A Latent Representation Model for Sentiment Analysis in Heterogeneous Social Networks

Debora Nozza[1], Daniele Maccagnola[1(✉)], Vincent Guigue[2], Enza Messina[1], and Patrick Gallinari[2]

[1] DISCo, University of Milano-Bicocca, Milano, Italy
Daniele.Maccagnola@disco.unimib.it
[2] LIP6, Université Pierre et Marie Curie - UPMC, Paris, France

Abstract. The growing availability of social media platforms, in particular microblogs such as Twitter, opened new way to people for expressing their opinions. Sentiment Analysis aims at inferring the polarity of these opinions, but most of the existing approaches are based only on text, disregarding information that comes from the relationships among users and posts. In this paper we consider microblogs as heterogeneous networks and we use an approach based on latent representation of nodes to infer, given a specific topic, the sentiment polarity of posts and users at the same time. The experimental investigation show that our approach, by taking into account both content and relationship information, outperforms supervised classifiers based only on textual content.

1 Introduction

"What other people think" has always been an important piece of information during the decision-making process [1], and this lead to a growing need of methods that could infer the opinion of people. The field of Sentiment Analysis (SA) aims to define automatic tools able to extract opinions and sentiments from texts written in natural language. The growing availability and popularity of social media platforms, such as online review sites, personal blogs and microblogs, opened the way to new opportunities for understanding the opinion of people. Companies, advertisers and political campaigners are seeking ways to analyze the sentiments of users through social media platform on their products, services and policies.

Several works in Sentiment Analysis, however, suffer of important limitations. Most prior work on SA applied to social network data has focused on understanding the sentiments of individual documents (posts) [2–6].

The problem of inferring the sentiment of the users has been only recently addressed by some authors [7,8]. Smith et al. [9] and Deng et al. [10] study both post-level and user-level sentiments, assuming that a users sentiment can be estimated by aggregating the sentiments of all his/her posts. Although the sentiment of users is correlated with the sentiment expressed in their posts, such simple aggregation can often produce incorrect results, because sentiment

C. Canal and A. Idani (Eds.): SEFM 2014 Workshops, LNCS 8938, pp. 201–213, 2015.
DOI: 10.1007/978-3-319-15201-1_13

extracted from short texts such as tweets (which in Twitter are limited to 140 characters) will generally be very noisy and error prone.

All of these approaches do not consider that microblogs are actually networked environments. Early studies for overcoming this limitation exploit the principle of homophily [11] for dealing with user connections. This principle could suggest that users connected by a personal relationship may tend to hold similar opinions. According to this social principle, friendship relations have been considered in few recent studies.

In [12], the authors showed that considering friendship connections is a weak assumption for modelling homophily, as two friends might not share the same opinion about a given topic. Instead, they proposed to use approval relationships (e.g. in Twitter represented by "retweets" and in Facebook represented by "like") which better represent the sharing of ideas between two users. However, in [12], the sentiment of the posts is used to infer the sentiment of the users, but not vice versa.

In order to overcome this limitation, in our approach we consider social network data as a heterogeneous network, whose nodes and edges can be of different types. Inspired by the work of Jacob et al. [13], who introduced an innovative method for classifying nodes in heterogeneous networks, we propose an approach that can infer at the same time the sentiment relative to each post and the sentiment relative to each user about a specific topic. This algorithm learns a latent representation of the network nodes so that all the nodes will share a common latent space, whatever their type is. This ensures that the sentiment of the posts can influence the sentiment of the users, and in the same way the sentiment of the posts is influenced by that of the users.

For each node type, a classification function will be learned together with the latent representation, which takes as input a latent node representation and computes the sentiment polarity (positive or negative) for the corresponding node.

The paper is structured as follows. In Sect. 2 we introduce the basic concepts that are used in our model, while in Sect. 3 we describe the model and the learning algorithm. In Sect. 4 we test our approach on a case study, a Twitter network about the topic 'Obama', and finally in Sect. 5 conclusions are drawn.

