LIA@Replab 2014 : 10 methods for 3 tasks

Jean-Valère Cossu, Kilian Janod, Emmanuel Ferreira, Julien Gaillard and Marc El-Bèze

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. In this paper, we present the participation of the *Laboratoire Informatique d'Avignon* (LIA) to RepLab 2014 edition [2]. RepLab is an evaluation campaign for Online Reputation Management Systems. LIA has produced an important number of experiments for every tasks of the campaign: Reputation Dimensions and both Author Categorization and Author Ranking sub-tasks from Author Profiling. Our approaches rely on a large variety of machine learning methods. We have chosen to mainly exploit tweet contents. In several of our experiments we have also added selected meta-data. A fewer number of our proposals have integrated external information by using provided background messages.

1 Introduction

RepLab addresses the challenging problem of online Reputation analysis, i.e. mining and understanding opinions about companies and individuals by extracting information conveyed in tweets. Here, the end-user application is monitoring the reputation of several entities from Twitter messages. This year the organizers defined two tasks, namely Reputation Dimensions and Author Profiling. The last one is divided in two sub-tasks respectively Author Categorization and Author Ranking. In this context, LIA's participants have proposed several methods to automatically annotate tweets according to this problematic. We took part into each task. The rest of this paper is structured as follows. In section 2, we briefly discuss about data-set and RepLab tasks. In section 3, we present the LIA's submitted systems. Then in section 4, performances are reported before concluding and discussing some future works.

2 Tasks and Data

2.1 Reputation Dimensions

Data The corpus consists of the same multilingual collection of tweets as last edition [1] referring to a set of 61 entities spread in four domains: automotive, banking, universities and music/artists. Replab 2014 will use only the automotive

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

and banking subsets (31 entities). These tweets cover a period going from the 1st of June 2012 to the 31st of December 2012. Entitie's canonical names have been used as queries to extract tweets from a larger database. For each entity, at least 2,200 tweets have been collected. The 700 first tweets have been taken to compose the training set, and the other ones are used as test set. Consequently, tweets concerning each of the four tasks are not homogeneously distributed in the data-set. The corpus also provides additional background tweets for each entity (up to 50,000, with a large variability across entities). Each tweet is categorized into one of the following reputation dimensions: Products/Services, Innovation, Workplace, Citizenship, Governance, Leadership, Performance and Undefined

We have selected 3,000 tweets from the training collection to build a development set. As shown in table 1 there is bias with one class.

Table 1. Classes distribution in the training set.

Label	Number of tweets
Citizenship	2209
Governance	1303
Innovation	216
Leadership	297
Performance	943
Products & Services	7898
Undefined	2228
Workplace	468

The Reputation Dimensions is a classification tasks that consists in categorizing tweets according to their reputation dimension. The standard categorization provided by the Reputation Institute [1] is used as a gold standard. We may question about what is exactly the meaning of this task since there is a doubt on how the reference has been produced.

2.2 Author Profiling

Data For the author profiling task, the data set consists of over 8,000 Twitter profiles (all with at least 1,000 followers) related to the automotive and banking domains. Each profile consists of :

- author name
- profile URL
- the last 600 tweets published by the author at crawling time

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

Reputation experts have manually identified the opinion makers (i.e. authors with reputation influence) and annotated them as "Influencer". All those profiles that are not considered opinion makers were assigned the "Non-Influencer" label. Profiles for thoses it was not possible to perform a classification into one of these categories have been labeled as "Undecidable". Each opinion maker has been categorized as journalist, professional, authority, activist, investor, company, or celebrity. The data has been split into training and test sets, the proportion is respectively 30% and 70% .

Author Categorization goal's is to classify Twitter profiles by type of author: journalist, professional, authority, activist, investor, company or celebrity. The systems' output is a list of profile identifiers with the assigned categories, one per profile. Note that this sub-task has been evaluated only over the profiles annotated as "Influencer" in the "Author Ranking" gold standard.

