Automatic Detection of Influencers in Social Networks: Authority versus Domain signals

Javier Rodríguez-Vidal, Julio Gonzalo, Laura Plaza
Universidad Nacional de Educación a Distancia (UNED)

Henry Anaya Sánchez
Séntisis Analytics

Author Note

Javier Rodríguez-Vidal (jrodriguez@lsi.uned.es), Julio Gonzalo (julio@lsi.uned.es) and Laura Plaza (lplaza@lsi.uned.es), Department of Lenguajes y Sistemas Informáticos, UNED, Madrid, Spain.

Henry Anaya Sánchez (hanaya@sentisis.com), Séntisis Analytics, Madrid, Spain.

Corresponding author: Javier Rodríguez-Vidal

Abstract

Given the task of finding influencers (opinion makers) for a given domain (e.g. banking) in a social network, we investigate (i) what is the relative importance of domain and authority signals; (ii) what is the most effective way of combining signals (voting, classification, learning to rank, and other variants), and how best to model vocabulary signals; (iii) how large is the gap between supervised and unsupervised methods, and what are the practical consequences. Our best results on the RepLab Twitter dataset (which improves the state of the art) uses language models to learn the domain-specific vocabulary used by influencers, and combines domain and authority models using a Learning to Rank algorithm. Our experiments show that (i) both authority and domain evidence can be trained from the vocabulary of influencers; (ii) once the language of influencers is modelled as a likelihood signal, further supervised learning (via classifiers or learning to rank) and additional network-based signals only provide marginal improvements; (iii) although our best unsupervised system is only 7% worse than the best system at the RepLab competition, our best supervised system is 40% better, which indicates that the availability of training datasets is crucial to obtain competitive results in the task.

Our most remarkable finding is that influencers do use a distinctive vocabulary, which is a more reliable signal than non-textual network indicators such as the number of followers, retweets, etc.

Keywords: Learning to Rank, Web and social media search, Information extraction, Social Network Analysis, Natural Language Processing, Social Media Influencers

Automatic Detection of Influencers in Social Networks: Authority versus Domain signals

Introduction

According to the Cambridge dictionary, an *influencer* is the person who has the capacity to have an effect on the character, development, or behavior of someone or something. As stated in Biran, Rosenthal, Andreas, McKeown, and Rambow (2012), these kind of users must fulfill three properties: they have credibility inside the group, they are capable of being persuasive with other people even if some disagreements occur, and they can introduce new ideas that other components of the group select or support.

Before the advent of Social Media, people with the capacity of influencing the public opinion in a given domain were few and easy to identify: journalists from mass media, authorities with academic degrees and proved expertise, politicians, media owners, celebrities, etc. In practice, editorial boards and lobbies could effectively decide what information and what opinions reached the masses, and how. Public Relations (PR) for organizations and individuals were, then, a matter of addressing a few opinion makers to shape their reputation, i.e., how their image was projected to the public opinion.

Social Media has significantly complicated matters for organizations from the point of view of Public Relations. Monitoring and managing social media brings unprecedented opportunities to know and interact with clients and stakeholders, but it renders previous PR methodologies obsolete. One of the key aspects of Online Media, and of Social Media in particular, is that any citizen is a candidate to becoming influential: it is no longer possible to narrow the filter to media owners, journalists, academic experts and other standard profiles. In this context, one of the key aspects of Online Reputation Monitoring (ORM) is to detect which social media profiles have the capacity of influencing the public opinion and, therefore, creating opinion and shaping the reputation of organizations, companies, brands and individuals (Madden & Smith, 2010).

This paper is focused on the automatic detection of *influencers* (or *opinion makers*) in social networks and, in particular, in microblogging networks as exemplified by Twitter. A differentiating characteristic of Twitter, as a social network, is that breaking news appear and propagate first (it acts as the "neural spine" of online content). Therefore, issues that may affect the reputation of a company (and may require PR actions) can usually be spotted in Twitter earlier than in other social media. There are two other features that make us choose Twitter for our study: first, its contents are eminently public (unlike, e.g. Facebook); and second, there is already a suitable test collection available (the RepLab 2014 dataset (Amigó et al., 2014)).

We focus on a number of research questions that have not been previously addressed explicitly in the literature:

- **RQ1:What is the relative importance of authority signals vs domain expertise signals?** In order to be influential in a given domain, two major types of signal are involved: signals of authority (for instance, *does the user have many followers? Are her statements frequently retweeted?*) and signals of relevance to the domain (An influential voice in macro economics may have no influence at all when talking about music, for instance). We want to find out what is the relative role that each of the two types of signals play when detecting influencers and to discover which are the most useful specific signals in both cases.

- **RQ2:How best to combine signals?** We will explore several ways of combining signals to discover influencers: supervised classification, unsupervised signal voting, supervised learning to rank, and supervised classification combined with signal voting. We will also combine all these approaches with language models that learn both the domain language and the language of influencers, using them as the primary textual signals.

Our best results (which improve the state of the art) use language models to learn the domain-specific vocabulary used by influencers, and combine domain and authority models using a Learning to Rank algorithm. These results suggest that influencers do use a distinctive vocabulary, which is a more reliable signal than non-textual network indicators such as the number of followers, retweets, etc.