2 Preliminaries

In this section we introduce some preliminary concepts that will be used in our model. First, we give a definition of Heterogeneous Approval Network, which summarizes the structure of a social network and the information we require to determine the users' and posts' sentiment polarity. Then, we give a brief description of the techniques we use to represent and treat the textual data available in the posts.

2.1 Heterogeneous Approval Network

Following the work in [12], we assume that a user who approves a given message will share the same opinion with higher probability. Pozzi et al. defined as "approval network" a network where the nodes represent users of a social network, and a directed arc connects a user who has approved a post to the original author of that post. The most known example of approval relationship is the "retweet" feature in Twitter, which allows a user to share another user's post.

We start from the definition of "approval graph" in order to give a formal structure to our data.

Definition 1. *Given a topic of interest q, a* ***Directed Approval Graph*** *is a quadruple $DAG_q = \{V_q, E_q^{VV}, \mathbf{X}_q^V, \mathbf{X}_q^E\}$, where $V_q = \{v_1, \ldots, v_n\}$ represents the set of active users; $E_q^{VV} = \{(v_i, v_j) | v_i, v_j \in V_q\}$ is the set of approval edges, meaning that v_i approved v_j's posts; $\mathbf{X}_q^E = \{k_{i,j} | (v_i, v_j) \in E_q\}$ is the set of weights assigned to approval edges, where $k_{i,j}$ indicates the number of posts of v_j approved by v_i; $\mathbf{X}_q^V = \{c_i | v_i \in V_q\}$ is the set of coefficients related to nodes, where c_i represents the total number of posts of v_i.*

Starting from a DAG_q, the weight on the arc can be normalized to better reflect the importance of an approval.

Definition 2. *Given an Approval Graph $DAG_q = \{V_q, E_q^{VV}, \mathbf{X}_q^V, \mathbf{X}_q^E\}$, a* ***Normalised Directed Approval Graph*** *is derived as a triple $\text{N-DAG}_q = \{V_q, E_q^{VV}, \mathbf{W}_q^{VV}\}$, where $\mathbf{W}_q^{VV} = \{w_{i,j} = \frac{k_{i,j}}{c_j} | k_{i,j} \in \mathbf{X}_q^E, c_j \in \mathbf{X}_q^V\}$ is the set of normalised weights of approval edges.*

The N-DAG_q represents a network with a single type of node, the users. In [12], Pozzi et al. defined a heterogeneous graph which could represent both the user-user and user-post relationships. However, the network they defined does not consider relationships among posts. In this paper, we extend their Heterogeneous Normalized Directed Approval Graph (HN-DAG_q) so that post-post relationships can be taken in account as well (Fig. 1):

Definition 3. *Given a $\text{N-DAG}_q = \{V_q, E_q^{VV}, \mathbf{W}_q^{VV}\}$, let $P_q = \{p_1, \cdots, p_m\}$ be the set of nodes representing posts about q and $E_q^{VP} = \{(v_i, p_t) | v_i \in V_q, p_t \in P_q\}$ be the set of arcs that connect the user v_i and the post p_t. Then, let $E_q^{PP} = \{(p_{t_1}, p_{t_2}) | p_{t_1}, p_{t_2} \in P_q\}$ be the set of arcs that connect a post p_{t_1} to another post p_{t_2}, and $\mathbf{W}_q^{PP} = \{w_{t_1,t_2} | (p_{t_1}, p_{t_2}) \in E_q^{PP}\}$ is the set of weights of the post-post edges. An* ***Heterogeneous Normalised Directed Approval Graph*** *is a septuple $\text{HN-DAG}_q = \{V_q, P_q, E_q^{VV}, E_q^{VP}, E_q^{PP}, W_q^{VV}, W_q^{PP}\}$.*

2.2 Vector Space Document Representation

The field of Sentiment Analysis requires the analysis of text documents, where the words occurring in a document are used to determine the opinion expressed in it. As described in the previous section, our heterogeneous network is composed

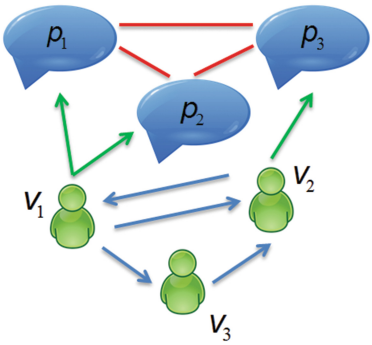

Fig. 1. Example of HN-DAG representing users and posts of a social network, connected by user-user (blue), post-post (red) and user-post (green) relationships (Color figure online).

not only by the users of a social network, but also by the textual posts every user has emitted.