Author Ranking objective's is to find out which authors have more reputation influence (who the influencers or opinion makers are) and which profiles are less influential or have no influence at all. For a given domain (e.g. automotive or banking), the system's output had to be a ranking of profiles according to their probability of being an opinion maker with respect to the concrete domain, optionally including the corresponding weights. Some aspects that determine the influence of an author in Twitter – from a reputation analysis perspective – can be the number of followers, the number of comments on a domain or the type of author.

3 Approaches

In this section we propose descriptions of the LIA's approaches used in this edition. Among our 10 approaches, note that parts were also used in the last edition [4]. As some systems are a combination of several methods our systems list can be found resumed in Table 2.

3.1 Cosine distance with TF-IDF and Gini purity criteria

We proposed a supervised classification method based on a cosine distance computed over vectors built using discriminant features like Term Frequency-Inverse Document Frequency (TF-IDF) [13], [12] using the Gini purity criteria [14]. This system consists in two steps. First the text is cleaned by removing hypertext links and punctuation marks and we generate a list of n-grams by using the Gini purity criteria. During this step stop-lists (from Oracle's website) [2] for both English and Spanish have been used. In the second step we creates terms (words or [2/3]-grams) models for each class by using term frequency with the TF-IDF

[2] `http://docs.oracle.com`

and Gini criterion. A cosine distance measures the similarity of a given tweet by comparing its bag of words to the whole bag built for each class and ranks tweets according to this measure. This classification process takes into account the following meta-data :

1. user id;
2. entity id / domain id;

3.2 Hidden Markov Models

Hidden Markov models (HMM) have been widely used for categorization [15]. For each class k, a language model Lmk is built from the train set. The language model Lmk is made of uni-gram probabilities and of probabilities $Pk(w — h)$, where histories h are obtained from chunks automatically selected . Conditional probabilities are estimated from the annotated tweets of the train set assuming that a term is considered as a unique event even though it is occurring several in a tweet (or used by an author). As before meta-data were included into the classification process.

3.3 Poisson modeling

Another approach inspired by the method used for the fast match component of a speech recognition system [3] has been also applied in parallel : although the corpus is not so small, it is interesting to use the Poisson law since it is well suited to take into account the sparse distribution of relevant features f mainly for the under populated classes.

3.4 Naive use of continuous Word2Vec model[8]

Word2vec is an unsupervised algorithm that give a fixed length vector representation for words. Word2vec proved their ability to extract semantics relation between words[9]. This mean that "king"'s vector is closer to "queen"'s vector than "cat"'s vector. We exploit naively this information to do an unsupervised classification. At first, two wor2vec models where built[11]. The first model was made for English from the Brown corpus and every English tweet contained in the background corpus. The second model was made for Spanish from various resources [7] and Spanish tweets in the background corpus. The label "Products & Services" was split in two during classification and re-merge later. Then a naive hypothesis was made.

The hypothesis was that the name of each class (citizenship,innovation ...) represents the meaning of the class and so the vector representation of a tweet wich can be classified must be somehow close to the vector representation of the class name. To achieve this class names were translated from english to spanish manualy and each tweets were preprocessed (like tockenization and stop word removing ...).

Then each words is labeled with the closest class and the majority class give the tweet a label.

3.5 Multilayer Perceptron

This classifier use two Word2vec models, one for English and one for Spanish and a multilayer perceptron (MLP) A multilayer perceptron is a feed-forward neural network model. In MLP each neurons use a nonlinear activation function. MLP are train with back-propagation. Our MLP used a 1 input layer with 2500 units, 1 hidden layout with 200 units , 1 output layout with 8 units and L2 normalization. The input was a 5 Words vectors concatenated. So each tweet had to be split with a five words sliding window. Each word is replaced by its Word2vec [8] representation inside of the sliding window. Then the MLP is trained with the concatenated vector made from the sliding window as intput and with the tweet's label as output. During the classification task the Multilayer Perceptron labeled each window. The final label for the entire tweet is chosen by majority rule from the different windows given a tweet.

