https://doi.org/10.1007/s12559-020-09793-7

Heterogeneous Graph Network Embedding for Sentiment Analysis on Social Media

Zhigang Jin[1] · Xiaofang Zhao[1] · Yuhong Liu[2]

Received: 13 March 2020 / Accepted: 14 November 2020 / Published online: 4 January 2021

Abstract

Nowadays, more people are used to express their attitudes on different entities in online social networks, forming user-to-entity sentiment links. These sentiment links imply positive or negative semantics. Most of current user sentiment analysis literature focuses on making a positive, neutral, or negative sentiment decision according to users' text descriptions. Such approach, however, often fails to retrieve users' hidden real attitudes. We design a powerful sentiment link analysis framework named graph network embedding for sentiment analysis (NESA). NESA first utilizes variational auto-encoder (VAE) to learn joint representations of users' social relationship by preserving both the structural proximity and attribute proximity. Then, a multi-view correlation learning–based VAE is proposed to fuse the joint representation and the user-entity sentiment polarity network. By jointly optimizing the two components in a holistic learning framework, the embedding of network node information and multi-network contents is integrated and mutually reinforced. The first experimental results verify the effectiveness of adopting user, entity attributes, and social relationships for sentiment link analysis. Then we demonstrate the superiority of NESA over state-of-the-art network embedding baselines on link prediction. The last experimental results further validate that NESA model outperforms the traditional text-based sentiment prediction methods. We propose to perform sentiment analysis from network perspective; the proposed NESA model applies heterogeneous graph network embedding to fuse multi-networks information with considering their correlations and then to retrieve users' hidden real attitudes in social networks. It provides a novel angle to resolve the sentiment analysis problem.

Keywords Social network · Sentiment analysis · Graph network · Embedding learning · Variational auto-encoder

Introduction

As a result of their massive, diverse, and increasing user bases, online social network platforms act not only as a way for interactions and communications among people but also as a way for people to share feelings and express attitudes towards other online entities, e.g., friends, movie stars, or politicians, resulting in sentiment links between these users and entities. These links with a large amount of up-to-date information have emerged as an important source for online opinion mining and sentiment analysis [1, 2], which has been successfully used in many tasks, such as stock market prediction [3], recommendation systems [4], and inference of public mood about social events [5].

Different from explicit social links indicating friend or follower relationship, sentiment links involving different types of links: positive links that express like, trust or support attitudes and negative links that signify dislike or disapproval of others, which are latent sentiment polarity by the semantic contents posted by users. In this paper, we explore the hidden sentiment polarity of sentiment links and provide a novel angle to resolve the sentiment analysis problem, which is different from traditional text-based method.

Traditional sentiment analysis (also known as opinion mining) that analyzes people's opinions and sentiments from text is an active research field in natural language processing (NLP) [6]. The existing text-based studies of sentiment classification, as an important part of sentiment analysis, can be mainly grouped into two categories: lexicon-based and corpus-based approaches. Lexicon-based approaches [7] mainly use the sentiment polarity associated with the

✉ Yuhong Liu
yhliu@scu.edu

[1] School of Electrical and Information Engineering, Tianjin University, Tianjin, China

[2] Dept. of Computer Science and Engineering, Santa Clara University, Santa Clara, USA

sentiment dictionary to calculate the sentiment polarity of each sentence or document. Corpus-based approaches take sentiment classification as a special case of text categorization problem, which utilize machine learning methods to extract reasonable features from texts and feed into a classifier to predict sentiments [8].

Nowadays embedding the network into a low-dimensional space has attracted extensive research interests. Such embedding can learn rich semantic information of nodes and enable a lot of applications, such as node classification and link prediction.

In this paper, we consider positive and negative sentiment polarity predictions as signed link predictions in the context of social networks. Such prediction is challenging mainly due to three reasons. (1) Users express their sentiment to entities with very limited words in online social networks (e.g., Sina Weibo, Twitter), where the sentiment polarity is likely to be hidden and not obvious from the short text. In this situation, it is very challenging for traditional sentiment classifiers to extract the user's attitude towards this entity. Besides this, if a user does not post any word concerning movie "Forrest Gump," it is hard for traditional sentiment analysis method to predict the user's attitude towards this movie. Therefore, it is a challenge to retrieve users' hidden real attitudes and predict the user's sentiment to an entity without any prior content information. (2) Existing network embedding methods have achieved significant performance on the task of link prediction; most of them can only train a classifier to predict whether a link exists or not, but cannot retrieve the sign of the relationship (i.e., positive or negative). Additionally, several studies focus on network embedding to signed link predictions; however, these methods mostly rely on the balance theory and the status theory and are not applicable to the problem of sentiment prediction in real-world signed and heterogeneous networks. (3) Most existing embedding methods only consider user structure information or attribute information, while ignoring the comprehensive connections between users and entities.

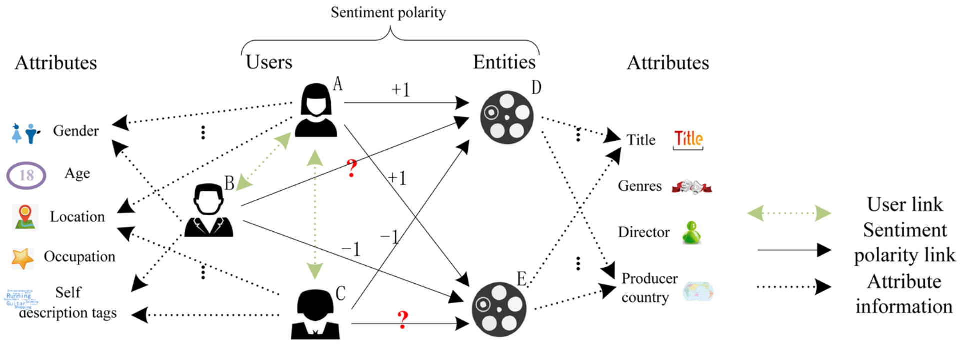
Figure 1 shows a simple example of a social network sentiment analysis scenario. We assume the existence of some latent relationship for each link in the users-entities sentiment polarity network. For example, users A, B, and C have social relationships among each other. User A holds definite positive attitudes to movie D and E, but the link between user B and movie D carries an unknown sentiment link. Our goal is to model the information network to retrieve unknown sentiment links based on users' social relationships, user attributes, and movie attributes. This example also demonstrates three types of networks that are helpful in sentiment sign link prediction: (1) user link network, (2) sentiment polarity network between users and entities, and (3) attributes network for user and entity.

Graph neural networks are often adopted by existing works to embed homogeneous networks into a low-dimensional space. However, due to the complexity of social networks, it is difficult to obtain the local and global network semantic information by using traditional graph neural network techniques. VAE is a kind of deep neural network, and motivated by its' excellent performance in avoiding overfitting and ensuring good properties of the latent space to enable generative process, as well as the flexible fusion of structural information and attribute information makes the learned representation of network nodes achieve satisfactory performance, we creatively propose a multi-view correlation learning model, which is capable of handling three kinds of graph network data as information sources to resolve the sentiment link prediction problem. To our best knowledge, this is the first work that VAE-based heterogeneous network embedding to fuse multi-networks information with considering their correlations and then to retrieve users' hidden real attitudes (sentiment polarity) in social networks.

Our main contributions of this work are summarized below:

- We propose and elaborate the problem of sentiment analysis in social networks, that is, to find the sentiment tendency hidden in user comments.
- By proposing heterogeneous graph network embedding to fuse user link network, attributes network, and sentiment polarity network with considering their correlations and then to retrieve users' hidden real attitudes in social networks, we design a powerful sentiment link analysis framework.

Fig. 1 A heterogeneous information network, which contains either multiple types of objects or multiple types of links



- We have carried out comprehensive experiments on real social network data to prove the effectiveness of the proposed method. The results also prove that the proposed method can effectively utilize users' and entities' information to achieve better accuracy in predicting users' sentiment polarity.

Related Work

Sentiment Analysis

We can divide text contents into two types: objective information mainly for events and products, and subjective information mainly generated by users' evaluation information of people, events, products, and so on. Subjective information expresses people's various sentiment polarity and tendencies, such as "positive," "negative," and "neutral."

