the "follow" relationships among users.

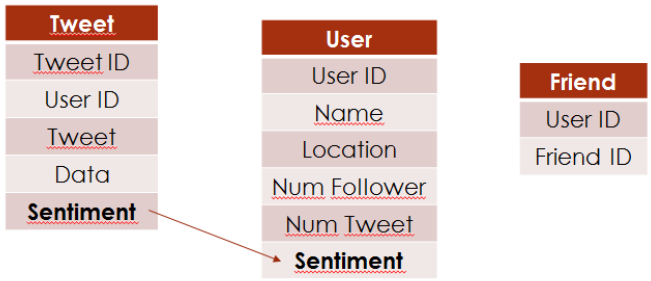


Fig. 1. The structure of the DataBase.

Apart from data obtained directly from Twitter, we added a field to both tweets and users, to associate a sentiment with them, according to the result of our SA. Currently if a user posts more than one tweet on the net, we decided to associate to him the sentiment of the last tweet that he posted.

3.2 Sentiment Analysis

As a communication medium, tweets have a quite peculiar nature. Some distinguishing features of communication on Twitter are related to technical aspects; those include length of text, tags, urls, etc. Other features may be classified as idiomatic use of the medium, and create a sort of Twitter culture.

As a start, a tweet may contain many elements that are not significant for our classification, and can thus be dropped though a filtering process. To polish the message, we defined various filters, that can applied in a customizable sequence. An example is shown in Fig.2.

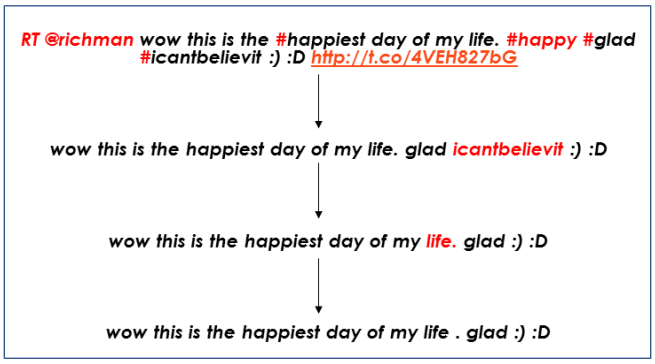


Fig. 2. The sequence of cleaning the tweet.

A first filter eliminates useless tokens such as: the "RT" sequence; the @ character and the whole following user name; the # symbol, but not the following topic name, which is kept in the message. The topic name is also removed, though, when it coincides with the name of the channel where tweets are collected from.

A second filter applies the language specific rules. It includes an orthographic correction of the message, which is used to remove unknown words (in the example: "icantbelievit") and other filtering processes for stemming and removal of stopwords.

Finally, another filter separates all punctuation symbols from the text, and organizes them as single-character words. Even if smiles sequences, repeated question and exclamation marks are kept as aggregates because they are important patterns for the classification.

The final result of the filtering process is a word vector, which is then submitted to a set of classifiers.

We use a set of classifier to identifying the following classes of messages: undiscriminated, objective, subjective, positive, negative. Moreover, there is a class in which the system put all the tweets that are too short to be classified. The system is organized as a simple hierarchy of agents, mimicking the hierarchy of sentiment classes. In fact, since objective messages have no polarity by definition, the classifier for positive and negative sentiments is only applied to subjective messages (see Fig. 3). One advantage of this framework for classifiers is the ease with which you can add classifiers trained to identify other emotions. In fact, hierarchical classification has been applied successfully in a number of studies, for information retrieval [23]. It has been proven effective especially in the case of classification over hierarchical taxonomies. Also in the case of sentiment analysis, a hierarchy of classes can be defined [12][5]. Accordingly, hierarchical classification has already been applied to sentiment analysis, too [23].

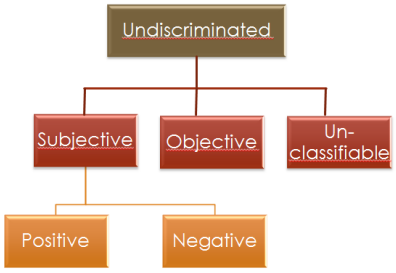


Fig. 3. Hierarchy of basic sentiment classes.

Each classifier is based on Multinomial Naive Bayes algorithm, that one of the most popular methods used in SA. We have selected it because it seems to be the most suitable to generate and process large sets of features. In fact, instead of generating a training set by hand, we aimed at realizing an automated (or at least semiautomated) process for obtaining good training sets. In our methodology, the training sets are obtained through the automatic elaboration of some particular streams of tweets and comments, obtained directly from Twitter, without any manual classification. Thus, each training set may contain an important number of wrong data. Nevertheless, we show that they can be used to obtain useful results.

About the objectivity/subjectivity classifier, we adopted a similar strategy to [20]. In fact, to obtain objective content, we gathered messages generated from popular news agencies. In our tests, we used the following list: *@ABC*, *@BBCNews*, *@BBCSport*, *@business*, *@BW*, *@cnnbrk*, *@CNNMoney*, *@fox32news*, *@latimes*, *@nytimes*, *@TIME*. To obtain subjective content, instead, we gathered comments directed to the same list of users.

About the polarity classifier, we used different sources, thus generating training sets which do not overlap with those about objectivity/subjectivity. In fact, we used sources of mostly positive or negative messages, respectively. On the one hand, those sources should fit the particular setting of Twitter (short messages, idiomatic expressions, smiles, etc.). On the other hand, they should not be specific to a particular topic or context (sport, music, etc.). Thus, we dropped the idea of collecting messages about particular events, mostly generating either positive or negative sentiments. Instead, we collected messages, using generic yet polar terms as queried hashtags. In particular, we used the following channels to gather positive content: *#adorable*, *#awesome*, *#beautiful*, *#beauty*, *#cool*, *#excellent*, *#great*. We used the following channels to gather negative content: *#angry*, *#awful*, *#bad*, *#corrupt*, *#pathetic*, *#sadness*, *#shame*. Actually, such terms have been chosen quite empirically, taking into account the quality of training sets they generated. But they could be selected from WordNet-Affect [24], SentiWordNet [3], and other affective lexicons, in a more systematic way.

In this way, the training set is generated in an automated fashion, as a list of tweets. Each tweet is associated with its supposed class, in accordance to its source. In fact, the training set is not perfect, as it contains messages gathered from public channels.

However, a training set of this kind can be generated easily and in a methodical way, from real and updated Twitter messages. Moreover it is possible to extend this approach to train a classifier to recognize feelings which are written in a particular slang.

Feature	$P(F_i \mid pos)$	$P(F_i \mid neg)$
:)	**0,0025**	0,00055
stupid	0,000098	**0,00065**
thank you	**0,0012**	0,00029
!!!	0,0028	0,0019

Feature	$P(F_i \mid obj)$	$P(F_i \mid sub)$
!!!	0,000031	**0,002**
:(	0,0000079	**0,00030**
%	**0,0013**	0,00080

Fig. 4. Generated model for the classifier: example of selected feature and their probabilities in the polarity (on the left) and subjective (on the right) classifiers.

In Fig. 4 there are some examples of features which are selected by the classifiers together their probabilities. It is worth noting that these are consistent with what we expected: the emoticons ':)' has a high probability of being in positive phrases, while the pattern '!!!' is very significant for the classifier of the subjectivity but it is a useless feature to determine the polarity of a tweet.

4 Experimental results

In this section, we will report the results of the classifiers and the analysis carried out on a couple of case studies. Using the methodology and the software which we described in Section 3, it is possible to obtain some generic training sets for the classifiers. This phase was carried out before selecting the final case studies. In our settings, they consist of:

- 86000 instances (polarity);
- 32000 instances (subjectivity).

These instances have been obtained by exploring more than 60 channels on the social network. In the generated models, the selected features are consistent with our expectations: the typical expressions of a certain feeling (such as smileys, or some words that express appreciation or disgust) show a higher probability of belonging to the class of that feeling, rather than to the class of the opposite sentiment.
The results obtained by the classifiers using cross-validation (with folds = 10) on the training sets showed an accuracy of:

- 77,45% (polarity classifier)
- 79,50% (subjectivity classifier)

These results show that the model of the classifiers contains effective features for the recognition of the sentiment of a message.

The case study which was considered in this work is the social network of the *#SamSmith* channel (the singer who won four awards at the Grammy Awards 2015). The choice of this channel is justified by the strong similarities found between the

type of the published tweets and the instances used for training the classifiers. All data were downloaded between 2015-02-02 and 2015-02-10. The awarding of the Grammy took place on 2015-02-08. The social network (shown in Fig. 5) consists of a total of 5570 nodes (users) and 6886 arcs ("follows" relationships). Nodes are deployed according to the ForceAtlas2 algorithm [15], which turns structural proximities into visual proximities, thus highlighting communities.

Looking at the figure, it is possible to notice that the network topology is consistent with the nature of the considered case. In fact, most of the channel consists of independent users (or small groups of users) that express their opinion about the artist; however, in the central part of the network there are some major communities.

As shown in Fig. 5, the prevailing sentiment detected from the classifier is the negative one. Performing an analysis on a sample of tweets in the network, we noticed that many sentences are actually quotes of songs. These messages contain melancholic and sad phrases, and are therefore classified as negative. Considering that a quote is generally an appreciation for the artist, most users classified as negative are actually positive users. This is a typical example of a classic problem of misunderstanding of the SA: the system, while classifying correctly the tweet, misses the assessment of the feeling because it can not evaluate the tweet together with its context.

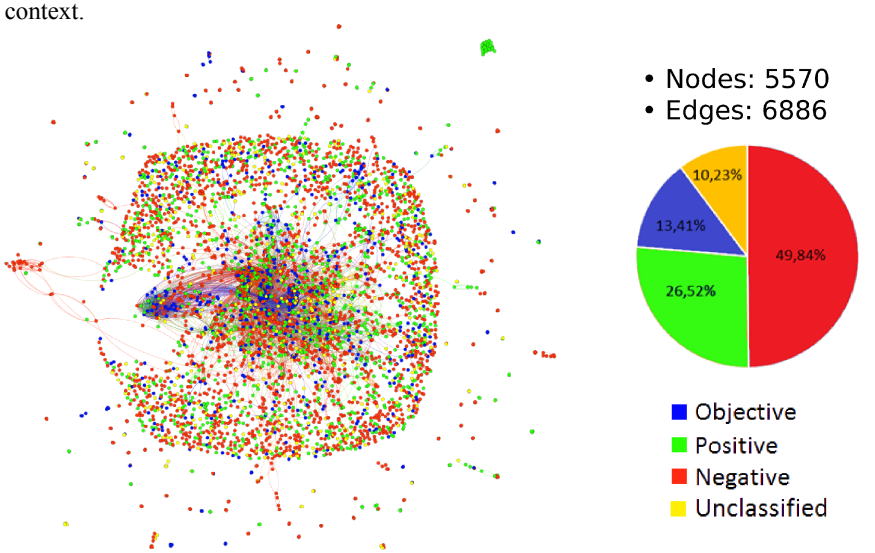


Fig. 5. Combined analysis of the *#SamSmith* channel.

For evaluating the performances of our system, we conducted a simple survey through a group of persons in our department. In this way, we selected and classified 100 messages that show a clear opinion on the singer. Then, we used those messages as a test. The results of the classifiers showed an accuracy of 84% for the polarity and 88% for subjectivity.

In the network periphery (at the top-right corner of Fig. 5), it is possible to notice a small group of users whose feeling is completely positive. After a careful analysis of users' tweets in this small group, it was found that these posts are mainly retweets and the original messages are only two. Of these two messages, the first is actually positive, while the other one is objective. This episode shows how some errors of assessment can have important impact on larger communities.

In addition to the *#Samsmith* channel, we considered the social network associated with the *#Ukraine* channel, trying to obtain some particularly significant results, above all from the point of view of network topology. In fact, the crisis in the region could lead to a quite sharp division on the Web. This work is still in progress, nevertheless we can show here some results which we already obtained.

At the moment we have downloaded the data, the network consisted of:

• 26131 nodes

• 1163588 edges

In Fig. 6, it is possible to see the main results of our analysis on the network.

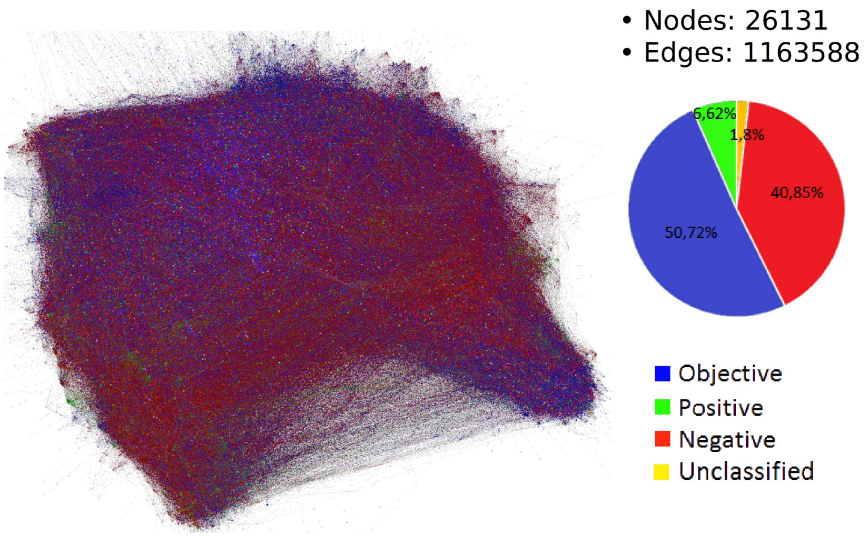


Fig. 6. Combined analysis of the *#Ukraine* channel.

The more evident thing to notice, is that the prevailing color in the network is blue (objective tweets), and the next one is red (negative tweets). Given the nature of the channel we are considering, which essentially reflects a social tragedy, the sentiment we have found through the analysis is quite plausible. However, analyzing some random messages, we have noticed a number of errors in the classification of these tweets. In particular, some objective sentences are often classified as negative ones, while some sentences expressing essentially hope (and thus positive) are classified as objective ones. In our opinion, the reason for these errors is related to the type of

features contained in the model of classifiers, which possibly are not a good fit for this particular case study.

The case of Ukraine has been discussed quite largely in traditional media, too, for the supposed role of "trolls" operating on new media to influence the public opinion [25]. In fact, this may represent, as a modern reproposition, the quite classical case of opposing propaganda campaigns, this time carried on through social media. Also for this reason, we analyzed the social communities participating in the channel. We focused on the most active users, who contributed with at least 6 tweets during the whole week we considered (mid July 2014). In fact, among those it is more probable to find candidate opinion makers. The analyzed subnetwork represents around a tenth of the original network, and precisely consists of:

- 3261 nodes
- 84307 edges

We used the community detection algorithm provided with Gephi, at various resolution levels [18]. Quite interestingly, we were able to identify quite clearly two major communities. Additionally, some much smaller communities were found.

	Full Network	**Community 1**	**Community 2**
Average degree	51.706	53.021	42.649
Diameter	7	7	6
Radius	1	4	4
Avg path length	2.511	2.248	2.334
Shortest paths	10591776	3152400	2014980
Graph density	0.016	0.030	0.030
Clustering coeff.	0.420	0.480	0.414
Total triangles	873460	540526	281524

Table 1. Features of the main communities detected on the *#Ukraine* channel.

Looking at data reported in Table 1, it is easy to notice that the two communities, corresponding to opposing factions in the crisis, have a quite similar size. Moreover, also their main features are quite similar. This seems to indicate that the two camps have a quite similar internal social organization, at least at the macroscopic level. Nevertheless, both the communities have high density, almost doubling the value of the whole network. This means that, in fact, there is a quite clear separation between those two communities, which have relatively few shared connections.

Our sentiment analysis has not highlighted significant differences in the emerging opinions in the two communities. In fact, they largely share the same negative outlook of the whole network. This is an issue that we plan to analyse in deeper detail in future. The emerging sentiment in each camp may also vary during time, and in particular in correspondance with major events and turnpoints in the crisis.

5 Conclusions

This study reports the initial results we obtained from the synthesis of Social Network Analysis and Sentiment Analysis. We experimented our approach on a couple of Twitter channels, as case studies. In particular, we considered the *#SamSmith* channel during the Grammy Awards in 2015, and the *#Ukraine* channel during the crisis of 2014. Apart from the particular results, a methodology and some guidelines for the automatic classification of Twitter content have been discussed.

The implemented software allows: (*i*) to get a training set for the classifiers that deal with Sentiment Analysis, and (*ii*) to make a thorough study of the network topology. The study of the global sentiment within the network has highlighted the typical problems of Sentiment Analysis (irony, sarcasm, lack of information, etc.). Additionally, some peculiar problems of the considered channel were also detected (such as the quotes of songs). Also, the analysis of biased channels, may pose additional difficulties.

The performances obtained by the classifiers during tests conducted on the training set and the analysis of the case studies have shown good and promising results.

