UNED at RepLab 2012: Monitoring Task

Tamara Martín, Damiano Spina, Enrique Amigó, and Julio Gonzalo

UNED NLP & IR Group
Juan del Rosal, 16
28040 Madrid, Spain
{tmartin,damiano,enrique,julio}@lsi.uned.es
http://nlp.uned.es

Abstract. This paper describes the UNED participation at RepLab 2012 Monitoring Task. Given an entity and a tweet stream containing the entity's name, the task consists on grouping the tweets in topics and then ranking the identified topics by priority. We tested three different systems to deal with the clustering problem: (i) an agglomerative clustering based on term co-occurrences, (ii) a clustering method that considers 'wikified" tweets, where each tweet is represented with a set of Wikipedia entries that are semantically related to it and (iii) Twitter-LDA, a topic modeling approach that extends LDA considering some of the intrinsic properties of Twitter data. For the ranking problem, we rely on the insight that the priority of a topic depends on the sentiment expressed in the subjective tweets that refer to it.
Although none of the proposed systems outperforms the official baseline in average, our systems obtain reasonable high precision results, (i.e. high Reliability scores). The average sentiment of a topic seems to be an useful indicator of priority, that merits further study. Finally, topics with high ratio of unrelated tweets are difficult to group correctly, suggesting the need of an explicit treatment of ambiguity.

1 Introduction

The enormous popularity of Social Media in the Web, such as blogs, forums, or real-time social networking's services offer a place for sharing information as it happens and for connecting with others in real time, often spreading a wealth of latest news about real-world events and topics dominating social discussions. This phenomenon has generated the opportunity - and the necessity of managing the online reputation of entities such as companies, brands and public figures. Online Reputation Management consists of monitoring and handling the opinion of Web users (also referred to as electronic word of mouth, eWOM) on people, companies or products [6].

Online reputation managers spend remarkable effort on continuously monitoring social streams such as Twitter[1] in order to early identifying the topics that may alter (either negatively or positively) the reputation of an entity of

[1] http://twitter.com

interest. The RepLab 2012 Monitoring Task [2] directly tackles this problem. Systems receive a stream of tweets containing the name of an entity, and their goal is to (i) cluster the most recent tweets in topics, and (ii) assign relative priorities to the cluster.[2]

In this paper, we present the results obtained from the systems proposed by UNED at the participation to the RepLab 2012 Monitoring Task. We tested three different approaches to deal with the clustering problem: (i) an agglomerative clustering based on term co-occurrences, (ii) a clustering method that considers 'wikified" tweets, where each tweet is represented with a set of Wikipedia entries that are semantically related to it and (iii) Twitter-LDA, a topic modeling approach that extends LDA considering some of the intrinsic properties of Twitter data. For the problem of the priority of a topic, we rely on the insight that the priority of a topic depends on the sentiment expressed in the subjective tweets that refer to it.

This paper is organized as follows. Section 2 describes the proposed systems. Section 3 gives details about the experiments and the obtained results. Finally, conclusions are presented in section 4.

2 Proposed Systems

We tested three different approaches to tackle the clustering problem in the monitoring task: (i) a two-step algorithm based on agglomerative clustering, that firstly it groups terms considering pair of co-occurrent terms in the tweets and then it assigns tweets to identified terms clusters, (ii) an agglomerative clustering of "wikified" tweets, where each tweet is represented with a set of Wikipedia entries that are semantically related to it and (iii) Twitter-LDA, a topic modeling approach that extends LDA considering some of the intrinsic properties of Twitter data. We also tested a method that relies to the polarity of tweets to deal with the priority problem.

2.1 Agglomerative Clustering Based on Term Co-occurrences

Let us assume that each topic discussed about an entity can be represented with a set of terms, that allow to the expert understand what the topic is about. Considering this, we define a two-step algorithm that tries to (i) identify the terminology of each topic, clustering the terms occurring in the input entity stream of tweets and (ii) assigning tweets to the identified clusters.

