Identifying and Classifying Influencers in Twitter only with Textual Information

Victoria Nebot[1], Francisco Rangel[2,3], Rafael Berlanga[1], and Paolo Rosso[2]

[1] Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I
Campus de Riu Sec, 12071, Castellón, Spain
Phone: (+34) 964 72 83 67. Fax: (+34) 964 72 84 35
[2] PRHLT Research Center, Universitat Politècnica de València,
Camino de vera s/n, 46022, Valencia, Spain
[3] Autoritas, Av. Puerto 267-5, 46011, Valencia, Spain

Abstract. Online Reputation Management systems aim at identifying and classifying Twitter influencers due to their importance for brands. Current methods mainly rely on metrics provided by Twitter such as followers, retweets, etc. In this work we follow the research initiated at RepLab 2014, but relying only on the textual content of tweets. Moreover, we have proposed a workflow to identify influencers and classify them into an interest group from a reputation point of view, besides the classification proposed at RepLab. We have evaluated two families of classifiers, which do not require feature engineering, namely: deep learning classifiers and traditional classifiers with embeddings. Additionally, we also use two baselines: a simple language model classifier and the "majority class" classifier. Experiments show that most of our methods outperform the reported results in RepLab 2014, especially the proposed Low Dimensionality Statistical Embedding.

Keywords: Online Reputation Management, Influencers Classification, Author Categorization, Textual Information, Twitter

1 Introduction

The rise of Social Media has led to the emergence and spread out of new ways of communicating, without borders or censorship. Nowadays, people can connect with other people anywhere and let them know their opinion about any matter. Furthermore, it is happening massively. The collective imaginary that is formed by the opinions expressed about a brand (organisational or personal) is what makes up its (online) reputation. That is why Online Reputation Management (ORM) has proliferated in parallel with the final goal of taking control of the online conversation to mold it at will. The first step is to find out where people are expressing their opinions about the brand, and what these opinions are. In recent years, research has been very prolific in sentiment analysis, irony detection, stance detection, and so forth. However, one of the main interests for brands is to know who is behind these opinions. Concretely, to know who are the

influencers[4] of their sector and what kind of influence they may generate (what kind of authority/auctoritas[5] they have over what they say).

1.1 Related work

Companies are interested in identifying influencers and classifying them. This problem has been addressed in Twitter in the framework of the RepLab[6] shared task in 2014 [2], as a competitive evaluation campaign for ORM.

At the RepLab the three approximations which obtained the best results were UTDBRG, LyS and LIA. In UTDBRG [1], the underlying hypothesis was that influential authors tweet actively about hot topics. A set of topics was extracted for each domain of tweets and a time-sensitive voting algorithm was used to rank authors in each domain based on the topics. They used quantitative, stylistic and behavioral features extracted from tweet contents. In LyS [14], features were extracted considering PoS tagging and dependency parsing. They used specific (such as URLs, verified account tag, user image) and quantitative (number of followers) profile meta-data. Finally, the authors of LIA [4] modeled each user based on the textual content, together with meta-data associated to her tweets.

Most of the current approaches rely on topology-based features. For example, the authors in [3] used a logistic regression method where all tweets from each user belonging to a given class were merged to create one large document. They employed 29 features such as user activity, stylistic aspects, tweets characteristics, profile fields, occurrence-based term weighting, and local topology. That is, they considered also features such as local topology (size of the friends set, size of the followers set, etc.) that are traditionally used by researchers from social network analysis. In this vein, the Collective Influence algorithm [10], which has been recently optimised [11], uses the "percolation" concept to isolate and identify the most influential nodes in a complex network. Recent investigations [7] also combine textual with topological features to find which tweets are worth spreading. Some findings suggest that influencers use more hashtags and mentions, as well as they follow more people on average.

However, our approximation aims at addressing the problem of identifying and classifying influencers in Twitter only on the basis of the textual information available in the tweets, without taking into consideration meta-data nor topological information from Social Network Analysis. We aim at investigating whether only textual features allow to find and classify such influencers in Twitter.

1.2 Our approach

In RepLab, several different categories are proposed to classify influencers. However, some of these categories are very low populated and the overall distribu-

[4] Influencer is a user (person or brand) who may influence a high number of other people.