For this reason, we require a way to model such text documents. The most common method applied in literature (in particular in the fields of information retrieval and text mining [14]) is the *bag of words* representation, where the words are assumed to appear independently and their order is not considered.

Given the set of posts P that are represented in our heterogeneous network, let $U = \{u_1, u_2, \ldots, u_m\}$ be the set of all the unique words occurring in P. Then, a post p_i can be represented by an m-dimensional vector $\overrightarrow{p_i}$. A usual document encoding for sentiment classification is $\text{tf}(i, u)$, which is the frequency of a word $u \in U$ in post p_i. Then, the vector representation of the post is:

$$\overrightarrow{p_i} = (\text{tf}(i, u_1), \text{tf}(i, u_2), \ldots, \text{tf}(i, u_m)) \tag{1}$$

In this work, we define the weights of the post-post edges as the value of similarity between each couple of posts. With document represented by vectors, we can measure the degree of similarity of two posts as the correlation between their corresponding vectors, which can be further quantified as the cosine of the angle between the two vectors (*Cosine Similarity*). Let $\overrightarrow{p_a}$ and $\overrightarrow{p_b}$ be the vector representation respectively of posts p_a and p_b. Their cosine similarity is computed as follows:

$$\text{similarity} = \frac{\overrightarrow{p_a} \cdot \overrightarrow{p_b}}{\|\overrightarrow{p_a}\| \|\overrightarrow{p_b}\|} = \frac{\sum_{j=1}^{l} p_{aj} \times p_{bj}}{\sqrt{\sum_{j=1}^{l} (p_{aj})^2} \sqrt{\sum_{j=1}^{l} (p_{bj})^2}} \tag{2}$$

3 Latent Space Heterogeneous Approval Model

Following the work of Jacob et al. [13], in this paper we propose a model that can, at the same time, learn the latent representation of the nodes and infer

their sentiment polarity. Differently from previous works, this model performs sentiment polarity classification on all the nodes of the network HN-DAG shown in Sect. 2.1, that means we can infer the polarity for both users and posts simultaneously.

Each of the nodes, whatever their type is, is represented by a vector space model so that all of them will share the same common latent space.

The model we propose will therefore learn the proper representation of each node, and at the same time it will learn a classification function on the latent space. This ensures that the sentiment of the posts can influence the sentiment of the users, and vice versa.

The classification function will take as input a latent node representation in order to compute the polarity (positive or negative) for the corresponding node.

The proposed approach can be summarized with the following steps:

- Each node is mapped onto a latent representation in a vector space $\mathbb{R}^Z$ where Z is the dimension of this space. This latent representation will define a metric in the $\mathbb{R}^Z$ space such that two connected nodes will tend to have a close representation, depending on the weight of the connection (*smoothness assumption*).
  The latent representation for the nodes is initialized randomly.
- A classification function for inferring the polarity of the nodes is learned on the network starting from the latent representations. Nodes with similar representations will tend to have the same sentiment polarity.

In other words, both graph and label dependencies between the different types of nodes will be captured through this learned mapping onto the latent space.

In the following we describe in details the components of the proposed approach.

Given the latent representation $z_i \in \mathbb{R}^Z$ of the node x_i, we want to predict the related sentiment y_i. We are therefore searching for a linear classification function f_θ, where θ are the parameters of the linear regression. This function is learned by minimizing the classification loss on the training data:

$$\sum_{i \in \mathcal{T}} \Delta(f_\theta(z_i), y_i) \tag{3}$$

where $\Delta(f_\theta(z_i), y_i)$ is the loss to predict $f_\theta(z_i)$ instead of the real label y_i, and $\mathcal{T}$ is the training set.