Related Work

Influencers are relevant authors in social networks, because they are trustworthy to the community. With their opinions they have the power to strengthen or harm the reputation of products and entire companies. There are different studies and techniques about how to detect influencers in social networks. Aral and Walker (2012) defend that, in order to predict the propagation of actions, is important to use jointly the influence, the susceptibility and the likelihood of spontaneous adoption in the local network around individuals. But, as the authors point out, it is not clear whether influence and susceptibility are general features or depend on the domain.

Sharma, Saha, and Dasgupta (2013) discuss the word-to-mouth marketing concept, where the brand that we are scouting is connected to certain subscribers which are grouped (all or a subset of them), as *influencers*, and their friends are treated as potential consumers. One of the implications of this model is that we have to identify the right kind of consumers; the fact that two people are connected does not mean that they have the same tastes. Squicciarini, Rajtmajer, Liu, and Griffin (2015) deal with identifying cyberbullies as influencers, and makes two main questions: how to measure the probability that a user engages in cyberbulling given her characteristics (social interactions and language used), the content posted and the social network metrics (centrality, etc.), and how to measure the influence of bullies on their peers. Other approach to find *influencers* is introduced in (Pope III et al., 2015), where the authors used classifiers based on fuzzy logic and linguistic features such as part of speech (POS) (in English noun, pronoun, adjective, determiner, verb, adverb, preposition, conjunction, and interjection) for the identification of *influencers*.

Bouguessa and Romdhane (2015) discussed some drawbacks about influencer identification approaches, such as the dependency on labeled data. They proposed a mixture-model-based of multivariate beta distributions approach to automatically identify authoritative users. Page, Brin, Motwani, and Winograd (1999) and Kleinberg (1999) take advantage of the network structure in order to identify authorities. In the first study, they calculate the relevance (the influence) of a web page (the Twitter users) according to the number of links received from other web pages and also, they check the kind of web page that provides the link. Meanwhile, the second study, authorities are identified as pages that were linked by many hubs, a hub is a page (directory) that pointed to many other pages.

Characterization and identification of authors in Twitter has its own characteristics. The first one is text sparsity due to Twitter's length restriction (140 characters) appearing many infrequent terms, some of them being considered noise. Also, the users follow very different rules of writing (e.g. "what" or "whaaaaatt"). Another issue is the large amount of features that Twitter provide[1] which are used and categorized in different ways. In Cossu, Labatut, and Dugué (2016), seven categories where the features can be grouped are introduced: user profile, publishing activity, local connections, user interaction, lexical aspects, stylistic traits and external data. Maleewong (2016) takes advantage of retweeter-based features for predicting the popularity of new tweets. This group of signals attempts to describe the main properties of the retweeters of a tweet. Another attribute that is widely used for author classification in Twitter, and in all Social Network in general, is the broad number of subjects that users could talk about. Bigonha, Cardoso, Moro, Gonçalves, and Almeida (2012) analyze the users behaviors, interactions and connections in order to determine their influence in Twitter. The main idea is that key users tend to be the leaders of conversations and actions for a given topic.

With respect to author profiling in Twitter, the problem can be modeled as a classification or as a ranking task. The first approach is an intuitive choice because we have different items (authors), and we want to put them in different groups. In classification techniques we use features such as number of followers, number of published tweets, etc. in order to learn to predict which users are *influencers* or not. Pennacchiotti and Popescu (2011) observed that linguistic features (e.g. prototypal words, typical lexical expressions and hashtags (for people with similar interests), generic LDA, domain-specific LDA and sentiment words) are reliable in order to distinguish political affiliation. This conclusion is aligned with (Conover, Gonçalves, Ratkiewicz, Flammini, & Menczer, 2011), where using Twitter features as retweets also provides competitive accuracy.

But detecting influencers in ORM has two distinguishing features: first, the number of influencers is orders of magnitude lower than the number of non-influencers;

i.e. the classes are highly unbalanced. Second, potential influencers are usually scanned by reputation experts, which use automatic filters as a preliminary step. Both features are characteristic of search problems, where ranking is the most natural way of presenting results to the users (in this case, the reputation experts). This is the approach taken in RepLab 2014[2]. RepLab was an evaluation campaign for reputation monitoring systems, which in 2014 included an author profiling task on Twitter where detecting and characterizing influencers was the main challenge. This is the dataset used in our experiments and it is described in detail in the next section. The best system in the competition was (Aleahmada, Karisania, Rahgozara, & Oroumchiana, 2014), which implemented the idea that people who are influencers will talk more about hot topics. Using the same dataset, Cossu et al. (2016) obtained the best results up-to-date, and they concluded that users from particular domains behave and write in their own specific way and using only text-based features is enough to detect domain influencers. Furthermore, the authors did not model the authority and domain vocabulary which seems to be an useful information for *influencers* detection. Furthermore, no previous work has employed Learning to Rank techniques to locate influencers in Twitter, which seem a natural choice to address a ranking problem when there is training material.

Methods

In this section we present the signals and algorithms used for the automatic detection of influencers in Twitter.

Signals

One of our main goals is to compare the utility of authority versus domain signals. To this end, we have experimented with a wide range of signals, that we have classified in two categories: User signals and Textual signals.

User Signals. We have used both raw Twitter features and meaningful combinations of them, as described below.

