UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Feature selection and data sampling methods for learning reputation dimensions: The University of Amsterdam at RepLab 2014

Gârbacea, C.; Tsagkias, M.; de Rijke, M.

Publication date
2014

Document Version
Final published version

Published in
Working Notes for CLEF 2014 Conference

Link to publication

Citation for published version (APA):
Gârbacea, C., Tsagkias, M., & de Rijke, M. (2014). Feature selection and data sampling methods for learning reputation dimensions: The University of Amsterdam at RepLab 2014. In L. Cappellato, N. Ferro, M. Halvey, & W. Kraaij (Eds.), *Working Notes for CLEF 2014 Conference: Sheffield, UK, September 15-18, 2014* (pp. 1479-1490). (CEUR Workshop Proceedings; Vol. 1180). CEUR-WS. http://ceur-ws.org/Vol-1180/CLEF2014wn-Rep-GarbaceaEt2014.pdf

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:13 Oct 2026

Feature Selection and Data Sampling Methods for Learning Reputation Dimensions

The University of Amsterdam at RepLab 2014

Cristina Gârbacea, Manos Tsagkias, and Maarten de Rijke
{G.C.Garbacea, E.Tsagkias, deRijke}@uva.nl

University of Amsterdam, Amsterdam, The Netherlands

Abstract. We report on our participation in the *reputation dimension* task of the CLEF RepLab 2014 evaluation initiative, i.e., to classify social media updates into eight predefined categories. We address the task by using corpus-based methods to extract textual features from the labeled training data to train two classifiers in a supervised way. We explore three sampling strategies for selecting training examples, and probe their effect on classification performance. We find that all our submitted runs outperform the baseline, and that elaborate feature selection methods coupled with balanced datasets help improve classification accuracy.

1 Introduction

Today's growing popularity of social media requires the development of methods that can automatically monitor the reputation of real world entities in a social context. Even though reputation management is currently witnessing a shift from the traditional offline environment to an online setting, the algorithmic support for processing large amounts of user generated data created on a daily basis is still narrow and limited. For this reason, computational tools that can instantly extract and analyze the relevant content expressed online are in high demand.

In this paper we present our contribution to RepLab 2014 [3], an evaluation initiative promoted by the EU project LiMoSINe,[1] which focuses on monitoring the reputation of entities (companies, organizations, celebrities, universities) on Twitter. In previous years RepLab mainly addressed tasks like named entity disambiguation, reputation polarity, topic detection and topic ranking. This year, RepLab has introduced two new tasks: (i) *reputation dimensions* and (ii) *author profiling*. We describe each of them.

The *reputation dimensions task* aims at classifying tweets into eight reputation dimensions. These dimensions are defined according to the RepTrak framework,[2] which aims at facilitating reputation analysis. According to RepTrak, inside each dimension lie specific attributes that can be customized for clients in order to allow for program and message-ready analysis. An overview of these categories is presented in Table 1. For example, the tweet "We are sadly going to be loosing Sarah Smith from HSBC Bank,

[1] http://www.limosine-project.eu

[2] http://www.reputationinstitute.com/about-reputation-institute/the-reptrak-framework

Table 1: The eight reputation dimensions according to the Replab 2014 challenge.

Dimension	**Gloss**
Products & Services	Related to the products and services offered by the company and reflecting customers' satisfaction.
Innovation	The innovativeness displayed by the company, nurturing novel ideas and incorporating these ideas into products.
Workplace	Related to the employees' satisfaction and the company's ability to attract, form and keep talented and highly qualified people.
Governance	Capturing the relationship between the company and the public authorities.
Citizenship	The company's acknowledgement of community and environmental responsibility, including ethic aspects of the business: integrity, transparency and accountability.
Leadership	Related to the leading position of the company.
Performance	Focusing on long term business success and financial soundness.
Undefined	In case a tweet cannot be classified into none of the above dimensions, it is labelled as "Undefined."

as she has been successful in moving forward into a... http://fb.me/18FKDLQIr" belongs to the "Workplace" reputation dimension, while "HSBC to upgrade 10,000 POS terminals for contactless payments... http://bit.ly/K9h6 QW" is related to "Innovation."

