UAMCLyR at Replab2013: Monitoring Task *

Notebook for RepLab at CLEF 2013

Christian Sánchez-Sánchez, Héctor Jiménez-Salazar,
Wulfrano Arturo Luna-Ramírez

Departamento de Tecnologías de la Información
Universidad Autónoma Metropolitana, Unidad Cuajimalpa,
Vasco de Quiroga 4871 Col. Santa Fe, México D.F.
{csanchez,wluna,hjimenez}@correo.cua.uam.mx

Abstract. In this article we deal with the Topic Detection and Priority Detection subtasks from RepLab 2013, trying clustering and classification methods as well as term selection techniques in order to know its performance in two sub collections of tweets: single and extended (single tweet plus derived tweets). Our tests show good performance in spite of we used very few resources.

Keywords: Tweet Clustering, Tweet Classification, Term Selection Techniques

1 Introduction

Twitter has become a very popular social interaction place where the users give their opinions about the companies and their products via tweets. Because of that Twitter has become a significant repository of opinions on companies, brands, and persons, such entities have interest to protect their reputation and try to deal with non founded gossips that can affect their image and incomes. RepLab 2013 [12] faces some research challenges on Twitter, one of these is the Monitoring task which consists on:

- clustering tweets based on their attributes
- ordering tweets by priority for each entity

Performing such a monitoring of tweets is a significantly challenging task given that the tweet messages are very short and noisy. Some authors try to face these problems through the idea of concept term expansion in tweets performing one or more clustering phases of and priority level, as well as unsupervised clustering techniques. Additionally, some authors use supervision for priority level assessment [1]. The main motivation of our experiments was:

- To evaluate how much previous topics and terminology (supervised approach) can help to identify new topics.

* This work was partially supported by CONACyT México Project Grant CB-2010/153315, and SEP-PROMEP Project Grant UAM-C-CA-31/10847.

- To estimate how much derived conversations from one tweet can ease to detect topics.
- To determine which kind of terms can improve Topic Detection.
- To evaluate how much features extracted from tweets' metadata are useful to determine priority of tweets.

In this paper we explored clustering and classification methods as well as term selection techniques in order to know its performance over two sub collections of tweets: single and extended (single tweet plus derived tweets). Our tests show good performance in spite of we used very few resources. In the following section we describe the data and its preprocessing. Section 3 outlines the methods applied to data collection for Topic Detection subtask. At Section 4 we detail attributes and our approach followed for tweet Priority Detection subtask. The results are presented at Section 5, and finally, our conclusions are depicted in Section 6.

2 Data description and preprocessing

The corpus of tweets that was used in the experiments was formed considering two kinds of texts: main text of the tweet (Main) as well as the derived conversation from the main text (All) Thus, from everyone, training and testing collections, we build up two collections, namely:

- Main-Training,
- All-Training,
- Main-Testing, and
- All-Testing.

These collections allow us to organize the experiments on pairs: given a particular method for some subtask, it was applied on each collection as two independent runnings.

In order to perform Topic Detection subtask it is important to mention that in all experiments it was only taken into account the information contained in the text of the tweets. It is worth mentioning that we classify and cluster tweets for each entity in order to ease the subtask. Additionally, for each entity we work on two subsets: English and Spanish tweets, filtered through the Language attribute.

All experiments for Topic Detection subtask carried out the same preprocessing: removing stop words, morphological and inflectional endings (Porter Stemming)[2], as well as internal links and user names from tweets were removed. Furthermore, text representation of tweets was supported by some term selection techniques and with the purpose of clustering and classifying the information, when WEKA[3] was used, we model it through Bag of Words (BOW) representation with boolean weighting scheme.

Also, Priority Detection subtask was applied to each entity, considering some attributes extracted from the Training and Testing collection; it was not used

Main/All collection as in Topic Detection subtask. Furthermore, at Priority Detection subtask the tweet text and some of the related attributes were extracted from the tweet `html` file and they were stored as plain text file. When there were responses to the tweet text, they were attached to the text plain text file. At this process, we observed, in the gold standard, some tweets were not related to any entity and then, they were discarded.

3 Topic detection subtask

At the first four experiments of Topic Detection subtask, we pretended to know how useful is training set aiming to identify topics in twitter. Experiments seven and eight are completely unsupervised and try to compare its performance to the previous first to sixth tests.

3.1 Supervised detection

We applied two classification algorithms, Naive Bayes and Sequential Minimal Optimization Support Vector Machines (SMO SVM) [4][5]. After testing some combination and configuration of the aforementioned algorithms, using Main-Training and All-Training collections, the best configuration was SMO SVM (with a polykernel and standardized data), then this election was applied to Testing collection, for classifying its elements.

3.2 Unsupervised detection

We performed three pairs of experiments on an unsupervised manner, two of them select terms accordingly with the percentage of terms which obtains the best performance on the training dataset. A final pair of experiments applies a method to automatically select the set of terms used at the representation of tweets.

