LIA@RepLab 2013

Jean Valère Cossu, Benjamin Bigot, Ludovic Bonnefoy, Mohamed Morchid, Xavier Bost, Grégory Senay, Richard Dufour, Vincent Bouvier, Juan-Manuel Torres-Moreno, Marc El-Bèze

LIA@RepLab 2013

Jean-Valère Cossu*, Benjamin Bigot*, Ludovic Bonnefoy*,**, Mohamed Morchid*, Xavier Bost*, Grégory Senay*, Richard Dufour*, Vincent Bouvier***, Juan-Manuel Torres-Moreno*, and Marc El-Bèze*

* LIA/Université d'Avignon et des Pays de Vaucluse **
firstname.name@univ-avignon.fr
** iSmart
*** LSIS/Aix-Marseille University
firstname.name@lsis.org

Abstract. In this paper, we present the participation of the Computer Science Laboratory of Avignon (LIA) to RepLab 2013 edition. RepLab is an evaluation campaign for Online Reputation Management Systems. LIA has produced a important number of experiments for every tasks of the campaign: filtering, topic priority detection, Polarity for Reputation and topic detection. Our approaches rely on a large variety of machine learning methods. We have chosen to mainly exploit tweet contents. In several of our experiments we have also added selected metadata. A fewer number of our proposals have integrated external information by using provided links to Wikipedia and users homepage.

1 Introduction

RepLab addresses the challenging problem of online Reputation analysis, i.e. mining and understanding opinions about companies and individuals by extracting information conveyed in tweets. In this context, LIA's participants have proposed several methods to automatically annotate tweets.

The rest of this article is structured as follows. In section 2, we briefly discuss about datasets and RepLab tasks. In section 3, we present the LIA's submitted systems. Then in section 4, performances are reported before concluding and discussing some future works.

2 Data and Tasks

2.1 Data

The corpus is a multilingual collection of tweets referring to a set of 61 entities. These entities are spread into four domains: automotive, banking, universities and music/artists. These tweets cover a period going from the 1^{st} of June 2012 to the 31^{st} of December 2012. Entities' canonical names have been used as queries

** http://lia.univ-avignon.fr/

to extract tweets from a larger database. For each entity, at least 2,200 tweets have been collected. The 700 first tweets have been taken to compose the training set, and the other ones are for the test set. Consequently, tweets concerning each of the four tasks are not homogeneously distributed in the datasets. We have selected 8,000 tweets from the training collection to build a development set.

2.2 Filtering

The *Filtering* task consists in identifying, in a stream of tweets, those which are referring or not to a given entity and label these tweets as *related* or *unrelated*. For instance in the tweets written in English, systems have to distinguish if a tweet containing the word "U2" correctly refers to the famous music band or not. The lack of context is one of the main issue while processing tweets. These messages count only 140 characters and in many cases the text content is not sufficient to correctly classify a tweet as related or not.

2.3 Polarity for Reputation

The goal of the task *Polarity for Reputation* is to find if a tweet contains a positive, negative or neutral statement concerning the reputation of a company. This task is significantly different from a standard sentiment analysis since the objective is to find a polarity about a reputation, without considering if tweet contents are opinionated or not. For example, sentiments known as negative do not always imply a negative polarity for reputation characterization in tweets. We observed that the tweet "We'll miss you R.I.P. Whitney" has been associated with a *negative* label (the writer is sad because of someone's death), but this is undoubtedly a positive tweet about the reputation of Whitney Houston. Finally, polarity's definition may be really different depending on the considered entity.

2.4 Topic Priority Detection

In the *Topic Priority Detection* task, we look for the priority level (alert, mildly important, unimportant) of a topic. Priority classes have been defined as follow:

1. alert : the topic deserves immediate attention of reputation managers;
2. mildly relevant : the topic contributes to the reputation of the entity but does not require immediate attention;
3. unimportant : the topic can be neglected from a reputation management perspective;

It seems possible to detect priority levels without processing any new clustering task. Indeed, negative messages typically concern an information requiring a high priority reaction. Negative tweets may therefore be highly correlated with the higher priority level. Again many factors play a role on the understanding of the proposed priority level.