The existing text-based studies of sentiment classification can be mainly grouped into two categories: lexicon-based and corpus-based approaches. Lexicon-based approaches are relatively early for sentiment analysis, and it has an excellent effect on the fine-grained sentiment analysis of text. Park et al. [9] propose a method to use three online dictionaries to collect thesauruses based on the seed words, and store only co-occurrence words into the thesaurus lexicon in order to improve the reliability of the thesaurus lexicon for sentiment classifications. Zhang et al. [10] propose a method to extend the sentiment dictionary by extraction and construction of degree adverb dictionary, network word dictionary, negative word dictionary, and other related dictionaries for Chinese micro-blog text sentiment analysis. Tang et al. [11] cast sentiment lexicon learning as a phrase-level sentiment classification task and propose to build large-scale sentiment lexicon from Twitter with a representation learning approach. Felipe et al. [12] propose a novel, simple, and yet effective lexicon-expansion strategy based on semantic relationships extracted from word embeddings.

Corpus-based approaches are often preferred as compared with lexicon-based approaches. Corpus-based approaches have two main directions: traditional feature-based methods and deep learning methods. Feature selection has been studied by many researchers, which extracts manually designed features, such as Part-of-Speech (POS), term frequency-inverse document frequency (TF-IDF), and N-gram (unigrams, bigrams and trigrams), from the text, and applies classification models, such as Naive Bayes, Support Vector Machine [8], support vector machine with Naive Bayes features [13], and Ensemble learning [14], to complete the sentiment classification task.

Deep learning models have achieved remarkable results in computer vision and speech recognition in recent years. Within NLP, many works with deep learning models have focused on learning word embeddings [15–18] with neural language models, such as word2vec [14], GloVe [15], the latest pre-training ELMO [16], and BERT [17] models. The word vector obtained by these methods has better representation and reasoning ability in the semantic space and can be used as the input of various deep learning models. These models have achieved great success on sentiment analysis tasks.

Recently, neural network models, such as Convolutional Neural Network (CNN) [19], Recursive Auto-encoders [20], Long Short Term Memory (LSTM) [21], and many more obtained from an unsupervised neural language processing models, are applied for sentiment classification, and have achieved excellent results on multiple benchmarks.

As online users' opinions can be influenced by their social relationships and users, entities attributes [22–31], some researchers combine users' preferences and products' characteristics for sentiment classification. For example, User Product Neural Network (UPNN) [22] captures user- and product-level information for sentiment classification. User Product Attention (UPA) [23] takes global user and product information into consideration. In Chen et al. [24], a sequence modeling–based neural network approach is proposed to embed temporal relations of reviews into the categories of distributed user and product representations learning for the sentiment classification of reviews. User Product Deep Memory Network (UPDMN) [25] applies a deep memory network for the first time to capture the user and product information for document-level sentiment classification. The last model, Hierarchical User Attention and Product Attention neural network (HUAPA) [26], applies two individual hierarchical neural networks with user attention or with product attention to encode user and product information, and then design a combined strategy to make full use of the two representations for training and final prediction, achieving high performances.

These above literatures demonstrate the effectiveness of internal and external connections of users and products for sentiment classification, which is actually an active direction in sentiment analysis and opinion mining. But these methods rely heavily on the concrete content posted by users, and cannot detect the existence of sentiment links without any prior content information. Based on the above facts, in this paper we proposed a sentiment analysis model mainly focusing on predicting the sign of a given sentiment link without any prior content information.

In addition to the above analysis, as people often express their opinions on social networks, where similar hobbies and concepts group users into personalized social communities, some studies have shown that social computing theory has a guiding role in short text sentiment analysis, which can further improve the prediction performance. For example, Wang et al. [27] define user to user topic inclusion and have constructed its sparse network. Xiao et al. [28] quantify the dependency of common neighbors and analyze user correlations in the structural space. Wei et al. [29] study the role of users in the sentiment diffusion of social networks. Zhao et al. [30] propose a deep neural network based model to take user social relations into consideration for sentiment classification.

Different from the above text-based sentiment analysis methods, we propose to perform sentiment analysis from network perspective, where the user-entity network is modeled as a signed network with either positive or negative links indicating users' different sentiments on entities.

Network Embedding

Traditional network embedding methods, such as locally linear embedding (LLE) [39] and Laplacian Eigenmaps (LE) [40], are mainly based on the calculation of matrix eigenvectors. In particular, one critical component is the definition and construction of an incidence matrix, which needs to calculate the eigenvalues of the association matrix. However, due to the high time and space complexity of such calculation, this type of approach is not suitable for large-scale network embedded learning.

In recent years, network embedding based on neural networks has attracted many research attentions. Inspired by the neural network language model Word2vec [14], Perozzi et al. [33] propose a DeepWalk method, which uses a truncated random walk sequence to represent the neighbors of a node and obtains the embedded representation of network information through Skip-Gram model. Because random walk only depends on local information, the DeepWalk model requires less time and space to calculate. Later, Node-2vec has improved the random walk strategy of DeepWalk by introducing depth first search and width first search to capture the structural equivalence and homogeneity of nodes [32]. LINE formally establishes the first- and second-order neighboring features of the network by optimizing the first- and second-order similarity of network nodes and proposes to use the second-order proximity to compensate for the data sparsity issue [34]. In order to obtain higher-level feature representations of nodes, GraRep has modeled higher-order similarity relationships on the basis of LINE [41]. SDNE uses auto-encoder to capture first-order and second-order network structures and learn user representations [42].

Although maintaining network structure is an important requirement for network embedding, these above methods only preserve the pairwise relationship between nodes, while ignoring useful community structures, large amount of users' relation and attribute information. MNMF integrates community structures into the network embedding, and then jointly optimizes the NMF-based embedding representation learning model and the module-based community detection model in a unified framework. As a result, the learned node representations can preserve the network's microstructure and community structure. However, this method is based on matrix factorization, which has high complexity and cannot handle large-scale networks [43]. GEMSEC introduces clustering constraints into the node embedding optimization goal. While learning the features of the nodes, it performs *k*-means clustering on the nodes. The clustering results are used to influence the node embedding, so that the latent clustering structure of the network can be preserved [44]. GENE integrates the structures of the inner-group and cross-group information. The network structure and label information can be integrated by adding nodes containing label information in the walk sequence [45]. A maximum margin–based DeepWalk model (i.e., MMDW) is proposed to integrate the label information of nodes into the node embedding vector to learn the differentiated node embedding representation [46].

Additionally, several studies focus on representation learning in special scenarios [35–37, 47, 48]. Chang et al. [47] design a deep embedding algorithm for networked data. A highly nonlinear multilayered embedding function is used to capture the complex interactions between the heterogeneous data in a network. Xiao et al. [48] investigate how labels can be modeled and incorporated to improve attributed network embedding. Yuan et al. [35] develop a signed network embedding model called SNE, which is the first research on signed network embedding. Negative links have distinct properties and added values besides positive links, which bring both challenges and opportunities for signed network embedding. Wang et al. [36] propose a deep learning framework SiNE, which is a linearly scalable method that leverages balance theory along with random walks to obtain low-dimensional vectors for directed signed networks embedding. Shen et al. [37] propose graph representation learning in signed networks based on stacked auto-encoder and the structural balance. However, these methods are specialized in only one particular type of networks, which is not applicable to the problem of sentiment prediction in real-world signed and heterogeneous networks. SHINE proposes deep auto encoders to map each node into a low-dimension feature space and demonstrates superiority over state-of-the-art baselines on link prediction and node recommendation in two real-world datasets, but the shortcoming of this model is that its individual models are trained independently without considering their correlations, and results are simply combined after training [38].

Based on the fusion of network structure and node information, this paper proposes three heterogeneous graph network embedding learning representations, and a multi-view learning method of VAE to learn the representations of each node. By jointly optimizing these components, a holistic learning framework is formed, which realizes an enhanced network embedding by integrating network structure, attribute information, and sentiment information.