References

1. A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau. “Sentiment analysis of Twitter data”, Procs of the Workshop on Languages in Social Media (LSM '11), Association for Computational Linguistics, USA, pp. 30-38, 2011.
2. L. Allisio, V. Mussa, C. Bosco, V. Patti, and G. Ruffo, “Felicittà: Visualizing and Estimating Happiness in Italian Cities from Geotagged Tweets,” Proc. of the 1st Int. Workshop on Emotion and Sentiment in Social and Expressive Media: Approaches and Perspectives from AI (ESSEM 2013), Turin, Italy, 2013.
3. A. E. S. Baccianella and F. Sebastiani, “Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining,” Proc. of the 7th Conf. on Inter. Language Resources and Evaluation (LREC’10), ELRA, 2010.
4. F. Bergenti, A. Poggi, and M. Tomaiuolo, “An Actor Based Software Framework for Scalable Applications,” Lecture Notes in Computer Science, 8729, pp. 26-35. 7th International Conference on Internet and Distributed Computing Systems (IDCS), 2014.
5. M. Baldoni, C. Baroglio, V. Patti, and P. Rena, “From tags to emotions: Ontology-driven sentiment analysis in the social semantic web,” Intelligenza Artificiale, vol. 6(1), pp. 41-54, 2012.
6. E. Franchi, A. Poggi, and M. Tomaiuolo, “Information and Password Attacks on Social Networks: An Argument for Cryptography,” Journal of Information Technology Research (JITR), 8(1), 25-42, 2015. doi:10.4018/JITR.2015010103
7. D. Boyd, N. Ellison, “Social Network Sites: Definition, History and Scholarship,” Journal of Computed-Mediated Communication, vol. 13 (1), pp. 210-230, 2008.
8. K. Ca, S. Spangler. Y. Chen; L. Zhang, "Leveraging Sentiment Analysis for Topic Detection," Web Intelligence and Intelligent Agent Technology (WI-IAT '08), IEEE/WIC/ACM Int. Conf. on, vol.1, pp. 265-271, 2008.
9. E. Cambria, B. Schuller, Y. Xia, C. Havasi , "New Avenues in Opinion Mining and Sentiment Analysis", IEEE Intelligent Systems, vol.28, no. 2, 2013.

10. E. Franchi, A. Poggi, and M. Tomaiuolo, “Open Social Networking for Online Collaboration,” Int. J. e-Collab, IGI Global Publisher, vol. 9(3), pp- 50-68, 2013.
11. E. Franchi, and M. Tomaiuolo, “Distributed social platforms for confidentiality and resilience,” Social Network Engineering for Secure Web Data and Services, IGI Global Publisher, pp 114-136, 2013.
12. V. Francisco, P. Gervas, and F. Peinado., “Ontological reasoning to configure emotional voice synthesis,” Procs of Web Reasoning and Rule Systems, vol. 4524 of LNCS, pp. 88–102. Springer, 2007.
13. A. Go, L. Huang, and R. Bhayani, “Twitter sentiment analysis,” Final Projects from CS224N for Spring 2008/2009 at The Stanford Natural Language Processing Group, 2009.
14. S. Hassan, F. Miriam, H. Yulan, and A. Harith, “Evaluation datasets for Twitter sentiment analysis: a survey and a new dataset, the STS-Gold,” Proc. of 1st Int. Workshop on Emotion and Sentiment in Social and Expressive Media: Approaches and Perspectives from AI (ESSEM 2013), Turin, Italy, 2013.
15. M. Jacomy, T. Venturini, S. Heymann, M. Bastian, “ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software,” PLoS ONE 9(6), 2014. doi:10.1371/journal.pone.0098679
16. S. Kiritchenko, X. Zhu, and S. Mohammad, “Sentiment analysis of short informal texts,” J. Artif. Int. Res., vol. 50(1), 2014.
17. A Kowcika, A. Gupta, K. Sondhi, N. Shivhre, R. Kumar, “Sentiment Analysis for Social Media”, Int. J. of Advanced Research in Computer Science and Software Engineering, vol. 3(7), 2013.
18. R. Lambiotte, J.C. Delvenne, M. Barahona, “Laplacian dynamics and multiscale modular structure in networks”. arXiv preprint arXiv:0812.1770, 2008.
19. S. Mohammad, X. Zhu, S. Kiritchenko, J. Martin, “Sentiment, emotion, purpose, and style in electoral tweets”, Information Processing & Management, Elsevier, vol 50 (1), 2014.
20. B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: sentiment classification using machine learning techniques”. Procs of the ACL-02 conf. on Empirical methods in natural language processing (EMNLP '02), Association for Computational Linguistics, vol. 10, USA, pp. 79-86, 2002.
21. A. Pak and P. Paroubek, “Twitter as a corpus for sentiment analysis and opinion mining,” in Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC’10). Valletta, Malta: European Language Resources Association (ELRA), May 2010.
22. M. Shams, M. Saffar, A. Shakery, and Faili,. “Applying sentiment and social network analysis in user modeling” Procs of the 13th int. conf. on Computational Linguistics and Intelligent Text Processing – Vol. Part I (CICLing'12), Springer-Verlag, Berlin, 2013.
23. Jr. Silla, and A. Freitas, “A survey of hierarchical classification across different application domains,” Data Mining and Knowledge Discovery, vol. 22(1-2), pp. 31-72, 2011.
24. C. Strapparava and A. Valitutti. “WordNet-Affect: an affective extension of WordNet,” Procs of 4th Int. Conf. on Language Resources and Evaluation (LREC’04), vol. 4, pp 1083–1086, 2004.
25. S. Walker, “Salutin' Putin: inside a Russian troll house”, The Guardian, 2015-04-02.
26. X. Zhu, S. Kiritchenko, and S. Mohammad, “NRC-Canada-2014: Recent improvements in sentiment analysis of tweets,” Procs of the Int. Workshop on Semantic Evaluation, Dublin, Ireland, 2014.

When food matters: identifying food-related events on Twitter

Eleonora Ciceri, Ilio Catallo, Davide Martinenghi and Piero Fraternali*

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy
first.last@polimi.it

Abstract. Food communities in Twitter are growing every year, and food-related content permeates everyday conversations. Users meet on Twitter to share recipes, give cooking advices or simply inform others about what they are eating. While some of these food-related conversations are not associated with any special occurrence, many conversations take place instead during specific events. The detection of food-related events gives interesting insights: people do not talk only about Halloween and Easter, but they also create their own food-related events, such as the promotion of products (e.g., an online petition to propose the production of bacon-flavored chips) or themed home-made recipes (e.g., a day of recipes dedicated to chocolate). In this paper, we propose an approach that accurately captures food-related content from the tweet live stream, and analyze the detected conversations to identify food-related events. The proposed technique is general as it can be applied to the identification of other thematic events in digital streams.

1 Introduction

Recently, Twitter has received much attention from the research community. It is reported[1] that 500 million tweets are published on a daily basis. Tweets cover a variety of topics, ranging from personal status updates (e.g., *"going to the gym"*) to local and global news (e.g., *"FBI investigating possible corruption at New York prison"*). Tweets may contain hashtags, i.e., words prefixed with the hash symbol `#`, which allow tweets with similar topics to be identified. Users interested in specific topics can search for relevant tweets by hashtags, which make it particularly easy for users to create conversations about specific events. In the following, we denote by *event* a recognizable happening of limited duration [7]. While some topics are extemporary, news-based, or tied to some specific real-world occurrence, others are always discussed, permeating from everyday conversations and involving large communities. An example is *food*: food bloggers, food celebrities, media channels and common users discuss about themes such as food for holidays, cooking advices for singles, and virtual recipe sharing

* This work is partly funded by the EC's FP7 "Smart H2O" project, and the EU and Regione Lombardia's "Proactive" project

[1] `https://about.twitter.com/company`

parties. Food conversations, as for other topics with a wide coverage in social media, permeate several events, which originate either within the boundary of the digital community (e.g., `#TacoTuesday`) or in the real world (e.g., `#easter`). Despite the huge adoption of Twitter as a platform for publishing and talking about events, their automatic detection still remains an open problem [4]. Indeed, given the availability of such a diverse assortment of tweets, it is still not completely clear how to automatically recognize a given hashtag (and its related stream of tweets) as being associated with an event.

In this paper, we propose a technique for the automatic detection of *topic-related events*, i.e., events pertaining to a given topic of interest. More precisely, we devise a two-step detection procedure: we first identify hashtags related to a given topic of interest, and then analyze them in order to extract the associated topic-related events. We show that, when applied to food-related events, our method is able to successfully identify relevant events among the top-1000 hashtags, attaining 100% Precision@10, and 80% Precision@172. Moreover, in addition to common food-related celebrations such as `#easter`, the proposed technique also manages to identify more Twitter-specific initiatives, such as `#MeatlessMonday`. Nevertheless, note that our technique is applicable to several other contexts, including disaster management, breaking news and political events.

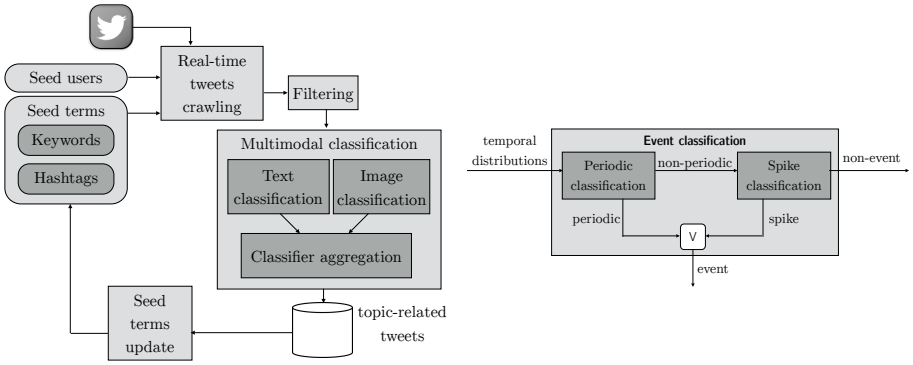
The remainder of this paper is organized as follows. We formally introduce the *topic-related event detection problem* in Section 2. In Section 3, we introduce our process for the retrieval and subsequent identification of topic-related tweets. In Section 4, our approach for the detection of events is presented. In Section 5, we demonstrate the effectiveness of our method on a real-world scenario. In Section 6, we discuss about the related works in the literature, just before our final conclusions and discussion of future works in Section 7.

2 Topic-related event detection: problem statement and proposed approach

Let $\mathcal{T} = \{\theta_1, \ldots, \theta_N\}$ denote the tweet set obtained from observing the tweet live stream for a certain amount of time. Each tweet $\theta_j = \langle \omega_j, I_j, \mathcal{H}_j \rangle$ is composed of a textual component ω_j, a (possibly empty) image component I_j, and a set of related hashtags $\mathcal{H}_j$. Moreover, let τ denote a topic of interest. If we indicate with $\mathcal{Y} = \{\mathtt{Y}, \mathtt{N}\}$ the set of relevance classes for the topic τ, we can associate each tweet θ_j with a label $y_j \in \mathcal{Y}$, such that $y_j = \mathtt{Y}$ if tweet θ_j is related to topic τ, and $y_j = \mathtt{N}$ otherwise. By considering the set of the sole relevant tweets $\mathcal{T}^R = \{\theta_j : y_j = \mathtt{Y}\} \subseteq \mathcal{T}$, and defining $\mathcal{H}^R = \bigcup_{j:\theta_j \in \mathcal{T}^R} \mathcal{H}_j$ as the set of hashtags extracted from $\mathcal{T}^R$, we can therefore formulate the *topic-related event detection problem* as that of finding a set of topic-related hashtags $\mathcal{F} \subseteq \mathcal{H}^R$ that are also associated with an event.

In order to solve the event detection problem, we devise the following two-step procedure.

1. **Topic-related tweet retrieval**. Each tweet entering our system is classified as relevant/non-relevant for the topic τ. Specifically, to determine the rele-



(a) Topic-related tweet retrieval pipeline (b) Event detection pipeline

Fig. 1. Conceptual model of the two-step procedure for the detection of topic-related events

vance to topic τ we adopt a multimodal classification approach [23], which combines textual and image classification.

2. **Event classification**. For each hashtag in $\mathcal{H}^R$, we count its daily occurrences to obtain its temporal distribution (which conveys the change of its usage over time). Temporal distributions are used to classify the hashtags as either event-related or event-unrelated.

These two phases are implemented as independent processes, discussed in Section 3 and Section 4, respectively.

3 Topic-related tweet retrieval

The process for the retrieval and classification of topic-related tweets is illustrated in Figure 1(a). The system identifies topic-related tweets from the live stream in three phases: *crawling*, *filtering* and *classification*. Thanks to the presence of a feedback loop, the system automatically follows the topics users are currently discussing, thus adapting the crawling step to the emerging trends in conversations. Let us comment in greater detail on each such step.

3.1 Crawling phase

Let a *seed user* denote a user which was identified by a domain expert as relevant to topic τ. Moreover, let $\mathcal{S}$ be a set of tweets manually labeled as relevant/non-relevant to topic τ. A *seed term* (either keyword or hashtag) is a term that appears frequently in positively labeled tweets and rarely in negatively labeled tweets in $\mathcal{S}$. The crawling module monitors the tweet live stream[2], and retains tweets meeting at least one of the following selection criteria: *i)* authored by a seed user; or *ii)* containing at least one relevant seed term.

[2] Within the limitations of the Twitter's terms of service.

3.2 Filtering phase

The collected tweets proceed in input to the filtering module, which discards a tweet if at least one of the following conditions holds: *i)* the tweet content is not written in English, *ii)* the tweet contains inappropriate words, or *iii)* the tweet contains words belonging to a topic-dependent set of stop words (e.g., "apple" in the case of food).

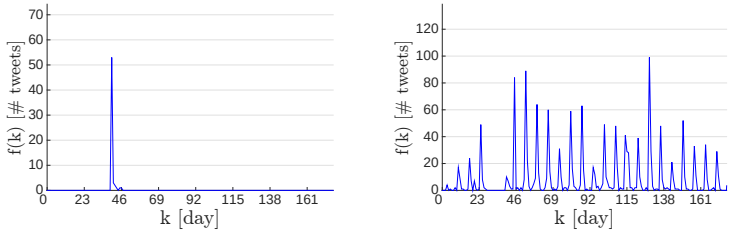
3.3 Classification phase

This step consists of a classification phase, at the end of which each tweet is labeled as relevant/non-relevant to the topic τ. We first disaggregate each tweet $\theta_j \in \mathcal{T}$ in its constituting components ω_j and I_j, and then use a textual and an image classifier to obtain two independent opinions on the relevance of ω_j and I_j to the topic τ. Finally, we merge these opinions to obtain a unique relevance label y_j for the tweet θ_j.

Text classification. We collected a dataset of tweets $\mathcal{T}^\omega$ (such that $\mathcal{T}^\omega \cap \mathcal{T} = \emptyset$) and manually annotated their textual components ω_j with a label $y_j^\omega \in \mathcal{Y}$, which specifies the relevance of ω_j w.r.t. topic τ. Each textual component ω_j is subdivided in terms. User mentions (written as `@username`) and stop words are deleted from the list of extracted terms, since they are not attributable to a specific topic. On the contrary, hashtags (after trimming the `#` symbol off) are kept as discriminative features. Finally, terms are normalized by lowercasing letters and applying Porter stemming [20], and the feature vector x_j^ω is computed according to a TF-IDF approach. To train the classifier and assess its performance, we split $\mathcal{T}^\omega$ in training set $\mathcal{T}^\omega_{\text{train}}$ (60%), cross-validation set $\mathcal{T}^\omega_{\text{CV}}$ (20%) and test set $\mathcal{T}^\omega_{\text{test}}$ (20%). An SVM classifier with RBF kernel is trained on the set $\{(x_j^\omega, y_j^\omega)\}_{j:\theta_j \in \mathcal{T}^\omega_{\text{train}}}$. The combination of the classifier parameters (i.e., the regularization parameter C and the kernel width σ) that guarantees the best performance on the cross validation set $\mathcal{T}^\omega_{\text{CV}}$ is selected, and the classifier performance is computed on the test set $\mathcal{T}^\omega_{\text{test}}$.

Image classification. We collected a dataset of tweets $\mathcal{T}^I$ (such that $\mathcal{T} \cap \mathcal{T}^I = \emptyset$) and manually annotated their image component I_j with a label $y_j^I \in \mathcal{Y}$, which specifies the relevance of I_j w.r.t. topic τ. An equal (and small) amount of positive and negative samples is extracted from $\{I_j\}_{j:\theta_j \in \mathcal{T}^I}$, and their key-points together with the related SIFT descriptors [16] are computed. By applying k-means clustering, we aggregate the extracted descriptors in K clusters, and use the centers of the learned clusters as representative terms: they characterize the visual dictionary $\mathcal{W}$. Each image I_j is then analyzed to extract its feature vector: i) we extract the key-points of I_j and the related descriptors; ii) for each key-point, we select from $\mathcal{W}$ the three most similar terms; iii) we build a histogram of occurrences of the selected terms; iv) we normalize the histogram x_j^I, which represents the feature vector for the image I_j. The set of collected visual samples is subdivided in training set $\mathcal{T}^I_{\text{train}}$ (60%), cross validation set $\mathcal{T}^I_{\text{CV}}$ (20%) and test set $\mathcal{T}^I_{\text{test}}$ (20%). An SVM classifier with RBF kernel is finally trained on the available training set $\{(x_j^I, y_j^I)\}_{j:\theta_j \in \mathcal{T}^I_{\text{train}}}$, and performance is computed on $\mathcal{T}^I_{\text{test}}$.

