Article

Journal of Information Science
2016, Vol. 42(5) 659–674

DOI: 10.1177/0165551515605217
jis.sagepub.com
SAGE

OLFinder: Finding opinion leaders in online social networks

Abolfazl Aleahmad
University of Tehran, Iran

Payam Karisani
University of Tehran, Iran

Maseud Rahgozar
University of Tehran, Iran

Farhad Oroumchian
University of Wollongong in Dubai, UAE

Abstract
Opinion leaders are the influential people who are able to shape the minds and thoughts of other people in their society. Finding opinion leaders is an important task in various domains ranging from marketing to politics. In this paper, a new effective algorithm for finding opinion leaders in a given domain in online social networks is introduced. The proposed algorithm, named OLFinder, detects the main topics of discussion in a given domain, calculates a competency and a popularity score for each user in the given domain, then calculates a probability for being an opinion leader in that domain by using the competency and the popularity scores and finally ranks the users of the social network based on their probability of being an opinion leader. Our experimental results show that OLFinder outperforms other methods based on precision-recall, average precision and P@N measures.

Keywords
Opinion leader identification; people retrieval; social networks

1. Introduction

Finding influential nodes in different kinds of networks ranging from heterogeneous scholarly networks [1–3] to protein interaction networks [4] is an important topic in the field of information science. Social networks are also a kind of network which is made up of a set of people and a set of relations between them. There are numerous reasons for identifying influential people in social networks; advertising [5, 6], direct marketing [5, 7], predicting customer equity [6, 8], selling premium services [6, 9], analysing political and social intentions [10, 11], adopting new technologies [12] and improving brand awareness [13] are some of the major motivations for studying this area of science.

Opinion leaders are the influential people in a society that are able to shape ideas of other people. They are frequently asked for their opinions and advice by other members of their society. In other words, they serve as a medium that transmits information from different media to the people of their society. The concept of opinion leadership was introduced by Paul Lazarsfeld and Elihu Katz in the theory of 'The Two Step Flow of Communication' [14] for the first time. Since its introduction, many researches have been studying different aspects of this phenomenon. For example, different

Corresponding authors:
Abolfazl Aleahmad and Maseud Rahgozar, Database Research Group, Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, North Kargar St, University of Tehran, Tehran, Iran.
Emails: aleahmad@ut.ac.ir; rahgozar@ut.ac.ir

characteristics of opinion leaders have been studied by researchers in the field of sociology such as [15] or [16]. Elihu Katz and Paul Lazarsfeld have summarized some of the most important characteristics of opinion leaders as:

- Personification of certain values: Opinion leaders express the main values of their society and the opinion followers want to be like them.
- Competence: Opinion leaders have competency in their profession. Opinion followers prefer an opinion leader with the knowledge, familiarity or expertise on the matter.
- Strategic social location: Opinion leaders have more strategic position than others in their social network. Opinion leaders have many relations with others within/outside of their community.

The above characterization covers a combination of both personal and social aspects of opinion leaders that is broadly accepted by sociologists [15]. In this study, we have utilized the above characterization in our algorithm for finding opinion leaders in online social networks. The problem can be stated as the following: given a set of users A, and a set of domains D (e.g. automotive, banking, etc.), it is desirable to identify the opinion makers in each domain and rank them based on their influence.

Furthermore, we will use a standard dataset sampled from Twitter to evaluate the proposed algorithm based on the commonly known evaluation metrics: average precision and precision-recall. The dataset is prepared and verified in the Replab 2014 campaign of the CLEF conference [17]. In comparisons, our proposed algorithm can identify opinion leaders considerably better than other state-of-the-art algorithms.

The rest of this paper is organized as follows: section 2 describes the related works, section 3 presents the proposed algorithm, section 4 reports our experimental environment setup and the evaluation of our proposed algorithm, section 5 discuses some important issues about the algorithm and finally section 6 concludes the paper.

2. Related work

Many researchers from a variety of disciplines have contributed to understanding peoples' behaviour in social networks; for example, from psychology and sociology [16, 18], from business and marketing [19, 20], from discrete mathematics [21, 22] and from computer science [23, 24].

Opinion leaders in the two-step flow theory [16] are the people who possess exceptional qualities; ideas often flow from media to opinion leaders and then they are passed to the rest of their society. Although there are some other researchers that criticize the role of opinion leaders [20, 25], the importance of such influential people is often emphasized [6, 26–28]. There are a number of models which try to simulate the diffusion of information in social networks for identification of influential people. The Linear Threshold Model [29] and Independent Cascade Model [30] are the most common models; the majority of other approaches are based on these two models [31]. In both models, a set of initial nodes are assumed to be active, and in each step, based on some specific criteria, a number of inactive nodes may become active [32]. These models describe why and how an influential node can affect other nodes in a social network. Influence maximization algorithms [20] use these models to maximize the propagation of an idea in a social network [31]. Generally influence maximization algorithms make use of the structural location of people in social networks.

Most studies about finding influential people in online social networks are carried out on Twitter. Different features of Twitter are studied for this purpose; Cha et al. [33] compare the effects of three different measures (i.e. the number of followers, number of retweets and number of mentions) and conclude that each measure leads to a different group of users. Also, they analyse the role of each measure and demonstrate that, in-degree, the number of followers is related to popularity. Kwak et al. [34] compare the number of followers, the number of retweets and the result of PageRank [23] and reconfirm that each criterion leads to a different group of users.

Also, there are a number of graph-based approaches that combine network structure and network content to find influential people. Weng et al. [35] state that the 'following' relationship can be justified by the phenomenon of homophily; that is, some users may follow back their followers due to the common topic that they share. Based on this idea, they propose a PageRank-based approach named TwitterRank which considers the topical similarity between users in addition to their link structure. Bakshy et al. [28] restrict their definition of influential users and propose a model to predict influential users based on the cascade size of the retweeted tweets which contain a URL. Cataldi et al. [36] propose an unsupervised system to estimate the influence of a user in a community. Using an N-gram classifier, they classify all tweets and construct a domain exchange graph for each of the classes; then, they analyse the diffusion of information in these graphs and estimate the influence of users on each community. Apart from these works – and including other online social networks like Facebook – features like time [37, 38], read and reply count [37], login count [5] and some external factors [39] have been investigated too.

From a different perspective, we can categorize the state of the art into two categories: Those researchers who try to identify global influential users regardless of a specific domain [40, 41] and those other researchers that focus on finding influential users in a given domain [36, 37]. Some studies in the second category are more related to our work. Vilares et al. [42] use a set of features to train a lib-linear classifier. After the classification step, they sort the users based on a confidence factor and build their final ranked list. They hypothesize that since the task is domain specific, the biography of the profiles must be useful. Therefore, they use the user descriptions as a bag-of-words representation for the feature vectors that are fed to their classifiers. They also investigate various features some of which are binary. For example: the URLs that exist in the profiles, verified accounts, background image of the profiles. Additionally, they use some numerical features such as favourite count, follower count, friend count and status count.

Cossu et al. [43] have investigated different machine learning approaches and their combinations by using tweet contents to build an effective binary classifier. They interpret the problem as a binary classification task for each user. The most notable approaches that they use are Cosine distance with TF-IDF [44] and Gini purity criteria [45], Hidden Markov model [46], Poisson modelling [47] and Word2Vector model [48]. Their proposed systems, determine whether each tweet is opinionated or not; then, having the tweet labels, the systems decide whether the user is an opinion maker or not. The final rank is calculated using the probability of positive tweets in each user's profile. Their best result is achieved using the combination of HMM, Poisson model and Cosine distance.