In order to make sure that connected nodes have similar representations, we introduce the other following loss:

$$\sum_{i,j:w_{i,j} \neq 0} w_{i,j} \|z_i - z_j\|^2 \tag{4}$$

which forces the approach of the latent representation of connected nodes. The complete loss function is the aggregation of the classification and similarity loss:

$$L(z, \theta) = \sum_{i \in \mathcal{T}} \Delta(f_\theta(z_i), y_i) + \lambda \sum_{i,j:w_{i,j} \neq 0} w_{i,j} \|z_i - z_j\|^2 \tag{5}$$

This loss will allow us to find the best classification function and, at the same time, improve the meanings of the latent space.

In the original work of [13], the authors fixed a value of λ for all the possible edges. In our work, we decided to model the problem with three different parameters to give different weights to different types of edge, instead of a single parameter λ. Three new parameters are introduced: λ_{pp} refers to the edges connecting two posts, λ_{pv} refers to the edges connecting a post to a user and λ_{vv} refers to the edges connecting two users.

Following this idea, the loss function in Eq. 5 can be rewritten as follows:

$$
\begin{aligned}
L(z,\theta) = & \sum_{i \in \mathcal{T}} \Delta(f_\theta(z_i), y_i) + \underset{vv}{\lambda} \sum_{\substack{i,j:w_{i,j}\neq 0 \\ i\in V \wedge j \in V}} w_{i,j} \|z_i - z_j\|^2 \\
& + \underset{pv}{\lambda} \sum_{\substack{i,j:w_{i,j}\neq 0 \\ i\in V \wedge j \in P}} w_{i,j} \|z_i - z_j\|^2 \\
& + \underset{pp}{\lambda} \sum_{\substack{i,j:w_{i,j}\neq 0 \\ i\in P \wedge j \in P}} w_{i,j} \|z_i - z_j\|^2
\end{aligned} \tag{6}
$$

The minimization of the loss function (Eq. 6) is performed by exploiting a Stochastic Gradient Descent Algorithm (see Algorithm 1). The algorithm first chooses a pair of connected nodes randomly. After that, if the node is in the training set $\mathcal{T}$ it modifies the parameters of the classification function and the latent representation according to the classification loss following Eq. 3. Successively, it updates the latent representation of both the nodes depending on the difference between the two representation presented in Eq. 4.

Algorithm 1. Learning$(x, w, \varepsilon, \lambda)$

for A fixed number of iterations **do**
  Choose (x_i, x_j) randomly with $w_{i,j} > 0$
  if $x_i \in \mathcal{T}$ **then**
    $\theta \longleftarrow \theta + \varepsilon \nabla_\theta \Delta(f_\theta(z_i), y_i)$
    $z_i \longleftarrow z_i + \varepsilon \nabla_{z_i} \Delta(f_\theta(z_i), y_i)$
  end if
  if $x_j \in \mathcal{T}$ **then**
    $\theta \longleftarrow \theta + \varepsilon \nabla_\theta \Delta(f_\theta(z_j), y_j)$
    $z_j \longleftarrow z_j + \varepsilon \nabla_{z_j} \Delta(f_\theta(z_j), y_j)$
  end if
  $z_i \longleftarrow z_i + \varepsilon \lambda \nabla_{z_i} w_{i,j} \|z_i - z_j\|^2$
  $z_j \longleftarrow z_j + \varepsilon \lambda \nabla_{z_j} w_{i,j} \|z_i - z_j\|^2$
end for

4 Experiments

4.1 Dataset

In order to evaluate the proposed approach, we used a dataset that contains enough information about users and posts to build a heterogeneous network as described in Sect. 2.1. Every user and post in the network has been labelled with its polarity (positive or negative).

We used the 'Obama' dataset available in [12], which has been collected from Twitter and contains the following data:

1. A set of users and their sentiment labels about the topic 'Obama' (obtained by manual tagging);
2. Tweets (posts) written by users about the topic 'Obama' with their sentiment labels (obtained by manual tagging);
3. The users' retweet network, which represent the approval connections between users.

This dataset contains 61 nodes and 187 tweets, and a total of 252 arcs representing retweet connections.