Classifier aggregation. In case tweet θ_j is made of a single component (i.e., either ω_j or I_j) the aggregation is not necessary. When both text and image



(a) `#NationalCheesecakeDay` (b) `#MeatlessMonday`

Fig. 2. Temporal distributions of a spike event (a) and a periodic event (b)

content exist, we aggregate the classifiers opinions, with the method proposed in [24], which applies Bayesian formalism and belief functions to estimate the aggregated label y_j.

4 Event classification

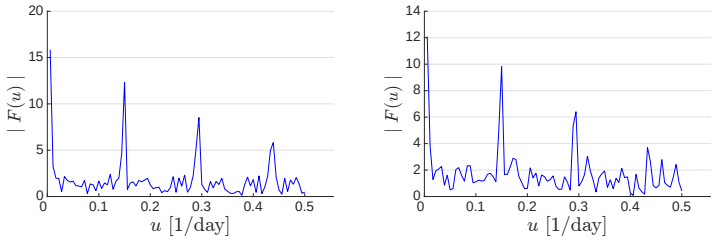
Twitter users track content related to specific topics using hashtags. Some tags are just used to describe content, so that it can be easily classified and retrieved in the future. Other hashtags are meant to track *real-world events* (e.g., earthquakes, holidays, elections) and *social events* (e.g., birthday of a social community).

When an event occurs and users start talking about it, the rate of usage of the related hashtag(s) increases rapidly, and it stays off-the-scale with respect to other common hashtags until either the event ends or the community loses interest in it. To study the rate of usage of hashtags, one can analyze their *temporal distributions*. A temporal distribution is a K-dimensional histogram associated with hashtag H, where the k-th component indicates the number $f(k)$ of tweets produced during day k that contain H. Two examples of temporal distribution are shown in Figure 2.

In this paper, we identify topic-related events by tracking temporal variations in the usage of hashtags. We start from a collection of tweets related to topic τ downloaded as described in Section 3. For each hashtag in the collection, we extract its temporal distribution, and use a supervised approach to decide if the hashtag is related to an event.

4.1 Tracked events

Events discussed on Twitter have different natures. Some events happen once, and generate a large interest (although limited in time). For these events, which we call *Spike Events*, there is a single (and strong) perturbation in the usage of related hashtags. An example of spike event is shown in Figure 2(a). Here, a single activity peak on the hashtag `#NationalCheesecakeDay` was detected, as the Twitter food-related community joined the event by massively publishing cheesecake recipes in a limited amount of time.



(a) `#MeatlessMonday` (b) `#TacoTuesday`

Fig. 3. Fourier transforms of two periodic events

On the other hand, some events are recurring periodically. For these events, which we call *Periodic Events*, there are multiple perturbations in the usage of related hashtags, such that the interest in the event raises periodically, and is null (or low) during the other days. An example of periodic event is shown in Figure 2(b). Here, an activity peak on the hashtag `#MeatlessMonday` can be detected on each Monday, since the event is joined by people that meet virtually every Monday to discuss about meatless recipes.

Figure 1(b) depicts our event classification process. The temporal distributions associated with the hashtags we want to classify as event-related/event-unrelated are fed as an input to a chain of two binary classifiers, the first dedicated to spike event detection, while the second dedicated to periodic event detection. A hashtag (or equivalently its temporal distribution) is labeled as event-related if at least one the classifiers recognizes it.

Feature set. Spike and periodic events have a peculiar temporal distribution which is common for all the events of the same class. However, when it comes to training a classifier for the recognition of event classes, temporal distributions cannot be used as feature vectors: they suffer from *temporal dependence* of subsequent components, and consequently events that clearly belong to the same class but happened in different periods of time would have completely different feature vectors and thus would not help the classifier learn the underlying model. For this reason, we used as feature vector the spectrum of the *Fourier transforms* $|F(u)|$ of the normalized temporal distribution, which describe the frequency components of the signal and are agnostic with respect to the actual time of the events. As an example, Figure 3 shows the Fourier transforms of two periodic events, which happen in different periods but have similar spectrum.

Event classifiers. When it comes to building an annotated dataset to train the classifier, we come up with an *unbalanced training set*, since events are rare if compared to the total number of produced hashtags. Due to the lack of positive samples (i.e., temporal distributions corresponding to events), the classifiers could easily fall into the problem of *overfitting the data*. Thus, we applied the EasyEnsemble algorithm [13], which uses undersampling to rebalance the training set, combined with AdaBoost classifiers [11], since boosting is often robust to overfitting. Finally, to assess the performance of the classifiers on the training and test sets, we applied K-fold cross validation, with $K = 10$.

(a) Topic classifier

Text samples $\mathcal{T}^{\omega}$	Dictionary size	12988
	Positive samples	14234
	Negative samples	14218
	Total samples	**28452**
Image samples $\mathcal{T}^{I}$	Dictionary size	5000
	Positive samples	11759
	Negative samples	11746
	Total samples	**23505**

(b) Spike and periodic event classifier

Samples in $\mathcal{H}_s^R$	Positive samples	2030
	Negative samples	4870
	Total samples	**6900**
Samples in $\mathcal{H}_p^R$	Positive samples	5000
	Negative samples	5890
	Total samples	**10890**

Table 1. Dataset cardinalities

5 Experimental Evaluation

In this section we assess the performance of the proposed topic-related event detection approach. We first show how we can correctly identify topic-related tweets captured from the tweet live stream. Then, we apply event detection to the resulting tweet set, showing that our approach is capable of attaining good performance (measured as Precision@K).

5.1 Topic-related tweet retrieval

In the following, we illustrate the characteristics of the datasets we used to assess the multimodal classifier performance and report classification performance.

Dataset description. We trained the text and image classifiers on, respectively, the textual and image datasets $\mathcal{T}^{\omega}$ and $\mathcal{T}^{I}$, whose cardinalities are reported in Table 1(a). To test our classification approach, we randomly extracted and manually annotated the following sets of samples: *i)* $\tilde{\mathcal{T}}^{\omega}$, composed of 1900 tweets containing only text; *ii)* $\tilde{\mathcal{T}}^{\omega+I}$, composed of 1900 tweets containing both text and images, where $\mathcal{T}^{\omega}$, $\mathcal{T}^{I}$, $\tilde{\mathcal{T}}^{\omega}$, $\tilde{\mathcal{T}}^{\omega+I}$ are all disjoint. Note that some tweets are characterized by ambiguous content, and thus annotating them as relevant or not relevant is difficult for a human annotator too. On our dataset, the inter-annotator agreement is 93.86%.

Classifiers performance. Multimodal classification improves performance with respect to text classification on $\tilde{\mathcal{T}}^{\omega}$ and $\tilde{\mathcal{T}}^{\omega+I}$. Table 2 shows how accuracy, precision, recall and $F1$-measure increase in this scenario.

Text classification performance is insufficient when images are involved, because it is not able to interpret visual content and may misinterpret the text associated with images.

5.2 Event classification

In this section, we assess the performance of the proposed event detection technique on the food-related tweets.

Dataset description. We ran our topic-related tweet retrieval process from June 1, 2014 to June 10, 2015. During that period, the system processed more

than 15 million tweets, 9 millions of which were labeled as food-related. The corresponding number of relevant hashtags was 171451. However, only 21451 were associated with a temporal distribution comprising more than 5 tweets and were included in the final set of topic-related hashtags $\mathcal{H}^R$. In order to train the spike classifier, we took a random sample $\mathcal{H}^R_s$ of size 6900 from $\mathcal{H}^R$. Then, we performed a data annotation campaign on the crowdsourcing platform Champagne [6], to label them as event-related/unrelated. Crowd workers were prompted with a sequence of temporal distributions (similar to those in Figure 2), and asked to identify spike events. A different approach was instead required for training the periodic classifier, due to the fact that periodic events are quite rare in $\mathcal{H}^R$. We compensated for this unfavorable situation as follows. We first identified 10 periodic events in $\mathcal{H}^R$. We then used such events to synthetically generate 5000 new positive instances by combining each periodic event with a Gaussian process with mean 0 and variance 0.03, and randomly shifting the temporal distribution within a period of 7 days. Such procedure is similar to what is done in the literature (see, e.g., [18]). Let us denote the resulting dataset as $\mathcal{H}^R_p$. The cardinalities of the two datasets are reported in Table 1(b).

Classifiers performance. The performance of the spike and periodic event classifiers are reported in Table 3. As shown, both classifiers attain high values of $F1$-measure and accuracy, on both the training and test set. In order to further evaluate the effectiveness of our approach, we also tested the proposed event detection technique against a gold standard dataset $\mathcal{H}^R_g$, which we obtained by first ordering hashtags in $\mathcal{H}^R$ by total number of tweets, and then providing a gold label for the first 1000 hashtags. In particular, each hashtag has been assigned a gold label by analyzing different factors, such as the name of the hashtag, its current use on Twitter, the shape of its temporal distribution, and the content of tweets collected by the process. Since the total number of tweets might be intended as a proxy for the success of an event, we believe that testing the proposed technique against the top-1000 hashtags can provide a meaningful insight on its effectiveness in detecting successful events. Since the test was performed against a top-K ranked list, we measured performance by means of a Precision-Recall curve, which depicts the attained precision-recall values as K increases. Figure 4 reports the performance of our technique on $\mathcal{H}^R_g$. As shown, our method correctly identifies the first 14 food-related events. Overall, our method labels 172 events as food-related, which leads to a final precision-recall value of $(0.80, 0.67)$.

	$\tilde{\mathcal{T}}^{\omega}$	$\tilde{\mathcal{T}}^{\omega+I}$
Text classification	$\texttt{Accuracy} = 75.22\%$ $\texttt{Precision} = 65.67\%$ $\texttt{Recall} = 80.54\%$ $\texttt{F1} - \texttt{measure} = 72.35\%$	$\texttt{Accuracy} = 73.47\%$ $\texttt{Precision} = 63.61\%$ $\texttt{Recall} = 80.30\%$ $\texttt{F1} - \texttt{measure} = 70.99\%$
Multimodal classification		$\texttt{Accuracy} = 82.23\%$ $\texttt{Precision} = 79.13\%$ $\texttt{Recall} = 97.22\%$ $\texttt{F1} - \texttt{measure} = 87.24\%$

Table 2. Performance of text classification and multimodal classification

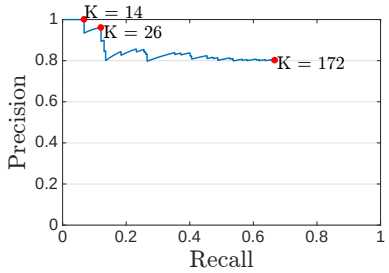


Fig. 4. Recall-Precision curve

Discussion. Table 4 shows the top-10 food-related hashtags retrieved by our event detection pipeline, together with tweet samples showcasing their usage. The list reports: *i)* food-centered social events that are confined in the Twittersphere (`#foodiechats`, `#MeatlessMonday`, `#bandwiches`); *ii)* holidays (`#halloween`, `#easter`) and periodic calendar-based events (`#sunday`, `#tbt` [that is: '*Throwback Thursday*'], `#tgif` [that is: '*Thank God it's Friday* ']) during which users share themed recipes; *iii)* media events (`#espys`), during which people have dinner in front of the TV and share comments about the show and their food; *iv)* food-centered advertising campaign (`#TeamWalmartProduce`). Finally, Figure 5 shows a set of images retrieved by our pipeline and related to the periodic calendar-based events `#TacoTuesday` and `#NationalCheesecakeDay`. This sample shows how our pipeline is able to retrieve high quality multimedia content (thanks to multimodal classification), which could be used, e.g., to summarize the contents shared by Twitter users during the detected events.

6 Related work

A number of recent works in the literature cover the problem of event detection on Twitter. The work in [22] builds a spatiotemporal model to estimate where and when events happened, with specific focus on earthquakes and typhoons. The work in [10] applies a state-of-the-art earthquake detection algorithm to detect earthquake-related tweets in real-time. The demo in [17] proposes a system which identifies in real-time real-world events by detecting bursty keywords. The work in [14] detects unusually crowded regions that can eventually suggest the

	Training set	**Test set**
Spike classifier	`Accuracy` = 92.48% `F1-measure` = 87.99%	`Accuracy` = 92.11% `F1-measure` = 88.14%
Periodic classifier	`Accuracy` = 99.77% `F1-measure` = 99.75%	`Accuracy` = 99.45% `F1-measure` = 99.42%

Table 3. Performance of spike and event classifiers when tested against the training and test set

Hashtag	# tweets	Representative tweet
#foodiechats	28845	@Foodiechats We have Smoked Turkey Sliders, Tandoori Chicken Flatbread Panko Sesame Fish Skewers, and Peach Shortcake! #foodiechats
#MeatlessMonday	26643	Spicy black bean burgers. #MeatlessMonday #food
#TeamWalmartProduce	22421	There's nothing better than a dessert with delicious stone fruit! #ad #TeamWalmartProduce
#sunday	19966	Photo: Sushi treats at the Spice Haat Sunday Brunch #sunday #brunch #sushi
#halloween	16201	Strawberry Ghosts – are these cute! Love the little ghost "tails" on them #halloween #partyfood
#espys	10002	first time i've ever cried while eating pizza. love you, Stuart Scott. #stay$TRONG #espys
#tbt	9268	RT @Justelise97: Pancakes + Vanilla Ice Cream #tbt #throwback #foodporn
#easter	8964	RT @FoodEmbassy_: This #Italian #pie has #easter written all over it! Torta Pasquale!! @BBCFood
#bandwiches	7978	Peanut butter and Pearl Jam #bandwiches @midnight
#tgif	6903	Egg whites and PB toast. #postworkout #breakfast #daymaker #tgif #riseandshine #todayisagoodday #smile

Table 4. Top-10 food-related hashtags based on the total number of tweets

occurrence of geo-social events. The work in [1] identifies local events by dividing the timeline of a potential event in time frames, extracting bursty keywords in each time frame and selecting only the keywords that have local spatial distribution. The work in [8] retrieves tweets that contain drug-related keywords and identifies drug-related events as spikes in the number of collected tweets. The work in [25] classifies social events by clustering temporal series having similar shapes. The work in [9] applies a similar approach, with the strong assumption that no event can transgress the boundaries of a day. The work in [7] sequentially retrieves tweets from Twitter and transform them in lists of words, which are then used to cluster keywords according to their density and filter non-local events. The work in [12] manually identifies hashtags related to the *Je Suis Charlie* event and analyze how it relates to the raising counter-events (e.g., *Je Ne Suis Pas Charlie*). The work in [21] performs POS tagging, named entity extraction and extraction of temporal expressions to create classes of events, using unsupervised approaches, attaining a Precision@100 of 90%, a Precision@500 of 66% and a Precision@1000 of 52%. The work in [26] detects composite social events over streams, by using information deriving from similarity between messages in the social stream. The work in [19] analyzes the sentiment of produced tweets to discover real-world events, under the assumption that an event shifts

Fig. 5. Images from #TacoTuesday (left) and #NationalCheesecakeDay (right)

the sentiment toward a topic (represented by specific keywords in the content). Events are thus recognized as bursty keywords that shifted the mood of users. This approach achieves 60% recall if the objective is to discover the exact date of an event, and 90% recall if a tolerance of ± 1 day is allowed. The work in [2] uses several topic detection algorithms and an extension of the tf-idf approach over time to recognize emerging bursty topics. For this work, the Recall@N varies between 50% and 90% (depending on the used dataset). Although we rank favorably with comparable works such as [19] and [2], in many cases we cannot directly contrast our approach to what is present in the literature. Indeed, while our technique aims at identifying how hashtags relate to events, a significant percentage of previous works ([22], [10], [14], [1], [7], [26]) focus instead on the problem of spatially localizing such events. A direct comparison is also not possible for those works that try to identify open-domain events, such as [21].

Several works use supervised classification methods to state if content is related to an event. The work in [5] clusters similar messages to perform topic identification, and then classifies content as event-related/event-unrelated, based on temporal features (e.g., deviations from expected message volume), social features (e.g., retweets and mentions), topical features (e.g., focus on a topic) and Twitter-centric features (e.g., hashtag usage). In that work, the F1 measure achieves 83.7% on test set, while Precision@20 is 65%. The work in [3] uses an SVM classifier to select flu-related tweets, to track how flu moves over space and time. The F1-measure achieved by this method is 75.6%. The work in [15] identifies crime and disaster-related events via binary classification, based on Twitter-specific features (e.g., hashtags) and on the presence of event-specific text features (e.g., presence of happening time). Although on a different topic of interest, our approach is competitive with the afore-mentioned classification-based methods available in the literature. Moreover, note that none of the previous works deal with the problem of identifying periodic events, which we showed is an interesting problem in itself and permits unveiling a significant percentage of social events.

7 Conclusions

In this paper, we investigated the problem of topic-related event detection on Twitter, which we cast as a supervised learning problem. We focused on the concrete use case of identifying events that include a food-related component, such as holidays or commercial initiatives. We first induced a multimodal classifier capable of identifying tweets related to the topic of interest, which we used to isolate relevant tweets from the global tweet stream. Events were therefore identified by applying a chain of two classifiers, one for the identification of periodic events and one for the identification of spike events.

The experimental evaluation showed that our approach attains a Precision@10 value of 100%, and a Precision@172 value of 80%, proving therefore competitive with other state-of-the-art approaches available in the literature. Future work will focus on enriching the event classifier feature vector to capture social components, such as user profile characteristics (e.g., authority) and network characteristics (e.g., centrality) and on spatial distribution analysis.