3.6 Conditional random field [6]

CRFs represent a log-linear model, normalized at the sentence level. CRFs, though very comparable, have many advantages over hidden Markov models and maximum entropy Markov models (MEMM). HMMs model the joint portability between the observed sequences and tag sequences while CRFs are based on the conditional probability of tags considering the entire sequence. MEMM also maximize this conditional probability but only for local states. In our case, CRFs model the probability between class and words as follows:

$$P(c_1^N|w_1^N) = \frac{1}{Z}\prod_{n=1}^{N} H(c_{n-1}, c_n, w_{n-2}^{n+2}) \tag{1}$$

with

$$H(c_{n-1}, c_n, w_{n-2}^{n+2}) = \sum_{m=1}^{M} \lambda_m \cdot h_m(c_{n-1}, c_n, w_{n-2}^{n+2}) \tag{2}$$

Log-linear models are based on feature functions h_m representing the information extracted from the training corpus, λ are estimated during the training process, Z is a normalization term given by:

$$Z = \sum_{c_1^N}\prod_{n=1}^{N} H(c_{n-1}, c_n, w_{n-2}^{n+2}) \tag{3}$$

The tweets from the training set were used to train our CRF tagger with unigram (5 neighbors) and bigram features. Then a CRF tagged each unigram in every tweets and decision for the final tweet's label is made by majority

Table 2. LIA's systems for RepLab 2014

# Method	Description
1	HMM with TF-IDF and Gini purity criteria
2	Cosine distance with TF-IDF and Gini purity criteria
3	Poisson with TF-IDF and Gini purity criteria
4	Merge of HMM and Cosine (global models)
5	Merge of HMM, Poisson and Cosine (per lang specific models)
6	Multilayer Perceptron
7	Conditional random field
8	Naive Word2vec
9	Merge of Multilayer Perceptron, CRF, Naive and 4
10	Merge of 4 and 5

4 Submissions and results

4.1 Systems

Ten methods compose the LIA's set of submissions. For reading convenience, these methods are summed up in table 2 and refer to a method number used in results table presented above. We now compare our result with regards to the baselines and also to the best score in a given task.

4.2 Reputation Dimensions

Table 3. Submitted runs to Reputation Dimensions Task ordered by F-Score.

#Run-ID	#Method	F-Score	Accuracy
-	**Best**	**0,489**	**0,695**
-	*SVM Baseline*	0,380	0,622
Run 2	6	0,258	0,612
Run 1	7	0,258	0,607
Run 5	9	0,238	0,595
Run 4	4	0,160	0,549
-	*Naive Baseline*	0,152	-
Run 3	8	0,121	0,356

As shown (in table 3) most of our runs, ranked according to F-Score are situated between the SVM and most frequent baselines. All our systems are under the SVM baseline. As our systems were biased by the most frequent class we mainly performed bad in term of per-class F-score (computed with precision and recall) although they are not so bad in terms of accuracy. Runs 2 and 1 used separate models for both English and Spanish languages while runs 4 and 3 used a global model. Run 1 also use the background tweets. The run 6 only used tweet's Word2vec information. Adding other source of information will make

the system do better decision. Likewise we can try to add more hidden layer now that we have more training data or add an unsupervised phase of pre-training. The Naive run (Run 3) did not perform well compared to others. On one hand its ability to infer meanings and semantic distance between words bring new information to the system. On the other hand due to our hypothesis this system bring a lot of noises. Word2vec have already proved that they are able to summarize information contain in a document[10] and thanks to the MLP we know that there is usefull information for this task in the Word2vec model. With this information there is many things we want to do in order validate/invalidate our usage of Word2vec model. The combination (run 5) has not been able to produce a good selection rules since it performances remains lower than the best system taken alone mostly due to the noise given by the Naive system.

Table 4. Classes distribution in gold-standard and systems output.

