

## Research Article

# A Sentiment Delivering Estimate Scheme Based on Trust Chain in Mobile Social Network

Meizi Li,<sup>1,2</sup> Yang Xiang,<sup>1</sup> Bo Zhang,<sup>2</sup> and Zhenhua Huang<sup>1</sup>

<sup>1</sup>College of Electronics and Information Engineering, Tongji University, Shanghai 200092, China

<sup>2</sup>College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 200234, China

Correspondence should be addressed to Bo Zhang; [sh.zhangbo@gmail.com](mailto:sh.zhangbo@gmail.com)

Received 5 July 2015; Revised 20 September 2015; Accepted 27 September 2015

Academic Editor: Jose Juan Pazos-Arias

Copyright © 2015 Meizi Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

User sentiment analysis has become a flourishing frontier in data mining mobile social network platform since the mobile social network plays a significant role in users' daily communication and sentiment interaction. This study studies the scheme of sentiment estimate by using the users' trustworthy relationships for evaluating sentiment delivering. First, we address an overview of sentiment delivering estimate scheme and propose its related definitions, that is, trust chain among users, sentiment semantics, and sentiment ontology. Second, this study proposes the trust chain model and its evaluation method, which is composed of evaluation of atomic, serial, parallel, and combined trust chains. Then, we propose sentiment modeling method by presenting its modeling rules. Further, we propose the sentiment delivering estimate scheme from two aspects: explicit and implicit sentiment delivering estimate schemes, based on trust chain and sentiment modeling method. Finally, examinations and results are given to further explain effectiveness and feasibility of our scheme.

## 1. Introduction

Recently, mobile social network has become one of the most popular platforms for people to daily communicate and share their information [1–3]. It allows users to post brief texts and express their opinions and then broadcasts them in public. Since the social network platforms are tied in mobile intelligent terminals, people can communicate through mobility of mobile social network platforms anywhere at any time with them. Such mobility of social network applications poses an ethereal new dimension that overcomes the limitations of time and space. That is, people can express their sentiments in mobile social network more freely than in other traditional media platforms. With the fission propagation pattern, sentiments would have larger and wider impacts in mobile social network, such as Twitter [2] and Sina Weibo [3] mobile social network. For example, a famous and reputational user's sentiment would have giant influences to his/her fans and followers. Furthermore, such impacts can be delivered through the relationship networks of the fans and bring wider and deeper impacts in mobile social network. That means sentiment influence in mobile social network is not

limited to direct impact but is more important on the indirect diffusion impacts among users. Under such explosion of public sentiment, there has been growing attention on the problem of sentiment analysis in mobile social network.

Traditionally, sentiment analysis studies have focused on mining sentiment from individual texts and the evaluation of their impact through users' direct relationships [4–6]. However, as sentiment is propagated and interacted mutually through user relationships, which is a significant manner of information and sentiment communication on social network platforms, we consider that this feature of sentiment delivering is essential and indispensable and cannot be ignored for the study of sentiment analysis in social networks. Thus, finding a way to estimate the sentiment delivering mechanism and evaluate the impact of such sentiment delivering is crucial for sentiment analysis in mobile social networks, which is the main goal of this paper.

In this work, there are two aspects which are taken into account for studying sentiment analysis: (1) sentiment in mobile social network, including explicit sentiment and implicit sentiment, can be delivered from one to another through user relationship and (2) the key manner which

drives sentiment delivering is the trustworthiness among users. Thus, by employing trust as the basis for our sentiment delivering evaluation, our main contributions in this work are as follows: (1) proposing an overview of sentiment delivering evaluation scheme is addressed and its related definitions, including trust chain, sentiment semantics, and sentiment ontology, are proposed; (2) presenting a formal trust chain model which links users and measures the trustworthiness is proposed; (3) presenting a sentiment modeling method is presented based on a set of rules which enable sentiment semantic to be calculated through definitions in sentiment ontology; and (4) proposing a sentiment delivering estimate scheme is therefore addressed, which is composed of explicit and implicit sentiment estimate based on trust chain and sentiment modeling.

The rest of the paper is organized as follows. Section 2 discusses related work of our study. Section 3 presents an overview of scheme and its related definitions. Section 4 addresses the trust chain model and its trust value calculation method. Sections 5 and 6 present the modeling method and delivering estimate schemes of explicit sentiment and implicit sentiment, respectively. Empirical results and discussion are given in Section 7. Finally, Section 8 concludes the paper.