3.3 DF term selection

In order to improve unsupervised Topic Detection, two term selection methods were tested: the well known *document frequency* index, DF, number of documents which contain the term; and the *transition point*, TP, frequency which divides into high and low the term frequencies [6]. After evaluating some combination and configuration of the aforementioned methods, over both collections (Main and All), the best results were gotten using DF with 43% of the terms of highest DF value (Spanish subset used only 34%).

3.4 Unifier term selection

This test was supported on the *diversification* and *unification* concepts proposed by Zipf [7], which have been used at clustering of web services [8]. Two measures were used aiming to select terms: unifier degree of a term, U, and the saturation of a set of terms, $\hat{S}$.

Given a collection of documents $\mathcal{C} = \{d_1, \ldots, d_n\}$, it is defined as

$$U(t_i) = \frac{1}{r} \sum_{j \neq i} sim(\bar{t_i}, \bar{t_j}),$$

and

$$\hat{S}(\mathcal{C}) = \frac{2}{n(n-1)} \sum_{i \neq j} sim(d_i, d_j),$$

where $\bar{t}$ is the representation of the term t given by the classes in which t occurs, $r = \#\{t_j | sim(\bar{t_i}, \bar{t_j}) \neq 0)\}$, and sim is a similarity measure. In our experiments we used Jaccard coefficient as similarity measure, and the classes, in order to represent terms ($\bar{t}$), were provided by the clustering of the tweets on the same working collection (without term selection). Here we used the K-Star clustering algorithm [9]. In these experiments we discard those tweets with no words contained in the term selection.

Summarizing, the method follows two steps:

1. Select terms basing on U and $\hat{S}$:
   (a) Calculate $U(t)$ for all terms of $\mathcal{C}$ and sort them in increasing order, namely $T_U = [U(t_1), \ldots, U(t_k)]$.
   (b) Divide T_U into m parts, in order to provide m sets of terms: $V_i, (1 \leq i \leq m)$ it represents the first i parts of terms (our experiments used $m = 10$).
   (c) Compute the array $[\hat{S}(\mathcal{C}_i)]$ whose elements correspond to each selection set V_i, and determine the index of the maximum descending value of $\hat{S}(\mathcal{C}_i)$: j.
2. Apply of the K-Star clustering algorithm to $\mathcal{C}_j$.

4 Priority detection subtask

4.1 Attributes used for priority subtask.

From the plain text files, a set of seven attributes were calculated, those are described as follows:

1. Referenced users: calculated from the number of user tags or email found within the tweet text, i.e. the number of tokens with the form `@string` are considered as referenced users.
2. Hashtags: the number of hashtags symbols are counted (#).
3. Web addresses: the number of `http` tokens are considered.
4. Tweet length in characters.

5. Frequency of retweets. This is a measure contained in the tweet information and it is considered as an attribute.
6. Frequency of `favorites`. This is a measure of popularity of the tweet, it is contained in the tweet information and also is considered as an attribute too.
7. Conversation Generated. This is a boolean attribute calculated from the presence or absence of responses to each tweet.

So, every tweet text file belonging to each entity was processed in order to calculate those seven attributes for the sake to classify them as MILDLY IMPORTANT, ALERT and UNIMPORTANT according to the training set.

4.2 Supervised detection

It was used the WEKA application to perform three runs in order to classify the test tweets as it was required for the Priority Detection subtask. So, three classifiers where applied to the files of attributes calculated for each tweet belonging to the entities: the tree inducer algorithm J48, the Naive Bayes and the SMO function (Support Vector Poly Kernel) [10] [4] [5] [11].

At it was stated, the training was executed over the training collection In some cases there were entities with missing tweets belonging to the ALERT class. Those files were filled with 30 tweets from all the entities for the sake to preserve the three classes in all the training set and, thus, to obtaining appropriated classifier's models. So, the three classifiers were applied to the test set as were mentioned.

5 Experimental results

5.1 Topic detection experiments

Four pair of experiments were carried out. Each pair deals with a pair of collections: Main-Testing (Main-Training), and All-Testing (All-Training), as described at Sec. 2. By instance, the Pair One consists of the runnings UAMCLyR_topic_detection_1 and UAMCLyR_topic_detection_2 which use the collections Main-Testing and All-Testing, respectively.

Table 1 depicts how each experiment pair was performed: method, approach (classification/clustering), and the term selection criterion.

Finally, at Table 2 we show for each run the used collection, and in descending order the F values based on Reliability and Sensitivity as well as the Baseline, defined in [12], for the Topic Detection subtask.

As we can see, all our experiments are above of the Baseline. We observed a better performance with clustering than classification. Term selection based on U and $\hat{S}$ provided the best result, however, this method was not able to determine best terms when the collection was extended from Main to All.

Table 1. Summary of Topic Detection experiments.