Starting from this dataset, we built a HN-DAG, where the set of nodes V_q represent the set of users who posted something about the topic 'Obama', and the set P_q represent the tweets that those users posted about 'Obama'.

We have three types of arcs connecting the nodes:

- the arcs connecting a user to another user, which weight is determined by the normalized number of retweets;
- the arcs connecting a user to a post, which in our case have 0/1 weights;
- the arcs connecting a post to another post, whose weight is determined by the cosine similarity between the two posts, as explained in Sect. 2.2.

4.2 Performance Evaluation and Settings

In order to assess the importance of relationships for determining the sentiment polarity of users and posts, we compare our method with two well-known approaches based only on the analysis of the textual data: a Support Vector Machine (SVM) and a L2-regularized logistic regression (LR). When only content is used, the posts are classified as positive or negative based on their content, while the users are classified based on the total polarity of their posts (the posts of a single user are merged and considered as a single document for determining the user's polarity).

We used the Support Vector Machine package available in LibSVM [15], using a linear kernel and default settings. The linear regression model was based on the library for large linear classification LibLinear [16].

We have considered as evaluation measures the well-known Precision(P), Recall(R) and F_1-measure:

$$P^+ = \frac{\#\text{ of instances successfully predicted as positive}}{\#\text{ of instances predicted as positive}} \tag{7}$$

$$R^{+} = \frac{\#\text{ of instances successfully predicted as positive}}{\#\text{ of instances effectively labelled as positive}} \tag{8}$$

$$F_1^{+} = \frac{2 \cdot P^{+} \cdot R^{+}}{P^{+} + R^{+}} \tag{9}$$

In the same way it is possible to compute the Precision, Recall and F-Measure for the negative class (P^{-}, R^{-}, F_1^{-}).

We also measured Accuracy as:

$$Acc = \frac{\#\text{ of instances successfully predicted}}{\#\text{ of instances}} \tag{10}$$

The performance of the proposed model can be affected by the randomness of the learning algorithm, leading to less-than-optimum results. In order to reduce this effect and improve the robustness of the classification, we used a majority voting mechanism to label the instances. In particular we performed k = 1, 5, 11, 15, 21 and 101 runs to get k predictions (votes) and we took a majority vote among the k possible labels for each node. For each k, we performed 100 experiments and considered their average performance. In the following, we report the results for $k = 21$, which show a good trade-off between the performance variability and the computational complexity.

The total number of iterations of the learning algorithm has been set to 4000000, while the gradient step ε have been set to 0.1. The size of the latent representation has been set to 40.

4.3 Results

Initially, we tested the performance of our approach by considering a case where 66 % of the nodes (randomly chosen) are considered as known. The proposed model is strongly influenced by the parameters λ_{pp}, λ_{pv} and λ_{vv} assigned to the different types of edges. Therefore, for each λ_i, where $i \in \{vv, pp, pv\}$, we investigated different values varying in the range $\{0.01, 0.05, 0.1\}$.

In Tables 1 and 2 we reported the best combinations of λ_i for classifying posts and users. The choice of the configuration is, at the current time, an empirical estimate. For the following experiments, we considered a trade-off between predicting the users and posts polarity, and therefore we chose as best configuration $\lambda_{pp} = 0.05$, $\lambda_{pv} = 0.05$, $\lambda_{vv} = 0.1$, as highlighted in the tables.

We compare the results obtained with these settings with the results achieved by the two textual approaches (see Table 3). The Latent space Heterogeneous Approval Model (LHAM) outperforms both Support Vector Machine (SVM) and Linear Regression (LR) when predicting the polarity of the posts (around 5 % improvement), and strongly outperforms them when predicting the polarity of users (more than 34 % of improvement in terms of accuracy).

In order to reduce the bias introduced by empirically choosing the values of λ_i, we computed the average performance over all possible combinations in the range $\{0.01, 0.05, 0.1\}$. The results (as reported in the last column of Table 3) show that our method still outperform the baseline algorithms when predicting

Table 1. Best configurations of λ_i for inferring the user polarity. The highlighted line represents the chosen configuration.