References

1. H. Abdelhaq et al. Eventweet: Online localized event detection from twitter. *VLDB*, 6(12), 2013.
2. L. M. Aiello et al. Sensing trending topics in twitter. *IEEE Transactions on Multimedia*, 15(6), 2013.
3. E. Aramaki et al. Twitter catches the flu: detecting influenza epidemics using twitter. In *EMNLP*, 2011.
4. F. Atefeh and W. Khreich. A survey of techniques for event detection in twitter. *Computational Intelligence*, 2013.
5. H. Becker et al. Beyond trending topics: Real-world event identification on twitter. *ICWSM*, 11, 2011.
6. C. Bernaschina et al. Champagne: a web tool for the execution of crowdsourcing campaigns. In *WWW Companion volume*, 2015.
7. A. Boettcher and D. Lee. Eventradar: A real-time local event detection scheme using twitter stream. In *GreenCom*, 2012.
8. C. Buntain and J. Golbeck. This is your twitter on drugs: Any questions? In *WWW Companion*, pages 777–782, 2015.
9. C. De Boom et al. Semantics-driven event clustering in twitter feeds. In *Making Sense of Microposts (# Microposts2015) (WWW)*, 2015.
10. P. S. Earle et al. Twitter earthquake detection: earthquake monitoring in a social world. *Annals of Geophysics*, 54(6), 2012.
11. Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *JCSS*, 55(1), 1997.
12. F. Giglietto and Y. Lee. To be or not to be charlie: Twitter hashtags as a discourse and counter-discourse in the aftermath of the 2015 charlie hebdo shooting in france. In *Making Sense of Microposts (# Microposts2015) (WWW)*, 2015.
13. H. He et al. Learning from imbalanced data. *TKDE*, 21(9), 2009.
14. R. Lee and K. Sumiya. Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection. In *ACM SIGSPATIAL, 2010*, 2010.
15. R. Li et al. Tedas: A twitter-based event detection and analysis system. In *ICDE*, 2012.
16. D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999*, volume 2, 1999.
17. M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the twitter stream. In *SIGMOD*, 2010.
18. J. Nonnemaker and H. S. Baird. Using synthetic data safely in classification. In *IS&T/SPIE Electronic Imaging*, pages 72470G–72470G, 2009.
19. G. Paltoglou. Sentiment-based event detection in twitter. *Journal of the Association for Information Science and Technology*, 2015.
20. M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3), 1980.
21. A. Ritter et al. Open domain event extraction from twitter. In *SIGKDD*, 2012.
22. T. Sakaki et al. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW*, 2010.
23. M. Woźniak et al. A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16:3–17, 2014.
24. L. Xu et al. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE TSMC*, 22(3), 1992.
25. J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *WSDM*, 2011.
26. X. Zhou and L. Chen. Event detection over twitter social media streams. *The VLDB Journal*, 23(3), 2014.

The spider-man behavior protocol: exploring both public and dark social networks for fake identity detection in terrorism informatics

Matteo Cristani, Elisa Burato, Katia Santacá, and Claudio Tomazzoli

University of Verona
{matteo.cristani, elisa.burato, katia.santaca,
claudio.tomazzoli}@univr.it

Abstract. Hiding true personality behind a façade is one of the basic tricks adopted by humans who live double lives for illegal purposes. In particular terrorists have historically adopted the protocol of a façade behaviour coupled with a second life consisting mainly in illegal activities and their planning.
Nowadays a few cases of behaviours that hide a dangerous activity, possibly illegal, behind an apparentely neutral and mean public person, can be replicated, and sometimes just provided, by a social network profile. Recognizing that a social network profile is fake, in some extreme cases, a bot, and determining the contour relationships that limit such a condition is one of the most important weapons for terrorism fight.
In this paper we show that what we name the *Spider-man protocol*, a set of behaviour rules that bring to hiding a personality behind a façade, has several weaknesses, and it is prone to a set of attacks that permit to detect these behaviours. We provide the description of an experimental architecture that is used for determining violations of the protocol, and therefore breaches in the secrecy of the individual protection settled by the terrorists.

1 Introduction

In the recent past, it has been found that the web is also being used as a tool by radical or extremist groups and users to practice several kinds of mischievous acts with concealed agendas and promote ideologies in a sophisticated manner, so that several studies have been performed on how to understand and identify tension or deviant behaviors before these can lead to acts of terrorism. These investigations are paired by those studies, especially in the information security research area, that aim at determining cases of phishing, where people are showing off themselves as individuals different then they are, to obtain illegal profits.

To the best of our knowledge, however, only a few investigations have been carried out that combine these two aspects. It is clear that, when someone passes the border and becomes a terrorist, there is an observable phase in which part

of her life is still public though partly hidden, whilst after this phase that person becomes invisible. A few behaviours can be classified that correspond to *become clandestine* for illegal purposes, and, on the other hand, there are a few behaviours that can make such condition disclosed.

In this paper we study the ways in which the aforementioned transition happens (Section 4.1), how you can provide the recognition of a breach in such a protocol (Section 4.2) and present an architecture to deal with such a recognition need (Section 5). Before to do so, we need to model the behaviours (Section 2) and introduce a method, that is the extension of an existing approach [3] to more general cases (Section 3). At the end of the above presented studies, we review the recent literature (Section 6) and finally introduce some further perspective (Section 7).

2 How do terrorists behave on the web?

The majority of radical and extremist groups do not appear as regular individuals on the public web. They typically hide themselves under the level of publicly mapped web sites, the so-called Deep Web. This area of the web, often also known as *invisible web* is essentially identical to the visible part, aside from the lack of association of the web addresses to the web spiders of Google and other search engines. Within the Deep Web umbrella, many of those individuals interact in a social network that is totally hidden to the public web, the so-called darknets. This area of the web is known as *Dark Web*.

There are many hidden social networks, including, in particular, the well-known *AnonPlus* secret network, or the less known but very important *DarkNetMarket*, used to interact in the Dark Web by criminals, including drug marketers, pedophiles, terrorists. The majority of these web sites need specific tools to be used, as, for instance *Tor*, *Freenet* or *I2P*, and employ specific P2P protocol methods, including the files used for the specific P2P purpose, the *.onion* ones. The public part of the web is also referred to, in particular by the users of the Darknets, as *the Cleranet*.

The notion of a terrorist used in the current literature is that he is an individual who is acting in a public environment and secretly fighting for a social, political, religious, ethnic, or national cause. This definition implies that a terrorist can be in one of the following general conditions:

- *Fully clandestine*, the condition of terrorists acting completely on the secrecy, hidden in a place where they cannot be found. This is the case, for instance, of Al-Qaeda in certain areas, like Europe and the United States.
- *Rebel*, when a fighting individual lives separately from the counterpart, in a publicly known area, but protected by an openly fighting group. ISIS is acting in this way.
- *Double living*, when they act publicly as apparently harmless people, whilst living a second life of active fighters for some causes. This is the way in which Al-Qaeda members act in the same areas where fully clandestine members also exist.

3 Detecting terrorists: social media and dark web analysis

The basis of the web analysis we provide is a twofold approach: we aim at tracing individuals who act under the umbrella of the Dark Web in double living style as defined in Section 2. We trace individuals in the Darknets, and individuals in the Clearnet, and use a combination of Social Network and Sentiment Analysis for coupling profiles on the two sides.

The approach is based on the idea that when it is possible to establish a clear correspondence between an individual living a double lives style, it is also possible to mark that individual as a potential suspect, and therefore enshorten significanly investigative efforts. The potential is expressed in the duality of Darknet expression of ideas whose admissibility in public domain is deputable, especially when those ideas have a political origin, in a very general sense, including in this also religious, class, national and ethinc principles. The concept is that when it is possible to decontour two individuals that are likely to coincide in the reality, and one of these individuals have a specific interest in political issues, there is a suspect of terrorism (potentially).

To determine an individual to correspond in the Darknets and in the Clearnet, we use the *homophily* principle, namely we consider two invidiuals to be as close as their interests are in common. On the other hand, we make use of the so-called Social Network Analysis, considering two individuals to be as close as their reference networks overlap.

The difficulty in comparing individuals belonging to the two distinct sides of the web, is that they try to hide their correspondence, namely they try to make almost impossible to compare them. The behaviour of individuals that aim at avoinding any overlapping between their harmless public counterpart and their secret dark counterpart is here referred to as the *Spider-man protocol*. Clearly, if an individual is rigourous in keeping the two sides apart, and prevents any leak of information the protocol is respected, and no one can ever discover this secret. In Section 4, we analyse two situations in which it is possible to provide an attack to terrorism privacy, that can be used for useful purposes. The major weakness phase is the initial one, when an individual *becomes* a terrorist. Minor cases regard the preparation of a terroristic attack, and the phase in which an individual plays with the idea of exiting an organization.

3.1 Social network analysis: the social network measures of terrorists

A connection network of an individual i contains people that etiher have a personal relationship with i, or have a certain group of interests in common. When it is possible to detect the existence of interests in common (homophily), we can establish that u shares some interest with J.

Clearly, to share interest does not imply to share viewpoints, and thus an extremist can have a high homophily with a moderate person, being both interested in politics, and maybe being both on similar position, but still not sharing

the model of acting, as in particular, being different in acceptin or not acts of violence as means for making own ideas succed.

If an individual i is a terrorist and and an individual j is homophilic and connected to i it is plausible that also j should be suspect of terrorism. Therefore, once we know that an individual is connected to a potential terrorist, we attempt at determining connections that can be referential for other individuals.

3.2 Sentiment analysis: words of terrorists

Every terrorist organization employs a specific war lexicon, a sort of glossary of the fighter. The analysis of the posts of people close to terrorists, as well as many communications from self-declared terrorists, shows that there is combination of *extremism* and specificity of the referential ideology. Communist terrorist movements mixed up, for instance, words of war like fight, battle, kill, and many others with words of communism as working class, revolution, proletariat dictature, and others.

The common style of terrorist communication is also the usage of secret words, the so called *code language*. A famous example of this method of communication is the use of the term *pack* by tupamaros terrorists in South Amrrica in the Seventies, to refer a potential victim of a terrorist attack.

4 Weak passages of terrorism web behaviour

There are three phases of the terrorist activities in which the Spider-man protocol is weaker in resisting to attacks:

- In the phase in which an internaut becomes a terrorist, or in formal terms, enter a double lives behaviour;
- In the near temporal proximity of a terrorist attack, epecially during the preparation days;
- The phase in which an individual is planning to exit the terrorist organization he belongs to.

Majorly, during these phases, it is relatively easier that the terrorist makes errors, namely he breaches the protocol, by revealing directly or indirectly his identity.

4.1 The radicalisation phase

The radicalization phase is the period of time in which a person starts to move his political ideas close to those of an active terrorist group, or more generically, to the ideas of a political area where violence is considered an option.

From an use of the language viewpoint, it is relatively simpler to determine such a change of behaviour in those contour conditions where radicals exist and are contiguous to extremists and moderates in a general large organization. For

instance this happens for islamic terrorists, and to a more restricted extent, due to the reduction of size of the general movement, for revolutionary communist groups.

Analogously, the social network analysis of these groups reveals that the number of contacts of a newbie radical increase suddenly, during the radicalisation phase. This is due to entering the organization, and is also due to the attraction to other potential newbies generated by the appearance of the newbie in the panorama of radicals and extremists. After a phase like that, the radical pass to the double lives. When this happens, again relatively suddenly, the darknet side of them appear.

From a pure observational viewpoint, this is the phase in which apparent continuity of the Clearnet user is not anymore present: they need to be partly in the Darknets, and this absences are less justified than those of others, because the Clearnet radicals, not the terrorists, obviously, miss the presence of the newbie. In this phase, the number of posts, comments, sharing and other behaviours decrease. Simultaneously a newbie with some omophily of the Clearnet user appears. The radical who is moving to a terrorist group passes a phase in which his *above board* personality needs to be guarded, and therefore the Clearnet user appear often to become less aggressive, and less interested in establishing connections with other radicals. Recognizing these behaviour treats is a viable method to identify a potential terrorist in his initial phase.

4.2 Breaching protocols: when terrorists leave permanent traces on the web

During the preparation of a terrorist attack the members of an organization intensiify their darknet communications. Provided that you have connected a Clearnet personality to a hidden Darknet one, the hidden persobality can bring you the information (in this case, regarding an attack in the near future), and the Clearnet one can bring you to the terrorist actual life.

On the other hand, when a member of a terrorist organization is about to leave the organization itself, he tends to dissimulate his desire at most, being this passage much more dangerous in terms of personal freedom than it can be the opposite one, when someone enters an organization. However, a few errors are well-known as providing a view of this cases. In particular it is well known from military literature that moderate dissimulation that is a typical façade of terrorist with double lives on their Clearnet personality, decreases in those phases.

The ability of recognizing the aforementioned treats completely relies upon the combination of sentiment analysis and social network analysis. A flexible architecture for providing such a method is prsented in next section.

5 An architecture for the detection of potential terrorists

In this section we introduce an architecture for the detection of terrorists and potential ones called *DetectTerror*.

DetectTerror aim is to detect fake identity analyzing data coming from both public and dark social networks, as summarized in Figure 1.

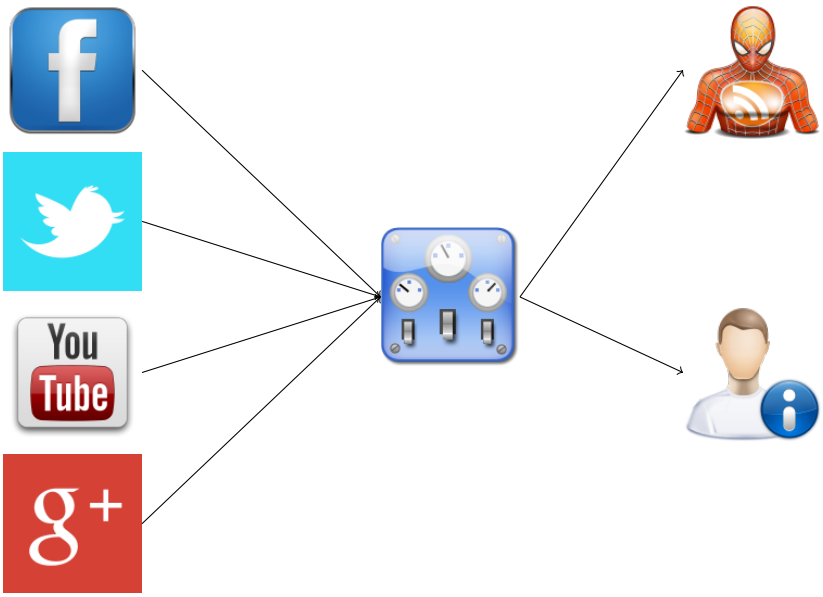


Fig. 1. The operative concept of DetectTerror architecture

DetectTerror is made of several modules, each one with a single responsibility; the logic model of DetectTerror is reported in FigureDetectTerror 2.

Every module is related to at leas one other, while all refer to one named *Orchestrator*:

Crawler: this component aims at the retrieval of raw informations form a specific source (i.e. Twitter, Facebook) following information structure to gather the correct piece of data. To add a new source to DetectTerror the only implementation regards this module and its related *Normalizer*

Normalizer: this component can analyze data and format them so that they are all in the same format and with a structure which makes them ready for the analysis

Analyzer: this module takes as input normalized data and gives as output a representation suitable to be later exposed to the *Reasoner*, a kind of digital identity fingerprint

Reasoner: this is the actual core of DetectTerror, the one which aim is to discover the relations between digital identities fingerprints to exploit where connections are

Orchestrator: this module is the "main app" of DetectTerror actually coordinating all others

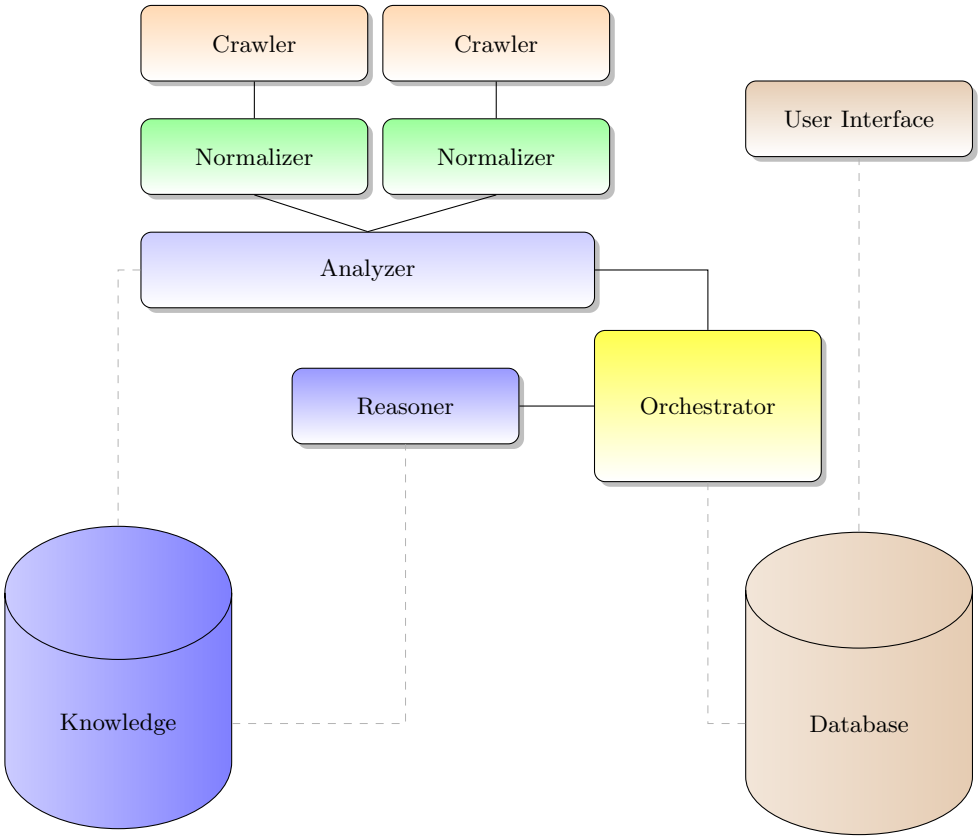


Fig. 2. Logic model of DetectTerror architecture

Knowledge base: where the knowledge base is stored; the *Reasoner* will access it for reasoning and the *Orchestrator* will increment it after evaluation of the results of the reasoner

Database: where all application data are stored, including partially evaluated retrieved data, configurations needed to effectively access information on social media, rules for data normalization, etc.