2.5 Topic Detection

Systems used for *Topic Detection* are asked, in a first time, to find out the main subject of a message and then to cluster related tweets. The objective is therefore to bring together tweets referring to the same subject with regards to a given entity.

3 Approaches

In this section we propose descriptions of the LIA's systems used in this edition.

3.1 TF-IDF-Gini approach with a SVM classification

We proposed a supervised classification method based on the Term Frequency-Inverse Document Frequency (TF-IDF) method using the Gini purity criteria coupled with a Support Vector Machines (SVM) classification. The system is composed of two main steps. The first one creates a vector representation of words using a term frequency Okapi/BM25 vector [9] with the TF-IDF-Gini method [10]. The second part uses the extracted vectors to learn SVM classifiers.

TF-IDF [9] has been widely used for extracting discriminative words from text. Several works have also reported improvements by using TF-IDF in association with the Gini purity criteria [10]. SVMs are a set of discriminative supervised machine learning techniques aiming at determining a separation hyperplane [1] that maximizes the *structural margin* between training samples.

Only tweet textual content is used with this approach. Classifiers have been trained with vectorial representation of words in order to automatically assign the most relevant class (for priority and polarity tasks) to a tweet. These tasks require a multi-class SVM classifier. We have chosen the *one-against-one* strategy and a linear kernel. This method have reported a better accuracy than the *one-against-rest* method [5].

3.2 Boosting classification approach

For the classification tasks, we propose to combine various features extracted from the tweets using a supervised machine learning meta-algorithm: the Boosting [6]. We chose to use the popular AdaBoost algorithm which is a variation of the classical boosting approach. AdaBoost is a multiclass large-margin classifier based on a boosting method of *weak* classifiers. The weak classifiers are given as input. They can be the occurrence or the absence of a specific word or n-gram (useful for linguistic features) or a numerical value. At the end of the training process, a list of selected rules is obtained as well as their weights. With this set of rules, a score for each class is computed on each data to classify. The classification tool used is IcsiBoost [8], an open source tool based on the AdaBoost algorithm such as the Boostexter software [7]. IcsiBoost presents the advantage to provide a confidence score between 0 (low confidence) and 1 (very confident)

for each instance to classify. This classification process proposes a categorization of the tweets according to its polarity and its priority. It takes into consideration information contained in the tweets:

1. user id;
2. tweet's textual content (bags of 3-grams max.);
3. language;
4. entity id;
5. category;
6. query string (bags of 3-grams max.);

Note that the tweet textual content has been normalized with some particular manual rules which mainly consist in separating punctuation from words (ex: "price!" becomes "price !""). We chose to not remove the punctuation from the tweet content because we assume that this information may be useful for polarity and priority classification.

3.3 Cosine distance with TF-IDF and Gini purity criteria

We proposed a supervised classification method based on a cosine distance computed over vectors built using discriminant features like Term Frequency-Inverse Document Frequency (TF-IDF) [12] using the Gini purity criteria [13]. This system consists in two steps. First the text is cleaned by removing hypertext links and punctuation marks and we generate a list of n-grams by using the Gini purity criteria. During this step stoplists (from Oracle.com) [1] for both English and Spanish have been used. In the second step we creates terms (words or n-grams) models for each class by using term frequency with the TF-IDF and Gini criterion. Models also contain specific tags when the second step has not been unable to properly produce feature from a training tweet. A cosine distance measures the similarity of a given tweet by comparing its bag of words to the whole bag built for each class and ranks tweets according to this measure. This classification process takes into account (depending on the task) one or several metadata among:

1. user id;
2. entity id;
3. language;

3.4 Continuous Context Models

Continuous Context Models (CCM) tend to capture and model the positional and lexical dependencies existing between a given word and its context. In this method, the presence in tweets of anchor words is required in order to build context vectors used in CCM. For every given entity of the data set, we consider a predefined set of words including hashtags, "@'s usernames" and other specific

[1] http://docs.oracle.com

terms. These words have been chosen on the training set in order to cover a large number of context examples for each entity.