Simple Signals. In this category we have selected the simplest features of the users' profiles: Feature 1 (*tweets*) number of tweets published by the author. Feature 2

(*RTs*) and 3 (*FAVs*) are the number of *retweets* and *favorites* respectively. Feature 4 (*Foll*) is the number of *followers*. Finally, Feature 5 (*Follees*) is the number of *followees*, the people that the user is following. All these features are summarized in Figure 1.

Combined Signals. Feature 6 (*DivFoll*) is the number of tweets published by the user's followers and by her followees. This signal can be understood as the level of publishing activity in the vicinity of the user's interests. Feature 7 (*DivFollees*) is the opposite to feature 6, and it measures the level of publishing activity of other non-related profiles. Features 8 (*DivRTFoll*) and 9 (*DivFAVFoll*) represent the level of interaction with the tweets published in the vicinity of the user's interests. Feature 10 (*DivRTFAVFoll*) is the total activity of the interaction with the tweets. Features 11 (*DivRTFollees*),12 (*DivFAVFollees*) and 13 (*DivRTFAVFollees*) represent the interaction carried out by the users' followees with their surroundings. Features 14 (*RVR*),15 (*FVR*) and 16 (*TVR*) are well-known features in the marketing field (Ramón & López, 2016). They measure how well the message, given by the user, spreads through the audience. Finally, Feature 17 (*Borda*) is a combination of the other features applying Borda voting algorithm (Saari, 1999). All these features are described in Figure 1.

Textual Signals. The textual content of a user's posts is a powerful distinguishing feature. First of all, it offers direct evidence of the relevance of the user's opinion with respect to the domain of interest. Second, it may also provide evidence for authority, under the hypothesis that authorities use a distinct vocabulary. For instance, an authority in the automotive domain may use technical words such as crankshaft, valves, etc. more often than regular users. In our work, we use textual signals in two ways, the simpler one only uses bag-of-words and the other one is more sophisticated and employs language models.

Bag of Words. Our baseline, a naive approach, uses a bag-of-words representation, where a text is represented as the bag (multiset) of its words, disregarding grammar and even word order. Note than, in the representation of the tweets, each word is a feature and therefore, the number of textual features is significantly larger than the rest of features. That might be a problem for learning algorithms, because the effect of non-textual features might be shadowed by the large textual signal.

Language Models. Our second approach to represent textual information solves this problem by using a Topic Modeling approach described in (Sánchez, 2016). In essence, we use (i) the training material to build a language model of authorities, and we use one single feature for each author that estimates to what extent her language is compatible with the language of authorities; and (ii) tweets from the domain to build a language model for the domain, and we use another feature to estimate how well the author's discourse fits that language.

Here we summarize the language modeling technique that we use. It learns a model of the language underlying a domain of authors, D, in the context of a reference collection of documents, $C = d_1, .., d_{|C|}$, with vocabulary $V = w_1, .., w_{|V|}$.

The aim of the model is to obtain a probability distribution of words, $p'(w)$, in which words likely to be accurately included in an author message in the domain D are assigned high probability values; whereas other words, including those that are very ambiguous or not domain-specific but occur in D, receive marginalized values.

Thus, by representing the context of the reference collection C with a probability distribution of words $p(w)_{w \in C}$, our method learns the distribution of words $p'(w)$ as the one that minimizes the cross entropy value, as expressed in Eq.1:

$$H_s = -\sum_{w \in V} p(w|L, D) log((1 - \lambda)p'(w) + \lambda p(w)) \tag{1}$$

where the argument of the logarithm is a mixture in which λ is the weight that accounts for the proportion of "context noise" in $p(w|L, D)_{w \in V}$, and $p(w)$ is the probability of word w under the reference model (i.e., the prior of w in C). The lower the value of λ $(0 \leq \lambda \leq 1)$ the more refined (i.e., the more content words having to do with both L and D will be weighted higher, while contentless or off topic words will be weighted lower) the model $p(w|L, D)$. In our experiments, we have used $\lambda = 0.2$, which properly marginalizes the scoring of general domain (frequent) contentless words (such as prepositions and broad/ambiguous verbs) in the model.

Therefore, the main idea behind is that we want to know the probability distribution of the words used by each author preferring those words that minimize the entropy value, in other words, that maximize the quantity of information.

From the above optimization equation, we base the learning of distribution $p'(w)$ on an Expectation Maximization procedure that starting from initial (uniform) values for $p'(w)_{w \in V}$, it iteratively approximates the values in $p'(w)_{w \in V}$ until convergence by means of the following updates in the r-th iteration:

$p'^{(r)}(w)$ is de Maximization-Step and is defined as:

$$p'^{(r)}(w) = \frac{p(w|L,D) * Z(w)}{(\sum_{w' \in V} p(w'|L,D)Z(w'))} \tag{2}$$

where $Z(w)$ is the Expectation-Step and is defined as:

$$Z(w) = \frac{(1-\lambda)p'^{(r-1)}(w)}{((1-\lambda)p'^{(r-1)}(w) + \lambda p(w))} \tag{3}$$

In our work, we define both $p(w|L)$ and $p(w|D)$ as follows:

$$p(w|L) = \frac{tf(w,L)}{\sum_{w' \in L}(tf(w'))} \tag{4}$$

$$p(w|D) = \frac{tf(w,D)}{\sum_{w' \in D}(tf(w'))} \tag{5}$$

The probability of an author a belonging to the language model D is finally computed as:

$$p(D|a) = \sum_{w}(p(D|w) * p(w|a)) \tag{6}$$

where

$$p(D|w) = Z(w)$$

$$p(w|a) \propto tf(w,Y)$$

being Y the set of tweets of the author a.