Villatoro-Tello et al. [49] propose a two-step method; the first step is supervised and the second step is unsupervised. In the first step, a ranked list of profiles is obtained by using a SVM classifier and two categories of features that are extracted from user profiles. Those features used in the first step are self-descriptions (such as the words in user profiles, mentions, the number of hashtags and the number of URLs), and user statistics (such as the number of tweets, followers, followings, the average of tweets per followers and the ratio of followings by followers). In the second step, they use a Markov Random Field configured as described in [50], to generate the final ranked list of opinion leaders. Also, they use other stylistic and behavioural features to estimate the similarities among the users (such as the number of URLs, hashtags, mentions, their vocabulary richness, the average number of favourites, re-tweets and the average posting frequency time with their respective standard deviations). Then, to configure the initial state of the MRF they use the results of the first step. Villatoro-Tello et al.'s results indicate that opinion makers have similar writing styles and behavioural patterns in Twitter. Also, Lomena and Ostenero [51] propose a method based on the number of followers and re-tweet speed.

It is often reported that many structural factors of social networks have not been helpful for finding influential people [33, 34, 52]; therefore, the primary motivation behind our work is to use another important feature for this purpose. In other words, we present an algorithm for using the posts written in online social networks for finding opinion leaders.

3. The proposed algorithm

The opinion leader finding problem is defined as: given a set of users that are active in a given domain, it is desired to find and rank opinion leaders based on their influence in that domain. Our algorithm that we call OLFinder has four steps. Table 1 provides the definition of the symbols used in the algorithm. Figure 1 and Figure 2 summarize the OLFinder algorithm for finding a ranked list of opinion leaders for a particular domain $d \in D$ (i.e. automotive and banking).

In the first step of the proposed algorithm, the main topics of a domain d are extracted. Usually users write about a number of topics using particular words from their vocabulary. Topic modelling is a statistical approach for detecting the main topics of a collection of documents. One of the well-received approaches in topic modelling is called LDA (Latent

Table 1. Symbols used in description of the proposed algorithm.

Symbol	Description
$A=\{a\}$	The set of users
$D=\{d\}$	The set of domains
$T(d)=\{t\}$	The set of topics in the domain d
$P(a)$	The set of posts published by the user a
$R(t,d)$	The set of posts ranked by DFR model [53] w.r.t. $t \in T(d)$
$\vec{W}=\{w_t\}$	The topic weighting vector; w_t is the weight of the topic t
$C(d)$	The collection of posts in the domain d
$G(d)$	The set of the users that are marked as opinion maker in our ground truth
$F(a)$	The in-degree centrality of the user a

Input: *A*: the list of users, $d \in D$: a domain of interest, *C(d)*: the collection of posts in the domain *d, F(a):* the in-degree centrality of the user *a*
Output: A ranked list of opinion leaders in the domain *d*

Step 1: Detection of the main topics discussed in the domain *d*:
Extract the set of the main topics T_d that are discussed in the domain *d*
Step 2: Computing competence of the users w.r.t. the topics set T_d:
Determine how much competency does each user $a \in A$ have with regard to each topic $t \in T_d$.
Step 3: Computing popularity of the users:
Compute popularity of each user $a \in A$ by the use of their in-degree centrality, *F(a)*.
Step 4: Provide the final ranking of the users:
Calculate a leadership score for each user $a \in A$ and provide the final ranking of the users in the given domain *d*

Figure 1. Summarization of the main steps of OLFinder.

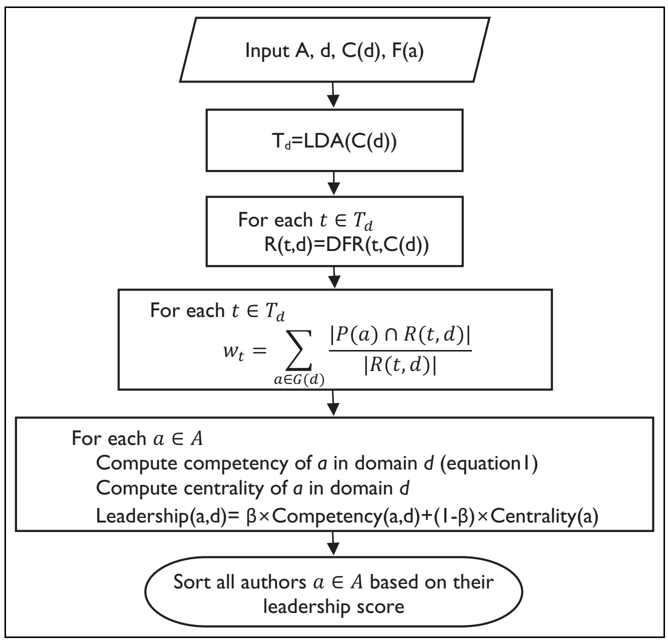


Figure 2. Flowchart summarization of the proposed algorithm.

Dirichlet Allocation) [54]. LDA considers documents as random mixtures of words over latent topics and tries to detect the topics using the following generative process:

(1) Choose $\theta \sim \mathrm{Dir}(\alpha)$.
(2) For each of the N_i words w_n:
(2.1) Choose a topic $z_n \sim \mathrm{Multinomial}(\theta)$.
(2.2) Choose a word w_n from $p(w_n \mid z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

Where α is the parameter of the Dirichlet prior on the per-document topic distributions, β is the parameter of the Dirichlet prior on the per-topic word distribution, N_i is the length distribution for the documents, θ is the topic distribution for the documents, z_n contains the topics for the nth document, and w_n is the words that are present in the nth document. The plate notation of LDA is depicted in Figure 3 [54].

If we consider each post published by the users in the domain *d* as a document then it is possible to use LDA for extracting the main topics from this collection of documents.

In the second step for calculating competence of each user with respect to each topic, we can use an information retrieval system to find a set of posts that are most relevant to the topic. Then we can calculate a score based on how often a user's posts appear in this set and how important they are for this topic. Let *T(d)* be the set of topics extracted by LDA in

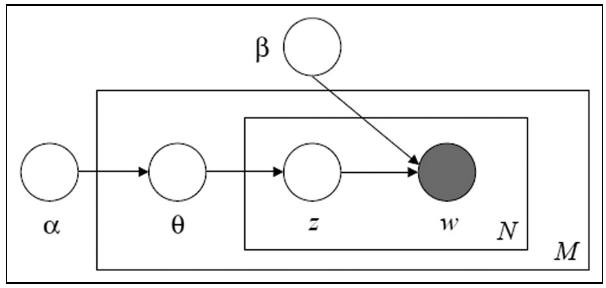


Figure 3. Plate notation of Latent Dirichlet allocation [54]: The larger box represents documents (M is the number of documents) and the smaller box represents the possible topics and words within a document (N is the number of words in a document).

the first step for the domain *d*. For each topic in *T(d)*, we retrieve a set *R(t,d)* of top thousand documents ranked by the Divergence From Randomness (DFR) retrieval model [53]. In other words, *R(t,d)* shows the relevance of the posts published by different users in the domain *d* with respect to each topic extracted for that domain. Then, the competency score of a user *a* in the domain *d* is calculated as follows:

$$Competency(a, d) = \frac{1}{|T(d)|} \sum\nolimits_{t \in T(d)} w_t \left(|R(t, d) \cap P(a)| \sum\nolimits_{p \in R(t, d) \cap P(a)} e^{DFR\_Score(p, t)} \right) \tag{1}$$

where *P(a)* is the set of posts written by the user *a*, $|R(t, d) \cap P(a)|$ is the number of the posts from the user *a* that also exist in *R(t,d)* and $DFR\_Score(p, t)$ is the normalized relevancy score of the content of the post *p* with regard to the topic *t* which is calculated using DFR model. It should be noted that the DFR model used in this research is based on the Poisson model with Laplace after-effect and the second normalization method of DFR described in [53]. Also, w_t is the weight of topic *t* that measures its significance in the domain *d*. w_t is estimated using formula (2) within a training dataset. Since not all the topics are equally important for representing a domain, w_t represents the value of a topic for a domain and it is calculated as the precision of the list of retrieved posts for the topic *t* within the domain *d*.

$$w_t = \sum\nolimits_{a \in G(d)} \frac{|P(a) \cap R(t, d)|}{|R(t, d)|} \tag{2}$$

where *G(d)* is the set of users that are marked as opinion makers in the domain *d* of the training set. Of course as any learned parameter in any data-based method, it is sensitive to training set quality. However, since the authors are ranked in a domain *d*, it is expected that the topics of the test set will not contain entirely new topics. In other words, both test and training sets belong to the same domain and contain the topics that are related to the domain *d*.