In the first step, we use Hierarchical Agglomerative Clustering (HAC) to build the clustering of terms. Obviously, not all the terms occurring in the tweets that we want to group belong to the terminology of the topics. For instance, stopwords and common terms across the topics are not representative of none of the topics. Since these terms are difficult to know a priori, we built a binary classifier that,

[2] Please refer to the RepLab Monitoring Task overview's paper [2] for detailed information about the task and the dataset.

given a pair of co-occurrent terms, it guesses whether both terms belong to the terminology of a same topic or not.

We use different families of features to represent the co-occurrent pair. We consider both the "labeled collection" and the "background collection" to compute the features. Besides the content of the tweets, we also use the meta-data such as the date of creation and the author. Finally, we apply regular expressions to extract named users (i.e. user), hashtags (e.g #apple) and URLs (e.g. http://www.google.com). Short URLs have been translated to long URLs using the conversion tables provided by the organizers. We define the following set of features:

- **Term features:** Features that describe each of the terms of the co-occurrence pair. These are: term occurrence, normalized frequency, TF.IDF and KL-Divergence –considering term frequency as the frequency of the term in a pseudo-document built from entity-specific tweets like in [12]. These features were computed in two ways: (i) considering only tweets on the labeled corpus, and (ii) considering tweets in both the labeled and background corpus. Features based on the tweets meta-data where each term occurs are: Shannon's entropy of named users, URLs, hashtags and authors in the tweets where the term occurs.

- **Content-based pair term features:** Features that consider both terms of the co-occurrence pair, such as Levenshtein's distance between terms, normalized frequency of co-occurrences, Jaccard similarity between occurrences of each of the terms.
- **Meta-data-based pair term features:** Jaccard similarity and Shannon's entropy of named users, URLs, hashtags and authors between tweets where both terms co-occur.
- **Time-aware features:** Features based on the date of the creation of the tweets where the terms co-occurs. Features computed are median, minimum, maximum, mean, standard deviation, Shannon's entropy and Jaccard similarity. These features were computed considering four different time intervals: milliseconds, minutes, hours and days.

In our classification model, each instance corresponds to a pair of co-occurrent terms $\langle t, t' \rangle$ in the entity stream of tweets. In order to learn the model, we extract training instances from the trial dataset, considering the following labeling function:

$$label(\langle t, t' \rangle) = \begin{cases} \texttt{clean} \text{ if } & \max_j Precision(C_{t \cap t'}, L_j) > 0.9 \\ \texttt{noisy} \text{ in other case} & \end{cases}$$

where $C_{t \cap t'}$ is the set of tweets where terms t and t' co-occurs and L is the set of topics in the goldstandard, and

$$Precision(C_i, L_j) = \frac{|C_i \cap L_j|}{|C_i|}$$

Then, term pairs that co-occur in tweets where 90% were annotated with the same topic in the goldstandard, are considered as clean pairs. If the precision is below this threshold are then labeled as noisy pairs.

After training a binary classifier, we use the confidence of belonging to the "clean" class to build a similarity matrix between terms. A Hierarchical Agglomerative Clustering is then applied to cluster the terms, using the previously built similarity matrix. After building the agglomerative clustering, a cut-off threshold is used to return the final term clustering solution.

The second step of this algorithm consists on assigning tweets to the identified term clusters. This is carried out following a straightforward majority voting strategy: for each tweet, the final assigned cluster is the one that maximizes the number of terms in the tweet assigned to it.

2.2 Clustering Wikified Tweets

The second system we tested relies on the hypothesis that tweets sharing concepts defined in a knowledge base –such as Wikipedia– are more likely to belong to the same cluster than tweets with none or less concepts in common. In this approach, each tweet is linked to a set of Wikipedia entries that semantically represents the concepts that are related to it. We use the COMMONNESS probability presented in [9] to identify the relevant concepts to a given tweet. Based on the intra-Wikipedia hyperlinks, it computes the probability of a concept c been the target of a link with anchor text q in Wikipedia:

$$\text{COMMONNESS}(c, q) = \frac{|L_{q,c}|}{\sum_{c'} |L_{q,c'}|}$$

where $L_{q,c}$ denotes the set of all links with anchor text q and target c.