Experiment	Method	Approach	Selection
Pair One	K-Means	Classification	DF
Pair Two	SMO SVM	Classification	DF
Pair Three	K-Means	Clustering	DF
Pair Four	K-Star	Clustering	$U, \hat{S}$

Table 2. $F(R, S)$ values of the UAMCLyR Topic Detection subtask.

Run	Dataset	F
UAMCLYR_topic_detection_07	Main	0.238
UAMCLYR_topic_detection_05	Main	0.224
UAMCLYR_topic_detection_06	All	0.224
UAMCLYR_topic_detection_08	All	0.212
UAMCLYR_topic_detection_03	Main	0.198
UAMCLYR_topic_detection_04	All	0.192
UAMCLYR_topic_detection_01	Main	0.184
UAMCLYR_topic_detection_02	All	0.177
BASELINE	-	0.173

5.2 Priority Detection Experiments

Priority Detection experiments were carried out over the All-Testing collection using three classification methods as can be seen in Table 3

Table 3. Summary of Priority Detection experiments.

Experiment	Method
UAMCLYR_priority_detection_01	J48
UAMCLYR_priority_detection_02	Naive-Bayes
UAMCLYR_priority_detection_03	SMO SVM

The results of the three experiments of Priority Detection subtask can be seen in Table 4 following a descending order according to F value based on Reliability and Sensitivity and the Baseline, as defined in [12], for this subtask including the Accuracy measure for all the runs performed.

6 Conclusions and future work

In all cases of Topic Detection, clustering approach outperformed to classification approach. Additionally, we realized that when the derived conversation from tweets was included, the detection got worse. Particularly, it can be observed that

Table 4. $F(R, S)$ and Accuracy values of the UAMCLyR team on Priority Detection subtask.

Run	F	Accuracy
BASELINE	0.296	0.600
UAMCLYR_priority_detection_02	0.201	0.459
UAMCLYR_priority_detection_01	0.172	0.559
UAMCLYR_priority_detection_03	0.088	0.573

in all supervised Topic Detection experiments, the term selection method was unable to correctly discriminate the relevant terms when extending the corpus; i.e. from Main to All collection. The term selection based on unification provided the best results, perhaps because it was calculated directly from test collection. However, unification term selection it is not sensitive to the increasing of the vocabulary. We plan to carry out additional tests mainly to the term selection techniques using at clustering of tweets.

It can also be claimed that it is possible to detect priority in tweets based on models of classification that rely only in some attributes calculated from the metadata features of tweets. From this models it can be obtained acceptable results when classifying new instances. As further work, the method of Priority Detection could be tested in two ways in order to be improved:

- to separate the tweets by language
- and the extraction of models based on other attributes which take into account linguistic features of texts.

References

1. T. Martín, D. Spina, E. Amigó, & J. Gonzalo (2012) *UNED at RepLab 2012: Monitoring Task* . CLEF 2012 Working Notes.
2. Porter , M. F. (1997) An algorithm for suffix stripping. Morgan Kaufmann Publishers Inc. pp. 313-316.
3. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann & I. H. Witten (2009) *The WEKA Data Mining Software: An Update*, SIGKDD Explorations, Volume 11, Issue 1.
4. Steve R. Gunn (1998) *Support Vector Machines for Classification and Regression.*University of Southamptom, Technical Report.
5. Platt, John C. (1998) *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*, Technical Report MSR-TR-98-14.
6. H. Jiménez, D. Pinto & P. Rosso (2005) Uso del punto de transición en la selección de términos índice para agrupamiento de textos cortos, Revista Procesamiento del Lenguaje Natural No. 35, pp 416-421, España.
7. G. K. Zipf (1949) *Human Beahaviour and the Principle of Least-Effort.* Addison-Wesley, Cambridge, MA.
8. H. Jiménez, Ch. Sánchez, C. Rodríguez & W. Luna (2011) Modelación léxico semántica de descripciones de servicios web. 8o. Taller de Tecnologías del Lenguaje Humano, Complejo Cultural Universitario. BUAP, Puebla.

9. K. Shin & S.Y. Han (2003) *Fast clustering algorithm for information organization*, Lecture Notes in Computer Science, Vol. 2588, pp 619-622, Springer.
10. J. Ross Quinlan (1993) *C4.5: programs for machine learning*. Morgan Kauf-mann Publishers Inc., San Francisco, CA, USA.
11. Mitchell, T. (2006) *The discipline of machine learning* (Technical Report CMU-ML-06-108). Carnegie Mellon University.
12. E. Amigó, J. Carrillo de Albornoz, I. Chugur, A. Corujo, J. Gonzalo, T. Martín, E. Meij, de M. Rijke & D. Spina (2013) *Overview of RepLab 2013: Evaluating Online Reputation Monitoring Systems*. In Proceedings of the Fourth International Conference of the CLEF initiative, CLEF 2013. Springer LNCS, Valencia, Spain