The *author profiling task* aims at profiling Twitter users with respect to their domain of expertise and influence for identifying the most influential opinion makers in a particular domain of expertise. The task is further divided into two subtasks: (i) *author categorization*, and (ii) *author ranking*. The first subtask aims at the classification of Twitter profiles according to the type of author, i.e., journalist, professional, authority, activist, investor, company or celebrity. The second subtask aims at identifying user profiles with the biggest influence on a company's reputation.

We focus on the reputation dimensions task. Our main research question is how we can use machine learning to extract and select discriminative features that can help us learn to classify the reputation dimension of a tweet. In our approach we exploit corpus-based methods to extract textual features that we use for training a Support Vector Machine (SVM) and a Naive Bayes (NB) classifier in a supervised way. For training the classifiers we use the provided annotated tweets in the training set and explore three strategies for sampling training examples: (i) we use all training examples for all classes, (ii) we downsample classes to match the size of the smallest class, (iii) we oversample classes to match the size of the largest class. Our results show that our runs consistently outperform the baseline, and demonstrate that elaborate feature extraction and oversampling the training data peak classification accuracy at 0.6704.

The rest of paper is organized as follows. In Section 2 we present related work, in Section 3 we introduce our feature extraction approach, in Section 5 we describe our experimental setup, in Section 6 we report on our results. We follow up with an error analysis and reflections in Section 7 and conclude in Section 8.

2 Related Work

The field of *Online Reputation Management* (ORM) is concerned with the development of automatic ways for tracking online content that can impact the reputation of a company. This involves non-trivial aspects from natural language processing, opinion mining, sentiment analysis and topic detection. Generally, opinions expressed about individuals or organizations cannot be structured around a predefined set of features/aspects. Entities require complex modeling, which is a less understood process, and this turns ORM into a challenging field of research and study.

The RepLab campaigns address the task of detecting the reputation of entities on social media (Twitter). Each year there are new tasks defined by the organizers. Replab 2012 [4] focused on *profiling*, that is filtering the stream of tweets for detecting those microblog posts which are related to a company and their implications on the brand's image, and *monitoring*, i.e., topical clustering of tweets for identifying topics that harm a company's reputation and therefore, require the immediate attention of reputation management experts. Replab 2013 [2] built upon the previously defined tasks and proposed a full reputation monitoring system consisting of four individual tasks. First, the *filtering* task asked systems to detect which tweets are related to an organization by taking entity name disambiguation into account. Second, the *polarity detection for reputation classification* task, required systems to decide on whether the content of a social media update has positive, neutral or negative implications for the company's reputation. Third, the *topic detection* task aimed at grouping together tweets that are about the same topic. Four, the *priority assignment* task aimed at ranking the previous topics based on their potential for triggering a reputation alert.

Replab proposes an evaluation test bed made up of multilingual tweets in English and Spanish with human annotated data for a significant number of entities. The best systems from previous years addressed the majority of the above presented tasks as classification tasks by the use of conventional machine learning techniques, and focused on the extraction of features that encapsulate the main characteristics of a specific reputation related class. For the filtering and polarity detection tasks, Hangya and Farkas [9] reduce the size of the vocabulary following an elaborate sequence of data preprocessing steps and create an n-gram based supervised model, which was found previously successful on short messages like tweets [1]. Graph-based semantic approaches for assembling domain specific affective lexicon seem not to yield very accurate results given the inherent short and noisy content of social media updates [20]. The utility of topic modeling algorithms and unsupervised techniques based on clustering are explored in [7,5], both addressing the topic detection task. Peetz et al. [15] show how active learning can maximize performance for entity name disambiguation by systematically interacting with the user and updating the classification model.

The reputation dimensions task stems from the hypothesis that customer satisfaction is easier to measure and manage when we understand the key drivers of reputation that actively influence a company's success. The Reptrak system was designed to identify these drivers by evaluating how corporate reputation emerges from the emotional connection that an organization develops with its stakeholders. In this scenario, reputation is measured on a scale from 0–100 and considers the degree of admiration, trust, good feeling and overall esteem investors display about the organization. Reptrak defines

seven key aspects that define reputation and the reputation dimensions task uses them to define the reputation dimensions that we listed in Table 1 except the "Undefined" category, which is an extra class local to the reputation dimensions task.