## 2. Related Work

*2.1. Trust Computation.* Trust, as an inherent willingness of human beings, shows the emotional and rational confidence in between people. It is derived from judging trustworthiness by evaluating various facts which can lead to either the confidence or distrust. Over the past few years, many works have focused on computing trusted paths in network environment, such as P2P and mobile social networks (SNS) [6–10]. In trust evaluation, there are two core kinds of trust as direct trust and indirect trust. Direct trust is used for reflecting the trustworthiness between direct connected users, while indirect trust is used widely in long path connected users through intermediate users [6]. Typically, many historical data based on direct interactions, such as behaviors, comments, or other various evidences, are used to calculate the direct trust degree among users [6]. In most cases, the direct trust reflects the subjective trustworthy feelings towards the targets through the historical direct interactions. See-To and Ho [7] propose a method to evaluate the influence of trust on oral comment in mobile social network. Wu and Chiclana [8] present an approach to computing trust that users have all agreed on in mobile social network. Wang and Gui [9] select transaction nodes in mobile social network and compute trust between them. Based on real-time content, Li et al. [10] develop a community information recommendation system which takes advantage of user experience. Peiyun et al. [11] propose an algorithm for computing direct trust value and furthermore develop a Web service trust model based on mobile social network dynamic feedback. Qiao et al. [12] propose a context-based trust computation method, where the trust generation principle in the psychology is employed. In addition, the indirect trust is to evaluate trustworthiness through a deliverable perspective of trust, concerning the indirect connected relationships among users [13–15]. From

such point, direct trust evaluation is also the foundation of indirect trust evaluation. In most cases, the calculation of direct and indirect trust is discussed based on a graph model: direct trust can be seen as a direct connected edge between vertices, while the indirect trust is seen as a long path composed of direct trust in graph model. Therefore, the path among indirect connected users is an essential factor for indirect trust evaluation. Javier Ortega et al. [13] propose a method to compute a ranking of the users in a mobile social network and propagate both positive and negative opinions of the users. Then the opinions from each user about others can influence their global trust score, including direct and indirect trust views. Qureshi et al. [14] propose a decentralized framework and the related algorithms for trusted information exchange and social interaction among users based on the dynamicity aware graph relabeling system. The Bellman-Ford algorithm computes trust based on direct witness interaction trust judgments [15]. It generates a trust graph on the basis of the trust link between two peers who have direct interaction. Each peer can submit or renew their trust judgments of others based on new direct interactions. Further, the trust between peers is constantly updated by compounding old and new trust judgments. In addition, the algorithm admits the most trustable path for trust computation; it deems a long path to be untrustworthy. Golbeck proposed TidalTrust that gets trust in mobile social networks using numeric trust values [16]. It utilized the shortest path based on the breadth-first search. Further, TidalTrust can be used to retrieve accurate information from the highest trust adjacent nodes.

However, most of these above methods focus on either direct trust or reputation computation and do not take into account the chain relationship of trust among users and its complex path topology. Different from traditional trust modules, there are the following considerations in this study: (1) since the sentiments among users are considered to be delivered through the user relationships, the proposed trust model in this study therefore takes chain relationships among users into account to reflect the indirect confidence which can be used to measure the probability of sentiment delivering through users by their mutual trustworthiness; (2) for direct trust calculation, that is, atomic trust chain in this work, we also use the weighted average method to calculate them, which is similar to other traditional works. However, we introduce a new factor of community, which widely exists in mobile social network, for weights evaluation. That is, the direct trust calculation is impacted by the facts whether the users are in same community or not; and (3) with respect to indirect trust, the proposed trust model calculates it through user indirect relationships. The relationship composition among users, that is, the path topological information among users in social network graph, is complex and various. Unfortunately, few attentions have been paid to the complex path composition for indirect trust. That is, there are many different accessible paths between users and each accessible path manifests a delivering confidence. Then, the proposed trust model aims to calculate the trust chain with taking all paths confidence into account comprehensively in mobile social network, which is also one of the main motivations which we work in this study.

**2.2. Sentiment Analysis.** Sentiment analysis has been extensively studied in recent years and various methodological schemes have been proposed for sentiment extraction in Web, image, text, or other areas [3, 4]. Sentiment analysis has been utilized in applications such as news tracking and summarizing, online forums, file sharing, chatting rooms, and blogging [17]. There are two typical classes for sentiment analysis as machine learning based approach and semantic based approach [17]. In machine learning based approach, a mass of material with real emotions are provided to machine for training machine to learn the rule of understanding and detecting sentiment [4, 17–19], such as support vector machine (SVM), k-means method, Naive Bayesian (NB), LDA, and Max Entropy (ME). In semantic based approaches, semantic analysis related methods are utilized for understanding the characters of sentiment [5, 20–24]. Typical methods of semantic based approach include natural language processing method (NLP), keyword based method, emotion dictionary based method, ontology based method, and computational linguistics for sentiment analysis. In the above semantic based sentiment analysis methods, knowledge about sentiment classification and representation is essential and indispensable for detecting and identifying sentiment. In addition, some combined methods, which use the above machine learning and semantic analysis based method together, are proposed for sentiment analysis to improve the precision of results [25]. Besides explicit sentiment mining, implicit sentiment analysis has been paid many attentions in mobile social network [26–28]. Su et al. propose a mutual reinforcement principle, which is based on opinion evaluation through product feature words and opinion words, to mine the hidden sentiment associations [26]. Athar and Teufel explore methods to automatically identify all mentions of a paper and then classify explicit and implicit mentions for sentiment detection [27]. Balahur et al. present an approach towards automatically detecting implicit sentiment from contexts in which no clues of sentiment appear based on a commonsense knowledge base (named EmotiNet) [28, 29]. However, all these methods are still based on existing materials and cannot estimate implicit sentiments through a delivering view.