We used the web service described in [8] to compute the COMMONNESS probability. Given a tweet, the web service extract all possible term n-grams. Each n-gram is then matched to articles in Wikipedia, returning the COMMONNESS probability of each linked Wikipedia article. Then we filter out n-grams that are contained in longer n-grams. The Wikipedia article with highest COMMONNESS probability of each of the longest n-grams are then considered to the "wikified" representation of the tweet. Depending on the language of the tweet, we use either the English or the Spanish Wikipedia. Spanish Wikipedia articles are then translated to the corresponding English Wikipedia article by following the interlingual links, using the Wikimedia API[3].

Table 1 shows some tweets that have been "wikified" using the method described above.

After wikifying the tweets, the clustering is done by grouping together those tweets having more than 40% of the identified concepts in common.

[3] http://www.mediawiki.org/wiki/API:Properties

Tweet	Wikified representation
Les presento el nuevo producto de la marca Apple...El iMeestabajando. pic.twitter.com/JPdR5Oct	Brand, Product (business), Apple Inc.
Apple ya ha comenzado con iOS 6 en el iPad 3 !!!! http://goo.gl/fb/aYOcO #rumor #ios6 #ipad #ios #ipad3g #apple	IOS, Rumor, Apple Inc., IPad
Server logs show Apple testing iPads with iOS 6, possible Retina Displays http://bit.ly/ysaFUA (via @appleinsider)	Software testing, Retina, IOS, Display device, Apple Inc., IPad

Table 1. Examples of tweets represented with the Wikipedia entries identified using the COMMONNESS probability.

2.3 Identifying Trivial Clusters

Retweets and automatic generated tweets (by clicking "share" buttons in news or blog posts, tweets generated by third services like Foursqurare, etc.) are frequent in the trial data.

Moreover, tweets sharing a high percentage of words are very likely to belong to the same cluster. In both co-occurrence graph-based and commoness-based systems, tweets with a term overlap higher than 70% are grouped a-priori. These tweets are then removed from the input, except of one representative tweet for each of the trivial clusters. After running the system, we merge the output with the a-priori trivial clustering. Finally, each cluster is joined with the cluster in the system output that contains the representative tweet of the trivial cluster.

2.4 Twitter-LDA Approach

Twitter-LDA is a variant of LDA proposed by Zhao et al. [13] that is adapted to the characteristics of Twitter: tweets are short (140-character limit) and a single tweet tends to be about a single topic. Like Latent Dirichlet Allocation [4], it is an unsupervised machine learning technique which discovers the latent topics distributed across the documents of a given corpora.

The model is based on the following assumptions. There is a set of topics T in Twitter, each represented by a word distribution. Each user has her topic interests modeled by a distribution over the topics. When a user wants to write a tweet, first chooses a topic based on his topic distribution. Then chooses a bag of words one by one based on the chosen topic. However, not all words in a tweet are closely related to the topic of that tweet; some are background words commonly used in tweets on different topics. Therefore, for each word in a tweet, the user first decides whether it is a background word or a topic word and then chooses the word from its respective word distribution.

The generation process of tweets is described in Figure 1, where: ϕ^t denote the word distribution for topic t; $t^{\mathcal{B}}$ the word distribution for background words; θ^u denote the topic distribution of user u and π a Bernoulli distribution that governs the choice between background words and topic words.

1. Draw $\phi^{\mathcal{B}} \sim \text{Dir}(\beta), \pi \sim \text{Dir}(\gamma)$
2. For each topic $t \in \mathcal{T}$,
   (a) draw $\phi^t \sim \text{Dir}(\beta)$
3. For each user $u \in \mathcal{U}$,
   (a) draw $\theta^u \sim \text{Dir}(\alpha)$
   (b) for each tweet $d_{u,m}$
      i. draw $z_{u,m} \sim \text{Multi}(\theta^u)$
      ii. for each word $w_{u,m,n}$
         A. draw $y_{u,m,n} \sim \text{Bernoulli}(\pi)$
         B. draw $w_{u,m,n} \sim \text{Multi}(\phi^{\mathcal{B}})$ if $y_{u,m,n} = 0$ and $w_{u,m,n} \sim \text{Multi}(\phi^{z_{u,m}})$ if $y_{u,m,n} = 1$

Fig. 1. The generation process of tweets.