According to the procedure formerly presented in [11], for one occurrence of a given entity in a tweet, we build one vector. This vector is filled with the relative positions of words in the entity's neighbourhood with reference to the entity's position in the tweet. Vectors are then taken together in order to build a context-to-entity matrix on which we apply a dimension reduction using a Singular Value Decomposition for matrix sparseness reduction. The matrix is then used to train a 2-class SVM classifier [1] with a linear kernel.

Continuous Context Models have been used for the filtering task and for polarity and priority classification. For the filtering task, the two classes are respectively composed of vectors extracted from unrelated and related tweets. For the polarity and priority classifications, the strategy is different. For these 3-class problems we have built three classifiers. For example for the polarity classification we have built a positive-versus-not-positive model (no-positive corresponds to negative plus neutral tweets), negative-versus-not-negative and neutral-versus-not-neutral. The same procedure has been used for priority classification. Decision rules for the final class attribution has been learnt on the training data set. We only use tweet text content in this experiments. A normalization consisting in turning upper-case characters to lower-case and removing punctuation marks have been done.

3.5 k-Nearest-Neighbour with discriminant features

This method can be considered as a very improved version of the baseline. The system tries to match each tweet in the test set with the N most similar tweets in the training set. Tweet similarity is computed using Jaccard measure on the bag-of-words discriminant representation of the tweets. The representation being built from TF-IDF Term Frequency-Inverse Document Frequency [12] combined with the Gini purity criteria [13]. The process also takes into account tokens created from the metadata (author, entity-id). A stoplist for both English and Spanish has been used. It contains tool-words and ID from entities which obtained a score equal to 0 with official measures on the development set.

3.6 Adaptation of the LIA's system used in KBA 2012

In collaboration with the LSIS, we participated last year to the Knowledge Base Acceleration (KBA) task in TREC 2012 [16]. The KBA task is very similar to the RepLab filtering and priority sub-tasks: filtering a time-ordered corpus for documents that are highly relevant to a predefined list of 29 entities from Wikipedia and assigning them a degree of priority among central (alert), relevant (mildly important), neutral (related but unimportant) and garbage (unrelated). Even if the definitions are similar, the type of documents studied are different: blogs, forum posts, news and web pages vs. tweets.

For the KBA task we developed a state-of-the-art approach, which captures intrinsic characteristics of highly relevant documents by mean of three types

$TF(e,d)$	Term frequency of the entity e in d
$TF_{10\%}(e,d)$	Term frequency of e for each 10% part of d
$TF_{20\%}(e,d)$	Term frequency of e for each 20% part of d
$C(sent,e,d)$	Count of sentences mentioning e
$entropy(d)$	Entropy of document d
$length(d)$	Count of words in d
$SIM_{1g}(d,sd)$	Cosine similarity between d and the entity's Wikipedia article, based on unigrams
$SIM_{2g}(d,sd)$	Cosine similarity with bigrams
$TF(re,d)$	Term frequency of related entities in d
$TF(reL,d)$	Term frequency of related entities (embedded in links) in d
$TF(e,d).IDF(e,1h)$	Term frequency in d and inverse document frequency for an hour
$DF(e,1day)$	Number of documents with e this day
$DF(e,7d)$	Number of documents with e in 7 days
$Var(DF(e,7d))$	Variance of the DF in 7 days
$TF(e,7d)$	Term frequency of e in 7 days
$TF(e,title,7d)$	TF of e in titles in 7 days

Table 1. Document centric features, Entity related features and Time features. TFs are normalized by the size of the document if applicable.

of features: document centric features, entity's profile features, and time features [17]. This set of features is computed for each candidate document and, using a classification approach, used to determine if it is related or not to a given entity. A Random Forest classifier have been used in these experiments. One important point of this approach over most KBA 2012 systems is that only one classifier has been trained for all the entities and it has been proven to remains competitive without training data associated to a specific tested entity.