Note that both the Authority and Domain models are computed using the same procedure. The main difference is that to compute the authority model we use the training data (and hence it is a supervised process), while to compute the domain model we simply assume that all the tweets in the dataset belong to the domain (hence it is an unsupervised process with respect to the task).

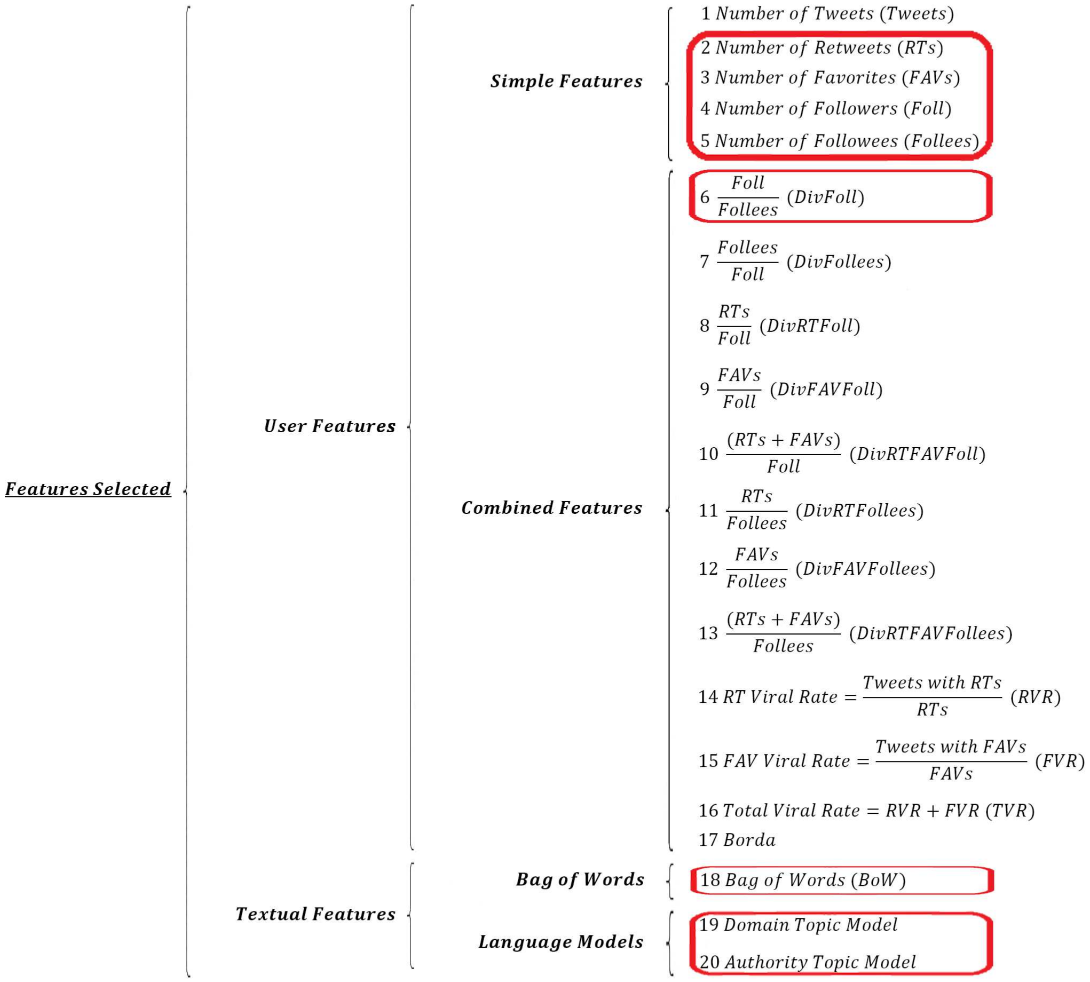
Features Selected
User Features
Simple Features
1 Number of Tweets (Tweets)
2 Number of Retweets (RTs)
3 Number of Favorites (FAVs)
4 Number of Followers (Foll)
5 Number of Followees (Follees)
Combined Features
6 $\frac{Foll}{Follees}$ (DivFoll)
7 $\frac{Follees}{Foll}$ (DivFollees)
8 $\frac{RTs}{Foll}$ (DivRTFoll)
9 $\frac{FAVs}{Foll}$ (DivFAVFoll)
10 $\frac{(RTs + FAVs)}{Foll}$ (DivRTFAVFoll)
11 $\frac{RTs}{Follees}$ (DivRTFollees)
12 $\frac{FAVs}{Follees}$ (DivFAVFollees)
13 $\frac{(RTs + FAVs)}{Follees}$ (DivRTFAVFollees)
14 $RT\ Viral\ Rate = \frac{Tweets\ with\ RTs}{RTs}$ (RVR)
15 $FAV\ Viral\ Rate = \frac{Tweets\ with\ FAVs}{FAVs}$ (FVR)
16 $Total\ Viral\ Rate = RVR + FVR$ (TVR)
17 Borda
Textual Features
Bag of Words
18 Bag of Words (BoW)
Language Models
19 Domain Topic Model
20 Authority Topic Model

Figure 1. Features Selected. The signals included in the red boxes are the filtered features.