Label	Run 1	Run 2	Run 3	Run 4	Run 5	Gold	Baseline
Citizenship	4578	3303	7485	855	3188	5027	3263
Governance	1209	1226	1372	465	507	3395	2131
Innovation	54	5	337	38	18	306	27
Leadership	286	46	117	72	120	744	352
Performance	916	1070	10765	266	1284	1598	668
Prod & Svc	20713	24513	12922	29233	25696	15903	19920
Undefined	2720	1186	6	567	383	4349	5303
Workplace	1154	281	58	136	434	1124	241

Classes distribution (in table 4) explains the low performance level our systems (shown in table 3) since they are all biased to Products&Services. As an interesting result we can notice that the Naive run (run 3) over-estimated the Performance class.

4.3 Author Profiling

Author Categorization Ranked according (table 5) to the average accuracy only one system is better than both "most frequent" and "Machine Learning" (SVM) baselines. One our of system is near the SVM baseline for "Automotive" accuracy while it outperforms the "Banking" accuracy of the baseline. A second system is far behind the baselines while the combination is worse.

Run 1 used two different systems combinations depending on the language. For English tweets HMM and Poisson were combined. Whereas in spanish Cosine was added to the above combination because there was less data.

In the second run combined Cosine and HMM where trained with global models without separating languages. Here again our combination (run 3) has

not been able to produce a good selection rules since it does worse than all systems taken alone.

Both baselines produced interesting results since they performed well. Since they are over all other candidates we can consider them as very strong baselines. Another interesting fact is that the "Stockholder" users were not found by any systems.

With regard to the label distribution in the training set, we decided to have an harmonization post-process of our systems output for this task. The post-process consist for each output to consider the second hypothesis of the system in the following case :

- The best hypothesis is an over populated class [3]
- The second hypothesis is an under populated class
- The score differential between the both hypothesis is not significant

In this case the system will full-up small classes despite it has a better confidence in a bigger class. Although this strategy implies as sacrifice some losses in terms of accuracy, it allows the system to be better with small classes. Depending on the chosen evaluation metric this strategy can perform well.

Author Ranking The run uses the same interesting double combination of Poisson and HMM for both English and Spanish tweets as in "Author Categorization" task. We interpreted this task as a binary classification problem for each author. System considered if each tweet in the author bag of tweets in opinionated or not. Considering now the majority label the system decides whether the user is "opinion maker" or not. To rank users we use the probability of the "opinion maker" label on his bag of tweets. In case of parity we add the probability of a HMM system trained with global models.

As in the Author Categorization task our Author Ranking output was post-processed in order the obtain an approaching ratio of "opinion maker" as the training set. Since there were only 2 classes in this task, our post-process can be considered as an offset and threshold set on the probability of one class.

[3] The notion of over or under population is considerd with regards to the class distribution in the training set.

Table 5. Submitted runs to Author Categorization Task ordered by Average accuracy.

#Run-ID	#Method	Automotive	Banking	Misc	Average	F-Score
Run 1	5	0,445	**0,502**	0,461	**0,473**	**0,319**
-	*Baseline-SVM*	0,426	0,494	-	0,460	0,302
-	*MF-Basline*	**0,450**	0,420	**0,51**	0,435	-
Run 2	4	0,356	0,397	0,376	0,377	0,294
Run 3	10	0,292	0,308	0,369	0,300	0,255

Table 6. Classes distribution in gold-standard and systems output.

Label	Run 1	Run 2	Run 3	Gold	Baseline
Public Institution	24	36	60	90	78
NGO	181	190	331	233	49
Stockholder	0	0	0	7	0
Sportsmen	157	219	364	208	7
Journalist	859	1407	1700	991	708
Employee	1	2	3	14	0
Undecidable	1972	1264	515	1412	2851
Celebrity	39	318	347	208	0
Professional	1492	1278	1291	1546	1144
Company	151	165	269	222	82

Table 7. Submitted run, best run and baseline to Author Ranking Task ordered by Average MAP.