User Interface: tho module provides visualization of all data, allow user to modify parameters

6 Related Work

There are numerous natural language processing applications for which subjectivity analysis is relevant, including information extraction and text categorization. According to Wiebe [36], the subjectivity of a text is defined as the set of elements describing the private state of the writer (emotions, opinions, judgments, etc.).

The term Sentiment Analysis has been introduced in 2001, in order to describe the process aimed at automatically evaluating the polarity expressed by a set of given documents [9]. The term Opinion Mining has been introduced in 2003 in order to describe the activity aimed at *processing a set of search results for a given item, generating a list of product attributes and aggregating opinions about each of them* [10]. While OM is mainly focused on recognizing opinions expressed in a given text with respect a specific attributes, SA is focused on classifying a given document according with the polarity [23].

There are several recent studies about sentiment analysis. A common approach for SA is to select a machine-learning algorithm and a method of extracting features from texts and then train the classifier with a human- coded corpus. The main features in representing documents are: bag-of-words as in [1, 17, 16, 33, 35] or tree sentence parsing as in [30, 19].

In [20], the authors show how the SA depends upon the adjectival and adverbial modification of nouns and verbs. Adjectives and adverbs are largely studied as word-sense modificators in the NLP community [28, 14, 7, 12, 21].

In [25] the authors show how to detect authorship by a similarity measure among documents represented by vector space model to identify fake content and fake users. The problem of authorship attribution is to identify the author of a new document having a corpora of documents of known authors [27, 32]. On the other hand, the authors of [26] present a web service that tracks the diffusion of a set of keywords to detect atroturfing and fake content by means of social network analysis procedures.

The authors of [34] present a survey of the issues in social interaction and the recognition of user behaviour in social channels. The analysis of social behavior and patterns of users is the main part in the identification of user groups, as in [22], and in [13].

Some scientist, thanks to the release in 2010 from the famous social network *Twitter* of remote stream APIs that enabled performing of real-time analytics, concentrated on extracting meaning from *tweets*[11].

The authors in [6] explored the frequency of retweets surrounding an event and the duration between the first and the last of these retweet to extract information on how people behave when confronted with both positive and negative events.

Using the aforementioned *Twitter* API, in [5] data have been used in the study of the spread of online hate speech, or *cyber hate*, and forecast the likely spread of cyber hate; a classifier was used based on Bag Of Word model and the presence of key terms.In [29] word are tagged using TreeTagger (Schmid, 1994) and interpreted the difference of tag distributions between sets of text (positive, negative, neutral or subjective, objective), while in [15] the authors make use of ontologies to enhance sentiment analysis and attach a sentiment grade for each distinct notion in *Twitter* posts.

Always analyzing *Twitter* data, in [4] there is an attempt at understanding tension at an early stage and evidence is given that a combination of conversation

analysis methods and text mining outperforms machine learning approaches at such task.

In [37] the whole chapter is related to topics of sentiment analysis based on visual and textual content, where information is extracted from meaning of words or images.

In [31] the authors search for Negativity, Fear, and Anger showing that fear and anger are distinct measures that capture different sentiments, and they achieve these results using dictionary-based sentiment analysis.

Mining opinions and sentiment from social networking sites is the aim in [18] where the tool used is a bag of words feature set enhanced by a statistical technique named *Delta TFIDF* to efficiently weight word scores before classification.

To exploit certain types of information from reports on terrorist incidents, the authors in [8] perform syntactic and semantic analysis and uses lexicons of various categories of terms.

In [24] the focus is the problem of real-time sub-events identification in social media data (i.e., Twitter, Flickr and YouTube) during emergencies, and the method used involved tracking the relevant vocabulary to capture the evolution of sub-events over time.

There is also a study [3] in which Social Network Analysis is combined with Sentiment Analysis to explore the potential for the possibility of individuals being radicalised via the Internet; key terms and their frequency are used in this analysis.

As a matter of fact, it is not only what is said that counts, but also who is speaking. There are people more likely to be listened to (or *followed*) than others and it can be of relevance to identify radically influential users in web forums, which the subject of other studies[2].

7 Conclusions

This paper describes an architecture that can be used for detecting terrorists when they use Darknets and the Clearnet in a substantially different and anyhow permeable way, breaching what we call the *Spider-man behaviour protocol*.

There are three different ways in which this research has to be taken further. First of all, we shall implement the technology in practice and experiment it with real-life cases, in order to provide a direct and verifiable example of what suggested in this paper. Secondly we need to refine both social and sentiment techniques in order to detect terrorists at different developing stages: early stage, namely when they enter the organization and pass to a clandestine (possibly partly) life, phase before exiting the organization (that can be used to prevent attacks). Finally it is of strong interest to provide a ranking, possibly regarding belonging to an organisation as well as a form of measure for the probability of an individual to enter an organisation.

References

1. Marilisa Amoia and Claire Gardent. Adjective based inference. In *Proceedings of the Workshop KRAQ'06 on Knowledge and Reasoning for Language Processing*, KRAQ '06, pages 20–27, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.
2. T. Anwar and M. Abulaish. Ranking radically influential web forum users. *Information Forensics and Security, IEEE Transactions on*, 10(6):1289–1298, June 2015.
3. A. Bermingham, M. Conway, L. McInerney, N. O'Hare, and A.F. Smeaton. Combining social network analysis and sentiment analysis to explore the potential for online radicalisation. In *Social Network Analysis and Mining, 2009. ASONAM '09. International Conference on Advances in*, pages 231–236, July 2009.
4. Pete Burnap, Omer F. Rana, Nick Avis, Matthew Williams, William Housley, Adam Edwards, Jeffrey Morgan, and Luke Sloan. Detecting tension in online communities with computational twitter analysis. *Technological Forecasting and Social Change*, 95(0):96 – 108, 2015.
5. Pete Burnap and Matthew L. Williams. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy and Internet*, 2015.
6. Pete Burnap, Matthew L. Williams, Luke Sloan, Omer Rana, and alt. Tweeting the terror: modelling the social media reaction to the woolwich terrorist attack. *Soc. Netw. Anal. Min.*, 2014.
7. Stergios Chatzikyriakidis and Zhaohui Luo. Adjectives in a modern type-theoretical setting. In Glyn Morrill and Mark-Jan Nederhof, editors, *Formal Grammar*, volume 8036 of *Lecture Notes in Computer Science*, pages 159–174. Springer Berlin Heidelberg, 2013.
8. Sumali J. Conlon, Alan S. Abrahams, and Lakisha L. Simmons. Terrorism information extraction from online reports. *Journal of Computer Information Systems*, 2015.
9. Sanjiv Das and Mike Chen. Yahoo! for amazon: Sentiment parsing from small talk on the web, 2001.
10. K. Dave, S. Lawrence, and D.M. Pennock. Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International World Wide Web Conference (WWW)*, pages 519–528, 2003.
11. Amir Hossein and Akhavan Rahnama. *Real-time Sentiment Analysis of Twitter Public Stream*. University of Jyvskyl, 2015.
12. Bjørn Jespersen and Giuseppe Primiero. Alleged assassins: Realist and constructivist semantics for modal modification. In *Logic, Language, and Computation - 9th International Tbilisi Symposium on Logic, Language, and Computation, TbiLLC 2011, Kutaisi, Georgia, September 26-30, 2011, Revised Selected Papers*, pages 94–114, 2011.
13. Nitin Jindal, Bing Liu, and Ee-Peng Lim. Finding unusual review patterns using unexpected rules. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1549–1552, New York, NY, USA, 2010. ACM.
14. Christopher Kennedy. Vagueness and grammar: the semantics of relative and absolute gradable adjectives. *Linguistics and Philosophy*, 30(1):1–45, February 2007.

15. Efstratios Kontopoulos, Christos Berberidisand Theologos Dergiades, and Nick Bassiliades. Ontology-based sentiment analysis of twitter posts. *Expert Systems with Applications*, 2013.
16. Zhaohui Luo. Formal semantics in modern type theories with coercive subtyping. *Linguistics and Philosophy*, 35(6):491–513, 2012.
17. Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 142–150, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
18. Justin Martineau and Tim Finin. Delta tfidf: An improved feature space for sentiment analysis. In *Proceedings of the Third International ICWSM Conference*, 2009.
19. Jeff Mitchell and Mirella Lapata. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1439, 2010.
20. A. Moreo, M. Romero, J.L. Castro, and J.M. Zurita. Lexicon-based comments-oriented news sentiment analyzer system. *Expert Systems with Applications*, 39(10):9166 – 9180, 2012.
21. Marcin Morzycki. Modification, 2013. Book manuscript. In preparation for the Cambridge University Press series *Key Topics in Semantics and Pragmatics*.
22. Arjun Mukherjee, Bing Liu, Junhui Wang, Natalie Glance, and Nitin Jindal. Detecting group review spam. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 93–94, New York, NY, USA, 2011. ACM.
23. Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January 2008.
24. Daniela Pohl, Abdelhamid Bouchachia, and Hermann Hellwagner. Online indexing and clustering of social media data for emergency management. *Neurocomputing*, 2015.
25. Tieyun Qian and Bing Liu. Identifying multiple userids of the same author. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1124–1135, 2013.
26. Jacob Ratkiewicz, Michael Conover, Mark Meiss, Bruno Gonçalves, Snehal Patil, Alessandro Flammini, and Filippo Menczer. Truthy: Mapping the spread of astroturf in microblog streams. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 249–252, New York, NY, USA, 2011. ACM.
27. Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press.
28. Susan Rothstein. Fine-grained structure in the eventuality domain: The semantics of predicative adjective phrases and be. *Natural Language Semantics*, 7(4):347–420, 1999.
29. Suprajha S, Yogitha C, Architha J Sanghvi, and Dr. H S Guruprasad. A study on sentiment analysis using tweeter data. *International Journal for Innovative Research in Science and Technology*, 1(9), 2015.

30. Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.
31. Stuart Soroka, LorI Young, and MeItal Balmas. Bad news or mad news? sentiment scoring of negativity, fear, and anger in news content. *Annals of the American Academy of Political and Social Science*, 2015.
32. Efstathios Stamatatos. A survey of modern authorship attribution methods. *J. Am. Soc. Inf. Sci. Technol.*, 60(3):538–556, March 2009.
33. Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173, 2012.
34. David C. Uthus and David W. Aha. Multiparticipant chat analysis: A survey. *Artificial Intelligence*, 199?200(0):106 – 121, 2013.
35. Anil Saroliya Vijay Dixit. A semantic vector space model approach for sentiment analysis. *International Journal of Advanced Research in Computer and Communication Engineering*, 2, 2013.
36. Janyce Wiebe, Theresa Wilson, Rebecca Bruce, Matthew Bell, and Melanie Martin. Learning subjective language. *Comput. Linguist.*, 30(3):277–308, September 2004.
37. Jianbo Yuan, Quanzeng You, and Jiebo Luo. Sentiment analysis using social multimedia. In Aaron K. Baughman, Jiang Gao, Jia-Yu Pan, and Valery A. Petrushin, editors, *Multimedia Data Mining and Analytics*, pages 31–59. Springer International Publishing, 2015.

Corpus Generation and Analysis: Incorporating Audio Data Towards Curbing Missing Information

Atiqah Izzati Masrani and Yoshihiko Gotoh

University of Sheffield, United Kingdom
{amasrani1,y.gotoh}@sheffield.ac.uk

Abstract. As video data becomes widely available, it is crucial that these videos are properly annotated for effective search, mining and retrieval purposes. Significant work has been done to explore natural language description as it can provide better understanding of the video content. Ideally, a summary should be informative and accurate in order for the users to have good understanding of the video content. An experiment has been conducted to evaluate the impact of audio information towards natural language summary annotations of a video content. The experiment proved that although events and human activities can be captured using visual features alone, key information of the video content would be missing without the audio information. Thus, future work on natural language summary generation should incorporate both visual and audio data to curb missing and erroneous information.

Keywords: Corpus generation, hand annotation, visual features, audio data

1 Introduction

Nowadays, there is an abundant of videos that are accessible online. The widespread use of the Internet has allowed videos to be accessed easily via video search engines such as from the YouTube or Daily Motion. The YouTube itself has more than 1 billion users and is estimated to have 300 hours of video uploaded every minute. It generates billions of views on a daily basis. Furthermore, the number of hours people spent on watching YouTube each month is up 50% year over year. [1] This raises the question on how the users can be more selective during video browsing and retrieval. Although some of these videos are well-organized with manually annotated tags or labels, some has no clear description of its content. Therefore, users may tend to skim through the video to grasp a hint of its semantic content.

Video summarization addresses this issue by providing brief information of the video. Significant work has been done in this area with a large part of it optimizes graphical representations. The graphical representations can be further

[1] https://www.youtube.com/yt/press/statistics.html

divided into two classes. The first class focuses on compressing the video into a shorter representation of the video that is also known as video skimming. This include works from [1], [2] and [3]. The second class uses image key frames extracted from the video stream to reflect the content or the highlights of the video [4] [5].

Natural language has also proven to be a popular choice to represent a video. It is an appealing option as it is less space consuming, has faster processing time for retrieval and is readable by both human and machine. Most early research such as works by [6] and [7] uses representative keywords. Using keywords can boost the potential for fast video retrieval because it helps efficient video categorization. However, using keywords alone may not be able to capture the whole key points of the video, as keywords tend to be ambiguous. This may affect the accuracy and effectiveness of video classification due to its ambiguity and lack of information. Natural language representation in the form of a "summary" or "abstract" is one way to address this. There has been significant works on creating a natural language summary that emphasizes on its coherency and informativeness. However, human's perspective when watching a video is subjective. Although presented with the same visual scene, one's interpretation may vary. This may influence on how they will write the summary of the video.

In this paper, an experiment was conducted to study the overlapped similarities of human's perspectives and also the impact of incorporating audio data during summary annotations. This paper aims to prove that the dissimilarity lies in the words used to semantically convey the meaning, and the similarity lies in the key information that is included in the summary. This paper also aims at proving that both the visual and audio data are important towards determining the key points of the video. Thus, using only one without the other towards natural language generation framework for video data may cause missing or erroneous information in the summary.

2 Corpus Generation

As video data becomes widely available, it is crucial that these videos are properly annotated for effective search, mining and retrieval purposes. Significant progress has been made to use natural language description as it can provide better understanding of the video content such as work by [9], [10], and [11]. Most of these works crafted their own video corpora that consist of the video data and its corresponding hand annotation. Each dataset are specifically designed with a certain prerequisites or constraints to fulfill a specific task or purpose.

In [8], the dataset is designed for the task of generating natural language descriptions of the video content. The work focuses on the natural language generation phase that is heavily dependent on the visual features extracted during the HLFs processing phase. The dataset is crafted from videos that consist of subjects, objects, actions and scene settings that can be easily identified using existing visual processing techniques. Therefore, the crafted videos are short and consist of a single shot or scene with minimal activity. Some other existing video

corpora are more domain-focused such as football, traffic, surveillance, cooking videos and so on [6], [12].

In this study, video clips from the BBC EastEnder series were selected. [2] It consists of approximately 244 episodes and each are associated with its own metadata and transcripts. This dataset is chosen because of its realistic elements with human subjects showing various activities, emotions and interactions with other objects. In this experiment, 5 episodes were chosen. These episodes were crosschecked with their metadata and transcripts. Each episode has a synopsis and description included in their metadata file. Assuming that the synopsis (summary) describes the highlight of its corresponding episode, these videos were cropped focusing on the episodes' highlight. The cropped video ranges between 4 to 20 minutes of playtime. Figure 1, shows the selected video with their synopsis, description and the duration of the cropped version.