In our study we follow [9], and in particular Gârbacea et al. [8], who presented a highly accurate system on the task of predicting reputation polarity. We build on their approaches but we focus on the task of reputation dimensions and we also explore the effect of balanced and unbalanced training sets on classification accuracy.

3 Feature Engineering

Classifying tweets by machine learning techniques imposes the need to represent each document as a set of features based on the presence, absence or frequency of terms occurring inside the text. Frequency distribution, tf.idf or χ^2 calculations are common approaches in this respect. In addition, identifying the semantic relations between features can capture the linguistic differences across corpora [21].

In our approach we consider textual features that we extract using corpus-based methods for frequency profiling. We build on the assumption that more elaborate feature extraction methods can help us identify discriminative features relevant for characterizing a reputation dimension class. We hypothesize that frequency profiling using the log-likelihood ratio method (LLR) [16], which is readily used to identify discriminative features between corpora, can also yield discriminative features specific to each of our reputation dimension classes. We extract unigrams and bigrams (the latter because they can better capture the context of a term) from our training data after having it split into eight annotated reputation dimensions, each corresponding to one of the given labels. Our procedure for extracting textual features is described in what follows.

Given two corpora we want to compare, a word frequency list is first produced for each corpus. Although here a comparison at word level is intended, part of speech (POS) or semantic tag frequency lists are also common. The log-likelihood statistic is performed by constructing a contingency table that captures the frequency of a term as compared to the frequency of other terms inside two distinct corpora. We build our first corpus out of all the annotated tweets for our target class and our second corpus out of all the tweets found in the rest of the reputation dimension classes. For example, for finding discriminative terms for the class "Products & Services," we compare pairs of corpora of the form: "Products & Services" vs. "Innovation" and "Workplace" and "Governance" and "Citizenship" and "Leadership" and "Performance" and "Undefined." We repeat this process for each of the eight classes and rank terms by their LLR score in descending order. We only keep terms that have higher frequency in the target class than for all the rest of the classes. This results in using as features only terms expected to be highly discriminative for our target class.

4 Strategies for Sampling Training Data

Machine learning methods are sensitive to the class distribution in the training set; this is a well described issue [22]. Some of RepLab's datasets, such as the one used for detecting reputation polarity, have different distributions among classes and between

training and test set. These differences can potentially impact the classification effectiveness of a system. To this extent, we are interested in finding out what the effect is of balancing the training set of a classifier on its classification accuracy. Below, we describe three strategies for sampling training data.

Unbalanced strategy. This strategy uses the original class distribution in the training data, and it uses all of the training data.

Downsampling. This strategy downsamples the training examples of each class to match the size of the smallest class. Training examples are removed at random. We evaluate the system using ten fold cross validation on the training data, and we repeat the process ten times. We select the model with the highest accuracy.

Oversampling. This strategy oversamples the training examples of each class to match the size of the largest class. For each class, training examples are selected at random and are duplicated. Similarly as before, we evaluate the system using ten fold cross validation on the training data, we repeat the process ten times, and we select the model with the highest accuracy.

5 Experimental Setup

We conduct classification experiments to assess the discriminative power of our features for detecting the reputation dimension of a tweet. We are particularly interested in knowing the effectiveness of our extracted textual LLR features for each of the eight reputation dimension classes, and the effect of the three sampling strategies for selecting training data.

We submitted a total of 5 supervised systems where we probe the usefulness of machine learning algorithms for the current task. We list our runs in Table 2. We train our classifiers regardless of any association with a given entity, since there are cases in the training data when not all classes are present for an entity (see Table 3). In UvA_RD_1 we choose to train an SVM classifier using all the available training tweets for each reputation dimension class, which implies that our classes are very unbalanced at this stage. In our next runs, UvA_RD_2 and UvA_RD_3, we randomly sample 214 tweets from each reputation dimension class and train a NB classifier, and respectively an SVM classifier. We also consider that using more training data can help our classifiers become more robust and better learn the distinguishing features of our classes. We explore a bootstrapping approach in runs UvA_RD_4 and UvA_RD_5, which train an NB classifier and an SVM classifier, respectively, using 7,738 tweets for each reputation dimension class. For under-represented classes we randomly sample from the labeled training data until we reach the defined threshold.