Algorithms

We have compared four algorithms that aim to generate a ranking of users according to their probability to be influencers. The first of them is very simple and lies in ordering the users' profiles by one or more chosen features. In the second approach, we uses the classifier confidence for ordering the users' profiles. The third one, is more sophisticated and uses Learning to Rank. The last one combines the first and second approaches.

- **Direct Signal Rank Strategy (DSR):** Each feature generates a ranking of users. When we use two or more features to produce a single rank, we apply a Borda voting step (Saari, 1999) to combine the ranks produced by each individual signal. The combined ranking is produced by adding the values assigned to each element by every rank, and using this number to produce the final ranking.

  Note that this is an unsupervised approach unless one or more signals are obtained using the training data. In our experiments, the only supervised signals are the language models, which compare the language of each user with the language models of authorities (authority model) or with the language models of tweets belonging to the domain (domain model).

- **Classifier Rank Strategy (CR):** Using classifiers is the most widely used technique for our problem in state of the art systems such as (Ramírez-de-la Rosa, Villatoro-Tello, Jiménez-Salazar, & Sánchez-Sánchez, 2014; Vilares, Hermo, Alonso, Gómez-Rodríguez, & Vilares, 2014). For each instance, the classifier makes a binary choice (opinion maker or non opinion maker) and provides a level of confidence for its choice. We take the confidence as the ranking signal, and we generate a ranking by ordering the instances in decreasing confidence order. We have experimented with several algorithms: SVM, Bayes Net, Naïve Bayes, Decision Trees and AdaBoost. This experimentation has been carried out using Weka tool (Hall et al., 2009) and the default parameters for each classifier.

- **Learning to Rank Strategy (L2R):** Instead of learning to classify in order to rank, we can directly learn to rank using Learning to Rank algorithms from the Information Retrieval field (Liu, 2009). These ranking models seek to optimize a chosen evaluation measure on the training data (in our case *Mean Average Precision* (MAP) (Manning, Raghavan, & Schütze, 2008), which is the one used in RepLab). We have focused on Pairwise approaches as our classification problem is binary, and we have experimented with three algorithms: **MART** (Friedman, 2001), **LambdaRank** (Wu, Burges, Svore, & Gao, 2008) and **RankBoost** (Freund, Iyer, Schapire, & Singer, 2003)

This experimentation was carried out using the RankLib tool (Dang, n.d.).

- **Direct Signal with Classification Filter Rank Strategy (DSCFR):** This strategy was first proposed in (Lomeña, 2014), and combines classification with signal rank. The output of the classifier is used to divide the profiles in two groups. Profiles classified as opinion makers all go first in the ranking, and non-opinion makers after them. Inside each group, profiles are ranked according to the values of ranking signals (instead of using confidence scores from the classifiers), combined via Borda voting as in our first ranking strategy. The idea is that the classification step provides useful information to rank profiles, but the confidence measure might not be as useful as the information that authority signals directly provide.

Experimental Framework

The primary focus of our experiments is to determine how our method, which relies on features extracted from the users' Twitter profile and published texts, compares to other methods that work under the same conditions. To do so, we perform experiments on the RepLab 2014 dataset, which is the largest expert annotated one for Author Ranking task. As we mentioned in the previous section, we select some features to determine whether an user is an *influencer* or not. Not all signals have the same strength when it comes to detect this type of users, so, we will filter those that are not useful for us and then, we will generate the different types of rankings (see section Algorithms).

Dataset

We have followed the guidelines of the RepLab 2014 competition and used the Author Ranking dataset for all our experiments. In this task (Amigó et al., 2014), systems are expected to "find out which authors have more reputational influence" for a given domain (automotive and banking). The systems' output is a ranking of Twitter profiles according to their probability of being opinion makers with respect to the domain.

The RepLab 2014 Dataset consists of 7,622 Twitter profiles (all with at least 1,000 followers) related to the automotive and banking domains (exists other domains such as universities, music/artists and miscellaneous but they were not used in this task). The profiles are divided in: 2500 training profiles, 4991 test profiles (for automotive and banking) and 131 additional test profiles which are domain-independent. Each profile consists of (i) author name; (ii) profile URL and (iii) the last 600 tweets published by the author at crawling time. Reputation experts manually assessed each profile as *opinion-maker* or *non-opinion-maker*.

Experiments

In this section we describe the feature selection process (using as input the features listed in Figure 1), and the algorithmic approach for ranking generation.

Feature Selection. Due to the large amount of features (see Figure 1) it is advisable to apply a previous narrow down step in order to select and study the features that fit better to our task. For this purpose we apply Feature Engineering, as described below:

- First, we performed a division of the RepLab dataset according to its different languages (English and Spanish) and domains (Automotive and Banking). We have four different ways to divide it: **separated by domain and language** (e.g. Automotive and English or Banking and Spanish), **separated by domai**n (e.g. Automotive and Banking), **separated by language** (e.g. English and Spanish) and **not separated**, without distinction between domain or language.
- With the input features of our algorithm and the different division strategies, we perform a *classifier step* using the training data. In this step, the different features are divided, according to Division Strategies, and they are used as a input to the classifiers presented in section Algorithms, and the best classifier is chosen.
- With the predictions, we generate intermediate rankings for the different features, in an independent way, and we evaluate each one of them using *Mean Average Precision* (MAP).
- With these results for each feature and ranking strategy, we compute the average between the best and the worst feature result, discarding each signal below the

average. Finally, we retain the features which perform above the average for all ranking strategies.