The next step of the proposed algorithm calculates the importance of the users based on their strategic location in the social network. Centrality indices are most commonly used as measures of the influence of users in a network. In order to choose a centrality measure, we experimented with the following measures: betweenness centrality, closeness centrality, degree centrality, HITS Authority, HITS Hub and PageRank. We ranked the people in our training dataset and compared the results with available data in the training dataset. Figures 4 and 5 depict the results of these experiments on automotive and banking domains.

As shown in Figures 4 and 5, the in-degree centrality, HITS Authority and PageRank are working better than the other centrality indices. Also, in [33] it is reported that in-degree centrality is a good indicator of people's popularity in social networks. Therefore for the rest of our experiments, we chose in-degree as a measure of centrality and popularity of the people in their networks.

Furthermore, it is commonly accepted that most social networks are scale-free networks [55]; that means their degree distribution follows a power law: $P(k) = k^{-\gamma}$ where *k* is the in-degree centrality of people in social networks and γ is a constant number (usually $2 < \gamma < 3$). The power law exponent γ can be estimated by a curve fitting approach like regression analysis [56]. We used the regression analysis to learn function, *P(k)*, in a way that given an in-degree value it can best estimate the number of people with that in-degree in the social network. Figure 6 (left) shows a number of curves for various values of γ.

A popularity score can be calculated based on the number of in-links (i.e. number of followers in Twitter) of the users as follows:

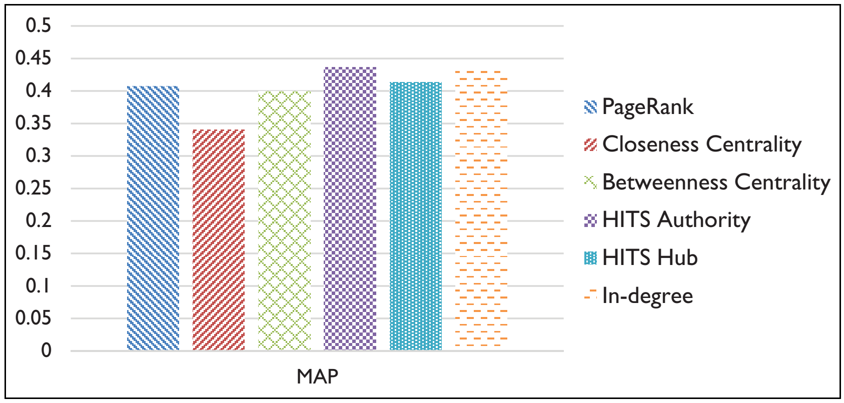


Figure 4. Comparison of different centrality measures based on AveP in automotive domain.

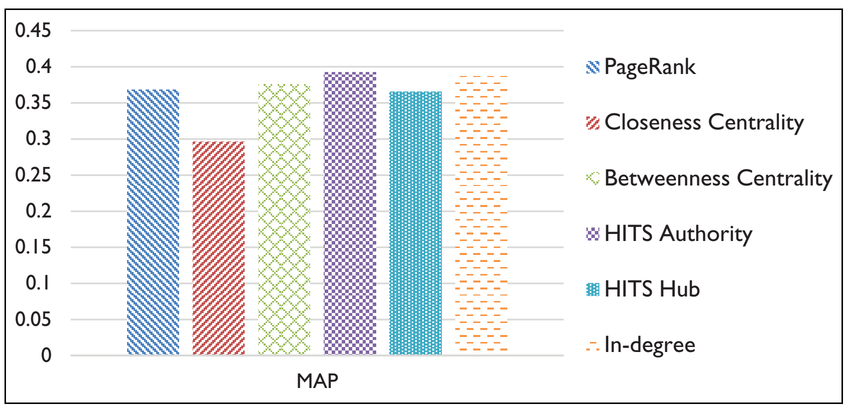


Figure 5. Comparison of different centrality measures based on AveP in banking domain.

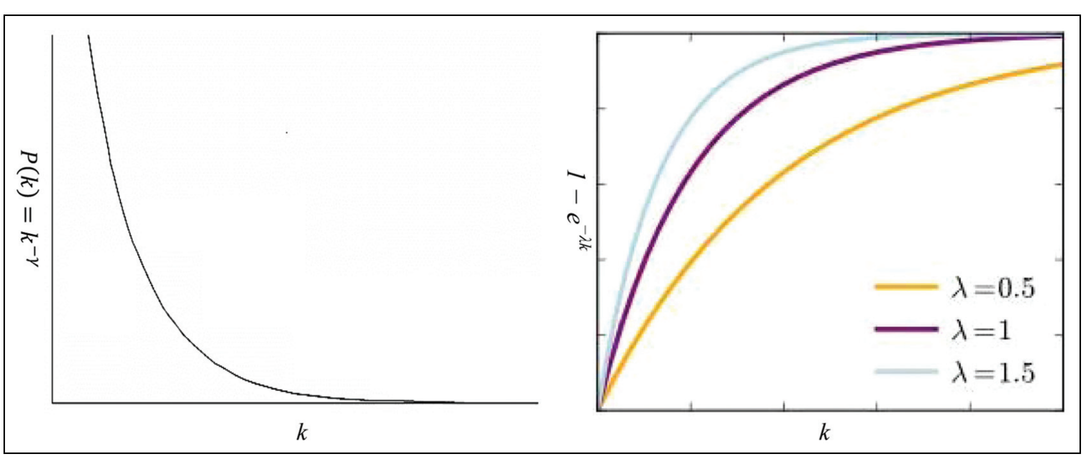


Figure 6. Comparison of in-degree distribution in scale-free networks (left) and the proposed popularity score (right) in which γ is a constant tuning factor and k is the in-degree centrality of people in social networks. The proposed popularity score is obtained by normalizing the left function by use of the right function.

$$Popularity(a) = 1 - e^{-\lambda F(a)} \tag{3}$$

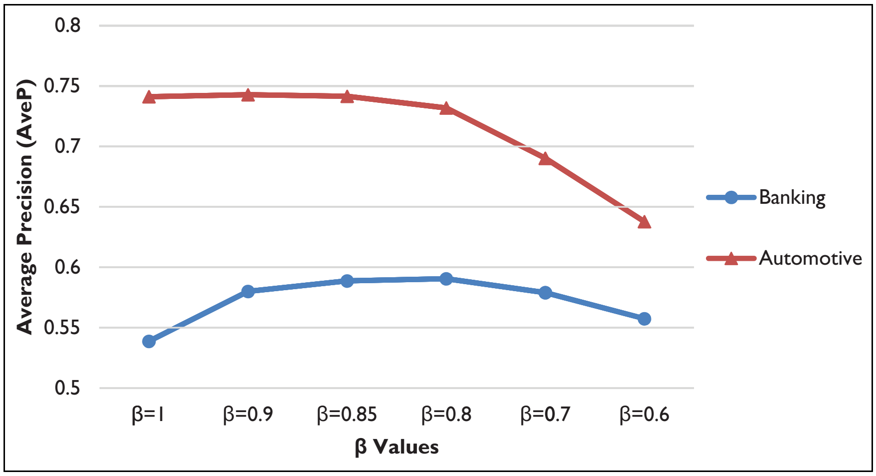


Figure 7. The performance of OLFinder system with various β values measured by AveP criteria.

where *F(a)* is the in-degree centrality (or the number of followers) of the user *a* and λ is a constant factor that is used for tuning the popularity score of the user as shown in Figure 6 (right). Comparison of the two sides of Figure 6 shows that as the in-degree centrality of users (k) grows, the popularity of the users (equation (3)) grows exponentially. In other words, the popularity equation is a normalized score which is obtained based on in-degree centrality and can be easily tuned by manipulating the constant λ.