λ_{vv}	λ_{pp}	λ_{pv}	P+	R+	F1+	P-	R-	F1-	Acc
0.01	0.01	0.01	0.91	0.841	0.873	0.887	0.93	0.907	0.895
0.01	0.05	0.01	0.91	0.841	0.873	0.887	0.93	0.907	0.895
0.05	0.01	0.01	0.91	0.841	0.873	0.887	0.93	0.907	0.895
0.05	0.01	0.05	0.91	0.841	0.873	0.887	0.93	0.907	0.895
0.05	0.01	0.1	0.91	0.841	0.873	0.887	0.93	0.907	0.895
0.05	0.05	0.01	0.905	0.836	0.868	0.89	0.933	0.91	0.895
0.05	0.05	0.05	0.91	0.841	0.873	0.887	0.93	0.907	0.895
0.05	**0.05**	**0.1**	**0.91**	**0.841**	**0.873**	**0.887**	**0.93**	**0.907**	**0.895**
0.05	0.1	0.05	0.91	0.841	0.873	0.887	0.93	0.907	0.895
0.05	0.1	0.1	0.91	0.841	0.873	0.887	0.93	0.907	0.895
0.1	0.01	0.05	0.91	0.841	0.873	0.887	0.93	0.907	0.895
0.1	0.01	0.1	0.91	0.841	0.873	0.887	0.93	0.907	0.895
0.1	0.05	0.01	0.925	0.839	0.878	0.913	0.953	0.932	0.914
0.1	0.05	0.05	0.91	0.841	0.873	0.887	0.93	0.907	0.895
0.1	0.05	0.1	0.91	0.841	0.873	0.887	0.93	0.907	0.895
0.1	0.1	0.05	0.91	0.841	0.873	0.887	0.93	0.907	0.895
0.1	0.1	0.1	0.91	0.841	0.873	0.887	0.93	0.907	0.895

the polarity of the users, maintaining a 33 % of improvement in terms of accuracy, while maintaining a comparable performance when predicting the polarity of the posts.

In order to fully validate our approach, we tested it with different sizes of training and test sets. Therefore, we randomly split our dataset with different percentages $\{20, 33, 50, 66, 80\}$. Given the small size of the dataset, we perform a cross-validation by repeating the random split 30 times for each percentage, and therefore obtain significant results.

Tables 4 and 5 show the results of posts and users classification, performed by our model and baseline models depending on training set percentage. It is clear from the tables that our model outperforms other approaches in most of the cases, in particular when the size of the training set has a larger number of instances. While the post classification shows a slight improvement by our model over SVM and Linear Regression, for user classification we are able to achieve far better results than text-only based approaches.

While our model improves its performance for larger training set sizes, the other methods do not improve, and their performance can even decrease. The most probable explanation of this behaviour is that short-text posts are very noisy: a text-only approach is therefore more affected by the introduction of more training instances (which are regarded as more noise), while our model is

Table 2. Best configurations of λ_i for inferring the post polarity. The highlighted line represents the chosen configuration.

λ_{vv}	λ_{pp}	λ_{pv}	P+	R+	F1+	P-	R-	F1-	Acc
0.01	0.01	0.01	0.673	0.819	0.738	0.763	0.587	0.661	0.705
0.01	0.05	0.01	0.677	0.819	0.74	0.762	0.594	0.666	0.708
0.05	0.01	0.01	0.629	0.806	0.699	0.643	0.477	0.528	0.644
0.05	0.01	0.05	0.677	0.806	0.734	0.755	0.6	0.666	0.705
0.05	0.01	0.1	0.68	0.819	0.741	0.769	0.6	0.671	0.711
0.05	0.05	0.01	0.639	0.863	0.727	0.813	0.465	0.533	0.667
0.05	0.05	0.05	0.678	0.825	0.743	0.772	0.594	0.668	0.711
0.05	**0.05**	**0.1**	**0.684**	**0.819**	**0.743**	**0.772**	**0.606**	**0.675**	**0.714**
0.05	0.1	0.05	0.671	0.813	0.734	0.756	0.587	0.658	0.702
0.05	0.1	0.1	0.678	0.825	0.743	0.772	0.594	0.668	0.711
0.1	0.01	0.05	0.669	0.794	0.724	0.743	0.594	0.657	0.695
0.1	0.01	0.1	0.676	0.806	0.734	0.755	0.6	0.666	0.705
0.1	0.05	0.01	0.606	0.869	0.707	0.826	0.394	0.481	0.635
0.1	0.05	0.05	0.666	0.806	0.728	0.751	0.581	0.652	0.695
0.1	0.05	0.1	0.669	0.806	0.73	0.751	0.587	0.656	0.698
0.1	0.1	0.05	0.673	0.819	0.738	0.761	0.587	0.661	0.705
0.1	0.1	0.1	0.673	0.806	0.732	0.753	0.594	0.661	0.702