Preliminaries: Variational Autoencoders

Variational autoencoders (VAEs) [49] is a generative model developed from standard autoencoders. It combines Bayesian inference with the flexibility of neural networks for robust representation learning. The traditional auto-encoder consists of two parts: encoder and decoder. Encoder encodes the input data to get the representation of the hidden layer. Decoder decodes the representation of the hidden layer to reconstruct the output $\hat{x}$ with the same node size as the input. The objective function of model optimization is to minimize the reconstruction error between x and $\hat{x}$, which can be expressed by cross entropy or square difference. From the principle of autoencoder, it can be seen that autoencoder is a hidden layer representation of the input data to learn directly, while VAE is not. Given the input data of one view x, the objective of VAE is to find a latent variable z to represent x. The latent variable z is generated from a prior distribution $p(z)$, where $p(z) \sim N(0, I)$ is a standard Gaussian distribution. VAE allows encoder to learn the latent variable z of input data, that is, to learn the parameters of Gaussian distribution of this set of latent variables: μ, σ^2 (the mean value μ and variance σ^2 of the Gaussian distribution of latent variable, respectively). The latent variable z can be sampled from this set of distribution parameters:$z \sim N(\mu, \sigma^2)$, and then the latent variable is decoded by the decoder to reconstruct the input.

However, in practice, VAE model does not really use $z \sim N(\mu, \sigma^2)$ to sample z variables, because after sampling, there is no way to directly derive the parameter $\theta(\mu, \sigma^2)$ and optimize the objective function through gradient descent algorithm. By applying the reparameterization trick, first, a standard Gaussian distribution (i.e., normal distribution) $\varepsilon \sim N(0, 1)$ is sampled, then latent variable z can be obtained by $z = \mu + \varepsilon \times \sigma$ and $z \sim N(\mu, \sigma^2)$. In this way, samples could be generated by decoding the sampling points in the Gaussian latent space. The objective function consists of two parts: reconstruction function and Kullback–Leibler divergence (denoted by KL):

$$L_{\text{loss}} = \mathbb{E}_{q_\phi(z|x_i)}\left[\log p_\theta(x_i|z)\right] + KL(q_\phi(z|x_i) \| p_\theta(z)) \tag{1}$$

where $q_\phi(z|x_i)$ and $p_\theta(x_i|z)$ are parameterized by the encoder network and the decoder network, respectively. The first term in the L_{loss} is a reconstruction loss that encourages the decoded $\hat{x}$ to be close to the observed x. The second term is a regularization term where the posterior distribution of z is pulled towards the prior, which is often a simple distribution.

Motivated by VAE's excellent performance in avoiding overfitting and ensuring good properties of the latent space to enable generative process, as well as the flexible fusion of structural information and attribute information makes the learned representation of network nodes achieve satisfactory performance, we creatively propose a multi-view correlation learning model, which is capable of handling three kinds of graph network data as information sources to resolve the sentiment link prediction problem.

Problem Formulation

In this section, we would like to formally model the problem that we aim to address in this work, including inputs and the desired outputs.

Input: When predicting user sentiments in social networks, if user relationships and users' sentiments on entities are used respectively to build a graph model, it will be more complicated. Therefore, we reconstruct three kinds of graph information network: (a) User link network (U-U), (b) User-Entity sentiment polarity network (U-E-P), and (c) User-Entity Attributes network, to predict users' hidden sentiments (U-E-A).

a. User Link Network

The user link network is denoted as G_U=(U, R). In particular, U=$\{u_i\}_{i=1}^{N}$ is user set, and $R = \{r_{ij}\}_{i,j=1}^{|R|}$ represents social connections between user pair i and j. If a connection exists between u_i and u_j, r_{ij} is set as 1. Otherwise, r_{ij} is set as 0, and then we can define user relation adjacency vector $x_U = \{r_{ij}|i \in U, j \in U\}$.

b. Sentiment Polarity Network

The sentiment network is denoted as G_P=(U, E, P), where U and E represent the set of users and entities contained in text, and then we define its sentiment polarity adjacency vector $x_P = \left\{ p_{ij} \middle| i \in U, j \in E \right\}$. Each p_{ij} can take the value of $+1, -1$, or 0, representing that user i holds a positive, negative, or unobserved sentiment towards entity j, respectively.

c. Attributes Network

User-Attribute Vector: In order to get the user-attribute network, we extract 5 attributes as user attribute information: age, gender, location, occupation, and self-description tags, since they can best represent users' sentiments towards different entities. Each value of these attributes is discretized as a vector and the concatenation of these vectors forms a one-hot vector. Then $\mathbf{A}_U = \{\mathbf{A}_U^1, \mathbf{A}_U^2, A_U^3, A_U^4, \mathbf{A}_U^5\}$ is used to denote the set of user attributes, and $\boldsymbol{A}_U \in \mathbb{R}^{N \times L_U}$, N is the number of users, and L_U is the total number of columns of $\mathbf{A}_U$. Figure 2a shows an example attribute feature vector consisting of these 5 feature entries.

Entity-Attribute Vector: Similarly, as shown in Fig. 2b, we also extract 4 attributes: title, genres, director, and producer country as entity attribute information. In the entity attributes vector, values of these attributes are discretized, similar with user-attribute network, we use $\mathbf{A}_E = \left\{\mathbf{A}_E^1, \mathbf{A}_E^2, A_E^3, A_E^4\right\}$ to denote the set of entity attributes, where $\boldsymbol{A}_E \in \mathbb{R}^{M \times L_E}$, M is the number of entities, and L_E is the total number of columns of $\mathbf{A}_E$. Then we combine $\mathbf{A}_U$ and $\mathbf{A}_E$ to define user-entity attribute matrix as $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_U & 0 \\ 0 & \boldsymbol{A}_E \end{bmatrix}$, where $\boldsymbol{A} \in \mathbb{R}^{(N+M)\times(L_U+L_E)}$ and $x_A = \left\{ a_{ij} \middle| i \in [1, N+M], j \in \left[1, L_U + L_E\right] \right\}$ represent user-entity attribute link. Each a_{ij} can take the value of 1 or 0. For a user node, when $1 \le i \le N$, $1 \le j \le L_U$, the value of a_{ij} is meaningful as the user's attribute. Similarly, when $N+1 \le i \le N+M$, $L_U + 1 \le j \le L_U + L_E$, the value of a_{ij} is meaningful as an entity attribute.

Output: Taking G_U, G_P, and G_A as inputs, our goal is to predict unobserved sentiment links in heterogeneous information networks. In other words, the output is the unobserved part of $\hat{G}_P$. Meanwhile, we will be able to retrieve a low dimensional vector z for each node in social networks, which is rich in underlying relational semantics and can be used for tasks such as link prediction, node classification, and node clustering.

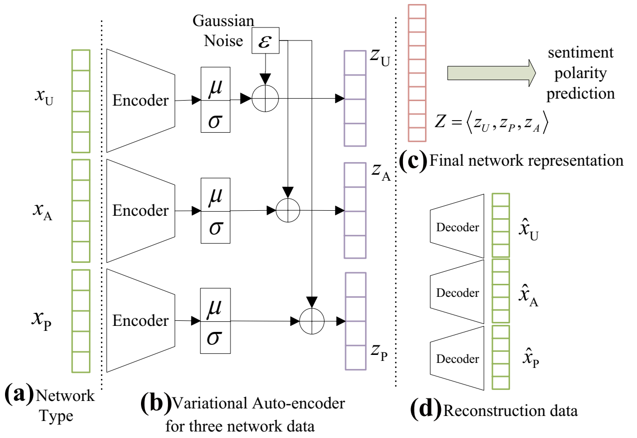


Fig. 3 Separate network embedding model for sentiment analysis

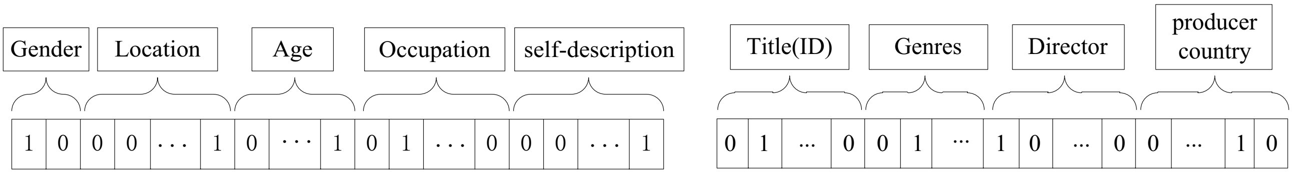
Proposed Method

In this section, we will introduce the proposed separate network embedding model for sentiment analysis (NESA-sep) and fused network embedding model for sentiment analysis (NESA-fus). For the NESA-sep model, we would like to first introduce its framework and then provide detailed discussions, including how to extract user-entity representations from the three input networks. However, the NESA-sep model ignores the correlation between the three input networks and cannot get complete embedding results, which contain both structure and attributes information of nodes. Therefore, we further propose the NESA-fus model and discuss it in details in "Fused Network Embedding Model (NESA-fus)." In particular, the NESA-fus model is based on VAE, which jointly considers user link network and attributes network to learn the representations of network nodes by preserving their structural proximity and attribute proximity. Furthermore, the sentiment polarity network and the above fusion results will be fed into the second VAE for feature extraction and retrieval of the network nonlinear fusion representations. The final embedding results will be used to predict unknown sentiment links.