Because each test case have few tweets to be annotated, we have to consider two sets of background. The first one is the background of the entity, consisting of 5000 tweets that refer to the entity that will provide additional information to cluster tweets that has the same topic. And the second set of 15000 tweets of a different entity will allow the model to differentiate between topics that do not refer to the entity.

2.5 Sentiment-based Priority Approach

For the priority of each topic we use a tweet-level sentiment analysis classifier [5]. The main idea of this method is to extract the WordNet concepts in a sentence that entail an emotional meaning, assign them an emotion within a set of categories from an affective lexicon, and use this information as the input to a machine learning algorithm. The strengths of this approach, in contrast to other more simple strategies, are: (1) use of WordNet and a word sense disambiguation algorithm, which allows the system to work with concepts rather than terms, (2) use of emotions instead of terms as classification attributes, and (3) processing of negations and intensifiers to invert, increase or decrease the intensity of the expressed emotions.

Given the polarity of each tweet we estimate the priority of a topic in the following manner:. Let T_i be a topic, N_{T_i} number of tweets in the topic T_i, we define three function: $Pos(T_i)$, $Neg(T_i)$, and $Neu(T_i)$ as the number of positive, negative and neutrals tweets of that topic, respectively. The priority of a topic can be defined as:

$$Priority(T_i) = \begin{cases} 3 \text{ if } & Neg(T_i) = N_{T_i} \quad or \quad Pos(T_i) = N_{T_i} \\ 2 \text{ if } & Neg(T_i) \geq Pos(T_i) \quad and \quad Neg(T_i) \geq Neu(T_i) \\ 2 \text{ if } & Pos(T_i) > Neg(T_i) \quad and \quad Pos(T_i) \geq Neu(T_i) \\ 2 \text{ if } & Pos(T_i) + Neg(T_i) \geq Neu(T_i) \\ 1 \text{ if } & Neu(T_i) = N_{T_i} \\ 1 \text{ if } & Neu(T_i) > Pos(T_i) + Neg(T_i) \\ 0 & \text{in other case} \end{cases}$$

3 Experiments and Results

In this section we describe the parameters used on each of the submitted systems and the obtained results. We report the scores obtained for the official metrics used to evaluate the monitoring task: Reliability & Sensitivity [3][4].

We submitted three runs in total:

- **wikified tweet clustering:** This run combines the wikified tweet clustering approach described in section 2.2 with the trivial clustering identification method described in section 2.3. This system corresponds to the `replab2012_monitoring_UNED_1` run.
- **co-occurrence clustering:** This run combines the agglomerative clustering based on term co-occurrences described in section 2.1 with the trivial clustering identification method described in section 2.3. This system corresponds to the `replab2012_monitoring_UNED_2` run.
- **Twitter-LDA:** This run uses Twitter-LDA, described in section 2.4 to identify the clusters and uses the sentiment-based priority approach described in section 2.5 to rank the clusters. This system corresponds to the `replab2012_monitoring_UNED_3` run.

In all runs, tweets were lowercased, tokenized using a Twitter tokenizer [11] and punctuation was removed.

The second run uses a Naïve Bayes classifier to learn the clean/noisy pair-terms classifier. We have experimented with several machine learning methods using Rapidminer[10]: Multilayer Perceptron with Backpropagation (Neural Net), C4.5 and CART Decision Trees, Linear Support Vector Machines (SVM), and Nave Bayes. We used a "leave-one-entity-out" strategy to evaluate the performance of the models on the trial data. On each fold, all the pair terms related to an entity are used as test data. All the term pairs related to the other entities are used as training data. This process is repeated 6 times (as many as entities in the trial corpus) and AUC is computed to evaluate the classifiers. Naïve Bayes significantly outperforms the other tested models, obtaining AUC values above 0.8 in all trial entities except of one, `Alcatel-Lucent` (entity id `RL2012E02`).

[4] In the context of clustering tasks, Reliability & Sensitivity are equivalent to BCubed Precision and BCubed Recall, respectively [1].