Table 3. Accuracy of users and post classification for different algorithms.

	LR	SVM	LHAM (Best λ_i)	LHAM (Average λ_i)
Users	0.467	0.552	0.895	0.886
Posts	0.66	0.657	0.714	0.680

Table 4. Accuracy of post classification for different sizes of the training set.

% Training set	LR	SVM	LHAM
20	0.613	0.597	0.542
33	0.629	0.620	0.662
50	0.642	0.641	0.718
66	0.679	0.679	0.722
80	0.660	0.669	0.739

Table 5. Accuracy of user classification for different sizes of the training set

% Training set	LR	SVM	LHAM
20	0.466	0.485	0.570
33	0.494	0.521	0.823
50	0.480	0.512	0.986
66	0.467	0.531	0.982
80	0.447	0.507	0.986

able to face this problem with the help of the additional information carried by the edges between different nodes.

The lower performance of LHAM for small percentages of the training set is explained by the behaviour of the Stochastic Gradient Descent Algorithm, which randomly chooses a pair of connected nodes at each iteration. When the number of training instances is small, the chance to pick nodes that are not in the training set will be higher. In this case, the latent representations will mostly depend on the similarity among connected nodes, and less on the correct sentiment polarity.

In order to tackle this problem, we modified Algorithm 1 as follows:

- At the beginning, starting from the training instances we create a list of "allowed" nodes;
- At each iteration, the algorithm must choose a pair of nodes where at least one of the nodes is in the list of "allowed" nodes;
- At the end of each iteration, if one of the chosen nodes was not in the list, it is added; if all the existing nodes have been added, the list is again initialized with the training instances.

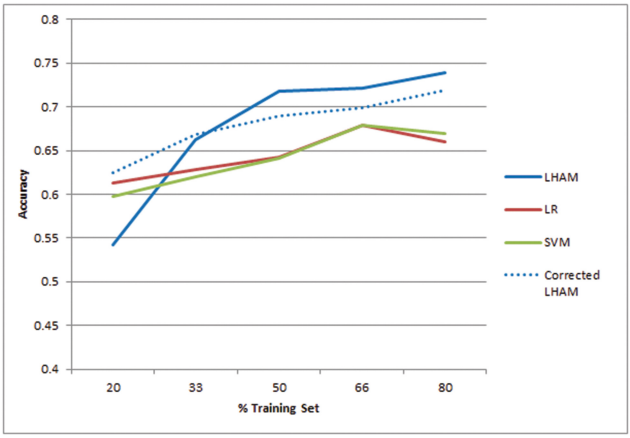

Fig. 2. Accuracy of post classification for different sizes of the training set.

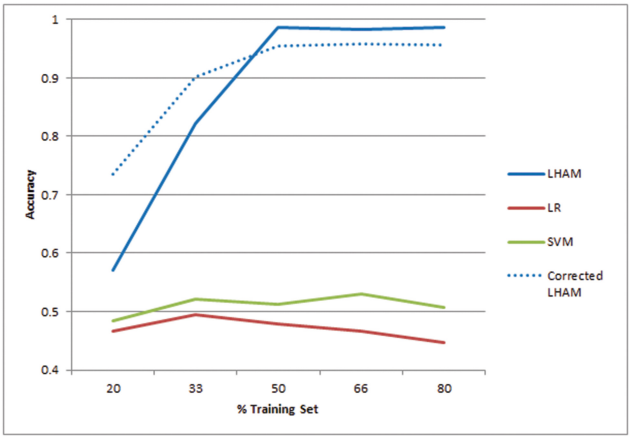

Fig. 3. Accuracy of user classification for different sizes of the training set

The corrected algorithm allows to spread the sentiment polarity information starting from the training nodes, and gradually towards the rest of the network. This permits to outperform the baseline algorithms even when dealing with small training sets both on posts and users classification. At the same time, we maintain a good performance when the training size gets larger (see Figs. 2 and 3).