Finally, the opinion leadership score of a user *a* in the domain *d* is calculated as a linear combination of the two scores as below:

$$Leadership(a, d) = \beta \times Competency(a, d) + (1 - \beta) \times Popularity(a) \tag{4}$$

where β is a scaling parameter between 0 and 1 for balancing the effect of the user's competency and popularity scores on his/her leadership score. The best value for parameter β has been obtained experimentally. The performance of the system with different β values on the training data has been measured by using the average precision score and the best β value has been selected for the evaluation of the system by the test set. As any learning method this approach leaves system vulnerable to variations in dataset. For example if the test dataset is significantly different from training data, one could expect significant drop in the performance of the system. In case the test data is radically different, then a new training set could be created and system parameters including β could be re-trained. However, in this case training and test data are quite similar. In our experiments depicted in Figure 7 and explained in the next section, it seems the optimum value of β should be closer to 1 (i.e. $0.5 < \beta < 1$). Note that in equation (4), it is assumed that the competency and popularity scores are normalized (i.e. between zero and one).

4. Experimental results

Most social networking sites provide very limited access to their users' information. For example Facebook's data is not accessible and Twitter provides limited access to its data through an API that allows only 180 calls every 15 minutes. Therefore creation of a reliable dataset for evaluation of any algorithm is a big challenge. However, a valuable standard dataset is created by the organizers of RepLab task in CLEF2014 [17]. The dataset contains a set of relevance judgments which has been obtained through a methodical process. This collection has been widely adapted by the researchers in this field. Therefore, we have also chosen this dataset as a platform to evaluate and compare our method with the other state-of-the-art methods.

The dataset consists of nearly 7500 English and Spanish Twitter profiles which are categorized into two broad domains: automotive and banking. Every profile has at least 1000 followers and at the crawling time, the last 600 published tweets of each profile were crawled. The dataset is split into two training and test sets that contain around 33% and 67% of the profiles, respectively. The training set consists of 1185 automotive and 1315 banking profiles. The test set contains 2345 and 2500 profiles from automotive and banking domains, respectively. Table 2 shows some useful features of the dataset.

Table 2. RepLab 2014 dataset features.

Feature	Description
Tweet_id	The related tweet id
Profile_id	The related profile id
Domain_id	Domain of the profile
Tweet_url	Tweet's URL
Language	Tweet's language
Timestamp	Tweet's published time

The evaluations are carried out based on manual relevance judgments provided by reputation experts. The outputs are stored in the standard TREC format and the traditional information retrieval criteria (AveP, R-Precision and P@N) are used to evaluate the performance of each algorithm.

The dataset contains approximately 4.5 million tweets for 7491 Twitter profiles. We downloaded the tweets directly from Twitter; Table 3 contains some properties of the crawled collection:

Twitter's standard API is used to get the number of the followers for each profile. Furthermore, because of the limitations of the Twitter API, we developed a tool to download the HTML version of each tweet and extract the tweets' text, retweet count and favourite count.

The tweet messages are stored in TREC format and indexed in Terrier platform1. In the retrieval step of our algorithm we used the PL2 [57] implementation of DFR model in the Terrier (i.e. creating *R(t,d)*). For the topic extraction step, we took advantage of the LDA implementation of Stanford Topic Modelling Toolbox2.

In addition, the performances of a number of other algorithms on this dataset have been compared with our proposed method. These algorithms are chosen because they solve the same problem as our algorithm does. They are listed in Table 4; an abbreviation is assigned to each algorithm that will be used in the rest of the paper for referencing purposes. As explained, we chose the in-degree (the number of followers of Twitter users) as a measure of importance and centrality of a user and also it is used as a base line in our comparisons.

In order to determine the best value for the β parameter in formula $Leadersh(\mathrm{a,d}) = \beta \times \mathrm{Competency}(\mathrm{a,d}) + (1-\beta) \times \mathrm{Popularity}(\boldsymbol{a})$ (4), we ran OLFinder with different values of β on the training set of the collection. Figure 7 depicts how the AveP criteria affected by various values of the β parameter.

As can be observed in Figure 7, a value of β which is about 0.85 produces the most optimal results on AveP (Average Precision) criteria. Therefore, we used this value for equation (4) throughout our experiments. Also, as suggested by Figure 7, it seems competency ($\beta = 0.85$) plays a more important role in overall performance than popularity ($1–\beta = 0.15$). In other words, a very popular user in a social network may not be considered as an opinion leader, if he/she has not enough competencies in that domain.

For comparing our algorithm with others we have used several widely used measures such as Precision and Recall, Average Precision (AveP) and P@N. Precision and Recall measures are defined as [58]:

$$Precision = \frac{Number\ of\ OLs\ found\ by\ the\ algorithm}{Number\ of\ people\ found\ by\ the\ algorithm} \tag{5}$$

$$Recall = \frac{Number\ of\ OLs\ found\ by\ the\ algorithm}{Number\ of\ OLs\ in\ the\ test\ collection} \tag{6}$$

Also, AveP measures the overall performance of an algorithm as defined in equation (7):

$$AveP = \frac{\sum_{k=1}^{n} P(k) \times Rel(k)}{Number\ of\ OLs\ in\ the\ test\ collection} \tag{7}$$

where, *P(k)* is the precision at cut-off *k* and *Rel(k)* is 1 if the people at rank *k* is an opinion leader and 0 otherwise.

Figure 8 shows a comparison of the performance of the systems on the automotive test dataset based on precision-recall measure. Figure 9 demonstrates the same for the banking domain.

Figure 10 shows a comparison of the performances of all of the systems on the both domains as measured by AveP criterion.

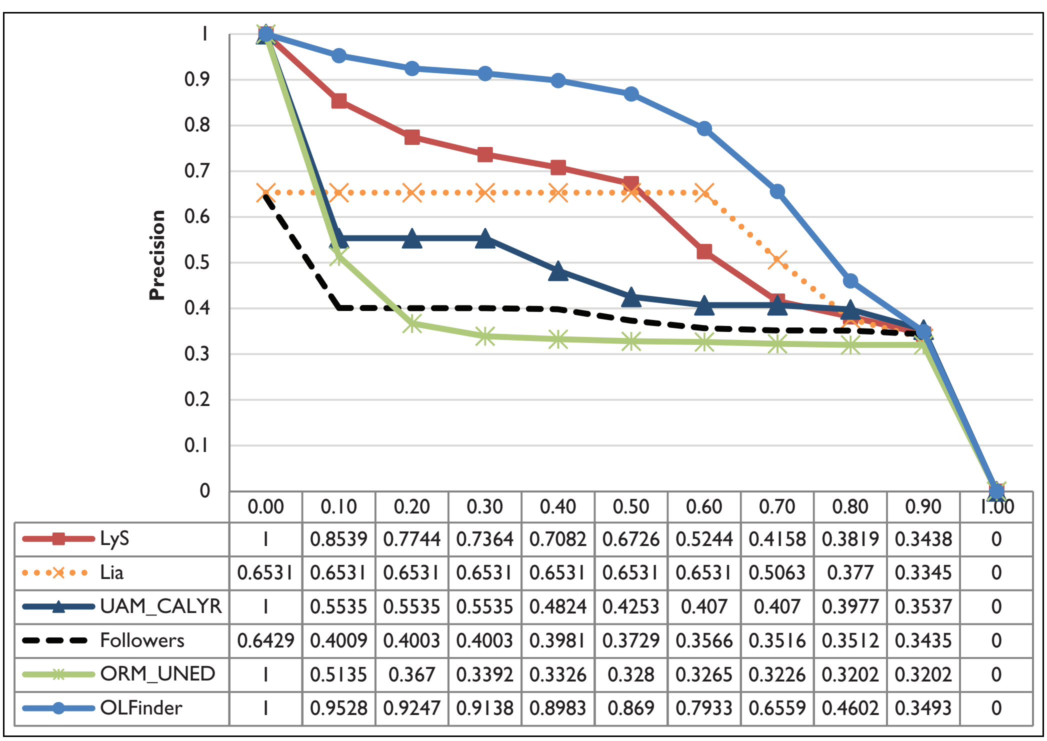
As seen in Figures 8 and 9, OLFinder outperforms all other systems in most recall values. On the automotive domain, OLFinder overall performs better in almost all recall levels and in banking it outperforms other systems until 0.7 recall

Table 3. Some statistics from the crawled collection.

