Towards the improvement of topic priority assignment using various topic detection methods for e-reputation monitoring on Twitter

J.-V. Cossu, B. Bigot, L. Bonnefoy, and G. Senay

LIA/Université d'Avignon et des Pays de Vaucluse
[**] 39 chemin des Meinajaries, Agroparc BP 91228, 84911 Avignon cedex 9, France
firstname.name@univ-avignon.fr

Abstract. Topic priority assignment is defined in *RepLab-2013* as labelling a topic according to its level of priority (ALERT, MILDLY IMPORTANT or UNIMPORTANT) in order to highlight topics requiring immediate attention for online reputation monitoring. Although they are strongly linked, topic detection and priority assignment have been previously treated as separate tasks. We study the impact of integrating topic detection outputs in the process of topic priority assignment.

1 Introduction

The amount and richness of the information collectively generated by users on online social networks have increased drastically during these last years. It is now well established that online social interactions often reflect in real-time the impact of real-world events on people opinions. Understanding social events is therefore crucial for persons and companies concerned with their online reputation. Companies typically spend a lot of money to get reliable satisfaction polls using call centers and surveys, and online social networks are certainly carrying key information to anticipate and react to the versatility of public opinions. Considering this amount of documents, automatic approaches are needed and have to deal with many sources of noise and perturbations. Noisy data mainly results from entity names ambiguities (e.g. jaguar: animal/car manufacturer), and an important number of linguistic variants and para-linguistic phenomena.

Replab 2013[1] provides a framework to evaluate Online Reputation Management systems on Twitter. The organizers have decomposed the monitoring issue into 4 subtasks: filtering, polarity classification, topic detection and priority assignment. In this paper, we are interested in 2 tasks: Topic Detection in which systems have to group together tweets related to one entity (a person, a company, etc.) by subject/event/conversation; and Priority Assignment consisting in ranking topics by priority (ALERT, MILDLY IMPORTANT and UNIMPORTANT). We will investigate the combination of these 2 subtasks in order to improve the quality of priority assignment for reputation monitoring.

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

[1] http://www.limosine-project.eu/events/replab2013

E. Métais, M. Roche, and M. Teisseire (Eds.): NLDB 2014, LNCS 8455, pp. 154–159, 2014.

2 Related work

Previous works on topic detection and characterization in tweet collections and streams aim at extracting messages requiring a attention from a user for instance by extracting new events [1], performing trend detection [2] or detecting late-breaking news [3] over the Twitter stream. To our knowledge, most of the contributions to reputation monitoring on Twitter have been proposed in the context 2012 and 2013 editions of Replab, with methods based on unsupervised clustering algorithms and supervised classification methods. Similarity between tweet content after a preprocessing consisting in a concept term expansion of filtered tweets words is used in [4]. Three topic detection approaches have been proposed in [5] and [6]: agglomerative clustering using term co-occurrences; agglomerative clustering using a wikified representation of tweet; and a Twitter-Latent Dirichlet Allocationused to discover latent topics in tweets. In [7], both supervised (Naive Bayes and Sequential Minimal Optimization Support Vector Machines) and unsupervised algorithms (K-star) combined with terms selection strategies are used. In [8], Social Network Analysis for tweets clustering is introduced.

Topic priority assignment for reputation monitoring in tweets is similar to topic characterization in Twitter [9]. Most of the contributions have been proposed in the context of Replab and mostly rely on supervised classification methods. In [5], authors use a tweet-level sentiment analysis classifier and exploit the link between priority and polarity values. In [7], three classifiers have been trained using features extracted from tweets content and meta-data.

3 Topic detection and priority assignment systems

To study the dependencies between the topic detection step and priority assignment, we first propose several systems based either on supervised classification methods or unsupervised clustering algorithms. We also use the **Replab2013 baseline** that consists in tagging the tweets of the test set with the label of the closest tweet (Jaccard word similarity) in the reference.

3.1 Topic Detection systems

The first method is a **K-means clustering using Jaccard similarity** [10] computed on the overall dataset (training and test tweets). The initial value of K is set to the number of clusters in the training set. As a preprocessing step we remove words appearing only once. The second method is a **Hierarchical clustering using Jaccard similarity** after the same preprocessing. The tree is cut according to the number of clusters in the training set. Our third system is based on a **Maximum *a posteriori* feature selection (MAP)**. This supervised method is based on [11]. Features are words, bigrams, distant bigrams (one gap) and tweet authors. It consists in selecting the most discriminant features for each topic using posterior probabilities of each term for a topic over the training dataset. Topic attribution is done by considering the maximum contribution of a tweet to a topic.

3.2 Priority Assignment systems

This first approach called the **KBA 2012 system** [12] has been proposed for the Knowledge Base Acceleration (KBA) task in TREC 2012 which is similar to RepLab priority assignment. The main difference lies in the kind of documents processed (web pages versus tweets). This method captures intrinsic characteristics of highly relevant documents using three types of features (document centric features, entitys profile features, and time features). We use two Random Forest classifiers (unimportant versus mildly important and important, them mildly important versus important). It matches a tweet in the test set with the k most similar tweets of the training set. Similarity is computed with Jaccard similarity on discriminant bag-of-words computed on tweet content and metadata (author, entity). k (equal to 6) has been fixed by cross-validation on the training set.