[5] In ancient Rome, auctoritas was the general level of prestige a person had. Due to that, authority in this context means prestige.

[6] http://www.clef-initiative.eu/track/replab

tion is very imbalanced. Furthermore, industry usually considers two types of influence: *i)* the capacity of influencing others with their own opinions (influence/authority); and *ii)* the ability to spread a message to the users with whom the user interacts (betweenness). Due to that, we have focused on an alternative way to detect and classify a subgroup of authority. Concretely, journalists who may spread a message, and professionals who have the authority. The proposed approach is based on the schema shown in Figure 1. In a first step, influencers are identified. Then, the identified influencers are classified in two ways. On the one hand, they are classified according to the RepLab taxonomy. On the other hand, influencers are firstly classified as belonging or not to the authority group and then, they are disaggregated into professionals or journalists.

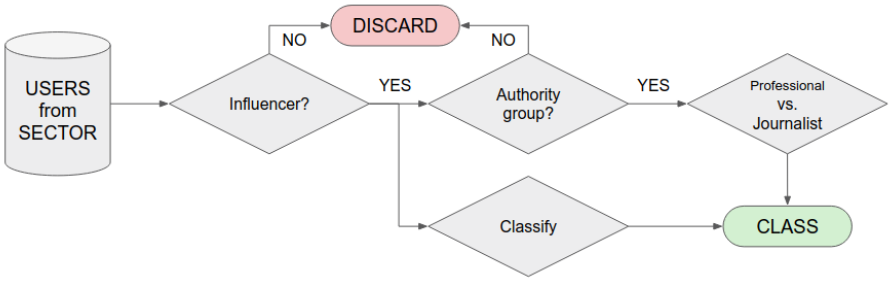

Fig. 1: Workflow to Identify and Classify Influencers.

The remainder of this paper is organized as follows. Section 2 describes our methodology. Section 3 discusses the experimental results, whereas the conclusions are drawn in Section 4.

2 Methodology

In this section, the RepLab 2014 shared task corpus is described. Then, we present the methods we propose to address the problem. Finally, the evaluation measures are discussed.

2.1 RepLab'14 Shared Task Corpus

The data collection contains over 7,000 Twitter profiles (all with at least 1,000 followers) that represent the automotive, banking and miscellaneous domains. We focus on the first two domains. Each profile contains the last 600 tweets published by the author at crawling time. Reputation experts performed manual annotations for two subtasks: Author Categorization and Author Ranking. First, they categorized profiles as company, professional, celebrity, employee, stockholder, journalist, investor, sportsman, public institution, and ngo. Those

profiles that could not be classified into one of these categories, were labeled as undecidable. In addition, reputation experts manually identified the opinion makers and annotated them as Influencer. Because the Author Categorization task is evaluated only over the profiles annotated as influencers in the gold standard, the exact number of profiles in training/test is shown in Table 1[7].

	Automotive			Banking			
Class	Training	Test	Total	Training	Test	Total	Total
Undecidable	454	-	454	556	-	556	1,010
Professional	312	358	670	279	286	565	1,235
Journalist	202	171	373	258	231	489	862
Company	94	119	213	51	33	84	297
Sportsmen	49	36	85	8	4	12	97
Celebrity	27	24	51	33	7	40	91
NGO	21	5	26	78	83	161	187
Public Inst.	12	3	15	27	30	57	72
Employee	3	8	11	1	6	7	18
Total	1,174	724	1,898	1,291	680	1,971	3,869

Table 1: RepLab'14 corpus statistics.

2.2 Methods

In this section we present all the approaches that we have evaluated and compared, along with their running parameters and configuration. We combine both traditional machine learning and deep learning methods with different word embedding techniques. We have classified the approaches based mainly on the type of embeddings they use, as we would like to explore the impact of the different types of embeddings on the classification tasks described above. We have also included a language model approach that was meant to serve as a baseline but has surprisingly given good results.

Glove Glove [12] is a word embedding model used for learning vector representations of words. Such representations are able to encode semantic relations between words, like synonyms, antonyms, or analogies. Therefore, they have been widely used for NLP tasks. It is a count-based model based on matrix factorization. We have evaluated the following approach using this type of word vectors:

– **LSTM with Glove Twitter embeddings (LSTM+Glove)**

Long short-term memory (LSTM) [5] is a deep learning technique that addresses the problem of learning long range dependencies and thus, is a state-of-the-art semantic composition model for a variety of text classification tasks.