Description	Value
Total number of profiles	7491
Total number of tweets	4,486,868
Number of English tweets	3,192,787
Number of Spanish tweets	1,211,714
Number of unknown language tweets	82,367
Number of tweets not crawled by our crawler	127,255
Tweets starting date	24 May 2010
Tweets ending date	8 February 2014

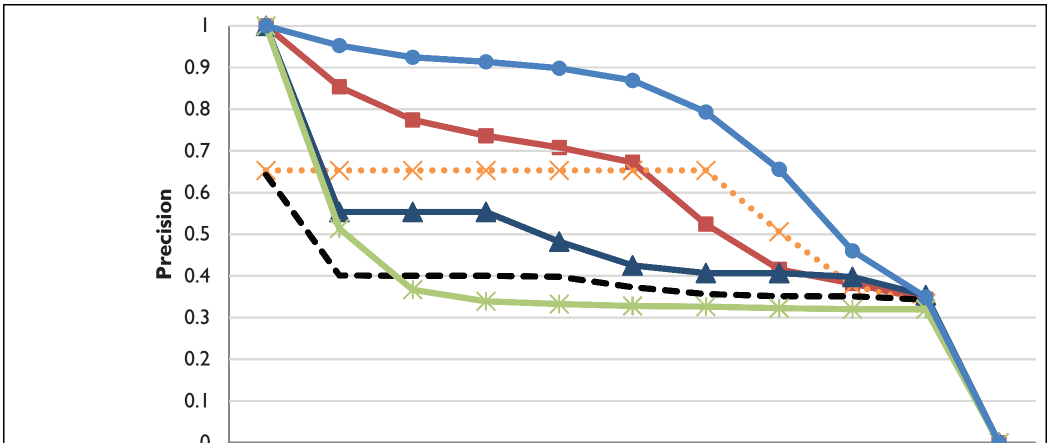
Table 4. The algorithms used in our experiments.

Reference	Abbreviation	Description
[43]	Lia	Hidden Markov Model (HMM) and Poisson classifiers are used to mark tweets as opinionated or not opinionated. Then the probability of being an 'opinion maker' is calculated based on the labels that are assigned to the user's tweets.
[51]	ORM_UNED	Tweets' text POS tags, number of hashtags and mentions, number of links, number of emoticons, number of followers and retweet speed are used in different weighting schemes
[42]	LyS	A Lib-Linear classifier uses biography that is created for each user. They use some meta-information like: listed count, favourites count, friends count, followers count, etc.
[49]	UAM_CLYR	Stylistics and behavioural attributes of users are used to compute a confidence score. Support Vector Machine and Markov Random Fields are used to rank the users



	0.00	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00
LyS	1	0.8539	0.7744	0.7364	0.7082	0.6726	0.5244	0.4158	0.3819	0.3438	0
Lia	0.6531	0.6531	0.6531	0.6531	0.6531	0.6531	0.6531	0.5063	0.377	0.3345	0
UAM_CALYR	1	0.5535	0.5535	0.5535	0.4824	0.4253	0.407	0.407	0.3977	0.3537	0
Followers	0.6429	0.4009	0.4003	0.4003	0.3981	0.3729	0.3566	0.3516	0.3512	0.3435	0
ORM_UNED	1	0.5135	0.367	0.3392	0.3326	0.328	0.3265	0.3226	0.3202	0.3202	0
OLFinder	1	0.9528	0.9247	0.9138	0.8983	0.869	0.7933	0.6559	0.4602	0.3493	0

Figure 8. Precision-recall comparison of the algorithms in automotive domain.



	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
LyS	1	0.8539	0.7744	0.7364	0.7082	0.6726	0.5244	0.4158	0.3819	0.3438	0
Lia	0.6531	0.6531	0.6531	0.6531	0.6531	0.6531	0.6531	0.5063	0.377	0.3345	0
UAM_CALYR	1	0.5535	0.5535	0.5535	0.4824	0.4253	0.407	0.407	0.3977	0.3537	0
Followers	0.6429	0.4009	0.4003	0.4003	0.3981	0.3729	0.3566	0.3516	0.3512	0.3435	0
ORM_UNED	1	0.5135	0.367	0.3392	0.3326	0.328	0.3265	0.3226	0.3202	0.3202	0
OLFinder	1	0.9528	0.9247	0.9138	0.8983	0.869	0.7933	0.6559	0.4602	0.3493	0

Figure 9. Precision-recall comparison of the algorithms in banking domain.

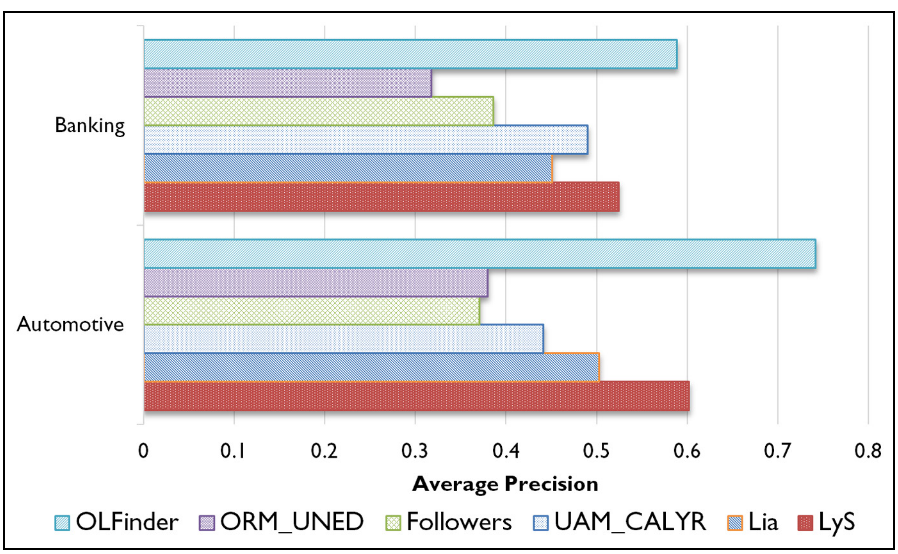


Figure 10. AveP comparison of the runs in both domains.

level. On AveP scale (Figure 10) OLFinder is considerably better than other systems (it is 12% and 23% better that the second best system in banking and automotive domains, respectively).

Another evaluation criterion is Precision at N cutoff (or P@N) which measures the precision of an algorithms when a specific number of opinion leaders (N) have been retrieved. This measure is a good indicator of the algorithms' performance when we need to pick N people from a social network (e.g. to start advertising a product).

Figures 11 and 12 depict the performance of the systems at different cutoff values (5, 10, 15, 20, 30, 100, 200, 500 and 1000). As can been seen in these figures, for small N ($N < 20$) OLFinder, LyS and ORM_UNED systems have comparable P@N values. However, as N grows ($N \geq 20$) OLFinder outperforms others. OLFinder maintains a precision of

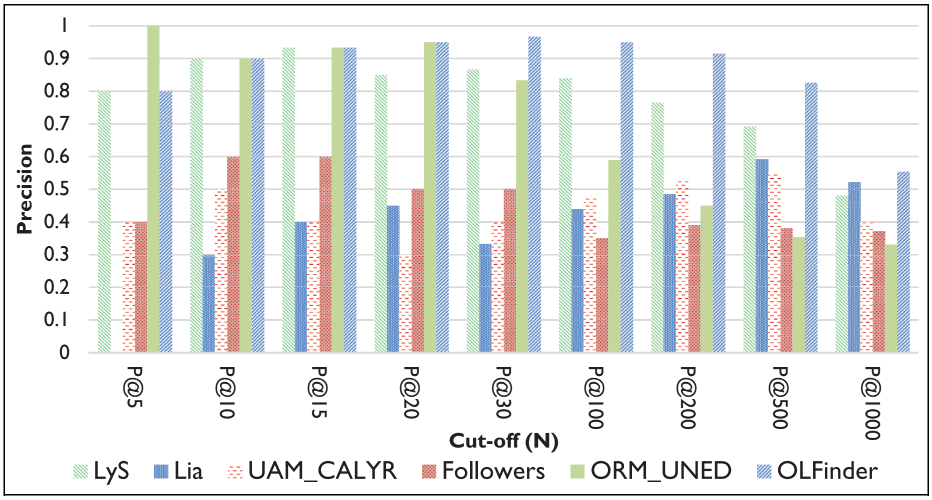


Figure 11. P@N comparison of the algorithms in automotive domain.