The Hierarchical Agglomerative Clustering was performed using average linkage (i.e. considering the mean similarity between elements of each cluster) and using the S-Space package implementation [7]. The cut-off threshold of the HAC was empirically established to 0.9999 after running some experiments over the trial data.

We ran Twitter-LDA with 500 iterations of Gibbs sampling. After trying a few different numbers of topics, we empirically set the number of topics to 100. We set α to $50.0/|T|$, β to a smaller value of 0.01 and λ to 20 as [13] suggested.

We also tried the standard LDA model (i.e. treating each tweet as a single document) and found that the Twitter-LDA model was better. In addition, Twitter-LDA is much more convenient in supporting the computation of tweet-level statistics (e.g. the number of co-occurrences of two words in a specific topic) than the standard LDA because Twitter-LDA assumes a single topic assignment for an entire tweet.

The official baseline consists of an agglomerative clustering algorithm that uses single linkage over Jaccard word distances. Different stopping threshold were used. Here we reports the results of considering 0%, 50% and 100% as stopping thresholds. For priority relations, the baseline assigns all non-single clusters to the same level, and single clusters are assigned to a secondary level.

Table 2 shows the results of the baseline and the proposed systems when considering clustering relationships.

System	Reliability (R)	Sensitivity (S)	$F(R,S)$
baseline_0%	0.4	**1**	**0.5**
Twitter-LDA	0.72	0.32	0.4
co-occurrence clustering	0.85	0.34	0.39
baseline_50%	0.97	0.22	0.33
wikified tweet clustering	0.9	0.2	0.3
baseline_100%	**0.98**	0.17	0.26

Table 2. Performance of the proposed systems only considering clustering relationships.

With regards to F-Measure, baseline_0% obtains the highest score. The baseline with 0% stopping threshold assigns all the tweets to a single cluster, corresponding to the so-called *all-in-one* system. This system reaches perfect recall, and in precision is relatively high on entities with few topics on the set of tweets. More precisely, this system achieve Reliability score above 0.95 in five of the 24 test cases (slightly more than 20%).

Although the clustering based on co-occurrences outperforms Twitter-LDA in R and S, the latter obtains 0.01 higher F-1 score, suggesting that Twitter-LDA is more R/S balanced across test cases than co-occurrence clustering.

Remarkably, in some test cases where most of the tweets are not related to the entity of interest such as `Indra` (`RL2012E12`), `ING` (`RL2012E15`) or `BP` (`RL2012E27`), all of the proposed systems obtain F-1 scores below 0.25. This

suggests that an explicit treatment of ambiguity is needed, at least when the entity's name may refers to multiple entities or concepts (e.g. acronyms).

Table 3 shows the results obtained by the proposed systems considering only priority relationships.

System	R	S	$F(R,S)$
baseline_50%	**0.31**	**0.31**	**0.27**
Twitter-LDA	0.25	0.3	0.26
baseline_100%	0.28	0.27	0.24
baseline_0%	0	0	0
co-occurrence clustering	0	0	0
wikified tweet clustering	0	0	0

Table 3. Performance of the proposed systems only considering priority relationships.

Note that only the run that uses Twitter-LDA incorporates the sentiment-based priority approach. The runs using Co-occurrence clustering and wikified tweet clustering return all clusters with the same priority. These systems are considered as non-informative by the used evaluation measures, obtaining the minimum score in both R and S. As baseline_0% group all tweets in one cluster, no single clusters are returned and then R and S is also 0. The baseline using a stopping threshold of 50% obtains the highest scores for all the reported metrics. However, the sentiment-based priority approach obtains competitive results, suggesting that the overall sentiment of the topic is a helpful variable to assign relative priorities.

Finally, Table 4 shows the performance of the proposed systems in the RepLab monitoring task, considering both clustering and priority relationships.

System	R	S	$F(R,S)$
baseline_0%	0.4	**0.43**	**0.41**
Twitter-LDA	0.32	0.26	0.29
baseline_50%	0.35	0.21	0.26
baseline_100%	0.3	0.16	0.2
co-occurrence clustering	0.85	0.09	0.14
wikified tweet clustering	**0.9**	0.05	0.1

Table 4. Performance of the proposed systems for the monitoring task, considering both clustering and priority relationships.