#Run-ID	#Method	Automotive	Banking	Average MAP
Best	-	**0,721**	0,410	**0,565**
Run 1	5	0,502	**0,450**	0,476
Baseline	-	0,370	0,385	0,378

5 Conclusions and perspectives

In this paper we have presented the systems as well as the performances reached by the *Laboratoire Informatique d'Avignon* to RepLab 2014. We have presented a large variety of approaches and observed logically a large variety of system performances even about one system in several tasks. Our results are good in both subtasks of "Author Profiling" but it seems like we missed something in the "Reputation Dimensions" We have also proposed several combinations of systems in order to benefit from the diversity of information considered by our runs but it did not worked as expected. Sign that our results could still be improved by looking for another way of considering the data and our systems output during both classification and merging processes. While the mass of data has caused us many troubles, in a future work, we will propose to automatically summarize tweets clusters or users profiles in order to reduce our representation and perform a faster classification. As we have already done on the ImagiWeb dataset [5] we intend to apply an active learning strategy to answer the Replab issue.

References

1. Amigó, E., Corujo, A., Gonzalo, J., Meij, E., de Rijke, M. *Overview of RepLab 2013: Evaluating Online Reputation Management Systems* CLEF 2013 Labs and Workshop Notebook Papers (2013).
2. Amigó, E. ,Carrillo-de-Albornoz, J., Chugur, I., Corujo, A., Gonzalo, J., Meij, E., de Rijke, M., and Spina, D *Overview of RepLab 2014: author profiling and*

reputation dimensions for Online Reputation Management In Proceedings of the Fifth International Conference of the CLEF initiative, 2014, sep, Sheffield, UK
3. Bahl, R.L. and Bakis, R. and De Souza, P.V. and Mercer, R. *Obtaining candidate words by polling in a large vocabulary speech recognition system* In Proceedings of ICASSP 1988 (pp 489-492 vol.1).
4. Cossu J.-V., Bigot B., Bonnefoy L., Morchid M., Bost X., Senay G., Dufour R., Bouvier V., Torres-Moreno J.-M., El-Bèze M. *LIA@RepLab 2013* An evaluation campaign for Online Reputation Management Systems (CLEF'13), 23-26 September 2013.
5. Cossu J.-V., El-Bèze M., Sanjuan E., and Torres-Moreno J.-M *E-reputation monitoring on Twitter with active learning automatic annotation* Techreport hal-01002818, April 2014.
6. Lafferty, J., McCallum, A., and Pereira, F. C. *(2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.*
7. Lara, L.F. and Chande, R.H. and Hidalgo, M.I.G. *Investigaciones lingüísticas en lexicografía, 1979, Colegio de México, Centro de Estudios Lingüísticos y Literarios 89.*
8. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. *Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.*
9. Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. *Linguistic Regularities in Continuous Space Word Representations. In Proceedings of NAACL HLT, 2013.*
10. arXiv:1405.4053 *Quoc V. Le, Tomas Mikolov, Distributed Representations of Sentences and Documents*
11. Radim Řehůřek and Petr Sojka *Software Framework for Topic Modelling with Large Corpora, Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks,2010, p45-50, ELRA, http://is.muni.cz/publication/884893/en*
12. Robertson,S. *Understanding inverse document frequency: on theoretical arguments for IDF* In Journal of Documentation, 60, 5, pp 503-520, 2004, Emerald Group Publishing Limited.
13. Salton, G. et Buckley, C. *Term weighting approaches in automatic text retrieval* In Information Processing and Management 24, pp 513–523, 1988.
14. Torres-Moreno, J.-M., El-Beze, M., Bellot, P. and Bechet, *Opinion detection as a topic classification problem* In in Textual Information Access. Chapter 9, pp 337, John Wiley & Son. 2013
15. Wang, L., and Li, L. *Automatic Text Classification Based on Hidden Markov Model and Support Vector Machine* In Proceedings of The Eighth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 2013 (pp. 217-224).