In mobile social network, sentiment has been extracted for further analyzing in different platforms, such as Twitter [23, 30, 31], MySpace [32], Facebook [33], YouTube [34], and other platforms [26, 35, 36]. However, most works of sentiment analysis focus on detecting and mining explicit sentiments from objects of texts, videos, or audios. In our consideration, sentiment analysis should be studied in a view of its dynamic characteristic. Different from other works, there are the following considerations for sentiment analysis in our study: (1) the communication nature of mobile social network enables users to interact and share their opinions and emotions freely in the platforms, which means that sentiment interactions with the effect of interactive influence are one of the most significant functions of mobile social network; (2) sentiment can be delivered through users relationships and then how to model the delivering process and evaluate the degree of delivering is a crucial challenge, and (3) besides explicit sentiment, there are many implicit sentiments which

are kept by users and are not expressed explicitly, and then these implicit sentiments should be recognized in sentiment analysis.

### 3. Overview and Related Definitions

The guiding principles we take in this work are as follows: (1) there are two types of sentiment in mobile social network as explicit sentiment and implicit sentiment; (2) explicit sentiment, which is expressed in users posts, comments, or other ways directly, can be modeled through a formal format and calculated for its delivering estimate; (3) implicit sentiment, which has not been expressed visibly in mobile social network texts, can be estimated by explicit sentiment delivered through users trustworthy relationships.

Therefore, we address our scheme in mobile social network based on the trust chains among users. The proposed scheme in this study is composed of three phases as (1) a trust chain establishing, sentiment modeling, and sentiment delivering estimate. As shown in Figure 1(a), a network model of mobile social network is constructed based on graph theory. In such graph, users are regarded as vertices while their relationships are edges. Then, we can see that there would be trust chains formed by users' past interactions and related data in social graph, which is shown in Figure 1(b). Based on trust chains, sentiments toward topic of users can thereby be delivered in mobile social network, which reveal the impacts of sentiment in explicit and implicit ways. In Figure 1(c), the explicit sentiment, which comes from user A, can be delivered to B and C (in green solid lines) through the trust chain  $C \rightarrow B \rightarrow A$  (in blue solid lines). And then, such influential explicit sentiment would in turn induce others to express their different sentiments (in red solid lines). Likewise, Figure 1(d) shows an example of implicit sentiment delivering among users. Due to the fact that users of D and E have not expressed their sentiments toward a topic in explicit way, we can evaluate the implicit sentiments of them according to explicit sentiment from A because there is a trust chain  $E \rightarrow D \rightarrow A$  (in blue solid lines) between them. The explicit sentiment of A can be delivered to D and E in implicit way (in green dot lines) and then generate implicit sentiments of D and E towards the topic (in red dot lines).

Here, we first address related definitions in this work.

*Definition 1.* Trust chain model can be defined as  $\Omega = (N, F, C, T)$ , where  $N$  denotes nonempty set of user nodes in trust chain and the user nodes can be divided into three roles as: source user nodes  $N_S$ , intermediate user nodes  $N_I$ , and target user nodes  $N_E$ ;  $F \subseteq N \times N$  denotes the finite set of atomic trust chain;  $C \subseteq F \times F \cup F \parallel F$  denotes then combined trust chain which is composed of atomic trust chain and symbols of  $\times$  and  $\parallel$  denotes serial trust chain and parallel trust chain, respectively;  $T : F \rightarrow [0, 1] \cup C \rightarrow [0, 1]$  denotes trust value of atomic trust chain or combined trust chain.

Through Definition 1, we can describe the direct and indirect trust relationships between users according to the topology of chain combination formally.

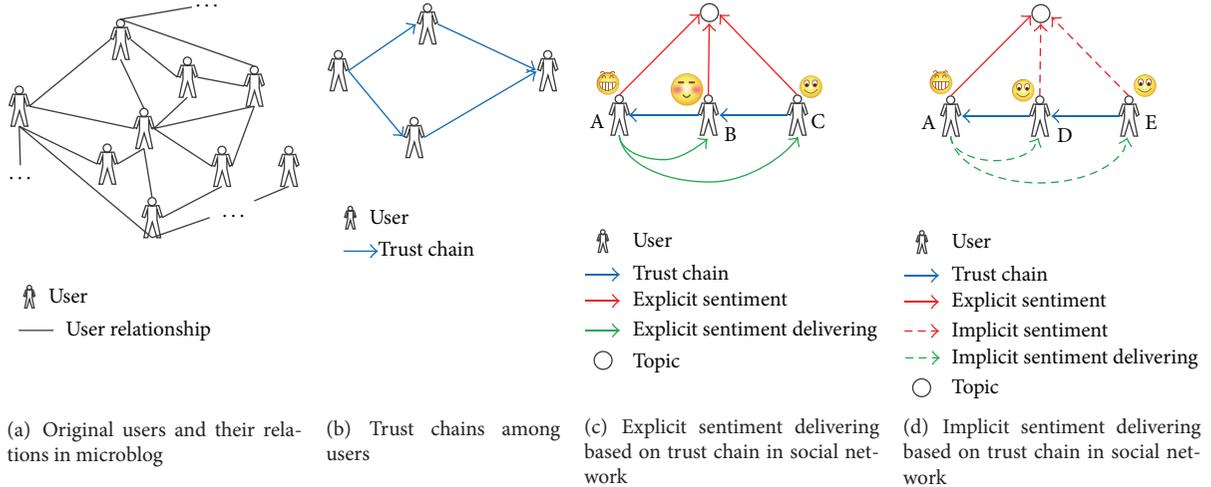


FIGURE 1: Example of sentiment delivering based on trust chain in mobile social network.