We want to measure the performances of this approach on another kind of documents and with a minimum of adaption. Features peculiar to the KBA corpus have been removed and no additional features have been built to match the specific features of the RepLab corpus. Feature set is listed in Table 1.

Filtering task: we have submitted 3 runs for the filtering task:

- Run 4: Tweets are cleaned : stop-words are deleted as well as @ before a user name and hashtag are split. A classifier is trained on all positive and negative examples for the all set of entities;
- Run 5: Similar to Run 4 but a new set of features is computed on web pages pointed by the URLs in the tweet. If the tweet do not contain an URL the value of the corresponding each feature are set to "missing";
- Run 6: Similar to Run 5 but one classifier is trained by type of entities (automotive, universities, banking and music/artists).

Priority task one run has been submitted for the priority task . It is similar to Run 5 presented above. Two steps are used to associate a priority level to a document: at first documents are tested with a classifier trained on unimportant vs. mildly important/alert examples; then documents which that have not been associated to the unimportant class go through a second classifier trained to separate mildly important documents to alert ones.

3.7 Ultrastemming + n-grams

For the filtering task, we proposed a supervised classification method based on word n-grams in [14] and n-ultra stemming in [15]. Tweets in English and/or Spanish are present in the RepLab corpus. In order to avoid the language detection or the specific strategies to process each language, we use the common information of each words, i.e. their ultra stem. For example, Information and Información share the common 5-ultra stem "Infor". n-ultra stemming is a method of words normalization to further reduce the space of documents representation. We propose to truncate each word to its five initial letters. The algorithm is very simple: we computed 5-ultra stems of i tweets in learning corpus. Then two simple probabilistic language models (LM_X) of n-grams ($n = 1, 2, 3$) for each class (X=related/unrelated) have been created. We classify each tweet j of the testing set by computing the argmax(x) value over each LM_X. The results show that 5-stemming preserves the content information of each tweet, regardless their language, in order to filter the tweets.

3.8 Maximum a Posteriori Feature Selection

LIA's topic detection system at first relies on the identification of headwords (HW) characteristic of one topic. HW are words, bigrams, distance bigrams and tweet author selected using a Maximum A Posteriori probability (MAP) estimator. For one theme, we compile one ordered list of HW, ranked considering a purity criterion. An initial choice of features for theme hypothesization is a set HW_k for each T_k of discriminative theme headwords. In order to have a fair characterization of themes with discriminative word vocabularies, all headword vocabularies have been formed with the same size $|HW_k|$. Vocabularies of different themes may share some headwords.

In order to attribute a topic to a tweet, we compute the topic contribution of a tweet Y_d in each topics T. This topic contribution $HW(T_k|Y_d)$ is a sum of contributions of the tweet in the topic coded by the features selected for it. The topic is attributed to the topic with maximum HW contributions. Systems proposed by the LIA for the topic detection vary by the number of features selected.

3.9 Merging algorithms

LIA's methods presented above rely on very different approaches and we except that combining systems outputs by the use of merging algorithms to improve

the performances of any system taken alone. To this purpose, we have applied merging methods at every tasks except for topic detection. We have used a linear combination of scores, as well as ELECTRE and PROMETHEE algorithms. Seven of our systems have been combined for polarity detection and filtering tasks and six for priority classification.

Linear combination of outputs scores: We dispose of N systems. For one tweet T of the set set, one system propose an entity label L_k with $k = 1 \dots 61$ and a corresponding output score $s_j(T_i, L_k)$. We first normalize to 1 the sum of scores provided by a system over the whole test set. The output entity label L is chosen according to

$$\gamma(T_i, L) = k = 1 \dots 61 arg\,max \left(\sum_{j=1}^{N} s_j(T_i, L_k) \right), \quad (1)$$

ELECTRE method: the objective of this method [18] is to chose the best system from the entire set of systems. This methods first consists in ranking entity labels comparing to each others by considering how an entity dominates another one. In a second time the method evaluates the rate of systems where this dominance between entity labels appears.