4 Relational Model, corpus and metrics

The corpus is a bilingual collection of tweets related to 61 entities from 4 domains: *Automotive*, *Banking*, *Universities* and *Music/Artists.* The tweets are labelled with 8 attributes: **tweet_id**, **author**, **entity**, **tweet_content**, **language**, **date**, **category** and **retweet**. The outputs are binary relations among tweet ids:

- **filter** $\subseteq$ **tweet_id** $\times$ **entity** $\cup$ {NULL} ,
- **opinion** $\subseteq$ **tweet_id** $\times$ {POSIVE, NEUTRAL, NEGATIVE},
- **priority** $\subseteq$ **tweet_id** $\times$ {NONE, MIDLY, ALERT},
- **topic** $\subseteq$ **tweet_id** $\times$ **tweet_id** is used to cluster the tweets by similarity.

The only defined functional dependency are **topic** → **entity** → **category** and **topic** → **filter** but **topic, opinion** → **priority** can also be assumed over more than 90% of records. The training set contains 34,496 tweets and the test set 70,412. Clearly, finding the appropriate **topic** relation is not a classification task but a clustering one since the training set contains 3,488 unique topics and the test set 5,343. However, record based NLP machine learning classification approaches appear to be efficient in providing a first approximation and additional attributes that can be further used in clustering.

Among the 3 priority levels, ALERT is the smallest with only 1,540 in the train set (3,161 tweets for test). We can find 17,954 MILDLY IMPORTANT (35,995 in the test) tweets and 31,256 tweets (15,378) are annotated as UNIMPORTANT. Note that most of the ALERT Tweets are related to *Banking.*

Metrics are Accuracy, Reliability (R), Sensitivity (S) and F-measure (based on R&S) [13]. Reliability and Sensitivity can be seen as precision and recall under the assumption that a test dataset can be seen as a bag of relationships (<, >, =) between the priority of test documents. Scoring is achieved by a comparison with the relations held in a gold standard. We have also computed classical F-measure (based on Precision and Recall) for each priority class.

5 Experiments

We consider the output of priority assignment and evaluate its improvement using additional information brought by topic detection. Performances of our topic detection systems, Replab2013's baseline and best system [6] are reported in Table 1. Our methods outperform the baseline and yield different values of Reliability (R) and Sensitivity (S). Two operating points have been set for the MAP (threshold on the number of words for the training) method in order to maximize either Sensitivity (MAP#1) or Reliability (MAP#2). The low performance of clustering methods is caused by significant differences of topics numbers in the both training and test set.

We now compare (cf. Table 2) the performances of priority assignment methods alone, and combined with the topic gold standard. In the first case, priority assignment based on KNN and KBA outperform the baseline and in the second case, adding the topic detection gold standard significantly improves topic priority. KNN now reaches an accuracy equal to 0.69 (+6 points comparing to KNN taken alone) and the best values of F-measure per class. F-measure (based on R&S) is also improved up to 0.387 instead of 0.335. This result proves good topic definitions do contain relevant information that improves priority assignment.

In the next experiment, we combine topic detection and priority assignment methods (cf. Table 3). Beyond the fact that results are lower than priority assignment system taken alone (F-measure=.335 for KNN cf. Table 2), it is

Table 1. Performances of topic detection systems

Method	Reliability	Sensitivity	F-Measure(R&S)
Replab baseline	.152	.217	.173
K-means clustering	.308	.157	.201
Hierarchic clustering	.261	.220	.227
MAP features selection #2	**.381**	.172	.238
MAP features selection #1	.193	**.497**	.266
Best@Replab2013	.462	.324	**.325**

Table 2. Priority assignment alone, with baseline and gold standard topics

	F-measure(Prec.&Rec.)						
Method	Alert	Mildly	Unimp.	Acc.	Rel.	Sens.	F-m(R&S)
Priority assignment only							
Baseline	.336	.643	.617	.530	**.403**	.248	.274
KNN	**.415**	**.684**	.646	**.627**	.387	**.315**	**.335**
KBA	.025	.560	**.705**	.585	.315	.276	.282
Priority assignment + gold standard topic detection							
Baseline	.441	.706	.703	.649	.511	.281	.326
KNN	**.514**	**.733**	.702	**.690**	**.549**	**.345**	**.387**
KBA	.002	.560	**.705**	.612	.532	.269	.329

very interesting to notice that except for the baselines combination, the performances respect **F-m(baseline)** < **F-m(MAP#1)** < **F-m(Hierarch.)** < **F-m(K-means)** < **F-m(MAP#2)**. F-measures of combined systems are ranked according to the values of topic detection method's Reliability (cf Table 1).