**Definition 2.** Sentiment in mobile social network is the opinion which user keeps to a topic or an event and can be defined as  $SE = \{se_1, se_2, \dots\}$ , in which  $se_i$  denotes one dimension of sentiment of user.

From Definition 2, we can see that the sentiment in this work is modeled as a multiple-dimension entity to describe user's diverse feelings. In the set of  $SE$ , we defined that each dimension of sentiment is a 3-tuple of  $se_i = \langle type, Related, level \rangle$ , where *type* is the sentiment type; *Related* is the set of sentiment which includes related sentiment and the calculating rules;  $level \in [0, 1]$  is the strong degree of sentiment.

For example, a user  $u_i$  has his/her sentiment semantics as  $SE(u_i) = \{se_1 : delight, se_2 : worry\}$  which includes two sentiment dimensions as  $se_1 : delight$  and  $se_2 : worry$ . According to Definition 2, the two sentiment dimensions can be formally described as follows:

$$\begin{aligned}
 se_1 : delight &= \langle type : happiness, Related : \\
 &\{joyful \mid \langle SMR, TAR \rangle, ecstasy \mid \langle SMR, TMR \rangle\}, \\
 &level : 0.9 \rangle, \\
 se_2 : worry &= \langle type : fear, Related : \\
 &\{sadness \mid \langle SMR, TAR \rangle, hate \mid \langle TMR \rangle\}, level : 0.3 \rangle,
 \end{aligned} \tag{1}$$

where the type of  $se_1 : delight$  is *happiness*; related sentiments are *joyful* with calculating rules of SMR and TAR and *ecstasy* with calculating rules of SMR and TMR; the level value of  $se_1 : delight$  is 0.9. Likewise, the type of  $se_2 : worry$  is *fear*; related sentiments are *sadness* with calculating rules of SMR and TAR and *hate* with calculating rules of TMR; the level value of  $se_2 : worry$  is 0.3. Then, the sentiment dimension calculation can be executed based on given rules, such as SMR, TAR, or TMR, in the formal definitions. Details of calculation rules are discussed in later sections.

Similar to the above example, sentiment can reflect user's multiple-dimension feeling or emotion focusing on a specific event or topic. It is apparently easy to evaluate one's explicit sentiment according to his explicit posting text, behaviors, comments, or other direct evidences. For example, if a user approved, posted positive comments, followed the topic, and forwarded posts to a topic in a near past, he might highly keep a positive sentiment about the topic. However, many users do not express their sentiments through their direct evidences. These potential sentiments, named implicit sentiment in this study, cannot be measured directly based on their past data. But, in our view, sentiment is delivered through users' relationships. That is, we can estimate implicit sentiment through relationships among users. For example, a user A, who has no direct evidence to express his sentiment to a topic, keeps a very high trust with his friend B, who keeps a strong positive sentiment to the topic. In such context, we can estimate that A might be likely to keep positive feeling because his trustworthy friend B does so. In this example, implicit sentiment can be delivered through users' trust relationship, which is also used for implicit sentiment estimation in this work.

From above consideration, we define two kinds of sentiment as explicit sentiment and implicit sentiment, which is reflected through users' explicit and implicit behaviors and evidences, respectively. Correspondingly, the calculation methods are based on the following rules: explicit sentiment degree is measured according to the direct sentiment evidences in past; implicit sentiment degree is calculated based on an estimate of trust chain delivery.

**Definition 3.** Sentiment ontology is defined to describe sentiment in a formal and normalized way for enabling sentiment to be understood by system automatically. Sentiment ontology is a 3-tuple as  $SO = \langle Class, Relation, Rule \rangle$ , where *Class* is a set of sentiment classes; *Relation* is the set of relationships among sentiment classes; and *Rule* is the set of representation and calculation rules for sentiment.

For example, there is a subpart of sentiment ontology as follows:

$$\begin{aligned}
 SO = \langle & \text{Class} : \{ \text{Root}, \text{happiness}, \text{sadness}, \text{anger} \}, \\
 & \text{Relation} : \{ \text{parent-child} \langle \text{Root}, \text{happiness} \rangle, \\
 & \text{parent-child} \langle \text{Root}, \text{sadness} \rangle, \\
 & \text{parent-child} \langle \text{Root}, \text{anger} \rangle \}, \text{Rule} : \{ \text{CMR}, \text{SMR}, \\
 & \text{TMR}, \text{TAR}, \text{MER} \} \rangle, \quad (2)
 \end{aligned}$$

where the subpart of sentiment ontology contains four classes as *Root*, *happiness*, *sadness*, and *anger*; the classes of *happiness*, *sadness*, and *anger* are all the direct children of *Root* according to the *parent-child* relation description; that is, *happiness*, *sadness*, and *anger* are brothers in ontology, and the calculation rules in sentiment ontology are five rules given in later section.