PROMETHEE method: The Preference Ranking Organisation METHod for Enrichment Evaluations [18] is a multi-criteria analysis method. It compare several alternative of actions taken by pair and measure the capacity of an entity label to dominates the others candidates and its capacity of being be dominated by the other ones. It finally creates a ranking of several alternatives.

4 Submissions and results

Eleven methods compose the LIA's set of submissions. For reading convenience these methods are summed up in Table 2 and refer to a method number used in results table presented above. We now compare our result with regards to the baseline and also to the median score computed over the scores obtained by all the RepLab particpant for a given task.

Filtering task: most of our runs, ranked according to F-measure in Table 3, are situated between the median and the baseline. Two systems (nb 1 and nb 4 with a F-measure scores of respectively 0.3819 and 0.3412) have reached performances greater than the baseline. The confidence interval (0.002) shows that in terms of accuracy many systems are equivalent despite of what can be seen according to F-measure. Merging strategies (methods 6, 7 and 8) have not been able to produce good selection rules since their performances remains lower than our best runs taken alone, a selection of best candidates before the merging would

# Method	Description
1	k-NN with discriminant features
2	Cosine distance with TF-IDF and Gini purity criteria
3	Continuous context models
4	Adaptation of the LIA's system used in KBA 2012
5	Ultrastemming + N-Grams
6	PROMETHEE
7	ELECTRE
8	Linear system combination
9	Boosting classification approach
10	TF-IDF-Gini approach with a SVM classification
11	Maximum a Posteriori Feature Selection

Table 2. LIA's systems for RepLab 2013

Run Id	# Method	ACCURACY	RELIABILITY	SENSITIVITY	F-MEASURE
Run 1	1	.8720	.6584	**.3566**	**.3819**
Run 6	4	.8764	.6195	.3311	.3412
Baseline		*.8714*	*.4902*	*.3200*	*.3255*
Run 10	8	**.8827**	.6805	.2906	.3127
Run 9	7	.8792	.6804	.2819	.3024
Run 8	6	.8745	**.6874**	.2661	.2962
Run 5	4	.8501	.4895	.3104	.2897
Run 4	4	.8501	.4895	.3104	.2897
Run 2	2	.8351	.4234	.3312	.2720
Median		*.8260*	*.4895*	*.2869*	*.2655*
Run 7	5	.8067	.2687	.2297	.1870
Run 3	3	.8000	.6319	.1241	.1265

Table 3. Submitted runs to Filtering Task ordered according to the F-Measure.

have been better. Moreover, the differences between entity label distributions of data in training and test sets may introduce some noise during the learning process. We have observed better performances by using a development set where entity label populations are more equally distributed in training and test set.

Polarity task: performances ranked according to the Pearson correlation are reported in Table 4. One important aspect of polarity systems consists of predicting the average polarity of an entity with respect to other entities. To cover this aspect, correlation is computed between the average polarity of entities versus the reference. This is therefore not necessary to capture the polarity of all tweets to correctly estimate the average polarity. In this task, most of our proposal performances are between the median and the baseline scores. One method (number 1) is over the baseline and reaches a correlation value equal to 0.8799. Here again, systems are very close according to accuracy while it can be re-

ally different with the others criterion. For some systems results are far from what was seen on the development set, theses differences come from the label distributions between the data set and rules learned from the training process.

Run Id	# Method	ACCURACY	RELIABILITY	SENSITIVITY	Correlation
Run 5	1	.6441	.4470	.2681	**.8799**
Baseline		*.5840*	*.3151*	*.2900*	*.8654*
Run 7	6	**.6477**	.4978	.1518	.8237
Run 8	7	.6467	.5125	.1393	.8203
Run 9	8	.6449	.5200	.1293	.8109
Run 1	9	.6152	.4779	.0824	.7778
Run 10	5	.5334	.5009	.0708	.7752
Run 2	10	.5942	.3410	.0802	.7698
Run 4	3	.5720	**.5509**	.0461	.7265
Median		*.5777*	*.4319*	*.2192*	*.7053*
Run 3	2	.5989	.3678	**.2709**	.6353

Table 4. Submitted runs to Polarity Task ordered with Pearson correlation.