Dataset The Replab 2014 dataset is based on the Replab 2013 dataset and consists of automotive and banking related Twitter messages in English and Spanish, targeting a total number of 31 entities. Crawling the messages was performed in the period June - December 2012 using the entity's canonical name as query. For each entity, there are around 2,200 tweets collected: around 700 tweets at the beginning of the timeline used as training set, and approximately 1,500 tweets collected at a later stage reserved as

Table 2: Description of UvA's five runs for the reputation dimensions task at RepLab 2014 using either a Support Vector Machine (SVM) classifier or a Naive Bayes classifier (NB) and three strategies for sampling training data: all training data (All), downsampling (Down), and oversampling (Up).

Run	Classifier	Sampling
UvA_RD_1	SVM	All
UvA_RD_2	NB	Down
UvA_RD_3	SVM	Down
UvA_RD_4	NB	Up
UvA_RD_5	SVM	Up

test set. The corpus also comprises additional unlabeled background tweets for each entity (up to 50,000, with a large variability across entities). We make use of labeled tweets only and do not process messages for which the text content is not available or users profiles went private. The training set consists of 15,562 tweets. Out of these, we can access 15,294 (11,657 English, 3,637 Spanish) tweets. The test set consists of

Table 3: Distribution of training (top) and test (bottom) data per reputation dimension class (excluding empty tweets).

	Prod./Serv.	Innov.	Work	Gov.	Citizen.	Leader.	Perform.	Undef.
Training set								
Total	7,738	214	459	1,298	2,165	292	931	2,197
Average/entity	249	6	14	41	69	9	30	70
Maximum/entity	563	42	80	358	461	99	81	387
Minimum/entity	12	0	0	0	2	0	0	0
Test set								
Total	15,670	305	1,111	3,362	4,970	733	1,584	4,284
Average/entity	505	9	35	108	160	23	51	138
Maximum/entity	1,183	113	223	932	1,230	158	184	480
Minimum/entity	10	0	0	0	3	0	1	1

32,446 tweets, out of we which we make predictions for 32,019 (24,254 English, 7,765 Spanish) non-empty tweets. Table 3 summarizes our training and test datasets.

Preprocessing Normalization techniques help to reduce the large vocabulary size of the standard unigram model. Social media posts are known for the lack of language regularity, typically containing words in multiple forms, in upper and lower case, with character repetitions and misspellings. The presence of blogging annotations, abundance of hashtags, emoticons, URLs, and heavy punctuation can be interpreted as possible indicators of the rich meaning conveyed. We apply uniform lexical analysis to English

Table 4: Distribution of extracted textual features per reputation dimension class using log-likelihood ratio (LLR) on the training dataset.

	Prod./Serv.	Innov.	Work	Gov.	Citizen.	Leader.	Perform.	Undef.
LLR Unigrams	2,032	9	45	218	360	34	105	223
LLR Bigrams	1,012	24	50	151	109	22	133	143

and Spannish tweets. Our preprocessing steps are basic and aim to normalize text content: we lowercase the tweets, remove language specific stopwords and replace Twitter specific mentions @user, URLs and numbers with the *[USER]*, *[URL]* and *[NUMBER]* placeholder tags. We consider hashtags of interest since users generally supply them to categorize and increase the visibility of a tweet. For this reason we delete hashmarks and preserve the remaining token, i.e., *#BMW* is converted to *BMW*, so that Twitter specific words cannot be distinguished from other words. We reduce character repetition inside words to at most 3 characters to differentiate between the regular and the emphasized usage of a word. All unnecessary characters `["#$%&()?!*+,./:;<=>\^{}~]` are discarded. We apply Porter stemming algorithm to reduce inflectional forms of related words to a basic common form.