Note that bag-of-words features do not provide a rank, which means that methods that use direct signal rank (*DSCFR* and *DSR*) cannot use them. The signals marked with a red square in Figure 1), are selected after the feature engineering process.

Rank Generator. Once we have selected the best features (see Figure 1), we generate and evaluate the final ranks produced by each approach. First of all, we have combined the filtered features in all possible ways in order to test if there exists a combination which gives the best result. Then we have split the *Dataset*, in the way explained in the previous section.

The different feature combinations are the input to the classifiers of section Algorithms. Finally, we have generated the final rankings and we have evaluated them using MAP (Manning et al., 2008). Note that some re-ranking strategies use features for this purpose and if we have a combination of two or more sortable features, we have to apply a previous Borda voting step in order to obtain a final re-ranking signal.

Metrics

MAP (Manning et al., 2008) is the official metric for RepLab 2014 competition so that, in order to compare us with the state-of-the-art systems, we also have used it. According to (Amigó et al., 2014) two main reasons exist to adopt it: it is well-known in information retrieval, and it is recall-oriented and lower ranking author's relevance is considered.

Results and Discussion

Table 1 summarizes the results (in terms of MAP) of all experiments using a simple approach to handle textual content (textual features are a bag of words extracted from the author posts). Table 2 summarizes the same set of experiments, but replacing bag-of-words features for a single feature that estimates the probability of the textual content belonging to the language model of the authorities (which is a supervised feature), and a similar feature that estimates the probability of the textual content belonging to the domain of interest (which is an unsupervised feature as it does not need hand-tagged training material). Both tables average the results obtained in the two RepLab domains: banking and automotive. Note that although results slightly vary across domains, the relative difference between strategies is relatively stable, with no major discrepancies. The appendix also includes results reported on precision at 10. Again, the results for each of the metrics are mostly consistent with each other, and therefore we focus on discussing MAP results.

	DSR*	**CR**	**DSCFR***	**L2R**
BoW with Foll	**0.38**	0.26	**0.54**	0.27
BoW with RTs	0.33	0.42	0.52	0.33
BoW with FAVs	0.32	0.42	0.51	0.33
BoW with DivFoll	0.34	**0.44**	0.44	0.32
BoW with FAVs and Foll	0.36	0.42	0.48	0.45
BoW with Foll and DivFoll	**0.38**	0.38	0.43	0.54
BoW with Follees, Foll and DivFoll	**0.38**	0.38	0.48	0.58
BoW with RTs, FAVs, Foll, Follees and DivFoll	0.36	0.38	0.44	**0.59**
Best Result Published	0.714			
Best RepLab 2014	0.57			
Baseline - Followers	0.38	0.34	0.38	0.30

Table 1

Overall MAP results using BoW to handle textual content

First of all, a direct comparison of Tables 1 and 2 reveals that our proposed topic model provides a substantial boost in performance with respect to the bag of words approach: learning topic models for the textual content is much better than applying any other learning algorithm using words as features, which means that there is a language model that characterizes influential people. In fact, it improves the state of the art results without using any additional signal. Therefore, we focus on the results using language models (Table 2) for the rest of the discussion.

According to the results in Table 2, we can conclude the following:

	DSR	**CR**	**DSCFR**	**L2R**
Best combination of user features (Twitter Auth.)	0.38	0.34	0.38	0.32
Domain Vocabulary	0.43	0.45	0.56	0.60
Twitter Auth. + Domain Voc.	0.53	**0.54**	0.57	0.63
Authority Vocabulary	**0.73**	0.40	**0.73**	0.70
Twitter Auth. + Authority Voc.	0.65	0.46	0.71	0.72
Domain Voc. + Authority Voc.	0.68	0.47	0.72	**0.74**
Best result published	0.714			
Best RepLab 2014	0.57			
Baseline - Followers	0.38	0.34	0.38	0.30

Table 2

Overall MAP Results using our Topic Model approach to handle textual signals

1. The vocabulary in the tweets is enough to learn domain and authority signals. Our best result is obtained with *L2R* over domain and authority signals learned from the training data (0.74), which also outperforms the best published result on the dataset (0.71 (Cossu et al., 2016)). The role of the domain signal, however, is only marginally relevant: the authority signal alone provides 0.73 (-1.3%) without using any ML algorithm. A possible explanation is that modeling the vocabulary of authorities in a given domain also, implicitly, includes domain information (we are comparing authorities in the domain with all other profiles in the dataset, which include people in and out of the domain).

2. Our exhaustive test of alternative ways of using signals indicates that no algorithm (classification, learning to rank or combined approaches) is able to improve significantly the raw use of signals to rank candidates. Surprisingly, ranking candidates according to how well their vocabulary fits into the models built with the training data provides results almost as good as any ML algorithm, even *L2R* which seems particularly well suited for the task.