In one last experiment we study the impact of an automatic topic detection on perfect priority assignment by combining our topic-detection methods with the priority gold standard (cf. Tab. 4). We considered the priority gold standard as a system output and tried to propagate the majority priority label to the

Table 3. Performances of priority assignment combined with topic detection methods

	F-measure						
Method	Alert	Mildly	Unimp.	Acc.	Rel.	Sens.	F-m(R&S)
Priority assignment + baseline topic detection							
Baseline	.336	.643	.617	.530	.403	**.248**	**.274**
KNN	**.376**	**.672**	**.633**	**.550**	.520	.136	.172
KBA	0	.478	**.661**	.489	**.578**	.071	.098
MAP features selection #1							
Baseline	.342	.657	.659	.628	.383	**.151**	**.195**
KNN	**.378**	**.660**	.646	**.632**	.413	.136	.181
KBA	0	.466	**.672**	.568	**.551**	.098	.126
MAP features selection #2							
Baseline	.329	.643	.628	.574	**.406**	.214	.261
KNN	**.373**	**.669**	.636	**.619**	.405	**.249**	**.288**
KBA	.069	.512	**.657**	.561	.361	.171	.217
Hierarchical clustering using Jaccard similarity							
Baseline	**.342**	.642	.631	.584	.378	.174	.214
KNN	.340	**.659**	.631	**.613**	.391	**.195**	**.239**
KBA	.126	.515	**.662**	.567	**.421**	.150	.192
K-means using Jaccard similarity							
Baseline	.338	.635	.625	.570	.392	.206	.253
KNN	**.365**	**.667**	.628	**.612**	**.416**	**.223**	**.269**
KBA	.130	.514	**.661**	.559	.409	.164	.212

Table 4. Impact of topic detection methods using priority assignment gold standard

	F-measure						
Method	Alert	Mildly	Unimp.	Acc.	Rel.	Sens.	F-m(R&S)
Topic detection + Gold standard Priority							
MAP feat. select. #2	.710	**.840**	**.823**	**.812**	**.756**	**.518**	**.602**
Hierarch. clust.	.712	.785	.769	.783	.696	.438	.519
K-means clust.	**.769**	.815	.791	.761	.655	.367	.437
MAP feat. select. #1	.551	.754	.743	.731	.666	.229	.311
Baseline	.535	.763	.727	.634	.657	.198	.262

whole topic cluster. It's interesting to check how much our clusters can degrade the priority gold standard. Again we observe that the order of the ranked F-measures is highlighting that Reliability of topic detection seems to have an important effect on the performances of combined systems.

6 Conclusion

We have studied the impact of combining priority classification methods with the outputs of topic detection approaches for the task of topic priority assignment for online reputation monitoring in tweets. Experiments have shown the relevance of this proposition, but actual methods are not yet mature enough to reach better performances than any priority assignment system taken alone. Since such a pipeline approach propagate early stage errors to the later stage we have to study how to tackle this issue with alternative combination strategies or by the use of an unified topic framework which can assign topic and priority in one pass by taking into account both topic and priority predictions.

References

1. S. Petrovic, M. Osborne and V. Lavrenko *Streaming first story detection with application to Twitter* in HLT-NACCL, pp 181-189, ACL. 2010
2. M. Mathioudakis and N. Koudas *TwitterMonitor: trend detection over the twitter stream* in SIGMOD 2010, pp. 1155-1158, ACM. 2010
3. J. Sankaranarayanan, H. Samet, B. Teitler, M. Lieberman and J. Sperling *TwitterStand: news in tweets* in SIGSPATIAL-GIS, pp 42-51, ACM, 2009.
4. M. Atif Qureshi, C. O'Riordan and G. Pasi *Concept Term Expansion Approach for Monitoring Reputation of Companies on Twitter* in CLEF 2012.
5. T. Martin-Wanton, D. Spina, E. Amigo. and J. Gonzalo *UNED at RepLab 2012: Monitoring Task* in CLEF 2012.
6. D. Spina, J. Carrillo-de-Albornoz, T. Martin, E. Amigo, J. Gonzalo and F. Giner *UNED Online Reputation Monitoring Team at RepLab 2013* in CLEF 2013
7. C. Sanchez-Sanchez, H. Jimenez-Salazar and W. Luna-Ramirez *UAMCLyR at Replab2013: Monitoring Task* in CLEF 2013.
8. J.-L. Berrocal, C. Figuerola and A. Rodriguez *REINA at RepLab2013 Topic Detection Task.* in CLEF 2013.
9. M. Naaman, H. Becker and L. Gravano *Hip and Trendy: Characterizing Emerging Trends on Twitter* in Journal of the American Society for Information Science and Technology, 62:5, 2007
10. F. Leisch *A toolbox for k-centroids cluster analysis* in Computational Statistics and Data Analysis, 2006
11. T. Hazen, F. Richardson and A. Margolis *Topic identification from audio recordings using word and phone recognition lattices* in ASRU, pp 659-664, IEEE, 2007.
12. L. Bonnefoy,V. Bouvier and P. Bellot *A Weakly-Supervised Detection of Entity Central Documents in a Stream*, in SIGIR, 2013
13. E. Amigo J. Gonzalo and F. Verdejo *A general evaluation measure for document organization tasks* in SIGIR, ACM pp. 643-652, 2013