Separate Network Embedding Model (NESA-sep)

In the NESA-sep model, in order to integrate the three information networks (G_U,G_P,G_A) for embedding, a simple method is to train a VAE model for each of the three networks, as shown in the Fig. 3. Then the three coding hidden variables corresponding to the three networks are integrated as a new data representation. The objective function of the joint optimization is established to realize linear fusion.

In Fig. 3, each of the three networks is encoded by a VAE. Take user link network as an example, the loss function of VAE is composed of the reconstruction term and the KL regularize:



(a) User attributes information

(b) Entity attributes information

Fig. 2 **a** User attributes information. **b** Entity attributes information

$$L(x_U) = KL\left(q(z_U/x_U)\middle\|p(z_U)\right) + \left\|(x_U - \hat{x}_U) \odot l_U\right\|_2^2 \quad (2)$$

where the first term $KL(\bullet)$ is the KL divergence function used as a measurement of the differences between two distributions. The second term is the function to measure the distance between x_U and $\hat{x}_U$, We define user link reconstruction weight vector $l_{U_{ij}}$. If there is a link between u_j and u_i, $l_{U_{ij}} = \alpha > 1$, else $l_{U_{ij}} = 1$.

Similar to user link network, we apply VAE to embed sentiment polarity network and attributes network. The loss functions of VAE for two networks are

$$L(x_P) = KL\left(q(z_P/x_P)\middle\|p(z_P)\right) + \left\|(x_P - \hat{x}_P) \odot l_P\right\|_2^2 \quad (3)$$

$$L(x_A) = KL\left(q(z_A/x_A)\middle\|p(z_A)\right) + \left\|(x_A - \hat{x}_P) \odot l_A\right\|_2^2 \quad (4)$$

The calculation of l_P and l_A, which represent the reconstruction weight vector of the sentiment polarity network and the attribute network respectively, is similar to l_U. In other words, the values of l_P and l_A depend on not only the sentiment polarity values between users and entities but also whether the users or entities have certain attributes or not. Then, the three encoded hidden variables z_U, z_P, and z_A corresponding to three information networks are concatenated as a new representation of the data.

The final representation z is the concatenation of the hidden vector $g(z) = \langle z_U, z_A, z_P \rangle$. Based on the heterogeneous network embedding z_i and z_j, the predicted sentiment $\hat{s}_{ij}$ can be calculated through similarity measurement function as follows:

$$f(U_i, E_j) = z_i^T z_j + b \quad (5)$$

The loss function between the predicted sentiment and the ground truth is calculated as follows.

$$L_{s_{ij}} = \left(s_{ij} - \hat{s}_{ij}\right)^2 \quad (6)$$

The joint optimization objective function is established as a loss function shown below to achieve linear fusion:

$$L_{\text{sep}} = L(x_U) + \lambda_1 L(x_P) + \lambda_2 L(x_A) + \lambda_3 L_{s_{ij}} \quad (7)$$

The complete objective function of NESA-sep model is as follows.

$$\mathcal{L}_{\text{sep}} = L(x_U) + \lambda_1 L(x_P) + \lambda_2 L(x_A) + \lambda_3 L_{s_{ij}} + \lambda_4 L_{reg} \quad (8)$$

where λ_1, λ_2, and λ_3 are balancing parameters, λ_4 is a trade-off parameter, and L_{reg} is an L2-norm regularizer term to prevent overfitting. However, this method neglects the correlation among the three information networks and is therefore not effective when there is a great difference on the vector lengths of three networks. In particular, the information network with longer vector is likely to dominate the objective functions of both KL divergence and data reconstruction. To address these issues of this method, we further propose the NESA-fus model (Fig. 4).

Fused Network Embedding Model (NESA-fus)

In social networks, the goal of network embedding is to project multiple information networks into a low-dimensional vector space (i.e., embedding space). To combine the sentiment polarity network, user link network, and attribute network, an intuitive way is to concatenate the learned embedding results from each part through the NESA-sep model. However, the main drawback of the NESA-sep model is that individual models are trained independently without considering their correlations, and results are simply combined after training.

In this section, we strive to develop a fusion model, which can effectively improve the prediction performance by considering the correlations among different networks. First of all, inspired by literature [50], which considers the user link network and attributes network as independent sources of information, we propose to integrate both networks to learn a

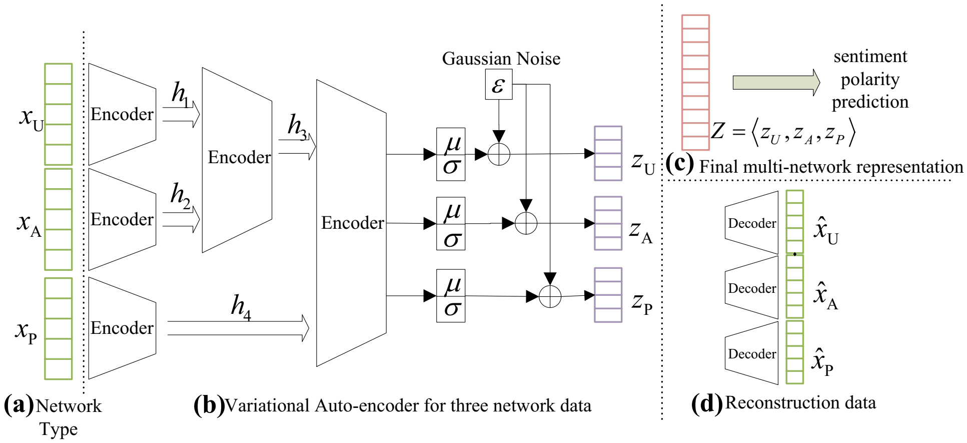


Fig. 4 NESA-fus model for three information networks embedding

comprehensive representation. In particular, we apply VAE to encode the two networks separately to obtain h_1 and h_2, respectively, then input them into the second encoder to obtain the joint representation h_3 of the two networks, which can preserve both the structural proximity and attribute proximity. Then, the sentiment polarity network and the learned joint representation h_3 are considered as two views. To further fuse the two views, a VAE-based multi-view correlation learning is proposed to learn the representation of each node. In the proposed NESA-fus model, to avoid differences on the vector lengths of two views, we first encode these two views into two vectors with the same length, and then concatenate these two vectors as inputs for the second encoder to jointly model the distribution of the two views.