Feature selection We select our textual features by applying the LLR approach; see Table 4 for the distribution of features over reputation dimensions in the training set. We represent each feature as a boolean value based on whether or not it occurs inside the tweet content. There is a bias towards extracting more features from the "Products & Services" reputation dimension class, since the majority of tweets in the training data have this label. At the opposite end, the "Innovation" and "Leadership" classes are among the least represented in the training set, which explains their reduced presence inside our list of LLR extracted features.

Training We use supervised methods for text classification and choose to train an entity independent classifier. For our classifiers we consider Naive Bayes (NB) and a Support Vector Machines (SVM). We motivate our choice of classifiers based on their performance on text classification tasks that involve many word features [10,11,12,19]. We train them using the different scenarios described in Section 4: making use of all training data, balancing classes to account for the least represented class ("Innovation", 214 tweets) and bootstrapping to consider for the most represented class ("Products & Services", 7,738 tweets). We conduct our experiments using the natural language toolkit [6] and the scikit-learn framework [14]. We use NB with default nltk.classify settings; for SVM we choose a linear kernel.

Evaluation Replab 2014 allowed participants to send up to 5 runs per task. For the Reputation dimension task systems were asked to classify tweets into 7 reputation dimension classes (see Table 1); samples tagged as "Undefined" according to human assessors are not considered in the evaluation. Performance is measured in terms of accuracy (% of correctly annotated cases), and precision, recall and F-measure over each class are

Table 5: Official results for our runs for the reputation dimension task at RepLab 2014.

System	Accuracy	Classified tweets (%)
UvA_RD_1	0.6520	0.9112
UvA_RD_2	0.6468	0.9494
UvA_RD_3	0.6254	0.9445
UvA_RD_4	0.6704	0.9526
UvA_RD_5	0.6604	0.9566

reported for comparison purposes. In the evaluation of our system we also take into account the predictions made for the "Unknown" class. The predictions for this class were ignored in the official evaluation, and therefore the absolute numbers between the two evaluations do not match.

6 Results

We list the performance of our official runs in Tables 5 and 6. All our runs perform better than the baseline (0.6221). We highlight the fact that we make predictions for all 8 classes, including the "Undefined" category which is not considered in the official evaluation. We also decide to ignore empty tweets, even though these are taken into consideration by the official evaluation script!

Our most effective run is UvA_RD_4, where we train a NB classifier using a bootstrapping approach to balance our classes. It is followed closely by UvA_RD_5, which suggests that oversampling to balance classes towards the most representative class is a more sensible decision than using all training data or downsampling classes towards the least represented one. When we use all training data (UvA_RD_1) we provide the SVM classifier with more informative features than when dropping tweets (in runs UvA_RD_2, UvA_RD_3), which confirms the usefulness of our LLR extracted features. Balancing classes with only 214 tweets per class can still yield competitive results, which are rather close to our previous approaches, and a lot more accurate in the case of the SVM classifier. We notice that NB constantly outperforms SVM. NB's better accuracy might be due to independence assumptions it makes among features, which is in line with other research carried on text classification tasks where NB classifiers output other methods with very competitive accuracy scores [17,18,8].

Looking at the performance of our system per class, we find the following. The "Citizenship" and "Leadership" reputation dimension classes present high precision, followed by "Governance" and "Products & Services." Recall is very high for the latter class, which comes as no surprise given the large number of features we extract with this label that tend to bias the predictions of our classifier towards "Products & Services." The F1-measure is remarkably low for the "Innovation" class, since there are only few "Innovation" annotated tweets in the training set.

Detailed statistics of the number of tweets classified per reputation dimension class by our best system are presented in Table 7.

Table 6: System performance for the reputation dimension task using log-likelihood ratio features. We report on precision, recall and F1-score for each reputation dimension class, averaged over all entities.