Priority Detection task: performances ranked according to F-measure are reported in Table 5. Most of our runs are situated between the median and the baseline values. Method number 1 based on k-NN classification method has obtained a F-measure equal to 0.3351 comparing to 0.2965 reached by the baseline system. Several of our proposal have reached accuracy scores over the baseline but here again merging strategies did not provide better results than the best system.

Run Id	#Method	ACCURACY	RELIABILITY	SENSITIVITY	F-MEASURE
Run 5	1	.6275	.3873	**.3155**	**.3351**
Baseline		*.6007*	*.3049*	*.3029*	*.2965*
Run 6	4	.5858	.3156	.2761	.2820
Run 1	9	.6405	.3760	.2364	.2680
Run 4	2	.6167	.3168	.2552	.2657
Run 8	7	.6514	.4129	.2210	.2530
Run 7	6	.6470	**.4349**	.2181	.2513
Run 9	8	**.6527**	.4143	.2167	.2510
Median		*.5734*	*.3639*	*.2069*	*.2496*
Run 2	10	.5758	.3094	.1089	.1457
Run 3	3	.5424	.2421	.1284	.1367

Table 5. Submitted runs to Priority Task ordered according to the F-Measure.

Run Id	# Method	RELIABILITY	SENSITIVITY	F-MEASURE
Run 3	11	.2187	**.3468**	**.2463**
Run 2	11	.2342	.2730	.2435
Run 1	11	**.3841**	.1724	.2280
Run 4	11	.2538	.2222	.2267
Median		*.3659*	*.2180*	*.1954*
Baseline		*.1525*	*.2173*	*.1735*

Table 6. Submitted runs to Topic Detection Task ordered according to the F-Measure.

.

Topic Detection task: one system has been submitted for this task. Performances of runs produced around this method are reported and ranked in terms of F-measure in Table 6. We can see that all our proposal are greater that the median and the baseline scores with a F-measure equal to 0.2463 for our best system.

As reported in Table 7 runs 1 & 2 yield a better classification for the class "other topics". and runs 3 and 4 do not consider "other topics" labels. Nevertheless performances are better even if runs 3 & 4 consider a lower number of tweets. In a complementary experiment realized after the campaign we have added a rule consisting in removing this "other topic" tweets from runs 1 & 2. This rules improves the performances and F-measure reaches now 0.2972 (R=0.4648, S=0.2307) for run 1 and 0.2928 (R=0.2763, S=0.3296) for run 2.

Run 1	679	Run 2	648
419	"other topics"	335	"other topics"
40	"mention of a product"	53	"u2 favourite songs"
36	"u2 favourite songs"	46	"jokes"
30	"second hand selling / buying"	39	"u2 fans"
25	"4square"	37	"4square"
21	"secondhand cars"	36	"second hand selling / buying"
19	"nowplaying"	35	"mention of a product"

Run 3	264	Run 4	193
75	"u2 fans"	36	"nowplaying"
32	"nowplaying"	36	"u2 favourite songs"
31	"u2 favourite songs"	25	"4square"
21	"4square"	24	"u2 fans"
13	"secondhand cars"	19	"mention of a product"
8	"praise for volvo"	17	"secondhand cars"
7	"mtv"	8	"lyrics"

Table 7. Number of tweet topic well classified in each runs.

5 Conclusions and perspectives

In this paper we have presented the systems as well as the performances reached by the Computer Science Laboratory of Avignon (LIA) to RepLab 2013. We have presented a large variety of approaches and observed logically a large variety of system performances. We have also proposed several combinations of systems by using different merging strategies in order to benefit from the diversity of information considered by our runs. Our results are globally good and are mostly situated between the median and the baseline, but could still be improved by considering a subset of systems instead of handling system outputs with an equal weight. In other words, new merging strategies will have to be explored. However we did not paid enough attention to label distribution while building this development set. This lead us to introduce some noise in our models and to produce "over-training" rules. Using cross-validation strategies with chopping development would avoid these problems.