[7] We did not include Stockholder and Investor classes because they did not have data for Automotive and in Banking they did not have either training or test data.

In our experiment, we use a combination of the Glove pre-trained word vectors on Twitter[8] and the learned vectors for the RepLab dataset as input sequences for LSTM. In particular, each tweet is represented as the sum[9] of its word embeddings of dimension 50. In order to generate longer sequences per user, we generate sequences of five consecutive tweets. The configuration of the network is as follows: LSTM has 128 hidden units, we add a dropout layer with rate 0.5 and a softmax layer. We set a batchsize of 128 and train during 50 epochs.

Doc2vec Doc2vec [8] is an extension of Word2vec [9]. Word2vec is another word embedding model but it differs from Glove in that it is a predictive model and uses a neural network architecture. Doc2vec learns to correlate labels and words, rather than words with other words. The purpose is to create a numeric representation of a document, regardless of its length. We have evaluated the following approaches using Doc2vec:

- **Logistic Regression with doc2vec (LR+d2v)**
- **Support Vector Machines with doc2vec (SVM+d2v)**
- **Multi Layer Perceptron with doc2vec (MLP+d2v)**
- **Convolutional Neural Network with doc2vec (CNN+d2v)**

We have evaluated both traditional machine learning methods (i.e., LR and SVM) and deep learning methods (i.e., MLP and CNN).

In the experiments, we consider each tweet as a document and obtain its vector representation of dimension 50. All the tweets of a user are aggregated by summing[10] the tweet vectors. This way, we obtain one feature vector per user that encapsulates the semantics of all her tweets. The feature vectors are passed to different classifiers for training. LR is configured with regularization 1e-5 and SVM uses a linear kernel. MLP is composed by 5 layers, a dense layer with 64 neurons, a dropout layer with rate 0.5, another dense layer and dropout layer with the same configuration and the output softmax layer. The batchsize is set to 128 and we train it for 20 epochs. For the CNN we basically use the configuration shown in [6].

Low Dimensionality Statistical Embedding (LDSE) LDSE[11] [13] represents documents on the basis of the probability distribution of the occurrence of their words in the different classes. The key concept of LDSE is a weight, representing the probability of a term to belong to one of the different categories: influencer vs. non-influencer, interest-group vs. others, professional vs. journalist. The distribution of weights for a given document should be closer to the weights of its corresponding category. Formally, we represent the documents following the next three steps:

[8] https://nlp.stanford.edu/projects/glove/
[9] sum has empirically given better results than average or concatenation
[10] sum has empirically given better results than average
[11] Previously in other tasks as Low Dimensionality Representation (LDR)

Step 1. We calculate the *tf-idf* weights for the terms in the training set D and build the matrix Δ, where earch row represents a document d, each column a vocabulary term t, and each cell represent the *tf-idf* weight w_{ij} for each term in each document. Finally, $\delta(d_i)$ represents the assigned class c to the document i.

$$\Delta = \begin{bmatrix} w_{11} & w_{12} & ... & w_{1m} & \delta(d_1) \\ w_{21} & w_{22} & ... & w_{2m} & \delta(d_2) \\ ... & ... & ... & ... & \\ w_{n1} & w_{n2} & ... & w_{nm} & \delta(d_n) \end{bmatrix}, \tag{1}$$

Step 2. Following Eq. 2, we obtain the term weights $W(t, c)$ as the ratio between the weights of the documents belonging to a concrete class c and the total distribution of weights for that term.

$$W(t, c) = \frac{\sum_{d \in D / c = \delta(d)} w_{dt}}{\sum_{d \in D} w_{dt}}, \forall d \in D, c \in C \tag{2}$$

Step 3. Following Eq. 3, these term weights are used to obtain the representation of the documents.

$$d = \{F(c_1), F(c_2), ..., F(c_n)\} \sim \forall c \in C, \tag{3}$$

Each $F(c_i)$ contains the set of features showed in Eq. 4, with the following meaning: *i)* average value of the document term weights; *ii)* standard deviation of the document term weights; *iii)* minimum value of the weights in the document; *iv)* maximum value of the weights in the document; *v)* overall weight of a document as the sum of weights divided by the total number of terms of the document; and *vi)* proportion between the number of vocabulary terms of the document and the total number of terms of the document.

$$F(c_i) = \{avg, std, min, max, prob, prop\} \tag{4}$$

As can be seen, this representation reduces the dimensionality to only six features per class by statistically embedding the distribution of weights of the document terms.