   With suboptimal signals, however, L2R and the combined strategy are sometimes able to boost performance (e.g. *L2R* improves 0.43 → 0.60 using a domain vocabulary signal; and the combined method improves 0.43 → 0.56 with the same signal).

Classification performs poorly, which is not surprising given that its objective optimization functions are classification evaluation metrics.

3. When comparing unsupervised vs supervised approaches, we extract the main following conclusions:

   (a) The best unsupervised method (0.53) is a Borda combination of the rankings provided by three unsupervised signals: number of followers, Followers/followees, and the domain signal. Note that this result is only 7% worse than the best system in the competition (0.57). But using the same signals, the best supervised method (*L2R*) provides a 19% relative improvement (0.63).

   (b) The best supervised method is L2R over the vocabulary signals modeling domain and authority (0.74). Note that the authority signal is itself supervised, as it models the vocabulary of Twitter profiles annotated as authorities in the training dataset. This represents a 40% improvement over the best unsupervised method.

4. The study of alternative evaluation measures (P@10) (see Tables 3 and 4) corroborates the results given by MAP.

	DSR*	**CR**	**DSCFR***	**L2R**
BoW with Foll	**0.60**	0.40	0.35	0.25
BoW with RTs	0.10	**0.50**	0.45	0.28
BoW with FAVs	0.25	**0.50**	0.35	0.15
BoW with DivFoll	0.45	0.35	0.25	0.25
BoW with FAVs and Foll	0.35	**0.50**	0.45	0
BoW with Foll and DivFoll	0.40	0.45	**0.60**	0.10
BoW with Follees, Foll and DivFoll	0.20	0.20	0.50	0
BoW with RTs, FAVs, Foll, Follees and DivFoll	0.30	0.45	0.40	**1**
Best Combination	**0.60**	**0.50**	**0.60**	**1**

Table 3

P@10 Average Results between Automotive and Banking Domains using BoW (not use BoW for ranking because it is not a sortable.)*

	DSR	**CR**	**DSCFR**	**L2R**
Twitter Authority	0.35	0.38	0.50	0.40
Domain Vocabulary	0.70	0.70	0.70	0.85
Twitter Auth. + Domain Voc.	0.90	0.40	0.65	0.95
Authority Vocabulary	0.90	0.45	0.90	**1**
Twitter Auth. + Authority Voc.	**1**	**0.80**	**1**	0.95
Domain Voc. + Authority Voc.	0.85	0.35	0.90	**1**
Best Combination	**1**	**0.80**	**1**	**1**

Table 4

P@10 Average Results between Modeled Automotive and Modeled Banking Domains

5. Even though results per domain are not presented in the paper, the evaluation has also shown that in the automotive domain is easier to identify influencers than in the Banking domain. This happens because the vocabulary concerning to banking is more specific (mortgages, stock values, etc.) and is always used meanwhile, the specific vocabulary used in automotive is used from time to time e.g. crankshaft, valves, etc.

Conclusions

Our main goal in this paper was to investigate and compare the role of domain and authority signals in the task of finding Twitter influencers for a given domain. In order to do so, we have experimented with several signals and Machine Learning algorithms, and we have proposed a topic modeling approach to handle the textual signal in tweets. Using the RepLab 2014 Author Profiling dataset (the largest of its kind known to us), the main results of our experiments are:

- Although it is common practice to assume that influencers are simply users with a number of followers above a certain threshold, our results indicate that reality is much more complex. The number of followers might be an initial filtering criterion (no one can be influential without an audience), but for users with more than 1,000 followers (all candidates in the dataset meet this criterion), the textual content signal is significantly more powerful than the number of followers and any other Twitter authority indicators (number of retweets, favorites, ratio of followers/followees, etc.).

- Our best result is obtained with L2R using the output of our domain and authority topic models (0.74), which also outperforms the best published result on the dataset (0.71 Cossu et al. (2016)). Note, however, that the difference with using exclusively textual topic models (0.73) is small and does probably not justify the use of Learning to Rank machinery and domain information. In fact, our exhaustive test of alternative ways of using signals indicates that no algorithm (classification, learning to rank, combined approaches) is able to improve significantly the raw use of signals to rank candidates.

- In the absence of training data, our best applicable method is a Borda voting combination of the rankings provided by three unsupervised signals: number of followers, Followers/followees, and the domain signal (which is modeled using the training part of the dataset, but does not use the authority labels or any other hand-tagged data).

The fact that the textual content is the key signal in our experiments, and makes Twitter signals unnecesary, is a positive result in terms of applicability of our findings, as we may expect to have competitive results in any other social network that includes textual content. In fact, in most other social networks there is no limit on the size of the posts (as there is in Twitter), and with longer posts and more textual content results might be even better.

Our study has some limitations. Most importantly, the RepLab dataset samples data from two domains, and we have seen that there is a substantial variability across domains. Our results should be confirmed in a larger and more diverse range of domains. Second, we have focused on supervised approaches (our unsupervised approaches are all Borda voting of primitive signal ranks). The difference between unsupervised and supervised approaches might stretch with more clever unsupervised approaches to the problem. Finally, we did not use in our study, for practical reasons, the topological properties of the Twitter network. For instance, profiling the followers of a given profile may help establishing its authority. Mining and including information from the Twitter graph is the next step that we intend to address.