The integrated loss function of the fusion model is formulated as follows.

$$\begin{aligned} L_z = & KL\Big(q\big(z_U/\left(x_U, x_P, x_A\right)\big)\Big\|p\left(z_U\right)\Big) + \Big\|\left(x_U - \hat{x}_U\right) \odot l_U\Big\|_2^2 \\ & +KL\Big(q\big(z_P/\left(x_U, x_P, x_A\right)\big)\Big\|p\left(z_P\right)\Big) + \Big\|\left(x_P - \hat{x}_P\right) \odot l_P\Big\|_2^2 \\ & +KL\Big(q\big(z_A/\left(x_U, x_P, x_A\right)\big)\Big\|p\left(z_A\right)\Big) + \Big\|\left(x_A - \hat{x}_A\right) \odot l_A\Big\|_2^2 \end{aligned} \tag{9}$$

The final representation z is the concatenation of the hidden vector $g(z) = \langle z_{\mathrm{U}}, z_A, z_{\mathrm{P}} \rangle$. Based on the heterogeneous network embedding z_i and z_j, the predicted sentiment $\hat{s}_{ij}$ can be calculated through similarity measurement function as follows.

$$f(U_i, \mathrm{E}_j) = z_i^{\mathrm{T}} z_j + b \tag{10}$$

The loss function between the predicted sentiment and the ground truth is calculated as follows.

$$L_{s_{ij}} = \left(s_{ij} - \hat{s}_{ij}\right)^2 \tag{11}$$

Based on the above analysis, we formulate the overall loss function of NESA-fus as the sum of the fusion model loss (Eq. (9)) and the sentiment prediction loss (Eq. (11)):

$$L = \sum_{i \in V} L_z + \beta \sum_{j \in E} \sum_{i \in U} L_{s_{ij}} \tag{12}$$

where V is the set of nodes in the entire social network. Then, the final loss function of the NEBA-fus model for training can be formulated as follows.

$$\mathcal{L}\ominus = \sum_{i \in V} L_z + \beta \sum_{j \in E} \sum_{i \in U} L_{s_{ij}} + \gamma L_{reg} \tag{13}$$

where L_{reg} is an L2-norm regularized term to prevent overfitting and γ is a trade-off hyper-parameter. The proposed NESA-fus model is briefly summarized in Algorithm 1.

Algorithm 1 The learning algorithm of NESA-fus

```
Input:
    Three information networks ( $G_U$ , $G_P$ , $G_A$ )

Output: Predicted sentiment polarity $\hat{G}_P$ between users and entities;

        Learned embedding: the concatenation of $z_{\mathrm{U}}$ $z_{\mathrm{P}}$ $z_{\mathrm{A}}$
repeat
for i in V do
    Obtain U-U,U-E-P,U-E-A information description $x_{\mathrm{U}}$ , $x_P$ , $x_A$ from $G_U$ , $G_P$ , $G_A$
    Apply VAE encoder obtain highly nonlinear representations $z_U$ , $z_{\mathrm{P}}$ , $z_A$
    Apply VAE decoder obtain reconstruction data $\hat{x}_U$ , $\hat{x}_P$ , $\hat{x}_A$
    Desine aggregation function $g(\cdot)$ obtain final heterogeneous embedding $z$
    Desine sentiment function $f(\cdot)$ calculate sentiment polarity $\hat{s}_{ij}$
    Feed-forward calculate the loss by Eq. (12).
end for
    Calculate the whole loss L= by Eq. (13)
    Update model parameters via SGD on the loss L⩦
until NESA converges
```

Table 1 Confusion matrix

		True	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

Experiment

In the experiments, we first evaluate the parameters setting and sensitivity, and then evaluate the impact of different embedding networks on sentiment link prediction accuracy. Next the problems of link prediction are used to validate the quality of the embedding representation results. The sentiment link prediction accuracy further validates that the proposed scheme outperforms the traditional text-based sentiment prediction methods.

Database and Evaluation Metrics

To evaluate the performance of the proposed scheme, we adopt two data sets as follows.

Weibo dataset: The Weibo towards Movies dataset consists of three heterogeneous networks with 120,752 texts, 11,675 users, 1576 movies, 65,568 social links, and 38,235 attributes values.

Wiki-RFA dataset: Each node represents a Wikipedia administrator candidate, and each link denotes a voter/votee pair. The sign of the link, i.e., positive or negative, represents the voter votes for or votes against the votee.

In order to evaluate the impact of different embedding networks on sentiment link prediction accuracy, the quality of the embedding representation and the sentiment link prediction accuracy, precision, recall, F1-measure, accuracy, and AUROC are used as evaluation metrics. Four parameters of confusion matrix are shown in Table 1.

The evaluation metrics are calculated as follows:

Precision

$$P_{\text{precision}} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

Recall

$$R_{\text{recall}}=\frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

F1:

$$F1 = 2 \times \frac{P_{\text{precision}} \times R_{\text{recall}}}{P_{\text{precision}} + R_{\text{recall}}} \tag{16}$$

Accuracy:

$$A_{\text{accuracy}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{17}$$

TPR:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{18}$$

FPR:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{19}$$

The ROC (receiver operating characteristic) curve is a curve based on FPR and TPR. The area under the ROC curve is called AUROC, AUROC $\in [0, 1]$, and the higher the AUROC value, the better the performance. $P_{\text{precision}}$, R_{recall}, and F1 are used as evaluation metrics in the second and the fourth set of experiments to evaluate the impact of different embedding networks on sentiment link prediction accuracy, and the sentiment prediction accuracy respectively, and A_{accuracy} and AUROC are used as evaluation metrics in the third set of experiments to validate the quality of the embedding representations.

Parameters Setting and Sensitivity

NESA-fus involves a number of hyper-parameters. In this subsection we examine how the different choices of parameters affect the Accuracy of NESA-fus on Weibo dataset. Please note that except for the parameters being tested, all other parameters are set as default.

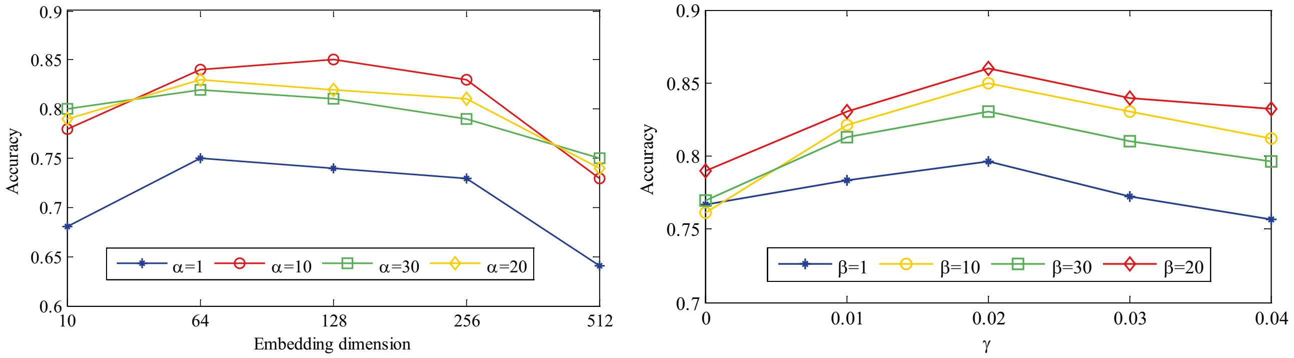
Embedding dimensions and reconstruction weight of non-zero elements α

By fixing $\beta = 20$ and $\gamma = 0.02$, Fig. 5 shows how the dimension of embedding layer in the VAEs of NESA-fus and the hyper-parameter α affect the performance.

From Fig. 5a we have the following two observations.

(1) The prediction accuracy increases at the beginning with the increase of dimensions, because more bits can encode more useful information. However, the performance slowly drops when the size of dimension further increases. This is because too large dimensionalities may introduce noises to mislead the subsequent performances.

(2) In VAE, non-zero elements carry more information than zero elements. In order to explain the control effect of α on non-zero elements, we have tested four values of α, which are $\alpha = 1, 10, 20, 30$, respectively. Figure 5a shows that the performance is initially improved with the increase of α from 1 to 10. This is because there are more non-zero elements when $\alpha = 10$. However, the performance drops when α continuously increases, because large α will lead to complete ignorance of the dissimilarity (i.e., zero elements) among



(a) dimension of embedding layers and α

(b) β and γ

Fig. 5 **a** Dimension of embedding layers and α. **b** β and γ. Parameter sensitivity: dimension of embedding layers, α, β, and γ

nodes. Based on the above analysis, we choose $\alpha = 10$ as the most appropriate value.