Ref_ID	Video_ID	Synopsis	Description	Duration (mins)
Video1	5082189274976367100	Dawn is finally able to escape May and Rob and is left holding the baby	Dawn is finally able to escape May and Rob and is left holding the baby. A persistent Bradley eventually wins Stacey's heart and Stella's past starts to catch up with her	04:52
Video2	5084819083455904024	As the Vic re-opens it looks like Pat may have got one over on Peggy	As the Vic re-opens it looks like Pat may have got one over on Peggy by offering a service you just can't get in the Vic. Ian discovers Jane has been living a lie	09:06
Video3	5087397352319500829	Max is determined to split up Stacey and Bradley. Phil has doubts about marrying Stella	Max is determined to split up Stacey and Bradley, forcing her to make a difficult decision. Phil's doubts about marrying Stella surface until Ben's return cements all their futures	05:31
Video4	5089967890246158488	Will Ben have the confidence to stop Phil and Stella's wedding?	After a failed attempt to run away, will Ben have the confidence to stop Phil and Stella's wedding? Jase Dyer arrives to claim his son and Yolande is shocked by an outburst from Jay	05:03
Video5	5092591256270557686	Stella's demise rocks the Mitchells' world, and Ian is beside himself with worry for Ben	Stella's demise rocks the Mitchells' world, and Ian is beside himself with worry for Ben. Bert feels pushed out by Jase. Bobby creates havoc at the Beales	07:05

Fig. 1. The selected video with their synopsis and description

3 Annotation Process

Figure 2, shows that the participants conducted two rounds of writing the summaries. In the first round, the participants were asked to watch each video without the audio and write the summary. In the second round, the participants were asked to watch each video with the audio and write the summary. No specific rules were imposed on how the summaries should be written. This is because the objective of the experiment is to obtain an unbiased (although varies) perspectives of the participants. The participants were given 2 weeks to complete the experiment. The experiment is conducted to answer these research questions:

[2] http://www-nlpir.nist.gov/projects/tvpubs/tv13.papers/tv13overview.pdf

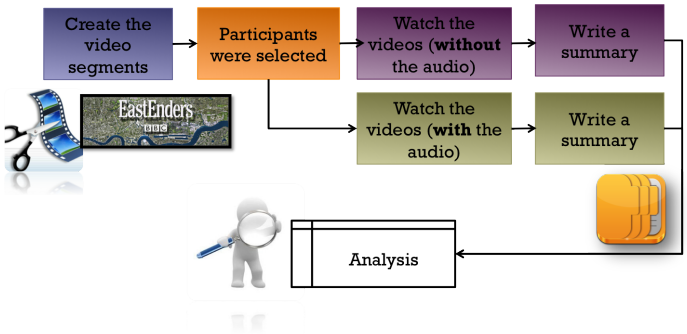


Fig. 2. The experimental setup

– Can the hand annotations consist of similarities that focus on the key interest points of the video?
– Can audio data help to reduce missing information?

4 Results

Figure 3 shows a selection of image key frames that depict the highlight of Video_ID:5082189274976367100 and an example of its corresponding hand annotations from one of the participant as shown in Figure 4.

The results will be presented in two corpora that is the hand annotations without audio and the hand annotations with audio. Total number of documents for both corpora was 50 (5 participants each created 2 summaries for 5 different videos). In each corpus, two classes[3] will be manually defined:

1. Human related: gender, age, body parts, identity, emotions, grouping, dressing, actions and activities numbers.
2. Non-human related: man-made objects, natural objects, scene settings, location, colours, size

4.1 Hand Annotations (Without Audio)

Total number of documents for this corpus was 25 (5 participants each created 1 summary for 5 different videos). The total number of words for the summaries was 1856, hence the average length of one document was roughly 74 words. Total number of unique words is 402. [4]

[3] Refers to the subclasses as defined in [8]
[4] This statistic is generated using www.linguakit.com

Fig. 3. A montage of the video highlights (Video_ID:5082189274976367100)

Human Related Features Figure 5 presents human related information observed in the hand annotations. The participants is shown to focus on identifying human's presence in the video because the top three most frequently used words (nouns) are woman with 41 occurrences, man with 31 occurrences and lady with 19 occurrences. For human related features, the human gender information has the highest number of occurrences: female with 77 occurrences and male with 54 occurrences. Related words such as 'lady' and 'woman' are combined into the same category 'female'. The same goes for 'male', which combines related words such as 'man' and 'boy'. Age information (e.g., old, young, child), identity (e.g., mother, nurse, groom) and grouping (e.g., one, two, crowd) are also often used. The words used to describe emotions are categorized into six basic emotions as described by Paul Ekman [5]. These six basic emotions are 'anger', 'disgust', 'fear', 'happy', 'sad', and 'surprise'. The least described features are body parts and dressing.

Non-human Related Features Figure 6 presents non-human related information observed in the hand annotations. The participants shown keen interest in identifying the location of a particular scene such as the hospital, restaurant,

[5] Paul Ekman is a psychologist and a co-discoverer of micro expressions with Friesen, Haggard and Isaacs

Hand Annotation 1	Without Audio	With Audio
	There is a girl in pain who had just given birth and she was arguing with an older woman who would have probably be her mother. The older lady then left the hospital but was stopped by another man. The older lady then rushed into her car and struggled for quite some time to get by the man. The lady back in the hospital got ready to leave the hospital but it seems that the nurse did not allow it. However, the lady insisted and the nurse left the room.	Dawn had just gave birth and was still in pain. A lady was trying to take her baby away and claimed that they had agreed on giving the baby to her in return of GBP10, 000. Dawn was furious and demanded her baby but the lady tried to increase the amount of money. Dawn took the baby from the lady and the lady left crying. She went into the car and a man tried to stop her to ask what had happened. The lady just drove away without any explanation. Dawn tried to contact her brother and stepfather to ask them to pick her up. The nurse stopped her but she still insisted because she did not feel secured in the hospital. She tried to reply on the nurse but the nurse failed to do so.

Fig. 4. An example of the hand annotation (Video_ID:5082189274976367100)

Table 1. Statistic and category frequency for the hand annotations without audio

Statistics	
Number of sentences	128
Number of words	1856
Number of unique words	402
Number of characters	8841
Characters no whitespace	7161

Category frequency	
Number of nouns	133
Number of adjectives	33
Number of verbs	129
Number of adverbs	30

church etc. They also showed interest in describing man-made objects involved (e.g., car, food, book etc.) and scene settings (e.g., ceremony, wedding, and outside). Natural objects and colours are rarely described. No word has been used to describe size.

4.2 Hand Annotations (With Audio)

Total number of documents for this corpus was 25 (5 participants each created 1 summary for 5 different videos). The total number of words for the summaries was 1983, hence the average length of one document was roughly 79 words. Total number of unique words is 426. [6]

Human Related Features Figure 7 presents human related information observed in the hand annotations. The participants is shown to focus on identifying human's presence in the video because the top three most frequently used words (nouns) are mother with 21 occurrences, baby with 20 occurrences and woman with 19 occurrences. For human related features, the human gender information has the highest number of occurrences: female with 75 occurrences and male with 75 occurrences. Identity features (e.g., mother, Dawn, nurse) also recorded high number of occurrences. Age information, emotions and grouping are described significantly. The least described features are body parts and dressing.

[6] This statistic is generated using www.linguakit.com

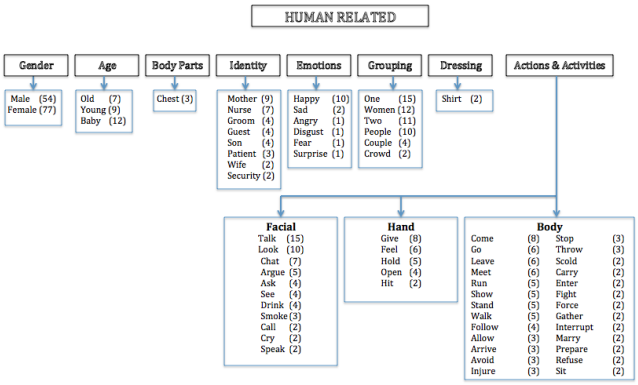


Fig. 5. Human related features in the hand annotations (without audio)

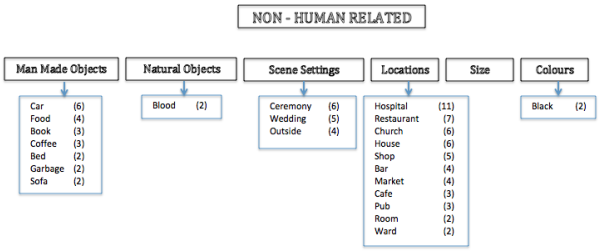


Fig. 6. Non-human related features in the hand annotations (without audio)

Non-human Related Features Figure 8 presents non-human related information observed in the hand annotations. The participants showed keen interest in identifying the location of a particular scene such as the pub, house, hospital etc. They also showed interest in describing man-made objects involved (e.g., rubbish, coffee, car etc.) and scene settings (e.g., ceremony, wedding, and outside). Natural objects and size are rarely described. No words have been used to describe colours.

5 Analysis and Discussion

The findings and analysis from this experiment will be presented in two subsections focusing on the two research questions.

Table 2. Statistics and category frequency for the hand annotations with audio

Statistics	
Number of sentences	127
Number of words	1983
Number of unique words	426
Number of characters	9500
Characters no whitespace	7713

Category frequency	
Number of nouns	138
Number of adjectives	31
Number of verbs	143
Number of adverbs	28

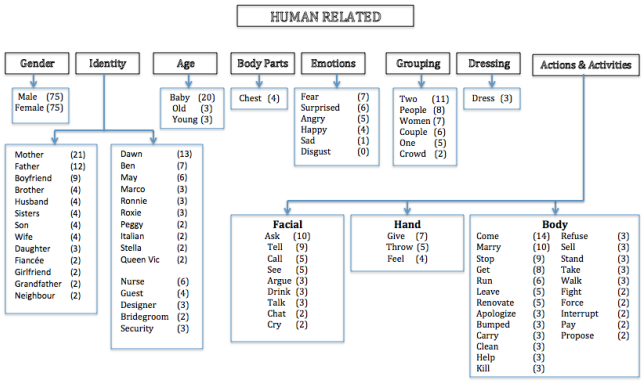


Fig. 7. Human related features in the hand annotations (with audio)

Finding Overlapped Key Interest Points The hand annotations are fed into an automatic summarizer tool[7] to identify the sentence relevance and the best keywords. This automatic summarization tool works in three phases. In the first phase, it will extract the sentences from the input text. Next, it will identify the keywords in the text and count each word's relevance. And in the final phase, it will identify the sentences with the most relevant keywords and displaying them based on the options selected. Table 3 and Table 4 shows the sentences with the highest relevance when the threshold[8] is set to 80. Based on these findings, the overlapped key interest points that have been identified for Video_ID: 5082189274976367100 are: they are two women having a conversation at the hospital; one of the woman ran out from the hospital crying after giving the baby; one of the woman argues with the nurse to get out from the hospital.

Using Audio Data to Reduce Erroneous and Missing Information Table 5 shows the best keywords that were identified. It is shown that when

[7] http://www.tools4noobs.com/summarize

[8] The value used to limit the sentences based on their relevance. The relevance is determined by the number or relevant words in it

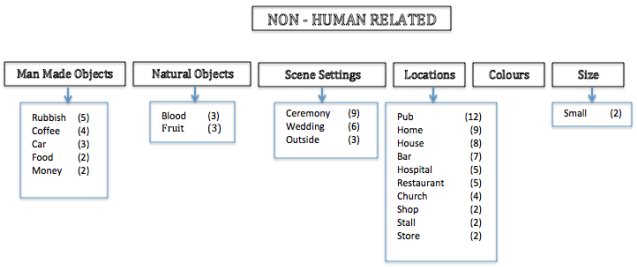


Fig. 8. Non-human related features in the hand annotations (with audio)

Table 3. Sentence relevance for hand annotations without audio (Video_ID : 5082189274976367100)

Summary	**Sentence Relevance**
There is one patient lying at the hospital bed and is talking with another woman who is holding a baby.	40
In the hospital, the young woman prepares to go back while a nurse came to talk to her.	37
One is a young woman sitting on the bed and one is middle age woman standing holding the baby.	32
A woman fought with nurse possibly about getting out from the hospital.	32
She ran out from the hospital and cried after gave the baby to that young lady.	32

audio data is present, the participants are keener towards identifying the identity of the human subjects (e.g., 'Dawn', 'mother'). Besides that, the key information of the video is also identified. In the hand annotations (without audio), although they managed to identify that the two women are having a conversation, the information regarding the conversation itself is missing. Keywords such as 'probably', 'possibly', and 'maybe' were often used. In the hand annotations (with audio), there is a substantial increment in relevance for the keyword 'baby'. This clearly shows that the participants have grasped the key information of the video that is about the two women arguing over the baby. Therefore, we can conclude that incorporating audio data may reduce erroneous and missing information.

This experiment also shows that there are a few challenges to be overcome when these two types of data are incorporated. First is to establish the relation between what is spoken and what is shown visually. The audio information extracted may or may not be related to the events or activities that are happening in that particular scene. For example, a conversation may be something about

Table 4. Sentence relevance for hand annotations with audio (Video_ID : 5082189274976367100)

Summary	Sentence Relevance
The mother begs and cries to get the baby and called the nurse to ask the middle age woman to leave.	45
Dawn was furious and demanded her baby but the lady tried to increase the amount of money.	44
May wants to take the newborn baby from Dawn who is the mother as they agreed before by paying some amount of money.	43
Dawn had just gave birth and a lady was trying to take her baby away and claimed that they had agreed on giving the baby to her in return of GBP10, 000.	41
Dawn want to leave the hospital but the nurse try to stop her because she need some rest.	37
Dawn wants to leave from the hospital with her baby.	37
The baby's mother want to go back home because she worries the middle age woman will come back and steal her baby.	37

Table 5. Best keywords for hand annotations without audio (left) and with audio (right) for Video_ID: 5082189274976367100

Keyword	Relevance
Woman	13
Hospital	10
Baby	8
Nurse	7
Lady	6

Keyword	Relevance
Baby	16
Dawn	11
Stop	6
Mother	6
Nurse	6

the past that differs (non-relevant) from what is visually shown. Future work should consider a decision-making process to filter non-relevant audio information by crosschecking it with the visual features and calculate their overlapped similarities.

Secondly, various audio processing tasks should be incorporated to get optimum results. For example, detecting a person's identity or relationship. Speech recognition alone is not sufficient to determine which spotted keyword can be associated with which detected person. It should include various cues in audio and video to determine either the keyword is referring to the person whom he/she is having the conversation with or a third person that may or may not be present in the video stream. Associating a detected person with a keyword that represents his identity or relationship is a challenge that is yet to be overcome.

Third, this experiment uses the Eastender dataset that has been crafted to include scenes with human activities and events. Thus, it is "rich" in both

audio and visual information to highlight the key interest points in the video stream. Different set of guidelines should be given to the participants depending on the type of the video dataset. For example, a lecture video may include a person presenting a PowerPoint slide. Although, the audio features may differ to their detected visual counterpart, in this context the information is relevant to describe the video content. For surveillance videos, the guideline should outline what is expected to be annotated. Due to the nature of this type of dataset that has no clear storyline or video highlights, a clear guideline is crucial to minimize hand annotations that are too diverse or subjective between one another.

Therefore, in order to incorporate audio data towards curbing missing information, these are the challenges that need to be put into consideration to achieve optimum results.

6 Conclusion

This paper has proven that although visual data is sufficient to detect humans, their interactions with related objects, actions, and scenes, using this information alone to generate natural language descriptions may not be able to capture the "key interest point" of the video content. An ideal video summary provides a brief overview of the video. It is not merely stating what is present (detected) in the video. Therefore, incorporating audio data is crucial towards curbing missing or erroneous information. Future work should consider the challenges that may arise when incorporating both of these data primarily the challenge of filtering relevant and non-relevant information. The corpus dataset (hand annotations) can also be used as a mean of evaluation against future works on natural language generation of a video stream.