Metric	Prod.&Serv.	Innovation	Workplace	Governance	Citizenship	Leadership	Performance
UvA_RD_1							
Precision	0.6134	0.1666	0.5901	0.5469	0.8176	0.7226	0.4043
Recall	0.9067	0.0130	0.1281	0.3192	0.4762	0.1155	0.1176
F1-score	0.7317	0.0241	0.2105	0.4031	0.6018	0.1991	0.1822
UvA_RD_2							
Precision	0.6317	0.2758	0.1110	0.4617	0.8228	0.7130	0.4120
Recall	0.8919	0.0261	0.2455	0.2612	0.5120	0.1102	0.1026
F1-score	0.7395	0.0476	0.1528	0.3336	0.6312	0.1908	0.1642
UvA_RD_3							
Precision	0.6678	0.0159	0.5232	0.4941	0.7965	0.5170	0.3460
Recall	0.8220	0.2026	0.2402	0.2871	0.5740	0.1223	0.1357
F1-score	0.7369	0.0294	0.3292	0.3631	0.6671	0.1978	0.1949
UvA_RD_4							
Precision	0.6041	0.2307	0.1300	0.6214	0.8553	0.7727	0.4660
Recall	0.9322	0.0098	0.1147	0.2698	0.5446	0.0913	0.0988
F1-score	0.7331	0.0188	0.1218	0.3762	0.6654	0.1633	0.1630
UvA_RD_5							
Precision	0.6144	0.0194	0.5253	0.6011	0.8015	0.7128	0.4018
Recall	0.9055	0.0947	0.0738	0.3360	0.5279	0.0967	0.1101
F1-score	0.7320	0.0322	0.1294	0.4310	0.6365	0.1702	0.1728

7 Analysis

In our analysis section, we perform a further experiment to assess how much including empty tweets in the evaluation and making predictions for the "Unknown" class influences results in terms of accuracy. We regenerated our top two best runs excluding the "Undefined" features and removing the 427 empty annotated tweets from the gold standard test file. We report on an almost 3% improvement in accuracy for run UvA_RD_4 (from 0.6704 to 0.6897) and a 2% increase in accuracy for run UvA_RD_5 (from 0.6604 to 0.6739).

On the one hand, we believe it is difficult to assess the performance of submitted systems and compare methods for the task of detecting Reputation Dimensions on Twitter data among RepLab participants since making predictions for only 7 reputation dimension classes outperforms systems that consider the "Undefined" category. We are not convinced that including empty tweets in the evaluation is a good idea and we were expecting the test corpus to be re-crawled beforehand so as to ignore non-relevant entries from the gold standard file.

Finally, our suggestion is that results could be more reliable and useful if the ratio of classified tweets would actually be considered when establishing a hierarchy of sub-

Table 7: Number of classified tweets for our best run, UvA_RD_4, per reputation dimension compared to the number of tweets in the gold standard.

Dimension	UvA_RD_4	Gold Standard
Products & Services	24,075	15,903
Innovation	13	306
Workplace	982	1,124
Governance	1,458	3,395
Citizenship	3,192	5,027
Leadership	87	744
Performance	332	1,598
Undefined	1,333	4,349

mitted runs. We were surprised to see systems with high accuracy scores ranking high up in the charts despite classifying fewer tweets than other runs with lower accuracy scores and more test set samples considered. It is well-known that accuracy is highly dependent upon the percentage of instances classified.

8 Conclusion

We have presented a corpus-based approach for inferring textual features from labeled training data in addressing the task of detecting reputation dimensions in tweets at CLEF RepLab 2014. Our results show that machine learning techniques can perform reasonably accurate on text classification if the text is well modeled using appropriate feature selection methods. Our unigram and bigram LLR features combined with an NB classifier trained on balanced data confirm steady increases in performance when the classification model is inferred from more example documents with known class labels. In future work we plan to use Wikipedia pages and incorporate entity linking methods for to improve the detection of concepts underlying the reputation dimensions inside tweets. We would also like to probe the utility of some other classifiers, like Random Forests, at an entity level, and consider tweets separately by language.

Acknowledgements

This research was partially supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 288024 (LiMoSINe) and nr 312827 (VOX-Pol), the Netherlands Organisation for Scientific Research (NWO) under project nrs 727.011.005, 612.001.116, HOR-11-10, 640.006.013, the Center for Creation, Content and Technology (CCCT), the QuaMerdes project funded by the CLARIN-nl program, the TROVe project funded by the CLARIAH program, the Dutch national program COMMIT, the ESF Research Network Program ELIAS, the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences (KNAW), the Netherlands eScience Center under project number 027.012.105, the Yahoo! Faculty Research and Engagement Program, the Microsoft Research PhD program, and the HPC Fund.