Balancing Parameter and Trade-off Parameter

By setting reconstruction weight $\alpha = 10$, embedding dimensions $d = 128$, Fig. 5b shows how the balancing parameter β and trade-off parameter γ in L2-norm regularization affects the performance.

We have tested four values of β, including $\beta = 1,\ 10,\ 20,\ 30$. Based on the curves in Fig. 5b, increasing the balancing parameter β from 1 to 20 can greatly improve the performance. The reason is that NESA-fus will concentrate more on the prediction error rather than the reconstruction error. However, when β is set to be a large value, such as $\beta = 30$, the accuracy decreases. This is because too large β breaks the trade-off among loss terms in the objective function.

The optimal parameters obtained from this section are used in the following experiments.

Network Embedding Analysis

In social networks, the purpose of network embedding is to project the information network into a low-dimensional vector space. In order to verify the effectiveness of adopting user, entity attributes, and social relationships for sentiment analysis, we conduct experiments to compare the sentiment analysis performance of several models. First, the U-E-P sentiment polarity network is considered as a common sign link prediction problem and used as a baseline model named as NESA$_N$. Furthermore, to validate the effectiveness of the social relationships on sentiment analysis, the U-U link information network is added on top of the baseline model as NESA$_S$. To validate the effectiveness of universal applicability of attribute information, the U-E-A attribute information network is added on top of the baseline model as NESA$_A$. At the end, both the U-U and the U-E-A networks are embedded on top of the baseline model as NESA-sep to further validate the superiority of combination. Model types are presented in Table 2.

As existing links in the original network are known and can serve as the ground truth, we randomly hide 50% of links in the U-E-P network and select a balanced test set (i.e., same number of positive links and negative links) out of them, while using the remaining network to train all models. We use Precision, Recall, and F1 as the evaluation metrics, and present the results in Table 3.

From Table 3, we have the following observations. First, with the addition of U-U link information network and U-E-A information network to the NESA$_N$ model, the accuracy of sign link prediction has been gradually improved. Compared with the NESA$_N$ model, the NESA$_A$ model has increased the Precision, Recall, and F1 values by 12.10%, 9.18%, and 10.61%, respectively. Compared with the NESA$_N$ model, the NESA$_S$ model has increased the Precision, Recall, and F1 values by 12.52%, 8.91%, and 10.75%, respectively. It is proved that the NESA$_S$ and NESA$_A$ models can improve the performance of sentiment link prediction, optimize the effect of sentiment analysis, and have universal applicability by introducing

Table 2 Comparison of different model types

	Model	Network Embedding		
		U-E-P	U-U	U-E-A
1	NESA$_N$	Yes	No	No
2	NESA$_S$	Yes	Yes	No
3	NESA$_A$	Yes	No	Yes
4	NESA-sep	Yes	Yes	Yes

Table 3 Results of different network embedding analysis

	Model			
	NESA_{N}	NESA_{S}	NESA_{A}	NESA-sep
$P_{\text{precision}}$ (Improved%)	0.719	0.809 (12.52)	0.806 (12.10)	0.874 (21.56)
R_{recall} (Improved%)	0.752	0.819 (8.91)	0.821 (9.18)	0.864 (14.89)
F1 (Improved%)	0.735	0.814 (10.75)	0.813 (10.61)	0.869 (18.23)

user relationships, user, and entity attributes. Second, for $\text{NESA}_{\text{-sep}}$ model, U-E-P sentiment polarity network, U-U link information network, and U-E-A attribute information network are introduced to realize the embedding and fusion of multiple networks for sentiment link prediction. When compared with the NESA_{N} model, NESA-sep has improved the Precision, Recall, and F1 values by 21.56%, 14.89%, and 18.23% respectively. Third, comparing the $\text{NESA}_{\text{-sep}}$ model with the NESA_{A} and NESA_{S} models, the Precision, Recall, and F1 values have been improved significantly.

In summary, these comparative experiments fully verify that the NESA-sep model proposed in this paper has significantly improved the sentiment analysis of Weibo by adopting user relationships and user entity attributes.

Link Prediction

In sentiment link prediction settings, our task is to predict the sign of an unobserved link between two given nodes. In this section, we use four state-of-the-art methods as baseline comparison schemes, in which the first three are network embedding methods, and the last one is a signed link prediction approach.

Unsigned Network Embedding

LINE [34]**:** Large-scale Information Network Embedding. It learns two embedding vectors for each node by preserving the first-order and second-order proximity of the network, respectively. Then the embedding vectors are concatenated as the final representation for a node.

Node2Vec [32]**:** Node2vec applies the Skip-Gram model on the node sequences generated by biased random walk and learns a mapping of nodes that maximizes the likelihood of preserving network neighborhoods of nodes.

SDNE [42]: Structural Deep Network Embedding is a semi-supervised deep model, which has multiple layers of non-linear functions, and is thereby able to capture the highly non-linear network structure. The second-order proximity and first-order proximity are used to capture the global network structure and to preserve the local network structure, respectively. By jointly optimizing them in the semi-supervised deep model, the method can preserve both the local and global network structure and is robust to sparse networks.

Signed Network Embedding

SHINE [38]**:** Signed Heterogeneous Information Network Embedding utilizes multiple deep auto-encoders to map each user into a low-dimension feature space while preserving the network structure.

Same as the previous set of experiments, we randomly hide 50% of links in the sentiment network and select a balanced test set out of them, while using the remaining network to train all baselines. We use Accuracy and AUROC as the evaluation metrics in link prediction task. The results are presented in Table 4.

As can be seen from Table 4, our proposed NESA-sep model always performs better than the baseline methods. Compared with baseline methods, the AUROC value of NESA-sep is increased by about 12.40%, 16.97%, 9.3%, and 5.04%, respectively. The accuracy is also better than that of other baseline methods, which proves the validity of the using three kinds of network in learning node

Table 4 Link prediction results (AUROC/Accuracy) on two datasets

Dataset	Metric	Unsigned network embedding			Signed network embedding	Our model	
		LINE	Node2vec	SDNE	SHINE	NESA-sep	NESA-fus
Weibo	Accuracy	0.793	0.788	0.802	0.815	0.827	0.852
	AUC	0.742	0.713	0.765	0.794	0.834	0.855
Wiki-RFA	Accuracy	0.802	0.791	0.807	0.825	0.837	0.863
	AUC	0.737	0.743	0.753	0.761	0.815	0.834

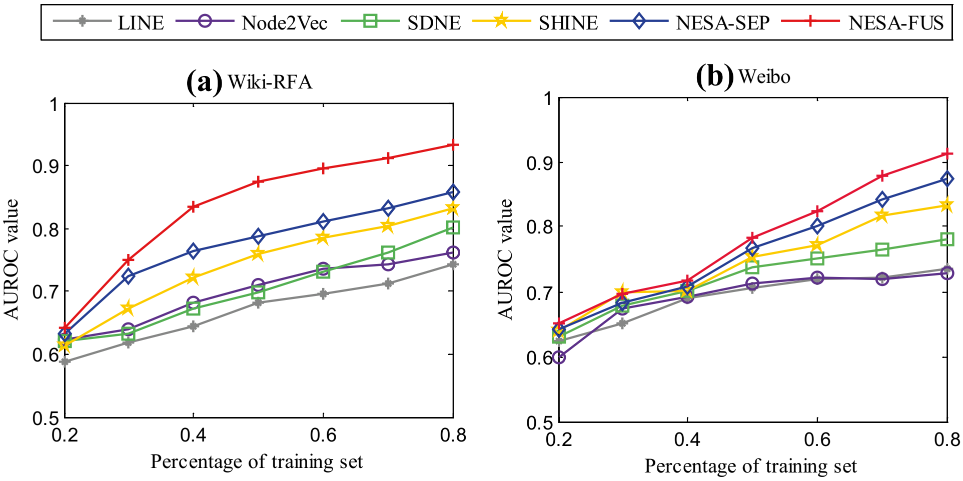


Fig. 6 Accuracy and AUROC on Weibo and Wiki-RfA for link prediction

representations for link prediction. Comparing with the NESA-sep model, the NESA-fus model achieves higher accuracy and AUROC values by 3.07% and 2.42%, respectively.