Here, we have the following related explanations for sentiment ontology:

- (1) Sentiment ontology describes sentiment in a tree structured model. In such tree model, each noneleaf class has just one parent and at least one child while each leaf class has just one parent and no child.
- (2) Each class has its layer number (*layer*) which indicates the distance from the class to the root of the tree.
- (3) There is a mapping function between sentiment dimension and sentiment ontology for describing the semantic relation as  $f: se_i.type \rightarrow SO.class_j$ .
- (4) The set of *Rule* in ontology aims to provide special calculation rules which can be used in sentiment migration or changing. The rule can be described as  $r(class_i, class_j) = class_k$ , which means that two classes,  $class_i$  and  $class_j$ , can be changed into a new class,  $class_k$ , under the above rule  $r(class_i, class_j)$ .

For the rules in ontology, we here have an example for its calculation as follows;

$$SMR : r(\text{terrified}, \text{worry}) = \text{disturbed}, \quad (3)$$

where the ontology defines a sentiment migration rule for calculating two classes, *terrified* and *worry*, and then gets a new class *disturbed*. Also, there are additional conditions for the above calculating rule and we will discuss them in later section.

Figure 2 shows an example of sentiment ontology with the above related explanations and its mapping with sentiment dimensions. In this example, we can see that the layer number of class 6 is 2 and it has a mapping function with sentiment

$se_3$ . Also, we can get the layer numbers and semantic relations as

$$\begin{aligned}
 \text{layer}(\text{class2}) &= 1 \\
 \text{layer}(\text{class5}) &= 2 \\
 \text{layer}(\text{class8}) &= 3, \\
 f: se1.type &\rightarrow SO.class2 \\
 f: se2.type &\rightarrow SO.class5 \\
 f: se4.type &\rightarrow SO.class8. \quad (4)
 \end{aligned}$$

Sentiment ontology can provide logic basis for sentiment evaluation and make the evaluation reasonable on knowledge layer. In this study, we assume that all sentiments can be described by sentiment ontology. Therefore, user's explicit sentiment can be modeled while the initial dimensions of sentiment are extracted from user's text, comments, and behaviors in mobile social network. Notice that, in this study, the work of analyzing initial dimensions of sentiment is not in our consideration and would be discussed in our other works.

#### 4. Trust Measurement of Trust Chain Model

Here, we propose the model of trust chains in detail based on their different network topologies and their trust value calculation methods. According to Definition 1 and the composition method of trust chain, we divide trust chain into four cases according to the path composition as atomic, serial, parallel, and combined trust chains.

**4.1. Atomic Trust Chain.** In atomic trust chain, there is no intermediate node between two nodes. That means the trust value is a kind of direct trust from one node to another. For our proposed atomic trust chain, we also use a weighted average method by past historical data, which is widely used in many existing trust models. However, different from the other existing trust model, we here introduce a new fact of community for weight evaluation in trust calculation. The factor of community is not included in most traditional direct trust computation methods since community is a specific entity in social network environment. It is very common that users locate in community of SNS environment and the identification of community implies one's trustworthiness to some degree. Therefore, trustworthiness level towards a user's community is crucial for direct trust calculation, including two cases: two users in the same community and two users in different communities. The former means that users have the same community identifications and their atomic trust chain is established based on similar past experiences directly, while the latter means that users have different identifications and their atomic trust chain is derived from the subjective trust of the whole group which the other belongs to.

Therefore, based on past historical data, trust value of atomic trust chain can be directly calculated by mutual interaction records, relationships, and community identities in atomic trust chain. Let there be two nodes  $N_1$ ,  $N_2$  and community set  $G_1, G_2$  ( $(N_1 \in G_1) \wedge (N_2 \in G_2)$ ),  $f_i(N_1, N_2)$

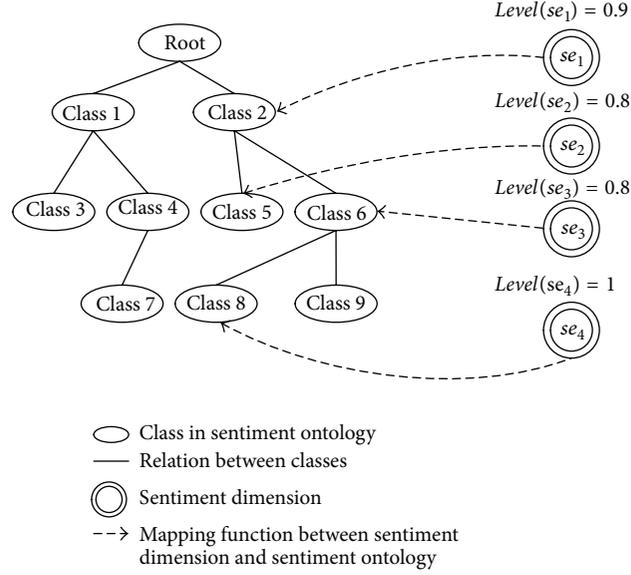


FIGURE 2: An example of sentiment ontology and its mapping with sentiment dimensions.

denotes judgment which is delivered from  $N_1$  to  $N_2$ . We denote  $T(N_1, N_2)$  as trust value of atomic trust chain from  $N_1$  to  $N_2$  after  $m$  times of judgment. And then,  $T(N_1, N_2)$  can be calculated as follows:

$$T(N_1, N_2) = \begin{cases} 0 & n = 0 \\ \frac{\sum_{i=1}^n (f_i(N_1, N_2) \times \bar{k}_i)}{n} & n \geq 1. \end{cases} \quad (5)$$

In the above equation, we give the impact factor  $\bar{k}_i$ , which denotes impact degree for each judgment  $f_i(N_1, N_2)$  to the accumulated trust value  $T(N_i, N_j)$ .