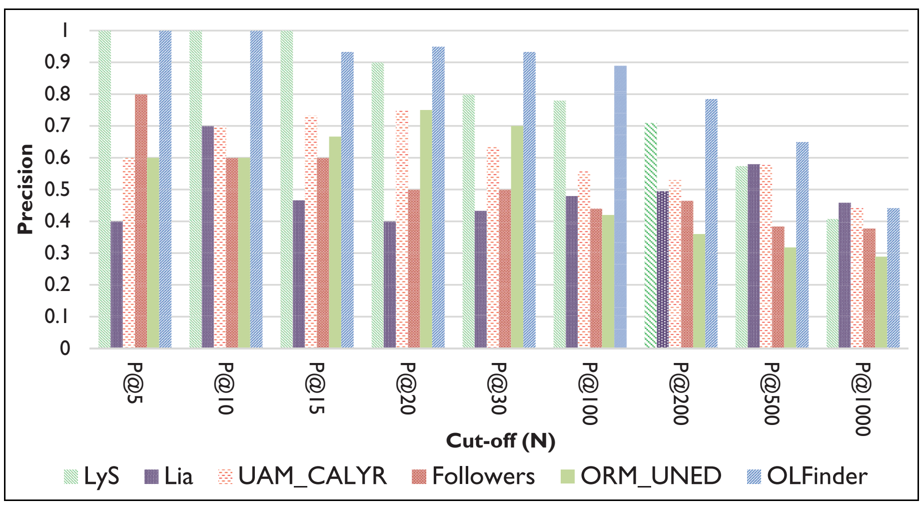


Figure 12. P@N comparison of the algorithms in banking domain.

above 80% for a list of up to 200 opinion leaders while for most of the other systems the precision drops sharply after retrieving the 30th opinion leader. At the cutoff of 500 our system is the only one with a precision above 60%. At around 1000 our system performs as good as the best systems. This shows how well our algorithm can furnish a list of opinion leaders.

As suggested by [33, 34, 52] the graph structure alone is not sufficient for effective leader finding. However, when we combined the number of followers with competency it helped the overall performance. Also, the comparisons state that all of the algorithms performed much better in the automotive domain compared with the banking domain. This suggests that domains do not lend themselves equally to the opinion leaders finding algorithms. Perhaps ambiguity of vocabulary plays a role in this.

5. Discussion

In this section, we would like to analyse the proposed algorithm in more detail. Specifically the following two challenging points will be discussed:

- As the presented competency score in formula 1 shows, the most challenging part of the algorithm is topic detection. So, it is necessary to analyse the effect of T in the proposed competency score.

Table 5. Comparison of the proposed algorithm using another topic extraction method based on AveP.

	OLFinder	Modified OLFinder	Difference
Automotive	0.7416	0.7047	5%
Banking	0.5887	0.3961	33%

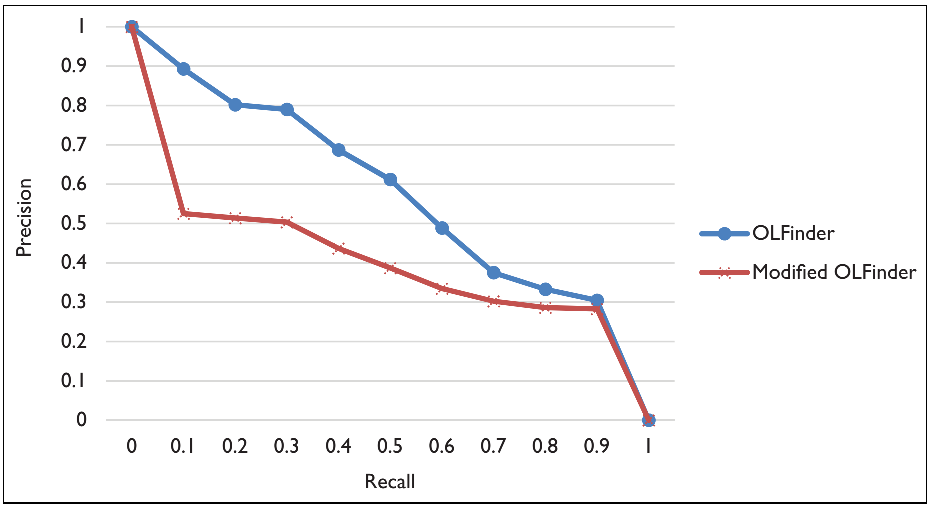


Figure 13. Comparison of OLFinder with its modified *tf-idf* version based on precision-recall in banking domain.

- What is the effect of tuning w_t parameter on performance? In other words, what is the effect of w_t in the proposed competency score?

In order to identify the effect of the quality of the extracted topics (T in formula 1) on efficiency of the proposed algorithm, another experiment was carried out using a different topic extraction method. Instead of LDA, a basic keyword-based topic creation method is used that works based on *tf-idf* (term frequency-inverted document frequency). *Tf-idf* is one of the most widely used weighting methods for determining most important keywords of a given document collection in the text mining field [59, 60]. The setup of the new experiment with *tf-idf* as a different method in the topic extraction step is as following: Let K_{di} be the set of top N_k keywords ranked by the *tf.idf* weighting method for each domain d_i. Then the set of topics for domain d_i is named T_{di} and calculated as below:

$$T_{di} = K_{di} - K_{dj} : for\ j \neq i \tag{8}$$

Table 5 and Figures 13 and 14 show the AveP and precision-recall comparison of the proposed algorithm and its modified *tf-idf* version, respectively. The comparison clearly demonstrates that the proposed algorithm is highly dependent on the method of creating the topic set T. However, this dependence is not uniform. It seems some domains such as banking are more affected by the choice of the topics and topic extraction method than others. This could be related to the ambiguity of vocabulary used in banking domain.

Furthermore, another experiment is carried out to find an answer for the second question, the effect of tuning of w_t parameter. Let's assume that there is no training set available; which means no previous knowledge is available for the topic set T and w_t cannot be estimated, therefore all the terms would have the same weight. So, in our new experiment, we considered $w_t=1$ for all $t \in T(d_i)$ (all topics are treated equally). Figure 15 depicts the precision-recall curves for this new system on banking and automotive domains.

Table 6 shows the AveP score of OLFinder system with and without training. It is obvious that the algorithm's performance is highly dependent on the quality of training. The performance highly decreases in both domains (–14% and –37%). As before, it can be seen that the banking domain is more sensitive than automotive. Since the banking domain is closely related with other domains such as economics, a broader range of topics are discussed in the banking domain.