Footnotes

[1] `https://dev.twitter.com/overview/api/users`

[2] `http://nlp.uned.es/replab2014/`

References

Aleahmada, A., Karisania, P., Rahgozara, M., & Oroumchiana, F. (2014). University of tehran at replab 2014.

Amigó, E., Carrillo-de Albornoz, J., Chugur, I., Corujo, A., Gonzalo, J., Meij, E., ... Spina, D. (2014). Overview of replab 2014: author profiling and reputation dimensions for online reputation management. In *International conference of the cross-language evaluation forum for european languages* (pp. 307–322).

Aral, S., & Walker, D. (2012). Identifying influential and susceptible members of social networks. *Science*, *337*(6092), 337–341.

Bigonha, C., Cardoso, T. N., Moro, M. M., Gonçalves, M. A., & Almeida, V. A. (2012). Sentiment-based influence detection on twitter. *Journal of the Brazilian Computer Society*, *18*(3), 169–183.

Biran, O., Rosenthal, S., Andreas, J., McKeown, K., & Rambow, O. (2012). Detecting influencers in written online conversations. In *Proceedings of the second workshop on language in social media* (pp. 37–45).

Bouguessa, M., & Romdhane, L. B. (2015, April). Identifying authorities in online communities. *ACM Trans. Intell. Syst. Technol.*, *6*(3), 30:1–30:23. Retrieved from `http://doi.acm.org/10.1145/2700481` doi: 10.1145/2700481

Conover, M. D., Gonçalves, B., Ratkiewicz, J., Flammini, A., & Menczer, F. (2011). Predicting the political alignment of twitter users. In *Privacy, security, risk and trust (passat) and 2011 ieee third inernational conference on social computing (socialcom), 2011 ieee third international conference on* (pp. 192–199).

Cossu, J.-V., Labatut, V., & Dugué, N. (2016). A review of features for the discrimination of twitter users: application to the prediction of offline influence. *Social Network Analysis and Mining*, *6*(1), 1–23.

Dang, V. (n.d.). The lemur project-wiki-ranklib. *Lemur Project,[Online]. Available: http://sourceforge. net/p/lemur/wiki/RankLib.*

Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, *4*(Nov), 933–969.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, *11*(1), 10–18.

Kleinberg, J. M. (1999). Hubs, authorities, and communities. *ACM computing surveys (CSUR)*, *31*(4es), 5.

Liu, T.-Y. (2009, March). Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, *3*(3), 225–331. Retrieved from `http://dx.doi.org/10.1561/1500000016` doi: 10.1561/1500000016

Lomeña, J. J. M. (2014). *Identificación y clasificación automática de creadores de opinión en Twitter* (Unpublished master's thesis). Universidad Nacional de Educación a Distancia.

Madden, M., & Smith, A. (2010). Reputation management and social media.

Maleewong, K. (2016). An analysis of influential users for predicting the popularity of news tweets. In *Pacific rim international conference on artificial intelligence* (pp. 306–318).

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval.* New York, NY, USA: Cambridge University Press.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The pagerank citation ranking: Bringing order to the web.* (Tech. Rep.). Stanford InfoLab.

Pennacchiotti, M., & Popescu, A.-M. (2011). A machine learning approach to twitter user classification. *ICWSM*, *11*(1), 281–288.

Pope III, F., Shirvani, R. A., Rwebangira, M. R., Chouikha, M., Taylor, A., Ramirez, A. A., & Torfi, A. (2015). Automatic detection of small groups of persons, influential members, relations and hierarchy in written conversations using fuzzy logic. In *Proceedings of the international conference on data mining (dmin)* (p. 155).

Ramírez-de-la Rosa, G., Villatoro-Tello, E., Jiménez-Salazar, H., & Sánchez-Sánchez, C. (2014). Towards automatic detection of user influence in twitter by means of stylistic and behavioral features. In *Mexican international conference on artificial intelligence* (pp. 245–256).

Ramón, A. E., & López, C. S. (2016). *Comunicación integrada de marketing.* ESIC Editorial.

Saari, D. G. (1999). Explaining all three-alternative voting outcomes. *Journal of Economic Theory*, *87*, 313–355.

Sánchez, H. A. (2016). *Discovering and describing coherent and meaningful topics from document collections* (Unpublished doctoral dissertation). Universitat Jaume I.

Sharma, D., Saha, D., & Dasgupta, P. (2013). *A graph-based scheme for brand promotion in social media platforms using influencer nodes* (Tech. Rep.). IIM Calcutta, Working Paper Series.

Squicciarini, A., Rajtmajer, S., Liu, Y., & Griffin, C. (2015). Identification and characterization of cyberbullying dynamics in an online social network. In *Proceedings of the 2015 ieee/acm international conference on advances in social networks analysis and mining 2015* (pp. 280–285).

Vilares, D., Hermo, M., Alonso, M. A., Gómez-Rodríguez, C., & Vilares, J. (2014). Lys at clef replab 2014: Creating the state of the art in author influence ranking and reputation classification on twitter. *CLEF (Working Notes)*, 1468–1478.

Wu, Q., Burges, C. J., Svore, K. M., & Gao, J. (2008). *Ranking, boosting, and model adaptation* (Tech. Rep.). Technical report, Microsoft Research.