$\bar{k}_i$  can be calculated as follows:

$$\bar{k}_i = \begin{cases} 1 & \text{evaluation } f_i(N_1, N_2) \text{ happened in the same community} \\ e^{-(1-|G_1 \cap G_2|/|G_1 \cup G_2|)} & \text{evaluation } f_i(N_1, N_2) \text{ happened in the different communities.} \end{cases} \quad (6)$$

**4.2. Serial Trust Chain.** Serial trust chain means that there would be a serial path from source node to target node and the path has the following features: (1) for source node, its out-degree is 1 and in-degree is 0; (2) for the target node, its out-degree is 0 and in-degree is 1; (3) for each intermediate node, its out-degree is 1 and in-degree is 1. Thereby, serial trust chain can be considered as the composition of atomic trust chain and the trust is transmitted one by one in such chain so that there would be an indirect trust from source node to target node.

Then, we propose the value calculation method for serial trust chain. In our trust model, serial trust chain reflects an indirect trust among users. We introduce the principle of Bellman-Ford algorithm [15], which deems a long path to be untrustworthy, for our proposed model of indirect trust calculation. That is, the trust value of serial trust chain would be decreased with the depth of trust chain increasing. Likewise, such rational is also applied for indirect trust of parallel trust chain and combined trust chain.

Let there be source node  $N_S$ , target node  $N_E$ , and intermediate node  $N_I^k$  in serial trust chain, and  $T(N_i, N_j)$  which denotes trust value of atomic trust chain in the serial trust chain. Therefore, we can calculate the trust value of serial trust chain as follows:

$$T_C(N_S, N_E) = \frac{1}{\text{depth}(N_E)} \left[ T(N_S, N_I^1)^{\text{depth}(N_E)} + \sum_{k=2}^{\text{depth}(N_E)-1} T(N_I^k, N_I^{k+1})^{\text{depth}(N_E)-k+1} + T(N_I^{\text{depth}(N_E)-1}, N_E) \right]. \quad (7)$$

Here function  $\text{depth}(N_E)$  denotes the depth of serial trust chain; namely,  $\text{depth}(N_E) = |\Omega \cdot N_I| + 1$ . We can see that the deeper the depth of trust chain is, the weaker the trust value among users is. That means longer trust chain would be

punished due to the fact that trust can damp with the number increasing of intermediate node.

**4.3. Parallel Trust Chain.** It is likely that there are two or more trust paths from source node to target node and there is no intersection node among the paths. In that case, we call the trust chain as parallel trust chain. That means parallel trust chain can be considered as composition of several serial trust chains between same source node and target node. Likewise, parallel trust chain has the following features: (1) for source node, its out-degree is  $m$  ( $m \geq 2$ ) and in-degree is 0; (2) for target node, its out-degree is 0 and in-degree is  $m$  ( $m \geq 2$ ); (3) for each intermediate node, its out-degree is 1 and in-degree is 1.

Due to multiple serial trust chains, we must consider all the impacts of them in calculation of trust value instead of calculating them in an average method simply. Here, we proposed a local credit trust evaluation method for trust value calculation. In this method, only those trust chains in which the direct neighbor nodes keep high trust values with source node would be considered total parallel trust value calculation.

Let there be  $m$  ( $m \geq 2$ ) serial trust chain between source node  $N_S$  and target node  $N_E$ .  $C_i$  denotes each serial trust chain and  $T(C_i)$  represents trust value of serial trust chain  $C_i$ . For each  $C_i$ ,  $T(N_S, N_i^1)_i$  denotes the direct trust value of atomic trust chain from  $N_S$  to its neighbor node  $N_i^1$  in  $C_i$ . We set a threshold  $\eta$  ( $\eta \in [0, 1]$ ) and define the serial trust chain which has condition  $T(N_S, N_i^1)_i \leq \eta$  as ignorable chain. All ignorable trust chains are excluded in our local optimized trust evaluation. Then, the trust value of parallel trust chain can be calculated as follows:

$$T(N_S, N_E) = \frac{1}{m} \left[ \left( \sum_{T(C_i) \geq 0.7} T(C_i)^{1/m_1} \right) + \left( \sum_{0.4 < T(C_i) < 0.7} T(C_i) \right) + \left( \sum_{T(C_i) \leq 0.4} T(C_i)^{m_2} \right) \right]. \quad (8)$$

In the above equation, serial trust chains are divided into three cases as follows: (1) trust chains whose values are greater than or equal to 0.7 are considered as positive views toward target node and would be enlarged by exponential weighting as  $1/m_1$  (here,  $m_1$  denotes the number of positive views); (2) trust chains whose values are between 0.4 and 0.7 are considered as neutral views toward target node and would be treated as their original values; (3) trust chains whose values are less than or equal to 0.4 are considered as negative views toward target node and would be weakened by exponential weighting as  $m_2$  (here,  $m_2$  denotes the number of negative views).