References

1. Smith, M.A. and Kanade, T.: Video Skimming and Characterization through the Combination of Image and Language Understanding Techniques. In: Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference, pp. 775–781. (1997)
2. Lienhart, Rainer and Pfeiffer, Silvia and Effelsberg, Wolfgang: Video Abstracting. In: Commun. ACM, vol. 40, pp. 54–62. New York (1997)
3. He, Liwei and Sanocki, Elizabeth and Gupta, Anoop and Grudin, Jonathan: Autosummarization of Audio-video Presentations. In: Proceedings of the Seventh ACM International Conference on Multimedia (Part 1), pp. 489–498., Orlando, Florida (1999)
4. Uchihashi, Shingo and Foote, Jonathan and Girgensohn, Andreas and Boreczky, John: Video Manga: Generating Semantically Meaningful Video Summaries. In: Proceedings of the Seventh ACM International Conference on Multimedia (Part 1), pp. 383–392., Orlando, Florida (1999)
5. Yeung, M.M. and Boon-Lock Yeo: Video Visualization for Compact Presentation and Fast Browsing of Pictorial Content. In: Circuits and Systems for Video Technology, IEEE Transactions, pp. 771–785., (1997)

6. Assfalg, Jürgen and Bertini, Marco and Colombo, Carlo and Bimbo, Alberto Del and Nunziati, Walter: Semantic Annotation of Soccer Videos: Automatic Highlights Identification. In: Comput. Vis. Image Underst., vol. 92, pp. 285–305. New York (2003)
7. Cui, Bin and Pan, Bei and Shen, HengTao and Wang, Ying and Zhang, Ce: Video Annotation System Based on Categorizing and Keyword Labelling. In: Database Systems for Advanced Applications, vol. 5463, pp. 764–767. Springer, Heidelberg (2009)
8. Khan, Muhammad Usman Ghani and Nawab, Rao Muhammad Adeel and Gotoh, Yoshihiko: Natural Language Descriptions of Visual Scenes: Corpus Generation and Analysis. In: Proceedings of the Joint Workshop on Exploiting Synergies Between Information Retrieval and Machine Translation (ESIRMT) and Hybrid Approaches to Machine Translation (HyTra), pp. 38–47., Avignon, France (2012)
9. Khan, Muhammad Usman Ghani and Lei Zhang and Gotoh, Yoshihiko: Generating coherent natural language annotations for video streams. In: Image Processing (ICIP), 2012 19th IEEE International Conference, pp. 2893-2896, (2012)
10. Niveda Krishnamoorthy and Girish Malkarnenkar and Raymond Mooney and Kate Saenko and Sergio Guadarrama: Generating Natural-Language Video Descriptions Using Text-Mined Knowledge, (2013)
11. Kojima, Atsuhiro and Tamura, Takeshi and Fukunaga, Kunio: Natural Language Description of Human Activities from Video Images Based on Concept Hierarchy of Actions. In: Int. J. Comput. Vision, vol. 50, pp. 171–184., Hingham, MA, USA (2002)
12. Das, P. and Chenliang Xu and Doell, R.F. and Corso, J.J. : A Thousand Frames in Just a Few Words: Lingual Description of Videos through Latent Topics and Sparse Object Stitching. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference, pp. 2634-2641., (2013)

A text classification framework based on optimized Error Correcting Output Code

Mario Locci and Giuliano Armano

DIEE Dept. of Electrical and Electronic Engineering, University of Cagliari,
Piazza dArmi 09123, Cagliari, Italy
locci.mario@gmail.com,giuliano.armano@diee.unica.it
http://www.diee.unica.it

Abstract. In recent years, there has been increasing interest in using text classifiers for retrieving and filtering infomation from web sources. As the numbers of categories in this kind of software applications can be high, Error correcting Output Coding (ECOC) can be a valid approach to perform multi-class classification. This paper explores the use of ECOC for learning text classifiers using two kinds of dichotomizers and compares them to each corresponding monolithic classifier. We propose a simulated annealing approach to calculate the coding matrix using an energy function similar to the electrostatic potential energy of a system of charges, which allows to maximize the average distance between codewords —with low variance. In addition, we use a new criterion for selecting features, a feature (in this specific context) being any term that may occur in a document. This criterion defines a measure of discriminant capability and allows to order terms according to it. Three different measures have been experimented to perform feature ranking / selection, in a comparative setting. Experimental results show that reducing the set of features used to train classifiers does not affect classification performance. Notably, feature selection is not a preprocessing activity valid for all dichotomizers. In fact, features are selected for each dichotomizer that occurs in the matrix coding, typically giving rise to a different subset of features depending on the dichotomizers at hand.

Keywords: ECOC classifiers, Simulated Annealing, Feature extraction

1 Introduction

Multi-class classification consists of assigning a given pattern x to a category taken from a predefined set, say $c \in C$, with $C = \{c_1, c_2, c_3, \ldots, c_m\}$. Several approaches have been devised to directly handle multi-class problems (e.g., decision trees [13] and CART [2]). Other algorithms, originally designed to handle binary problems have been extended to handle multi-class problems. Multi-class support vector machines (SVM) [**?**] are a notable example of this strategy. Other methods turn multi-class problems into a set of binary problems. Classical examples of this approach are: one-against-all and one-against-one. The former consists of handling multi-class problem with m binary classifiers, each trained

to discriminate the i-th class against the others. The latter uses a binary classifier to discriminate between each couple $\langle c_i, c_j \rangle, i \neq j$ of categories. In so doing, the overall number of classifiers ends up to $m \cdot (m-1)/2$.

An alternative approach to solve multi-class learning task is to adopt Error-Correcting Output Coding (ECOC). Error correcting codes are widely used in data transmission, being in charge of correcting errors when messages are transmitted through a noisy channel. A simple encoding strategy in data transmission is to add extra bits to any given message, so that the receiver will be typically able to correct it in presence of noise. A variation of this this basic principle is applied with success in the field of machine learning, to improve the performance of multi-class classifiers. The basic ECOC strategy is to assign a binary string of length n (i.e., a codeword) to each category, trying to separate as much as possible each codeword from the others. The set of codewords can also be viewed as a coding matrix, in which binary classifiers are related to columns, whereas categories are related to rows. Hence, the i-th classifier will consider samples taken from the j-th category as negative or posative depending on the value, i.e., -1 or 1, found at position $\langle i, j \rangle$ of the coding matrix. This approach was first used in the NETtalk system [15]. Dietterich and Bakiri[5] have shown that ECOC can improve the generalization performance of both decision trees (experiments have been made with C4.5) and neural networks (using backpropagation), in several benchmark datasets. They have also shown that ECOC is robust with respect to changes in the size of training samples as well as in changes of codeword assignments. Interesting experimental results has been obtained by Berger [1] on several real-world datasets of documents. The author has shown that ECOC can offer significant improvements in accuracy over conventional algorithms on tree over four datasets used for experiments. In this paper, the author used Naive Bayes (NB) [11] as base classifier, whereas the codeword assignments were chosen randomly.

1.1 Coding strategies

Since the first ECOC has been designed, many experiments have shown that, to achieve a good generalization capacity, codewords must be well separated, which implies that the corresponding binary classifiers are trained on different subsets of data. The most commonly used distance measure is the Hamming distance. Given two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, with components $x_i, y_i \in \{-1, +1\}$, the Hamming distance $d(\boldsymbol{x}, \boldsymbol{y})$ is defined as:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=0}^{n} \frac{|x_i - y_i|}{2} \tag{1}$$

In the training phase n binary classifiers are trained with samples relabeled in accordance with the coding matrix, say Ω. The trained classifiers have an output vector $\boldsymbol{y}$, y_j being the output of the corresponding j-th binary classifier. The decoding strategy is to assign to the output vector $\boldsymbol{y}$ the category that

corresponds to the closest codeword. In symbols (with ω_j = codeword of the j-th category and d = adopted distance function):

$$\arg\min_j d(\boldsymbol{\omega_j}, \boldsymbol{y}) \tag{2}$$

ECOC were used successfully in many application areas. It was shown that randomly generated matrices often perform well, sometimes even better than those generated by heuristic methods. Random codes were theoretically studied in [9], showing that the condition to obtain an optimal Bayes is to have equidistance between each pair of codewords, when the random code is large enough the ECOC classifier tends asymptotically to optimal Bayes if the base classifier is an optimal Bayes classifier.

Although maximizing the distance between any pair of codewords helps to remove individuals classification errors, still decoding errors may occur [16]. The effect on decoding error can be understood by analyzing the decoding strategy and the Bayes decision rule. An ECOC matrix Ω performs a linear transformation between spaces, the original output $\boldsymbol{q}$ of the optimal Bayes classifier is transformed by the ECOC matrix in the corresponding output $\boldsymbol{p}$. With $\boldsymbol{q}$ probability vector (i.e, q_i is the probability of the i-th class), the output vector $\boldsymbol{p}$ is:

$$\boldsymbol{p} = \Omega^T \boldsymbol{q} \tag{3}$$

When all pairs of codewords are equidistant, Equation 2 implies maximizing posterior probability:

$$\arg\min_j q_j \tag{4}$$

An interesting class of ECOC coding is BCH (from R. C. Bose and D. K. Ray-Chaudhuri), which form a class of cyclic error-correcting codes constructed using finite fields. The key feature of this type of coding is the precise control of correctable symbols. An example of algorithm for generating BHC codes is described in [12]. This algorithm uses a polynomial of degree m to build the Galois finite field $GF(2^m)$. The length L of the binary code fulfills the following constraints: $2^{m-1} - 1 < L <= 2^m - 1$. Moreover, given the parameter t, which represents the number of correctable error, the minimum distance between pairs of codewords is $d = 2\,t + 1$.

1.2 Feature selection

A characteristic of text categorization problems is the high dimensionality of the feature space. Each document is typically represented using a bag of words. Each word being a base vector that generates the space of features, a document can be represented as linear combination of these base vectors. A major problem is that there can be hundreds of thousands of terms even for small text collections. The amount of words is prohibitively high for many learning algorithms. Hence, reducing the original space without losing accuracy is highly desirable.

Many methods to reduce the dimensionality of the feature space have been devised. Most of the methods select words according to their score obtained

by means of a suitable performance measure devised to check to which extent the word at hand is in agreement (or disagreement) with the category under analysis. χ^2 [**?**] and Information Gain (IG) (e.g., [**?**]) are well known measures used to perform feature (i.e., word) ranking . Yang and Pedersen [17] measure the goodness of a term globally with respect to all categories on average defining a general version of IG and χ^2 for multi-class problems. They found IG and χ^2 most effective in aggressive term removal without losing accuracy in their experiments with kNN and LLSF. Rogati and Y. Yang [14] analyzed 100 variants of five major feature selection and found that feature selection methods based on χ^2 statistics outperformed those based on other criteria. The problem of selecting features for ECOC is not particularly addressed in the literature even though in our view it is very important.

The remainder of this paper is organized into five sections: Section 2 describes the proposed approach for code optimization; Section 3 introduces a selection method based on the configuration of the coding matrix; Section 4 explains the real dataset used and the experimental settings; Section 5 reports and discusses experimental results and Section 6 ends the paper.

2 The proposed Simulated Annealing approach for optimizing ECOC

In this section, we propose a method based on simulated annealing (SA) to optimize the coding matrix. SA is a very robust algorithm, often able to find a global optimum and less likely to fail on difficult tasks [3] and [7]. In our case, SA explores a space $\mathfrak{D}$ of coding matrices characterized by m rows (the set of codewords) and n columns (the number of binary classifiers). Let us denote with $\Omega^* \in \mathfrak{D}$ the optimal (or sub-optimal) coding matrix.

The standard SA algorithm starts with an initial temperature $T = T_0$ and moves randomly in the neighborhood of the current tentative solution ω. SA is a local search algorithm, whose strategy consists of always accepting any new solution improves the current one. However to avoid local minima, SA may also accept worse solutions, with a probability inversely proportional to the current value of the temperature T. The convergence of the algorithm is guaranteed by decreasing T as the search goes on. The search continues until the maximum iterations has been performed or no relevant changes has been observed between two consecutive steps.

A solution in the neighborhood of ω is calculated by the neighbor function, described by Equation 5 (with z uniform random variable and p_1, p_2, p_3 given constants). In the specific setting of searching for the (sub)optimal ECOC coding matrix, a neighbor is generated from ω i) randomly changing a bit from -1 to 1 or vice versa with probability p_1, ii) adding a column vector with probability

p_2, or iii) removing a random column vector with $p_3 - p_2$.

$$neighbor(\omega) = \begin{cases} \text{change ramdomly a bit of } \omega & \text{if } z < p_1 \\ \text{add a random column vector to } \omega & \text{if } z < p_2 \\ \text{remove a random column vector from } \omega & \text{if } p_3 > z > p_2 \end{cases} \tag{5}$$

In the proposed variant of the SA algorithm, the cost function is analogous to the potential energy of a particle system of electric charges, and is defined by Equation 6, where $\boldsymbol{\omega}_i$ and ω_j are codewords of Ω.

$$f(\omega) = \sum_{i=0}^{m} \sum_{j>i}^{m} \frac{1}{d(\boldsymbol{\omega_i}, \boldsymbol{\omega_j})^2} \tag{6}$$

The ECOC optimization method which makes use of SA will be denoted as SAE, hereinafter. Moreover, SAE which makes use of classifiers of type $\langle x \rangle$ will be denoted SAE$\langle x \rangle$.

3 Feature selection ECOC dependent

As text categorization has a very high feature space (a typical order of magnitude is $10,000$), a feature selection method is needed. Our approach is enforced after having found the coding matrix, as in our view each individual binary classifier should have its proper subset of features.

Many selection methods are based on the estimation of words probability, class probability and the joint probability of words and classes. These methods are usually computed considering only the corpus of documents, independently from the way classifiers group the data. This is reasonable if the adopted kind of classifier is inherently multi-class (e.g., NB classifiers). However, an ECOC classifier actually embodies a set of n dichotomizers (being n the length of the codewords). In particular, given a dichotomizer g_j, a category c_i can be considered as source of negative or positive samples, depending on which symbol appears at position $\langle i, j \rangle$ of the coding matrix (-1 for negative samples and 1 for positive samples). This is the reason why performing feature selection for each individual dichotomizer appears a reasonable choice. To help the reader better understand the whole process, let us summarize the whole procedure:

1. the coding matrix is calculated;
2. The given set of samples, say S, is split in two sets (i.e., S^+ and S^-), in accordance with the content of the coding matrix;
3. Features are ordered in descending order starting from the highest score;
4. The set of features is reduced by selecting the first K features (where K is a given constant).[1]

Feature ranking has been performed according to three measures of discriminant capability, which will be described in the next subsection.

[1] Typical values of K range from 5% to 40% of the original feature space dimension.

3.1 Measures of Discriminant Capability

Three measures of discriminant capability have been experimented to perform feature ranking: χ^2, IG, and δ. The first and the second measures are well known. Let us spend few words on the method denoted as δ. It originates from the proposal of Armano [**?**], focused on the definition of an unbiased [2]two-dimensional measure space, called $\varphi - \delta$. In particular, φ has been devised to measure the so-called characteristic capability, i.e., the ability of the feature at hand of being spread ($\varphi = 1$) or absent ($\varphi = -1$) over the given dataset. Conversely, δ has been devised to measure the so-called discriminant capability, i.e., the ability of the feature at hand of being in accordance ($\delta = 1$) or in discordance ($\delta = -1$) with the category under investigation. It is worth pointing out that the actual discriminant capability of a feature can be made coincident with the absolute value of δ, as the ability of separating positive from negative samples is high when $|\delta| \approx 1$, regardless from the fact that the feature is highly covariant or highly contravariant with the given category.

Focusing on the selected measure (i.e., δ), let us recall its definition:

$$\delta = tp - fp \tag{7}$$

where tp and fp are respectively true and false positive rates of the main class.

A definition of this measure in the event that samples are a corpus of documents and features the terms (or words) found in the corresponding dictionary, is the following:

$$\delta(t, c) = \frac{\#(t, c)}{|c|} - \frac{\#(t, \bar{c})}{|\bar{c}|} \tag{8}$$

where t denotes a word and c a category. Moreover, $\#(t, c)$ and $\#(t, \bar{c})$ denote the number of documents belonging to the main (c) or to the alternate ($\bar{c}$) category in which t appears, respectively. Of course, $|c|$ is the number of documents of the main category and $|\bar{c}|$ the number of documents of the alternative category.

4 Experimental settings

In all the experiments, base binary classifier were of two kinds: NB and SVM [6]. The following datasets have been selected:

– **Industry sector.** It is a collection of web pages extracted from the web site of companies from various economic sectors. The leafs of this hierarchy are web pages, the parent directory is an industry sectors or class. The data is publicly available at [8]. This dataset contains a total of 9555 documents divided into 105 classes. A small fraction of these documents (about 20) belongs to multiple classes, but in our experiments they have been removed from the corpus. Web pages have been preprocessed to filter out the HTML code.

[2] In the jargon of the author, a measure is "unbiased" when it is independent from the imbalance between positive and negative samples. Notable examples in this category of measures are sensitivity and specificity.

– **20 news groups dataset.** This is a well known dataset for text classification [10]. It is a collection of 20,000 messages posted by the users of UseNet, the worldwide distributed discussion system. The dataset collects posting messages taken from 20 different discussion groups. Each discussion group covers a topic: 5 groups are about companies and 3 are focused on religion topics. Other topics are: politics, sports, sciences and miscellaneous.
– **Library and multimedial materials.** It is a collection of library and multimedia materials classified manually by librarian. The dataset is a collection of recorded metadata that use the MARC format (MAchine-Readable Cataloging). MARC standards are a set of digital formats for the description of items catalogued by libraries. Each field in a MARC record provides particular information about the item the record is describing, such as author, title, publisher, date, language, media type, abstract, isbn, and subject. In this dataset each item is classified using the Dewey decimal classification taxonomy. The dataset contains 75207 items, of which 23760 are duplicated (abstracts and author fields and some other field are equals for duplicated items) and 11655 are unclassified. The remaining 39786, which are unique and classified, have been used in the experiments. We have performed experiments using a reduced form of the Dewey taxonomy, that considers the granularity of details from the root to the third level (the first three digits of the Dewey code). The resulting number of classes is 647.
– **The four universities dataset.** Four universities dataset is a collection of HTML web pages from computer science departments of various universities [4]. Documents that appear therein have been collected from January 1997 by the World Wide Knowledge Base (WebKb) project of the CMU text learning group. The dataset contains 8,282 Web pages divided into 7 classes, they are extracted from the Web sites of four universities. The data set is organized as a directory, each file is an HTML page. Web pages have been preprocessed also to remove the HTML code.

For each dataset we first processed the text of each document by removing puntuaction and stopwords. [3] For each experiment, we split the dataset at hand in two randomly-selected subsets (70% for training and 30% for testing). Classification accuracy has been used as performance measure. For each test, we ran 10 experiments using different data samples, then we computed mean and variance of the corresponding accuracies.