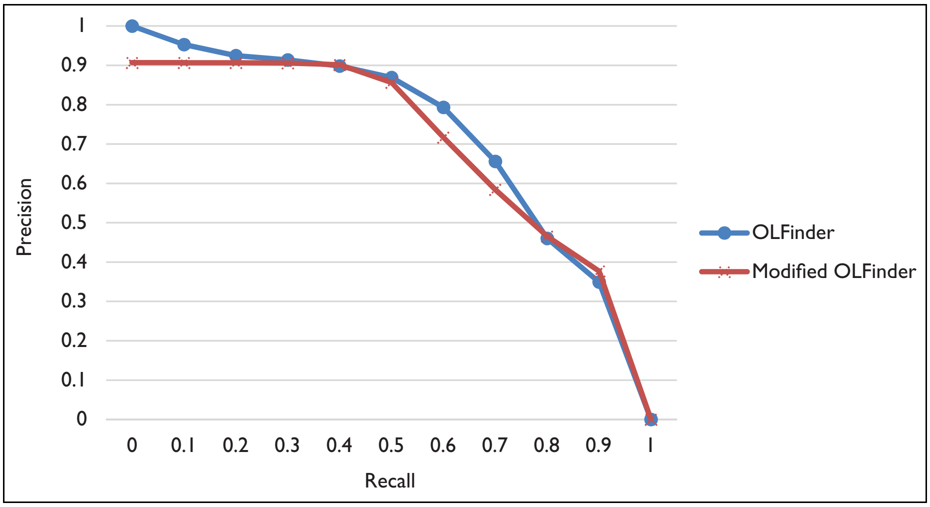


Figure 14. Comparison of OLFinder with its modified *tf-idf* version based on precision-recall in automotive domain.

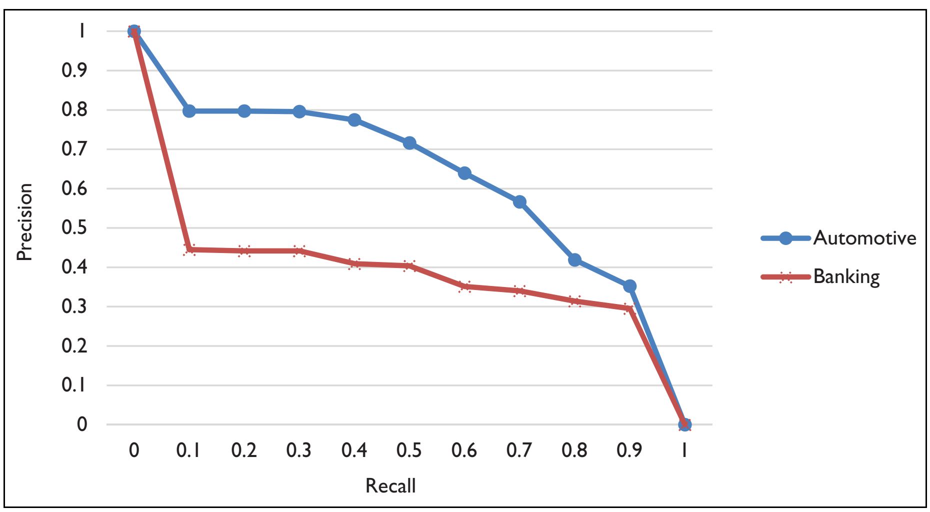


Figure 15. Precision-recall of OLFinder with and without training in both domains.

Table 6. Comparison of the proposed algorithm with/without usage of training set based on AveP.

	OLFinder with training	OLFinder without training	Difference percentage
Automotive	0.7416	0.6361	−14%
Banking	0.5887	0.3714	−37%

This could increase the ambiguity of topics and it would be more difficult to estimate the importance of topics for this domain. So, a more sophisticated training perhaps would be needed to be able to deal with this level of ambiguity. As a result, one needs to be more cautious when creating the training dataset. In practical applications, one could monitor the topic extraction part of the proposed algorithm and when feels there is major change in the topics set (i.e. many new topics added), can re-do the training using a new set of training data.

Furthermore, topic extraction is the time consuming part of the algorithm but this step can be computed offline and periodically. In this case, since the calculation of popularity and competency scores are very simple and fast, the algorithm will perform much faster and the end-users of a practical system will benefit from speedy turnaround of their result.

6. Conclusion and future works

In this paper, we introduced an algorithm for finding opinion leaders in social networks. The proposed algorithm first extracts the hot topics of discussion in a domain in the social network, then calculates two scores; a competency score based on those hot topics and a popularity score based on users' number of in-links. Finally, it computes the influence of a user based on a linear combination of the competency and the popularity scores. Our experimental evaluation of the algorithm showed that it outperforms the other existing algorithms in terms of average precision and precision-recall. In particular, P@N measure shows our algorithm produces lists of opinion leaders with higher precision for *N* as high as 500.

Our experiments showed that, as expected, the proposed algorithm is very dependent on the topics set that are extracted in the first step of the algorithm. Our future work would be to use other topic extraction methods and analyse their effect on the final performance of the algorithm. Also, it is worth mentioning that we used the number of retweets as a measure of the users' popularity. However, according to our experiments this feature was not as helpful; and we need to investigate this further in future. We can study using other methods such as influence maximization techniques that are better indicators of the users' strategic location in the social network.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Notes

1. Terrier IR Platform version 3.5: http://terrier.org.
2. Stanford Topic Modeling Toolbox: http://nlp.stanford.edu/software/tmt/tmt-0.4.

References

[1] West JD, Jensen MC, Dandrea RJ, Gordon GJ and Bergstrom CT. Author-level Eigenfactor metrics: Evaluating the influence of authors, institutions, and countries within the social science research network community. *Journal of the American Society for Information Science and Technology* 2013; 64(4): 787–801.
[2] Soulier L, Jabeur LB, Tamine L and Bahsoun W. On ranking relevant entities in heterogeneous networks using a language-based model. *Journal of the American Society for Information Science and Technology* 2013; 64(3): 500–515.
[3] Yan E, Ding Y and Sugimoto CR. P-Rank: An indicator measuring prestige in heterogeneous scholarly networks. *Journal of the American Society for Information Science and Technology* 2011; 62(3): 467–477.
[4] Khuri S and Wuchty S. Essentiality and centrality in protein interaction networks revisited. *BMC Bioinformatics* 2015; 16(1): 1–8.
[5] Trusov M, Bodapati AV and Bucklin RE. Determining influential users in internet social networks. *Journal of Marketing Research* 2010; 47(4): 643–658.
[6] Heidemann J, Klier M and Probst F. Identifying key users in online social networks: A PageRank based approach. In: *31st International Conference on Information Systems (ICIS)*. St. Louis, MO: ICIS, 2010, pp. 1–21.
[7] Cheung CMK and Lee MKO. A theoretical model of intentional social action in online social networks. *Decision Support Systems* 2010; 49(1): 24–30.
[8] Algesheimer R and Wangenheim Fv. A network based approach to customer equity management. *Journal of Relationship Marketing* 2006; 5(1): 39–57.
[9] Oestreicher-Singer G and Zalmanson L. 'Paying for content or paying for community?' The effect of social involvement on subscribing to media web sites. In: *Proceedings of International Conference on Information Systems (ICIS)*, 2009, pp. 1–17.
[10] Berger J and Strathearn B. Who matters online: Measuring influence, evaluating content and countering violent extremism in online social networks. In: *The International Centre for The Study of Radicalisation and Political Violence*, 2013, pp. 1–20.
[11] Wingyan C. Categorizing temporal events: A case study of domestic terrorism. In: *Intelligence and Security Informatics (ISI), 2012 IEEE International Conference*, 2012, pp. 159–161.
[12] Rogers EM. *Diffusion of Innovations*. 5th ed. New York: Simon and Schuster, 2003.
[13] Keller E and Berry J. *The Influentials: One American in Ten Tells the Other Nine How to Vote, Where to Eat, and What to Buy*. New York: Simon and Schuster, 2003.
[14] Lazarsfeld PF, Berelson B and Gaudet H. *The People's Choice: How the Voter Makes Up His Mind in a Presidential Campaign*. New York: Columbia University Press, 1948.