For a more fine grained analysis, we compare the performance by changing the percentage of training set from 20 to 80%. The results are presented in Fig. 6. As shown in Fig. 6, the NESA-fus model achieves significantly higher AUROC on two data sets when compared with the baseline algorithms. Specifically, for Weibo data, the AUROC of NESA-fus is 13.8%, 13.2%, 8.78%, and 6.34% higher than that of LINE, Node2Vec, SDNE, and SHINE, respectively. Among the three traditional network embedding methods, SDNE performs the best, while LINE and Node2Vec perform relatively poorly. The high performance of SDNE and SHINE, which also use autoencoder to learn the embedding of nodes, has further proved the advantages of autoencoder in extracting highly nonlinear representation of networks.

For fine-grained analysis of link prediction on the sparse network, we further tune the proportion of removed links from 50 to 100% in the original network, and then follow the aforementioned procedure to report the results of different network embedding methods over two datasets. The results are shown in Fig. 7. From Fig. 7, we can see that NESA-fus performs significantly and consistently better than baseline algorithms. And the margin between NESA-fus and other methods becomes larger when the network is sparser. It confirms the robustness of our proposed method in generating effective representations when the network is very sparse.

The above three groups of experimental results demonstrate that embedding results of our proposed methods are superior to LINE, Node2vec, SDNE, and SHINE models. In other words, integrating the sentiment polarity network, user link network, and attributes network for heterogeneous graph network embedding is effectively. At the same time, compared with NESA-sep, NESA-fus has jointly trained three information networks by considering their correlations, which can effectively improve the prediction performance.

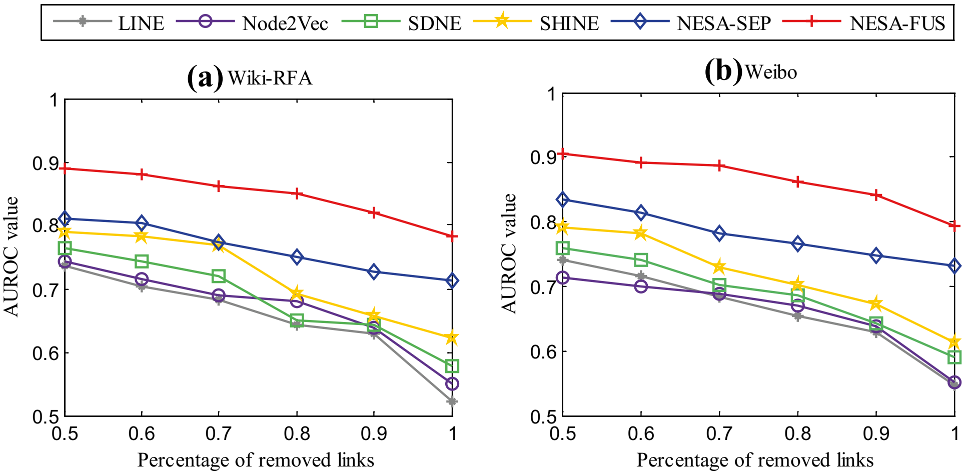


Fig. 7 Accuracy and AUROC on Weibo and Wiki-RfA for link prediction

Table 5 Setting of different model types

	Deep leaning	Social relation, Attributes
Word2Vec + CNN	Yes	No
Glove + CNN	Yes	No
ELMO	Yes	No
BERT	Yes	No
NESA-fus	Yes	Yes

Table 6 Parameter configuration for different algorithms

Algorithm	Parameter name	Parameter value
Word2Vec [14]	Window size	8
	training method	CBOW, negative sample
	Embedding dimension	200
GloVe [15]	Window size	5
	Embedding dimension	200
Elmo [16]	Learning rate	0.001
	Word embedding dimension	256
	Dropout rate	0.5
	Hidden layer size	128
	Batch size	128
BERT [17]	Dropout rate	0.1
	Hidden layer size	768
	Word embedding dimension	512
CNN [19]	Learning rate	0.001
	Word embedding dimension	200
	Dropout rate	0.5
	Filter windows	3,4,5
	Batch size	128

Sentiment Prediction Accuracy

In order to verify the effectiveness of the method based on the heterogeneous graph network embedding for latent sentiment link polarity prediction, in this section, we compare NESA-fus model with traditional deep representation models and the latest pre-training models.

The traditional depth representation model uses word vectors trained by Word2Vec [14] and GloVe [15], initializes words in sentences as model input, and then extracts sentence features by CNN [19]. Finally, via a full connection layer and a softmax layer, the model predicts the probability distribution of each sample in each category and takes the category with the highest probability as the classification result of the sample. The latest pre-training model uses ELMo [16] and BERT [17] two-stage models for sentiment classification. The first stage is the language model pre-training stage, where the word representation in sentences is obtained by unsupervised pre-training on a large amount of corpus. The second stage is the optimization stage of the downstream tasks, where through the supervised training on the sentiment analysis data, we can fine tune the word representation obtained in the first stage and complete the sentiment classification.

In this section, we also select a balanced sentiment data set to reduce the bias to a certain category, which ultimately affects the classification effect. The settings of different models are listed in Table 5. In the parameter setting process, we try to keep the consistency of the four text-based models in the parameter setting as much as possible; in the case where the consistency cannot be achieved due to the difference in the model structure, the optimal settings of the parameters are adopted. For the NESA-fus model, the optimal parameters obtained from Sect. 6.2 are adopted. The parameter configurations for different word embedding algorithms are listed in Table 6. We also use Precision, Recall, and F1 as the evaluation metrics, and present the results in Table 7.

The comparison results are listed in Table 7.

Through the comparative experiments among our proposed NESA-fus method, the traditional deep representation model (Word2Vec, GloVe), and the latest pre-training model (ELMO, BERT), we can get the following conclusions. First, the NESA-fus model is more favorable than traditional deep representation model (Word2Vec, GloVe) in Precision, Recall, and F1 values; compared with the pre-training ELMO model, the NESA-fus model has increased the Precision, Recall, and F1 values by 7.47%, 3.52%, and 5.90%, respectively. It is proved that the NESA-fus model can improve the performance of sentiment link polarity prediction. Second,

Table 7 Experiment results for sentiment prediction accuracy

	Models				
	Word2Vec + CNN	Glove + CNN	ELMO	BERT	NESA-fus
$P_{precision}$ (Improved%)	0.725 (15.03%)	0.773 (7.89%)	0.776 (7.47%)	0.838 (− 0.48%)	0.834
R_{recall} (Improved%)	0.733 (16.23%)	0.702 (21.36%)	0.823 (3.52%)	0.876 (− 2.73%)	0.852
F1 (Improved%)	0.729 (15.64%)	0.734 (14.85%)	0.796 (5.90%)	0.857 (− 1.63%)	0.843

when compared with the BERT model, the Precision, Recall, and F1 values of NESA-fus are slightly lower than that of BERT. We then compared the running time of different models subsequently.

In experiments, a machine with Nvidia Quadro P620 GPU and 48 CPU cores is used. For BERT and NESA-fus models the best performance on the validation set is chosen for evaluation. The runtime of BERT on Weibo data validation is 216 s in one epoch. Compared with Word2Vec, GloVe, and ELMO methods, BERT takes the longest time, due to its large amount of parameters. In comparison, the NESA-fus model is trained for a maximum of 100 epochs with a batch size of 256 nodes, and the runtime per mini-batch on the validation set is only 0.68 s.

In short, the Precision, Recall, and F1 values of NESA-fus model are slightly lower than that of BERT model, but it has a strong advantage in running time. The experiment results also validate the effectiveness of network embedding for sentiment sign link predictions.