5 Experimental Results

5.1 Comparison of base classifier to ECOC classifier

To show the advantages of ECOC classifiers whose codeword matrix has been optimized with SA, accuracy is reported together with the one obtained with

[3] As for stopwords, we used two different blacklists, one for the Italian and one for the English language, as part one corpus of documents (i.e., the one concerning libraries) is in Italian.

the corresponding base classifiers. Table 1 reports experimental results (the best results are highlighted in bold). In particular we observed that:

- ECOC classifiers generally perform better than base classifiers. However, better results are obtained with base classifiers in the four universities dataset. Let us also note that improvements are not statistically significant for the library dataset. These two data sets have in common the fact of being highly unbalanced.
- There are significant differences between the performance of the SVM and NB classifiers and this difference affects also the performance of the corresponding ECOC classifiers.

Table 1. Comparison among ECOC classifiers and base classifiers (Legenda: NB=Naive Bayes classifier; SAENB=ECOC based on NB; SVM=support vector machine; SAESVM=ECOC with SVM base classifier).

Dataset	NB	SAENB	SVM	SAESVM
4 universities	**.606**(1.62)	.584(1.56)	**.859**(2.29)	.851(2.27)
20 news	.868(2.32)	**.883**(2.35)	.896(2.39)	**.906**(2.42)
Ind. sector	.751(2.00)	**.844**(2.25)	.870(2.32)	**.879**(2.34)
library cat.	.588(1.57)	**.594**(1.58)	.625(1.67)	**.629**(1.68)

5.2 Comparative analysis of SAE, Random and BHC ECOC

In these experiments we imposed the same length of the codeword for all ECOC classifiers (i.e., 63 bits). Algorithms have been configured as follows:

- Random (RA): Random values -1 and 1 of the matrix bits are chosen with the same probability;
- BHC: the minimum value of the corrective capacity is chosen equals to $t = 6$, so that the minimum distance between codewords is $d = 2t + 1 = 12$;
- SA: The initial matrix state is obtained by using the algorithm RA. relevant parameters have been set as follows: $T_0 = f_0/5$, $T_{min} = 0.01$, $L_0 = 30$, and $N = 100$.

We used the same training partition of the data set to train the ECOC matrices obtained with three different algorithms. We ran ten experiment computing the mean and variance of the accuracy, Table 2 shows experimental results. We calculated also the mean (μ) and standard deviation (σ) between pairs of codewords for a matrix of size 100×104, the matrix calculated by the RA algorithm has $\mu = 49.96$ and $\sigma = 5.9$, whereas the one calculated by the SA algorithm has $\mu = 50.48$ and $\sigma = 2.77$. We observed that

– SA reduces the gap between minimum distance and maximum distance of codeword pairs, increases the minimum and the mean distance reducing the variance.
– SAE can achieve better performance than others for most of the datasets.

Table 2. Accuracy comparison of SA, Random and BHC ECOC.

Dataset	SAENB	SAESVM	RANB	RASVM	BHCNB	BHCSVM
4 univ.	.584±1.56	**.851**±2.27	.580±1.55	.842±2.24	**.590**±1.57	.850±2.27
20 news	**.883**±2.35	**.906**±2.42	.882±2.35	.902±2.41	.880±2.35	.899±2.40
Ind. sector	**.844**±2.25	**.879**±2.34	.832±2.22	.864±2.30	.839±2.24	.868±2.31
library cat.	**.594**±1.58	**.629**±1.68	.582±1.55	.624±1.66	.582±1.55	.627±1.67

5.3 Comparison Among χ^2, IG and δ

Selection the best terms able to ensure a good performance in terms of time and memory consumption plays a fundamental role in text classification, in particular when the selected corpus contains many documents and / or the corresponding dictionary is contains many words. This section reports a comparative assessment of the selected score functions. Table 3 reports experimental results, the best results being highlighted in bold. In particular, we found that the ordering among score function (from the best downwards) is the following: χ^2, δ and IG.

Table 3. Comparison between feature selection based χ^2, IG and δ.

Dataset Feature s.	SAENB δ	SAENB IG	SAENB χ^2	SAESVM δ	SAESVM IG	SAESVM χ^2
4 univ.	.598±1.59	.594±1.58	**.635**±1.69	.851±2.26	.849±2.26	**.861**±2.29
20 news	.883±2.35	.875±2.33	**.894**±2.38	.906±2.41	.905±2.41	**.909**±2.42
Ind. sector	.839±2.24	.811±2.16	**.854**±2.28	.877±2.34	.867±2.31	**.885**±2.36
library cat.	.564±1.50	.543±1.45	**.567**±1.51	**.614**±1.63	.608±1.62	.605±1.61

6 Conclusions and Future Work

In this paper a novel approach for building ECOC classifiers has been proposed. The corresponding algorithm is based on simulated annealing, whose energy function is anologous to the potential of a system of charges. Experimental results show that in the configuration of minimum energy the distances between codewords have high mean and low variance. A method for feature extraction based on the coding matrix has also been presented, three score functions for

selecting words have been compared. As for future work, more detailed experiments will be made on the ability of score functions to guarantee good classification performance. In particular, the generalized version of δ, able to deal with unbalanced datasets, will be experimented in a comparative setting.

References

1. A. Berger. Error-correcting output coding for text classification. In *IJCAI-99: Workshop on machine learning for information filtering*. Citeseer, 1999.
2. L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.
3. A. Corana, M. Marchesi, C. Martini, and S. Ridella. Minimizing multimodal functions of continuous variables with the simulated annealing algorithm corrigenda for this article is available here. *ACM Transactions on Mathematical Software (TOMS)*, 13(3):262–280, 1987.
4. M. Craven, A. McCallum, D. PiPasquo, T. Mitchell, and D. Freitag. Learning to extract symbolic knowledge from the world wide web. Technical report, DTIC Document, 1998.
5. T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *arXiv preprint cs/9501101*, 1995.
6. R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.
7. W. L. Goffe, G. D. Ferrier, and J. Rogers. Global optimization of statistical functions with simulated annealing. *Journal of Econometrics*, 60(1):65–99, 1994.
8. M. G. Inc. Industry sector dataset, 2011. on line 2015.
9. G. James and T. Hastie. The error coding method and picts. *Journal of Computational and Graphical Statistics*, 7(3):377–387, 1998.
10. K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th international conference on machine learning*, pages 331–339, 1995.
11. D. D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *Machine learning: ECML-98*, pages 4–15. Springer, 1998.
12. C. Lin. *Error Control Coding: Fundamentals and Applications*, volume 1. Prentice Hall, 1983.
13. J. Quinlan. C4. 5: Programs for empirical learning, 1993.
14. M. Rogati and Y. Yang. High-performing feature selection for text classification. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 659–661. ACM, 2002.
15. T. J. Sejnowski and C. R. Rosenberg. Parallel networks that learn to pronounce english text. *Complex systems*, 1(1):145–168, 1987.
16. T. Windeatt and R. Ghaderi. Coding and decoding strategies for multi-class learning problems. *Information Fusion*, 4(1):11–21, 2003.
17. Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420, 1997.

Modeling Socio-Psychological Behaviors in the Era of the WWW: a Brief Overview

Marco Alberto Javarone[1,2]

[1] Dept. of Mathematics and Computer Science, University of Cagliari, Cagliari Italy
[2] Dept. of Humanities and Social Science, University of Sassari, Sassari Italy
marcojavarone@gmail.com

Abstract. The World Wide Web has deeply changed our world and, over years, it has shaped our society. Social networks like Facebook strongly speed up the spreading of information among users, and allow people to communicate simultaneously with several individuals. As result, when studying sociological phenomena and socio-psychological behaviors, we have to consider the influence that the WWW has on people's life. In this work, we briefly present some computational models that can be adopted for representing socio-psychological behaviors in this scenario.

Keywords: evolutionary game theory, sociophysics, human behavior

1 Introduction

Sociophysics [1, 2] is a modern research field focused on investigations of socio-economic systems by means of computational and analytical models. Just to cite few, sociophysics deals with opinion dynamics [1, 2], language dynamics [3, 4], crowd dynamics [2], economy [5].Notably, simple models like the voter models [6] are able to represent simplified scenarios of opinion spreading, and to identify exact solutions. Although these analytical approaches often require a high level of abstraction compared to the real scenarios (e.g., electoral campaigns), they allow to introduce a mathematical formalism to study social issues. Moreover, agent-based models constitute a powerful framework —see [7] for modeling social dynamics, that can be combined with the modern network theory [8]. It is worth to recall that a list of qualitative models, developed in sociology and in social psychology, has been analyzed under the lens of statistical physics. In the last years, the WWW has deeply shaped our society and, in general, the life of several people.Therefore, both sociology and social psychology have to consider this modern world when studying social phenomena and behaviors. In the light of these considerations, in this work we report a brief summary of some socio-psychological behaviors analyzed in the context of complex networks [8]. In particular, we consider relevant to identify computational models able to describe human behaviors because, although a lot of data (currently defined 'Big Data'), a general mathematical framework to deal with them still lacks.

2 Models

We briefly present two different study-cases to show how human behaviors strongly affects dynamics in social systems, as social networks in the WWW.

Competitiveness. In the proposed model [9], we study a population whose agents play the Prisoner's Dilemma (hereinafter PD) in a continuous space (see also [10]). In so doing, agents play the PD with their neighbors computed according to an interaction radius. In principle, the PD is a very simple game where agents behave as cooperators or as defectors and, according to a payoff matrix, they compute their payoff after each challenge. Notably, the payoff matrix of the PD can be defined as follows

$$\begin{array}{c} \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cc} C & D \end{array} \\ \begin{pmatrix} 1 & S \\ T & 0 \end{pmatrix} \end{array} \tag{1}$$

The two strategies, i.e., cooperation (C) and defection (D), are grouped in the set $\Sigma = \{C, D\}$. Moreover, the parameter T represents the *Temptation*, i.e., the payoff gained by defectors when face cooperators, while parameter S the *Sucker's payoff*, i.e., the payoff obtained by cooperators when face defectors. Values of T and S are in the following range (in the PD): $1 \leq T \leq 2$ and $-1 \leq S \leq 0$. Results of numerical simulations can be studied by analyzing the TS-plane, computed on varying the value of S and T. In this scenario, it is interesting to analyze if a cooperative behavior emerges on varying S and T, when considering 'competitive' agents. Notably, agents have an interaction radius whose length depends on their payoff: as it increases/decreases their interaction radius increases/decreases. Thus, agents with high payoff become more competitive.Here, we consider the same geometrical framework of [11]) but with two main differences: *i)* agents cannot move and *ii)* agents may vary their interaction radius. Eventually, in all simulations we consider an equal initial distribution of strategies. Results, shown in panel **a** of Figure 1, suggest that 'competitiveness' strongly increases the level of cooperation in a population playing a game (i.e., the PD), characterized by an opposite Nash equilibrium (i.e., defection)

Group Polarization. Now we focus on the emergence of extreme opinions [12], by considering the theory of group polarization [13]. The latter is a collective phenomenon that occurs when groups of individuals are taking a decision. In order to model this phenomenon, in the context of social networks (and then of the WWW), we propose an agent-based model considering a system with 3 opinions: two opposite opinions and one representing the extreme form of one of them. For instance, opposite opinions may represent feelings pro-western (pw) and anti-western (aw), respectively, while the third opinion may represent the terrorist/passive supporter [14] ideal. Agents are arranged on a small-world network, so that they can interact with their neighbors; although we impose that they cannot change opinion from $+1$ (i.e., pw) to -1 (i.e., aw), and vice versa,

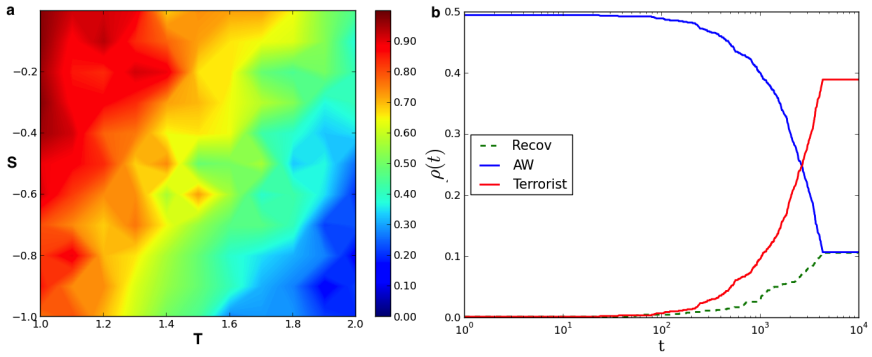


Fig. 1. a) Cooperation frequencies in the TS-plane achieved by agents provided with $k(0) = 4$. Colors indicate the averaged degree of cooperation achieved by the population. We recall that red indicates strong cooperation, while blue defection (i.e., no cooperation) —see [9]. **b)** Density of agents having the opinions aw and Terrorist, over time. The density of aw agents decreases until it reaches the density of recovered ones (i.e., those who quit secret meetings) see [12].

over time. We suppose that one agent of the network is a terrorist (hereinafter TL), with an opinion $s = -2$ representing an extreme form of the anti-western feeling. Then, at each time step, TL tries to convince other agents, among those with the aw feeling, in order to organize meetings. Each aw agent accepts the invitation with probability $p^r \in [0, 1]$ (equal for all aw agents). As aw agents accept to attend secret meetings, new connections emerge among them, giving rise to the emergence of a sub-community (having a structure similar to a fully-connected network). According to the theory of 'group polarization', a small set of people with the same idea can be lead to take the idea to the extreme level; hence, a small set of aw agents risks to become terrorist due to the intra-interactions. The recruiting of aw agents is the underlying mechanism responsible for the variation of the social network. Considering the i-th recruited agent (i.e., one of the meetings' participants), its p^t (i.e., probability to become terrorist) and p^{out} (i.e., probability to quit to attend secret meetings) are computed as follows: $p_i^t = f(\sigma_i^-, \sigma_i^{--})$ and $p_i^{out} = \sigma_i^+$, with σ_i^- and σ_i^{--} densities of aw and terrorist agents in the social circle of the ith agent, respectively; and σ_i^+ density of pw agents in the social circle of the ith agent. The function $f(\sigma_i^-, \sigma_i^{--})$ has been devised in order to consider the presence of both aw and terrorist agents among neighbors of the ith agent. Results (see panel **b** of Figure 1) show that a high fraction of agents which takes part to meetings undergoes the phenomenon of group polarization.

3 Conclusions

In the era of WWW, the studying of social behaviors recovers a particular importance. Notably, our lives are strongly affected by social networks and all

devices that are connected on Internet. Friendships and other human relations now are developed and supported by virtual connections that allow individuals to be connected with a wide list of people. As results, several socio-psychological behaviors must be analyzed in this new technological context. Moreover the increasing number of digital traces, currently defined as 'Big Data', still requires the definition of a formal mathematical theory. Thus, analytical and computational approaches for studying the evolution of social systems, considering human behaviors, may represent viable methods to investigate social network dynamics. With this idea in mind, we present a brief report about social behaviors modeled in the context of the WWW, showing their central role in the dynamics and in the evolution of a population.

Acknowledgments

The Author thanks Fondazione Banco di Sardegna for supporting his work.

References

1. Galam, S.: Sociophysics: a review of Galam models. *International Journal of Modern Physics C* **19-3** 409-440, 2008
2. Castellano, C. and Fortunato, S. and Loreto, V.: Statistical physics of social dynamics. *Rev. Mod. Phys.* **81-2** 591–646, 2009
3. Baronchelli, A. Felici, M., Loreto, V., Caglioti, E., Steels, L.: Sharp transition towards shared vocabularies in multi-agent systems. *JSTAT*, P06014, 2006
4. Javarone, M.A., Armano G.,: Emergence of Acronyms in a Community of Language Users. *EPJ-B*, 86-11, 474, 2013
5. Mantegna, R., Stanley H.E.: Introduction to Econphysics: correlations and complexity in finance. *Cambridge University Press* 1999
6. Sood, V. and Redner, S.: Voter Model on Heterogeneous Graphs. *Phys. Rev. Lett.* **94-17** 178701, 2005
7. San Miguel, M. and Eguiluz, V.M. and Toral, R.: Binary and Multivariate Stochastic Models of Consensus Formation. *Computing in Science and Engineering* **7-6** 67–73, 2005
8. Albert R. and Barabasi, A.L.: Emergence of Scaling in Random Networks. *Science* **286** (5439) 509–512, 1999
9. Javarone, M.A., and Atzeni, A.E.: The Role of Competitiveness in the Prisoner's Dilemma. *Computational Social Networks* **2:15**, 2015s
10. Szolnoki, A., Perc, M.: Conformity enhances network reciprocity in evolutionary social dilemmas. *J. R. Soc. Interface* **12** 20141299, 2015
11. Antonioni, A., Tomassini, M., Buesser, P.: Random Diffusion and Cooperation in Continuous Two-Dimensional Space. *Journal of Theoretical Biology* 344, 2014
12. Javarone, M.A., and Galam, S.: Emergence of Extreme Opinions in Social Networks. *Social Informatics - LNCS Springer*, 111 –117, 2014
13. Aronson, E., Wilson, T.D. and Akert, R.M.: Social Psychology. Pearson Ed, 2006
14. Galam, S, Mauger, A.: On reducing terrorism power: a hint from physics. *Physica A: Statistical Mechanics and its Applications* **323** 695–704, 2003