[15] Weimann G. *The Influentials: People Who Influence People*. Albany, NY: State University of New York Press, 1994.
[16] Katz E and Lazarsfeld PF. *Personal Influence, the Part Played by People in the Flow of Mass Communications*. New York: Collier-Macmillan, 1970.
[17] Amigó E, Carrillo-de-Albornoz J, Chugur I, Corujo A, Gonzalo J, Meij E et al. Overview of RepLab 2014: Author Profiling and Reputation Dimensions for Online Reputation Management. In: Kanoulas E, Lupu M, Clough P, Sanderson M, Hall M, Hanbury A et al. (eds) *Information Access Evaluation – Multilinguality, Multimodality, and Interaction*. New York: Springer, 2014.
[18] Troldahl VC. A field test of a modified ''two-step flow of communication'' model. *Public Opinion Quarterly* 1966; 30(4): 609–623.
[19] Dichter E. How word-of-mouth advertising works. *Harvard Business Review* 1966; 44(6): 147–160.
[20] Domingos P and Richardson M. Mining the network value of customers. In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. San Francisco, CA: ACM, 2001, pp. 57–66.
[21] Freeman LC. Centrality in social networks conceptual clarification. *Social Networks* 1978; 1(3): 215–239.
[22] Bonacich P. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology* 1972; 2(1): 113–120.
[23] Page L, Brin S, Motwani R and Winograd T. *The PageRank Citation Ranking: Bringing Order to the Web*. Stanford, CA: Stanford InfoLab, 1999, pp. 1–17.
[24] Kleinberg JM. Authoritative sources in a hyperlinked environment. *Journal of the ACM* 1999; 46(5): 604–632.
[25] Watts DJ and Dodds PS. Influentials, networks, and public opinion formation. *Journal of Consumer Research* 2007; 34(4): 441–458.
[26] Song X, Chi Y, Hino K and Tseng B. Identifying opinion leaders in the blogosphere. In: *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. Lisbon, Portugal, ACM, 2007, pp. 971–974.
[27] Probst F, Grosswiele L and Pfleger R. Who will lead and who will follow: Identifying influential users in online social networks. *Business & Information Systems Engineering* 2013; 5(3): 179–193.
[28] Bakshy E, Hofman JM, Mason WA and Watts DJ. Everyone's an influencer: quantifying influence on twitter. In: *Proceedings of the fourth ACM international conference on Web search and data mining*. Hong Kong, China, ACM, 2011, pp. 65–74.
[29] Granovetter M. Threshold models of collective behavior. *American Journal of Sociology* 1978; 8(6): 1420–1443.
[30] Goldenberg J, Libai B and Muller E. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters* 2001; 12(3): 211–223.
[31] Kempe D, Kleinberg J and Tardos E. Maximizing the spread of influence through a social network. In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. Washington, DC, ACM, 2003, pp. 137–146.
[32] Rahimkhani K, Aleahmad A, Rahgozar M and Moeini A. A fast algorithm for finding most influential people based on the linear threshold model. *Expert Systems with Applications* 2015; 42(3): 1353–1361.
[33] Cha M, Haddadi H, Fabricio B and Gummadi KP. Measuring user influence in Twitter: the million follower fallacy. In: *International AAAI Conference on Weblogs and Social Media, Association for the Advancement of Artificial Intelligence*, 2010, pp. 10–17.
[34] Kwak H, Lee C, Park H and Moon S. What is Twitter, a social network or a news media? In: *Proceedings of the 19th international conference on World wide web*. Raleigh, NC, ACM, 2010, pp. 591–600.
[35] Weng J, Lim E-P, Jiang J and He Q. TwitterRank: finding topic-sensitive influential twitterers. In: *Proceedings of the third ACM international conference on Web search and data mining*. New York, NY, ACM, 2010, pp. 261–270.
[36] Cataldi M, Mittal N and Aufaure M-A. Estimating domain-based user influence in social networks. In: *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. Coimbra, Portugal, ACM, 2013, pp. 1957–1962.
[37] Ding Z, Jia Y, Zhou B and Yi H. Mining topical influencers based on the multi-relational network in micro-blogging sites. *Communications, China* 2013; 10(1): 93–104.
[38] Gruhl D, Guha R, Liben-Nowell D and Tomkins A. Information diffusion through blogspace. In: *Proceedings of the 13th international conference on World Wide Web*. New York, NY, ACM, 2004, pp. 491–501.
[39] Ma N and Liu Y. SuperedgeRank algorithm and its application in identifying opinion leader of online public opinion supernetwork. *Expert Systems with Applications* 2014; 41(4): 1357–1368.
[40] Chen W, Wang Y and Yang S. Efficient influence maximization in social networks. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. Paris, France, ACM, 2009, pp. 199–208.
[41] Subbian K, Aggarwal C and Srivastava J. Content-centric flow mining for influence analysis in social streams. In: *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. San Francisco, CA, ACM, 2013, pp. 841–846.
[42] Vilares D, Hermo M, Alonso MA, Gómez-Rodríguez C and Vilares J. 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, pp. 1468–1478.
[43] Cossu J-V, Janod K, Ferreira E, Gaillard J and El-Bèze M. LIA@Replab 2014: 10 methods for 3 tasks. In: *Proceedings of the Fifth International Conference of the CLEF initiative*, 2014, pp. 1458–1467.

[44] Salton G and Buckley C. Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 1988; 24(5): 513–523.

[45] Torres-Moreno J, El-Beze M and Bellot P. Bechet, *Opinion detection as a topic classification problem. Textual Information Access*. Chichester: John Wiley & Sons, 2013.

[46] 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.

[47] Bahl R, Bakis R, De Souza P and Mercer R. Obtaining candidate words by polling in a large vocabulary speech recognition system. In: *International Conference on Acoustics, Speech, and Signal Processing*, IEEE, 1988, pp. 489–492.

[48] Mikolov T, Chen K, Corrado G and Dean J. Efficient estimation of word representations in vector space. In: *International Conference on Learning Representations*, 2013, pp. 1–12.

[49] Villatoro-Tello E, Ramírez-de-la-Rosa G, Sánchez-Sánchez C, Jiménez-Salazar H, Luna-Ramírez WA and Rodríguez-Lucatero C. UAMCLyR at RepLab 2014: Author Profiling Task. In: *Proceedings of the Fifth International Conference of the CLEF Initiative*, 2014, pp. 1547–1558.

[50] Villatoro E, Juárez A, Montes M, Villaseñor L and Enrique Sucar L. Document ranking refinement using a Markov random field model. *Natural Language Engineering* 2012; 18(02): 155–185.

[51] Jesus J, Lomena M and Lopez Ostenero F. UNED at CLEF RepLab 2014: Author Profiling. In: *Proceedings of the Fifth International Conference of the CLEF Initiative*, 2014, pp. 1537–1549.

[52] Kitsak M, Gallos LK, Havlin S, Lijeros F, Muchnik L, Stanley EH, et al. Identification of influential spreaders in complex networks. *Nature Physics* 2010; 6(11): 888–893.

[53] Amati G and Rijsbergen CJV. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems* 2002; 20(4): 357–389.

[54] Blei DM, Ng AY and Jordan MI. Latent dirichlet allocation. *Journal of Machine Learning Research* 2003; 3: 993–1022.

[55] Zhou T, Medo M, Cimini G, Zhang Z-K and Zhang Y-C. Emergence of Scale-Free Leadership Structure in Social Recommender Systems. *PLoS One* 2011; 6(7): e20648.

[56] Freund R, Wilson W and Sa P. *Regression Analysis*. Waltham, MA: Academic Press, 2006.

[57] He B and Ounis I. Term frequency normalisation tuning for BM25 and DFR models. In: *Proceedings of the 27th European conference on Advances in Information Retrieval Research*. Santiago de Compostela, Spain, Springer-Verlag, 2005, pp. 200–214.

[58] Singhal A. Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 2001; 24(4): 35–43.

[59] Lee K-P, Kim H-G and Kim H-J. A social inverted index for social-tagging-based information retrieval. *Journal of Information Science* 2012; 38(4): 313–332.

[60] Park J, Kim J and Lee J-H. Keyword extraction for blogs based on content richness. *Journal of Information Science* 2014; 40(1): 38–49.