**4.4. Combined Trust Chain.** Combined trust chain is composed of the above three kinds of trust chain. That is, there would be a complex path composition from source node to target node. Here, we address a scheme of combined trust chain value computation. In this method, several rules would be followed in calculation as follows:

- (i) Local credit rule (LCR): only those trust chains in which the direct neighbor nodes keep high trust values with source node would be considered total parallel trust value calculation. That is, the atomic trust chain whose trust value is lower than a threshold can be ignored in the combined trust chain. Meanwhile, their successor trust chains are ignored since the paths are broken.
- (ii) Serial calculating rule (SCR): if there is a combined trust chain from  $N_S$ 's indirect neighbor nodes to  $N_E$ , it would be eliminated as a serial one recursively.
- (iii) Parallel calculating rule (PCR): if there are two or more direct neighbor nodes of  $N_S$ , the combined trust chain would be reconstructed as parallel trust chain for  $N_S$  with its neighbors recursively.

An example of our scheme is shown in Figures 3(a)–3(d) and we can see how it works to calculate the combined trust chain. In original combined trust chain, there is a complex combined path from source node to target node through intermediate nodes of A–M. We use our proposed rules to reconstruct the combined trust chain as follows: (1) in Figure 3(b), the atomic trust chains, including source node  $\rightarrow$  B, source node  $\rightarrow$  C, D  $\rightarrow$  I, and F  $\rightarrow$  I, are ignored according to LCR (ignored chains are in dotted lines in Figure 3(b) and threshold  $\eta$  is set as 0.3). Meanwhile, atomic trust chains, including B  $\rightarrow$  G, C  $\rightarrow$  I, and I  $\rightarrow$  M, are ignored because their preorder trust chains are ignored; (2) the part of combined trust chain, E  $\rightarrow$  target node, is parallelized into two parallel trust chains by PCR: E  $\rightarrow$  G  $\rightarrow$  D  $\rightarrow$  target node and E  $\rightarrow$  K  $\rightarrow$  target node; (3) the part of E  $\rightarrow$  target node is eliminated as a serial trust chain as E  $\rightarrow$  K  $\rightarrow$  target node by SCR. Therefore, the original combined trust chain can be reconstructed as a parallel trust chain which is composed of source node  $\rightarrow$  A  $\rightarrow$  G  $\rightarrow$  K  $\rightarrow$  target node ( $C_1$ ), source node  $\rightarrow$  E  $\rightarrow$  K  $\rightarrow$  target node ( $C_2$ ), source node  $\rightarrow$  F  $\rightarrow$  H  $\rightarrow$  L  $\rightarrow$  target node ( $C_3$ ), and source node  $\rightarrow$  D  $\rightarrow$  J  $\rightarrow$  M  $\rightarrow$  target node ( $C_4$ ). Then, we can calculate the combined trust chain as follows:

$$\begin{aligned} T(C_1) &= \frac{1}{4} \times [1^4 + 0.9^3 + 0.7^2 + 0.8] = 0.75, \\ T(C_2) &= \frac{1}{3} \times [0.9^3 + 0.88^2 + 0.8] = 0.77, \\ T(C_3) &= \frac{1}{4} \times [0.7^4 + 1^3 + 0.9^2 + 1] = 0.76, \\ T(C_4) &= \frac{1}{4} \times [0.8^4 + 0.9^3 + 1^2 + 0.7] = 0.71, \\ T(N_S, N_E) &= \frac{1}{4} \times [0.75^{1/3} + 0.77^{1/3} + 0.76^{1/3} + 0.71] \\ &= 0.86. \end{aligned} \quad (9)$$