In a future work, we will propose some clustering strategies applied to labels co-occurrence and we will as well considered as a more important feature the users' influence sphere. Indeed, several tweet writer whose tweets are followed a large number of persons should be consider in a different manner than a user never read. Exploring how sentiment in web streams are affected by society and political events and their effects on topic and polarity trends, is also a very challenging question. Many situations may conduct to "swinging opinion states" for instance during a political campaign or depending of press coverage of an event.

References

1. Vapnik, V., *Pattern recognition using generalized portrait method*, in Automation and Remote Control, 24, pp 774-780, 1963
2. Boser B.E., Guyon I.M. and Vapnik V.N., *A training algorithm for optimal margin classifiers*, in 5th annual workshop on Computational Learning Theory, pp 144-152, 1992
3. Joachims T., *Transductive inference for text classification using support vector machines*, in international Machine learning conference, pp 200-209, 1999, Morgan Kaufmann Publishers, Inc.
4. Müller K., Smola A., Rätsch G., Schölkopf B., Kohlmorgen J. and Vapnik V., *Predicting time series with support vector machines*, in ICANN'97, pp 999-1004, 1997, Springer
5. Yuan, G-X, Ho, C-H and Lin, C-J,*Recent advances of large-scale linear classification*, in proceedinngs of the IEEE, 100, 9, pp 2584-2603, 2012, IEEE
6. Schapire, R. E., *The Boosting Approach to Machine Learning: An Overview*, in Workshop on Non-linear Estimation and Classification, 2002.
7. Schapire, R. E. and Singer,Yoram, *BoosTexter: A boosting-based system for text Categorization*, in Machine Learning, 39, 135-168, 2000
8. Favre, B. and Hakkani-Tür, D. and Cuendet, S., *Icsiboost: an opensource implementation of BoosTexter*, http://code.google.com/p/icsiboost, 2007

9. Robertson,S. *Understanding inverse document frequency: on theoretical arguments for IDF*, in Journal of Documentation, 60, 5, pp 503-520, 2004, Emerald Group Publishing Limited
10. Dong, T., Shang, W. and Zhu, H., *An Improved Algorithm of Bayesian Text Categorization*, in Journal of Software, 6, 9, pp 1837-1843, 2011
11. Bigot, B., Senay, G. , Linarès, G., Fredouille, C. and Dufour, R., *Person Name Recognition in ASR outputs using Continuous Context Models*, in Proceedings of ICASSP'2013, 2013
12. Salton, G. et Buckley, C., *Term weighting approaches in automatic text retrieval*, in Information Processing and Management 24, pp 513–523, 1988.
13. Torres-Moreno, J.-M., El-Beze, M., Bellot, P. and Bechet, F. *Opinion detection as a topic classification problem*, in Textual Information Access. Chapter 9, ISTE Ltd John Wiley and Son. 2012
14. Manning, C. D.and Schütze H., *Foundations of Statistical Natural Language Processing*, The MIT Press Cambridge, Massachusetts.
15. Torres-Moreno, J.-M., *Beyond Stemming and Lemmatization: Ultra-stemming to Improve Automatic Text Summarization* in C oRR, abs/1209.3126, 2012
16. Frank,J. R., Kleiman-Weiner, M., . Roberts, D. A, Niu, F., C. Zhang, C., Re,C. and Soboroff, I. *Building an Entity-Centric stream filtering test collection for TREC 2012*, in Proceedings of the Text REtrieval Conference (TREC), 2012.
17. Bonnefoy L., Bouvier V. and Bellot P., *A Weakly-Supervised Detection of Entity Central Documents in a Stream*, in SIGIR, 2013
18. Figueira, J., Greco, S. and Ehrgott, M., *Multiple Criteria Decision Analysis: State of the Art Surveys*, Springer Verlag, 2005