References

1. A. Agarwal, B. Xie, I. Vovsha, O. Rambow and R. Passonneau. Sentiment analysis of Twitter data. In *Proceedings of the Workshop on Language in Social Media*, 2011, 30–38.
2. E. Amigó, J. Carrillo-de-Albornoz, I. Chugur, A. Corujo, J. Gonzalo, T. Martin, E. Meij, M. de Rijke and D. Spina. Overview of Replab 2013: Evaluating online reputation monitoring systems. In *Information Access Evaluation. Multilinguality, Multimodality and Visualization*, volume 8138 of LNCS, pages 333–352, Springer, 2013.
3. E. Amigó, J. Carrillo-de-Albornoz, I. Chugur, A. Corujo, J. Gonzalo, E. Meij, M. de Rijke and D. Spina. Overview of RepLab 2014: author profiling and reputation dimensions for Online Reputation Management. In *Proceedings of the Fifth International Conference of the CLEF Initiative*, Sept. 2014, Sheffield, UK.
4. E. Amigó, A. Corujo, J. Gonzalo, E. Meij and M. de Rijke. Overview of Replab 2012: Evaluating online reputation management systems. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.
5. J. L. A. Berrocal, C. G. Figuerola and A. Z. Rodriguez. Reina at RepLab 2013 topic detection task: Community detection. In *CLEF 2013 Conference and Labs of the Evaluation Forum*, 2013.
6. S. Bird, E. Klein and E. Loper. *Natural Language Processing with Python*. O'Reilly Media Inc., Sebastopol, California, 2009.
7. A. Castellanos, J. Cigarran and A. Garcia-Serrano. Modelling techniques for Twitter contents: A step beyond classification based approaches. In *CLEF 2013 Conference and Labs of the Evaluation Forum*, 2013.
8. C. Gârbacea, M. Tsagkias and M. de Rijke. Detecting the reputation polarity of microblog posts. In *ECAI 2014: 21st European Conference on Artificial Intelligence*, 2014.
9. V. Hangya and R. Farkas. Filtering and polarity detection for reputation management on tweets. In *CLEF 2013 Conference and Labs of the Evaluation Forum*, 2013.
10. T. Joachims. Text categorization with Support Vector Machines: Learning with many relevant features. In *10th European Conference on Machine Learning*, pages 137–142, Springer Verlag, 1998.
11. T. Joachims. A statistical learning model of text classification for Support Vector Machines. In *24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 128–136, ACM, 2001.
12. A. McCallum and K. Nigam. A comparison of event models for Naive Bayes text classification. In *AAAI-98 Workshop on learning for text categorization*, pages 41–48, AAAI Press, 1998.
14. F. Pedregosa et al.. Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research*, 12, 2011, 2825–2830.
15. M. H. Peetz, D. Spina, J. Gonzalo and M. de Rijke. Towards and active learning system for company name disambiguation in microblog streams, In *CLEF 2013 Conference and Labs of the Evaluation Forum*, 2013.
16. P. Rayson and R. Garside. Comparing corpora using frequency profiling. In *Workshop on comparing corpora*, volume 9, pages 1–6, 2000
17. I. Rish. An empirical study of the Naive Bayes classifier. *International Joint Conference in Artificial Intelligence. Workshop on Empirical Methods in Artificial Intelligence*, Vol. 3, No. 22, 2001.
18. I. Rish, J. Hellerstein, J. Thathachar. An analysis of data characteristics that affect Naive Bayes performance. *International Conference on Machine Learning*. 2001.
19. K. M. Schneider. Techniques for improving the performance of Naive Bayes for text classification. In *CICLing*, pages 682-693, 2005.

20. D. Spina, J. C. de Albornoz, T. Martin, E. Amigo, J. Gonzalo and F. Giner. UNED online reputation monitoring team at RepLab 2013. In *CLEF 2013 Conference and Labs of the Evaluation Forum*, 2013.
21. C. Whitelaw and J. Patrick. Selecting systemic features for text classification. In *Australasian Language Technology Workshop*, 2004.
22. I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., 2005.