Finally, these weights are learned with a machine learning algorithm. Concretely[12]:

- **Naive Bayes & BayesNet** for influencers identification in the Automotive and the Banking domains, respectively.
- **Naive Bayes** for authority group detection.
- **SVM** for journalist vs. professional discrimination.
- **SVM** for RepLab task on classification.

[12] We have tested several machine learning algorithms and finally we report the ones with the best results.

Language Model (LM) Language models represent documents as a generative process of emitting words from each document $\mathbf{d}$, denoted $P(w_i|\mathbf{d})$. From the training set we can estimate the probabilities that each class c_j generates words w_i by simply applying Maximum Likelihood and Laplace Smoothing. The resulting model, denoted with $P(w_i|c_j)$, and the document model can be then factorized to get an estimate of the probability of each class to be generated by the document as follows:

$$P(c_j|\mathbf{d}) = \sum_{w_i} P(c_j|w_i) \cdot P(w_i|\mathbf{d})$$

Note that $\mathbf{d}$ can be either a tweet or a collection of tweets associated to a user. In this paper, $\mathbf{d}$ refers to the collection of user tweets. The resulting distribution $P(c_j|\mathbf{d})$ provides us the ranking of classes associated to each user to perform the classification.

2.3 Evaluation measures

In RepLab, because the number of opinion makers is expected to be low, the influencers task is modeled as a search problem rather than as a classification problem. Thus, the evaluation measure selected is MAP (Mean Average Precision). For the categorization task, the measure selected to be able to compare us against RepLab results is Accuracy, which for multiclass settings, is equal to micro-averaged precision. However, we also compute the macro averaged version to see the effectiveness on the smaller classes.

3 Experiments and discussion

This section presents the results of the different tasks defined in Figure 1. Firstly, the identification of influencers. Then, the detection of authority users and their classification in journalists or professionals. Finally, the classification of the influencers in the RepLab taxonomy.

3.1 Identification of Influencers

The first task is to identify whether a user is an influencer or not, which corresponds to the Author Ranking task at RepLab. Therefore, the results shown in Table 2 can be compared to the best ones obtained in the competition. As can be seen, the LDSE approach obtains the best results, followed by LR+d2v. Both of them outperform the best results obtained in the RepLab competition by more than 20% on average, demonstrating their competitiveness for the task. It has to be remarked that this was obtained on the basis of textual information of tweets only, without considering meta-data and Social Network Analysis information such as it was done by the best teams at RepLab and posteriorly by Cossu et al. [3].

	Automotive	Banking	Average
LSTM+Glove	0.663	0.654	0.659
MLP+d2v	0.674	0.718	0.696
CNN+d2v	0.785	0.718	0.752
LR+d2v	0.861	**0.816**	0.839
SVM+d2v	0.833	0.784	0.809
LDSE	**0.874**	0.810	**0.842**
LM	0.865	0.526	0.696
RepLab'14	0.720	0.520	0.620
Cossu'15	0.803	0.668	0.735

Table 2: Identification of Influencers (MAP score). RepLab'14 best results were obtained by UTDBRG in Automotive and LyS in Banking.

In this task, except in the case of LM in Banking, traditional approaches obtain a better performance than deep learning approaches, with average differences between 5.7% and 18.3%.

3.2 Identification of the Authority Group

In order to separate the authority group from the rest of users, we have grouped journalist and professional into one group, and the remaining classes in the second group. As shown in Table 3, the resulting dataset is imbalanced towards the authority group.

	Automotive			Banking			
Class	Training	Test	Total	Training	Test	Total	Total
Authority Group	514	529	1,043	537	517	1,054	2,067
Others	206	195	401	198	163	361	762
Total	720	724	1,444	700	878	1,415	2,829

Table 3: Authority vs. others corpus statistics.

Results are shown in Table 4. The best average results have been obtained by LDSE, which also demonstrates its robustness against the imbalance between classes with values about 77% in both micro and macro average. Deep learning methods such as MLP+d2v and CNN+d2v, and to a lesser extent LSTM+GLove, albeit they obtain a similar micro precision, show a trend to bias to the majority class, as shown by the low macro precision.