No considering priority relationships significantly drops Sensitivity scores. Note that in the case of baseline_0%, S decreases from 1 to 0.43. The co-occurrence clustering and the wikified tweet clustering goes below 0.1. As regards Twitter-LDA, Reliability and Sensitivity remains relatively close to the scores achieved by the baselines.

4 Discussion and conclusions

In this paper we have described the systems used in the runs submitted by UNED to the Monitoring Task of the RepLab 2012 evaluation campaign. Here, systems receive a stream of tweets containing the name of an entity, and their goal is to (i) cluster the most recent tweets in topics, and (ii) assign relative priorities to the cluster. We tested different clustering approaches, and a sentiment-based algorithm to predict the priority of the identified topics.

Results show the high difficulty of the monitoring task. In the case of the clustering problem, simple models such as the agglomerative clustering baseline are difficult to be outperformed by more elaborated systems. However, our proposed systems achieves reasonable high BCubed precision scores, suggesting that more information is needed in the representation of the tweets in order to solve joining gaps. With regarding of the priority problem, the sentiment expressed in tweets of a same cluster seems to be an useful indicator of the topic priority. However, there is still much room for improvement in this direction.

As future work we intend to take advantage of Twitter metadata to add new variables to the LDA model. As regards to co-occurence clustering, we plan to include distributional semantics in the co-occurrence similarity features. Finally, future work also includes incorporating a company name disambiguation component to our systems.

Acknowledgements. This research was partially supported by the Spanish Ministry of Education (FPU grant nr AP2009-0507), the Spanish Ministry of Science and Innovation (Holopedia Project, TIN2010-21128-C02), the Regional Government of Madrid and the ESF under MA2VICMR (S2009/TIC-1542) and the European Community's FP7 Programme under grant agreement nr 288024 (LiMoSINe).

References

1. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval 12(4), 461–486 (2009)
2. Amigó, E., Corujo, A., Gonzalo, J., Meij, E., de Rijke, M.: Overview of RepLab 2012: Evaluating Online Reputation Management Systems. In: CLEF 2012 Labs and Workshop Notebook Papers (2012)
3. Amigó, E., Gonzalo, J., Verdejo, F.: Reliability and Sensitivity: Generic Evaluation Measures for Document Organization Tasks. Tech. rep., UNED (2012)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (Mar 2003)
5. Carrillo de Albornoz, J., Chugur, I., Amigó, E.: Using an emotion-based model and sentiment analysis techniques to classify polarity for reputation. In: CLEF 2012 Labs and Workshop Notebook Papers (2012)
6. Jansen, B., Zhang, M., Sobel, K., Chowdury, A.: Twitter power: Tweets as electronic word of mouth. Journal of the American society for information science and technology 60(11), 2169–2188 (2009)

7. Jurgens, D., Stevens, K.: The s-space package: an open source package for word space models. In: Proceedings of the ACL 2010 System Demonstrations. pp. 30–35 (2010)
8. Meij, E.: LiMoSINe Deliverable 4.1: Initial Semantic Mining Module. Tech. rep., Univerity of Amsterdam (2012)
9. Meij, E., Weerkamp, W., de Rijke, M.: Adding semantics to microblog posts. In: Proceedings of the fifth ACM international conference on Web search and data mining (2012)
10. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: Yale: Rapid prototyping for complex data mining tasks. In: KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 935–940 (2006)
11. OConnor, B., Krieger, M., Ahn, D.: Tweetmotif: Exploratory search and topic summarization for twitter. Proceedings of ICWSM pp. 2–3 (2010)
12. Spina, D., Meij, E., de Rijke, M., Oghina, A., Bui, M., Breuss, M.: Identifying entity aspects in microblog posts. In: SIGIR '12: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval (2012)
13. Zhao, W.X., Jiang, J., Weng, J., He, J., Lim, E.P., Yan, H., Li, X.: Comparing twitter and traditional media using topic models. In: Proceedings of the 33rd European conference on Advances in information retrieval. pp. 338–349. ECIR'11, Springer-Verlag, Berlin, Heidelberg (2011)