Conclusion

Sentiment analysis is particularly important for psychology, education, sociology, business, politics, and economics. In this paper, we take sentiment analysis as a special sign link prediction problem and explore to combine User link information, User-Entity sentiment polarity information, and User-Entity attributes information for heterogeneous graph network embedding to predict latent sentiment links in social networks. In particular, a novel multi-network embedding method called NESA is proposed for more effective and efficient embedding. Our work is different from current sentiment classification researches which mainly determine the sentiment polarity of text from grammatical and semantic rules. In addition, it is different from the current multi-network embedding methods by considering interactions between the user links and user-entity attribute information. Furthermore, it also extracts users' highly nonlinear representations while preserving the structure of original networks. We conduct extensive experiments to evaluate the performance of the proposed scheme. Experimental results prove the competitiveness of NESA against multiple state-of-the-art baselines and demonstrate the effectiveness of adopting user link information and user-entity attribute information for embedding representation nodes in social networks to predict sentiment sign link polarities.

Funding This study was funded by the National Natural Science Foundation of China (No. 71502125).

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no competing interests.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Sánchez-Rada J, Iglesias C. Social context in sentiment analysis: formal definition, overview of current trends and framework for comparison. INFORM FUSION. 2019: 344–356.
2. Valdivia A, Luzn MV, Herrera F. Sentiment analysis in trip advisor. IEEE Intell Syst. 2017;32(4):72–7.
3. Jin Z, Yang Y, Liu Y. Stock closing price prediction based on sentiment analysis and LSTM. Neural Comput Appl. 2019:1–17.
4. Song K, Gao W, Feng S. Recommendation vs sentiment analysis: a text-driven latent factor model for rating prediction with cold-start awareness. The 26th International Joint Conference on Artificial Intelligence. 2017:2744–2750.
5. Sirisha DJ, Nandyala SP, Reddy PVB. A novel approach for sentiment analysis of public posts. Innovations in Computer Science and Engineering. 2019:161–167.
6. Brown PF, deSouza PV, Mercer RL, Pietra VJD, Lai JC. Class-based n-gram models of natural language. Comput Linguist. 1992;18(4):467–79.
7. Thewall M, Buckley K, Paltoglou G. Sentiment strength detection for the social web. J Am Soc Inf Sci Tec. 2012;61(1):163–73.
8. Pang B, Lee L, Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing. 2002:79–86.
9. Park S, Kim Y. Building thesaurus lexicon using dictionary based approach for sentiment classification. 2016 IEEE 14th International Conference on Software Engineering Research Management and Applications. 2016:39–44.
10. Zhang SX, Wei ZL, Wang Y. Sentiment analysis of Chinese micro-blog text based on extended sentiment dictionary. Future Gener Comp Sy. 2018;81:395–403.
11. Tang D, Wei F, Qin B, Zhou M, Liu T. Building large-scale twitter specific sentiment lexicon: a representation learning approach. In Proceedings of the 25th International Conference on Computational Linguistics. 2014:172–182.
12. Felipe V, Mario S, Sergio C, et al. Exploiting semantic relationships for unsupervised expansion of sentiment lexicons. Information Systems. 2020.
13. Wang S, Manning CD. Baselines and bigrams: simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. 2012. p. 90–94.
14. Wang G, Sun J, Ma J, et al. Sentiment classification: the contribution of ensemble learning. Decis Support Syst. 2014;57:77–93.
15. Mikolov T, Chen K, Corrado G., Dean J. Efficient estimation of word representations in vector space. In Proceedings of Workshop at International Conference on Learning Representations (ICLR) 2013.
16. Pennington J, Socher R, Manning CD. GloVe: Global vectors for word representation. In Proceedings of the Empirical Methods in Natural Language Processing. 2014:1532−1543
17. Peters M E , Neumann M , Iyyer M, et al. Deep contextualized word representations. In Proceedings of the56th Annual

Meeting of the Association for Computational Linguistics. ACL, 2018:2227–2237.

18. Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv: Computation and Language. 2018.
19. Kim Y. Convolutional neural networks for sentence classification. arXiv preprint arXiv. 2014:1746–1751.
20. Socher R, Perelygin A, Wu JY, Chuang J, Manning CD, Ng AY, et al. Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. 2013:1631–1642.
21. Zhu X, Sobhani P, Guo H. Long short-term memory over recursive structures. In Proceedings of the 32nd International Conference on Machine Learning, ICML. 2015: 1604–1612
22. Tang D, Qin B, Liu T. Learning semantic representations of users and products for document level sentiment classification. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. ACL, 2015:1014–1023.
23. Chen H, Sun M, Tu C, et al. Neural sentiment classification with user and product attention. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP. 2016: 1650–1659.
24. Chen T, Xu R, He Y, Xia Y, Wang X. Learning user and product distributed representations using a sequence model for sentiment analysis. IEEE Comput Intell Mag. 2016;11(3):34–44.
25. Dou Z. Capturing user and product information for document level sentiment analysis with deep memory network. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017: 521–526.
26. Wu Z, Dai X Y, Yin C, et al. Improving review representations with user attention and product attention for sentiment classification. Thirty-second AAAI conference on artificial intelligence. 2018.
27. Wang ZQ, Liang JY, Li R. Exploiting user-to-user topic inclusion degree for link prediction in social-information networks. Expert Syst Appl. 2018;108:143–58.
28. Xiao YP, Li XX, Liu YN, Liu H, Li Q. Correlations multiplexing for link prediction in multidimensional network spaces. Sci China Inform Sci. 2018;61(11):112103.
29. Huang W, Wang Q, Cao J. Tracing public opinion propagation and emotional evolution based on public emergencies in social networks. Int J Comput Commun. 2018;13(1):129–42.
30. Zhao K, Zhang Y, Zhang Y, et al. Learning from user social relation for document sentiment classification. International Conference on Database Systems for Advanced Applications. Springer, 2019: 86–103.
31. Leskovec J, Huttenlocher D, Kleinberg J. Signed networks in social media. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2010:1361–1370.
32. Grover A, Leskovec J, Node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016:855–864.
33. Perozzi B, Al-Rfou R, Skiena S. Deep walk: online learning of social representation. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2014:701–710.
34. Tang J, Qu M, Wang MZ, Zhang M, Yan J, Mei QZ. Line: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web. 2015:1067–1077.
35. Yuan S, Wu X, Xiang Y. SNE: signed network embedding. Pacific-Asia Conference on Knowledge Discovery and Data Mining. 2017:183–195.
36. Wang S, Tang J, Aggarwal C, Chang Y, Liu H. Signed network embedding in social media. In: Proceedings of the 2017 SIAM International Conference on Data Mining. 2017:327–335.
37. Shen X, Chung F. Deep network embedding for graph representation learning in signed networks [J]. IEEE Transactions on Cybernetics. 2020;50(4):1556–68.
38. Wang H, Zhang F, Hou M. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. 2018:592–600.
39. Wang J. Geometric structure of high-dimensional data and dimensionality reduction. Heidelberg: Springer; 2012. p. 203–20.
40. Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. Adv Neural Inf Process Syst. 2002;14(6):585–91.
41. Cao SS, Lu W, Xu QK. Grarep: Learning graph representations with global structural information. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. 2015:891–900.
42. Wang DX, Cui P, Zhu WW. Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016:1225–1234.
43. Wang X, Cui P, Wang J, Pei J, Zhu WW, Yang SQ, et al. Community preserving network embedding. In Proceedings of the 31st AAAI Conference on Aritificial Intelligence. AAAI, 2017:1–7.
44. Rozemberczki B, Davies R, Sarkar R, Sutton C. Gemsec: Graph embedding with self-clustering. In: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. 2019:65–72.
45. Chen JF, Zhang Q, Huang XJ. Incorporate group information to enhance network embedding. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. 2016:1901–1904.
46. Tu CC, Zhang WC, Liu ZY, Sun M. Max-margin deepwalk: discriminative learning of network representation. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence. IJCAI ,2016:3889–3895.
47. Chang SY, Han W, Tang JL, Qi GJ, Aggarwal CC, Huang TS, et al. Heterogeneous network embedding via deep architectures. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2015:119–128.
48. Huang X, Li JD, Hu X. Label informed attributed network embedding. In: Proceedings of the 10th ACM International Conference on Web Search and Data Mining. 2017:731–739.
49. Kingma DP, Welling M. Auto-encoding variational bayes arXiv preprint arXiv. 2013;1312:6114.
50. Liao L, He X, Zhang H, Chua TS. Attributed social network embedding. IEEE T Knowl Data En. 2018;30(12):2257–70.