	Automotive		Banking		Average	
	P-micro	P-macro	P-micro	P-macro	P-micro	P-macro
LSTM+Glove	0.724	0.472	0.760	0.589	0.742	0.531
MLP+d2v	0.731	0.365	0.760	0.380	0.746	0.373
CNN+d2v	0.731	0.365	0.760	0.380	0.746	0.373
LR+d2v	0.733	0.641	0.785	0.702	0.759	0.672
SVM+d2v	**0.751**	0.677	0.774	0.681	0.763	0.679
LDSE	0.745	**0.767**	**0.801**	**0.784**	**0.773**	**0.776**
LM	0.606	0.432	0.716	0.520	0.661	0.476
Majority class	0.731	0.365	0.760	0.38	0.746	0.373

Table 4: Identification of the Authority Group.

3.3 Discriminating between Journalists and Professionals

Once the authority influencers are identified, they should be separated into the corresponding class: journalist or professional. The corresponding dataset is also imbalanced towards the professional class, as shown in Table 5.

	Automotive			Banking			
Class	Training	Test	Total	Training	Test	Total	Total
Professional	312	358	670	279	286	565	1,235
Journalist	202	171	373	258	231	489	862
Total	514	529	1,043	537	517	1,054	2,097

Table 5: Journalists vs. Professionals corpus statistics.

	Automotive		Banking		Average	
	P-micro	P-macro	P-micro	P-macro	P-micro	P-macro
LSTM+Glove	0.673	0.574	0.625	0.619	0.649	0.597
MLP+d2v	0.677	0.338	0.553	0.277	0.615	0.308
CNN+d2v	0.677	0.338	0.584	0.617	0.630	0.478
LR+d2v	**0.745**	0.720	0.551	0.710	0.648	0.715
SVM+d2v	0.728	0.686	0.749	0.748	0.738	0.717
LDSE	0.742	**0.746**	**0.756**	**0.755**	**0.749**	**0.751**
LM	0.687	0.614	0,671	0.679	0.679	0.646
Majority class	0.677	0.338	0.553	0.277	0.615	0.308

Table 6: Discrimination between Journalist vs. Professional

The obtained results are shown in Table 6. Again, the LDSE obtains the highest results for both micro and macro precisions. In this task, traditional

approaches also obtain higher results than deep learning approaches, besides less bias to the majority class.

3.4 Author Categorization

In this section we discuss the results of the classification of the influencers in the RepLab taxonomy. Table 7 shows our results compared with the best accuracies reported in RepLab and the majority class prediction. This experiment includes all the classes defined in the dataset [13].

	Automotive		Banking		Average	
	P-micro	P-macro	P-micro	P-macro	P-micro	P-macro
LSTM+Glove	0.488	0.153	0.463	0.196	0.476	0.174
MLP+d2v	0.495	0.062	0.340	0.043	0.417	0.052
CNN+d2v	0.495	0.062	0.335	0.058	0.415	0.060
LR+d2v	0.524	0.228	0.594	0.330	0.559	0.279
SVM+d2v	0.526	**0.243**	0.579	0.279	0.553	0.261
LDSE	0.562	0.195	0.568	0.282	0.565	0.238
LM	**0.699**	0.205	**0.760**	**0.394**	**0.730**	**0.300**
Majority class	0.475	0.237	0.410	0.205	0.443	0.221
RepLab'14	0.450	-	0.500	-	0.475	-

Table 7: Results of the categorization task of the RepLab'14 shared task. RepLab'14 best results were obtained by LIA in both Automotive and Banking domains.

Results show that deep learning approaches (LSTM, MLP, CNN) have comparable performance to the best results of RepLab'14 in terms of micro precision. However, macro precision reveals poor performance regarding small classes. In particular, LSTM classifies great part of the test dataset to the majority class, basically following the training distribution. MLP+d2v and CNN+d2v also obtain very poor performance because they classify all the test data to the dominant class of the training set.