5 Conclusions

In this work, we proposed a classification approach that is able to infer the polarity of users and posts in a social network, particularly in the case of microblogs (such as Twitter).

We have shown that the exploitation of the information obtained from the heterogeneous network can improve not only the performance of the classification of users (as already proven in other works), but also the performance of the classification of posts. The results clearly show that the proposed model is promising and worth further investigation. In the future we plan to improve the robustness of the model by introducing a method for estimating the best parameter configuration.

Moreover, we want to compare our approach with other user-level polarity classifiers, and to focus on the development of larger datasets on different topics.

References

1. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retr. **2**(1–2), 1–135 (2008)
2. Hu, X., Tang, J., Gao, H., Liu, H.: Unsupervised sentiment analysis with emotional signals. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 607–618. International World Wide Web Conferences Steering Committee (2013)

3. Wang, X., Wei, F., Liu, X., Zhou, M., Zhang, M.: Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 1031–1040. ACM (2011)
4. Davidov, D., Tsur, O., Rappoport, A.: Enhanced sentiment learning using twitter hashtags and smileys. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 241–249. Association for Computational Linguistics (2010)
5. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. Technical report, Stanford (2009)
6. Pozzi, F.A., Fersini, E., Messina, E.: Bayesian model averaging and model selection for polarity classification. In: Métais, E., Meziane, F., Saraee, M., Sugumaran, V., Vadera, S. (eds.) NLDB 2013. LNCS, vol. 7934, pp. 189–200. Springer, Heidelberg (2013)
7. Tan, C., Lee, L., Tang, J., Jiang, L., Zhou, M., Li, P.: User-level sentiment analysis incorporating social networks. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2011, pp. 1397–1405 (2011)
8. Kim, J., Yoo, J., Lim, H., Qiu, H., Kozareva, Z., Galstyan, A.: Sentiment prediction using collaborative filtering. In: Seventh International AAAI Conference on Weblogs and Social Media (2013)
9. Smith, L.M., Zhu, L., Lerman, K., Kozareva, Z.: The role of social media in the discussion of controversial topics. In: 2013 International Conference on Social Computing (SocialCom), pp. 236–243. IEEE (2013)
10. Deng, H., Han, J., Ji, H., Li, H., Lu, Y., Wang, H.: Exploring and inferring user-user pseudo-friendship for sentiment analysis with heterogeneous networks. In: SDM, pp. 378–386 (2013)
11. Lazarsfeld, P.F., Merton, R.K.: Friendship as a social process: a substantive and methodological analysis. In: Berger, M., Abel, T., Page, C.H. (eds.) Freedom and Control in Modern Society, pp. 8–66. Van Nostrand, New York (1954)
12. Pozzi, F.A., Maccagnola, D., Fersini, E., Messina, E.: Enhance user-level sentiment analysis on microblogs with approval relations. In: Baldoni, M., Baroglio, C., Boella, G., Micalizio, R. (eds.) AI*IA 2013. LNCS, vol. 8249, pp. 133–144. Springer, Heidelberg (2013)
13. Jacob, Y., Denoyer, L., Gallinari, P.: Learning latent representations of nodes for classifying in heterogeneous social networks. In: WSDM, pp. 373–382 (2014)
14. Baeza-Yates, R., Ribeiro-Neto, B., et al.: Modern Information Retrieval, vol. 463. ACM Press, New York (1999)
15. Chang, C.-C., Lin, C.-J.: Libsvm: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 27:1–27:27 (2011)
16. Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J.: LIBLINEAR: a library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)