Traditional machine learning approaches using embeddings outperform the best results of the RepLab'14 competition, whose results are very similar to the majority class baseline. It is worth noting the scores given by LM, which was initially included as a baseline, but clearly outperforms the rest of the approaches in this task. LM has a similar performance in all the tasks and its performance is not downgraded in this multiclass classification task, unlike the other approaches. This is due to the fact that LM builds the model for each class independently.

Once more, we should highlight that our approach, which only uses textual information from tweets, obtains more than 25% better results than LIA where meta-data information associated to tweets was also used.

[13] As the Undecidable class only appears in training set, we have removed it to avoid noise in the training phase.

4 Conclusions

Companies are interested in identifying influencers and classifying them. This task was addressed in the framework of RepLab. In this paper, we were interested in approaching the problem both on the basis of deep learning techniques and with traditional machine learning. We compared the obtained results with the best systems in RepLab and the more recent approach of Cossu et al. On the contrary of the above systems, which used also meta-data and information from Social Network Analysis, we used only textual information and outperformed their results by more than 20% in influencer identification and more than 25% in influencer classification.

It is noteworthy that deep learning approaches obtained worse results than traditional techniques in all the previous tasks, where their results were biased to the majority class. Probably the low number of training samples (around 500 without undecidable class) explains part of these results. Low Dimensionality Statistical Embedding (LDSE) obtained the best results in almost all the tasks and it shows its robustness against corpus imbalance.

Due to the interest of the industry on identifying different kinds of influence, we have proposed an alternative methodology: first, we identify the influencer; then, we determine if the influencer belongs to the authority group; finally, we classify the influencer as journalist or professional. The results obtained show that the best approach (LDSE) is competitive: more than 84% identifying influencers, and 77% and 75% classifying them respectively in the authoritative group, and as the type of influencer. Nevertheless, future work should focus on combining it with generative models like LM as it shows best results in the categorization task, which is the only multi-classification task.

Acknowledgements The work of the last author was funded by the SomEMBED TIN2015-71147-C2-1-P MINECO research project. Authors from Universitat Jaume I have been funded by the MINECO R&D project TIN2017-88805-R.

References

1. Abolfazl AleAhmad, Payam Karisani, Masoud Rahgozar, and Farhad Oroumchian. University of tehran at replab 2014. In *Proceedings of the Fifth International Conference of the CLEF Initiative*, 2014.
2. Enrique Amigó, Jorge Carrillo-de-Albornoz, Irina Chugur, Adolfo Corujo, Julio Gonzalo, Edgar Meij, Maarten de Rijke, and Damiano 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*, September 2014.
3. Jean-Valère Cossu, Nicolas Dugué, and Vincent Labatut. Detecting real-world influence through twitter. In *Network Intelligence Conference (ENIC), 2015 Second European*, pages 83–90. IEEE, 2015.
4. Jean-Valère Cossu, Kilian Janod, Emmanuel Ferreira, Julien Gaillard, and Marc El-Bèze. Lia@ replab 2014: 10 methods for 3 tasks. In *Proceedings of the Fifth International Conference of the CLEF Initiative*, 2014.

5. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.
6. Yoon Kim. Convolutional neural networks for sentence classification. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL, 2014.
7. Eva Lahuerta-Otero and Rebeca Cordero-Gutiérrez. Looking for the perfect tweet. the use of data mining techniques to find influencers on twitter. *Computers in Human Behavior*, 64:575–583, 2016.
8. Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1188–II–1196. JMLR.org, 2014.
9. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.
10. Flaviano Morone and Hernán A Makse. Influence maximization in complex networks through optimal percolation. *Nature*, 524(7563):65, 2015.
11. Flaviano Morone, Byungjoon Min, Lin Bo, Romain Mari, and Hernán A Makse. Collective influence algorithm to find influencers via optimal percolation in massively large social media. *Scientific reports*, 6:30062, 2016.
12. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543. ACL, 2014.
13. Francisco Rangel, Paolo Rosso, and Marc Franco-Salvador. A low dimensionality representation for language variety identification. In *17th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing*. Springer-Verlag, LNCS, arXiv:1705.10754, 2016.
14. David Vilares, Miguel Hermo, Miguel A Alonso, Carlos Gómez-Rodríguez, and Jesús Vilares. Lys at clef replab 2014: Creating the state of the art in author influence ranking and reputation classification on twitter. In *Proceedings of the Fifth International Conference of the CLEF Initiative*, 2014.