## 5. Method of Sentiment Modeling

**5.1. Rules for Sentiment Modeling.** Sentiment is expressed in users' texts through evidences, such as approving and

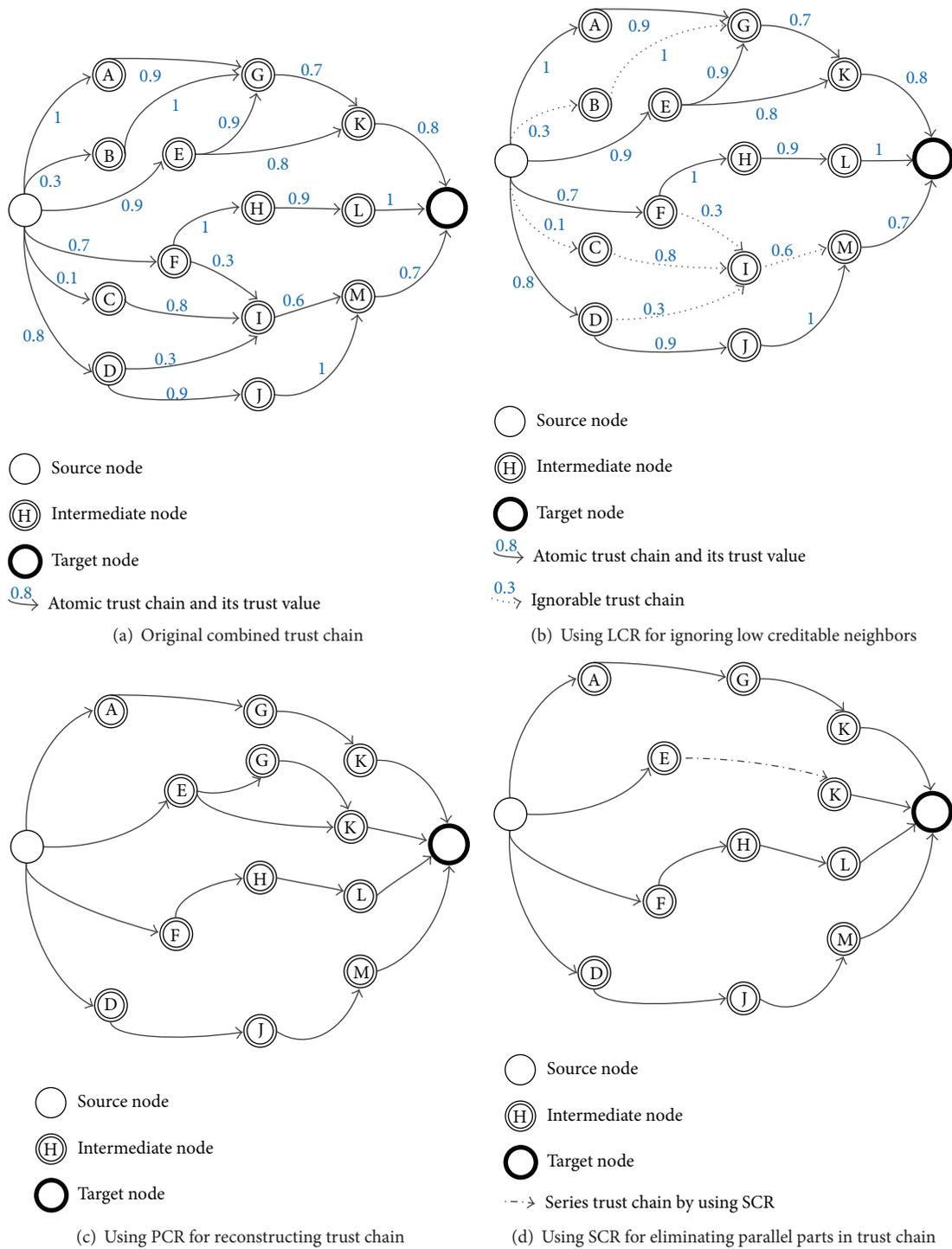


FIGURE 3: Example for combined trust chain calculation.

forwarding with positive comments. In this work, sentiment is described formally by a sentiment ontology which provides formal and normalized knowledge and then enables the sentiment to be understood by system automatically. The sentiment ontology is built up manually because there is no existing relevant ontology which is suitable for supporting the following sentiment calculation rules. All the classes and

their relationships are set manually according to knowledge of sentiment category.

Each dimension of sentiment,  $se_i = \langle type, Related, value \rangle$ , can find a corresponding ontology knowledge description through mapping function. These dimensions of sentiment are initial inputs for sentiment model. Then, we propose several rules to eliminate related dimensions

and ensure that the sentiment model is stable. Assume that the initial sentiment is  $SE.in = \{se_1, se_2, \dots\}$ . Then, the dimensions can be normalized and calculated by the following rules.

(1) *Class Merging Rule (CMR)*. Sentiment dimensions can be merged if and only if there are ancestor-descendant relationships or parent-children relationships between their classes in sentiment ontology.

Here, we first define an operator, denoted  $\odot$ , to calculate dimension changing based on TRP. Let there be two dimensions as  $se_i = \langle type, Related, level \rangle$  and  $se_j = \langle type, Related, level \rangle$ , which satisfy CMR calculation, and we denote the new dimension as  $se_i \odot se_j = se_k$ . Then, the calculation based on CMR can be composed of the following four phases: firstly, we get the sentiment ontology mapping functions  $f : se_i.type \rightarrow SO.class_i$  and  $f : se_j.type \rightarrow SO.class_j$ ; secondly, values of layer and level are obtained according to ontology tree, and then value of  $se_k$  is calculated as

$$\begin{aligned} value(se_k) &= value(se_i) \odot value(se_j) \\ &= \frac{(level(se_i) + lay(class_i)) + (level(se_j) + lay(class_j))}{2}. \end{aligned} \quad (10)$$

Then, we can update the layer value of  $se_k$  as  $layer(f(se_k)) = \lfloor value(se_k.type) \rfloor$ , and its level value as  $level(se_k) = value(se_k) - layer(f(se_k.type))$ ; further, the class of  $se_k$  is determined according to the layer value in the ontology tree and then the type of  $se_k$  is  $se_k.type = f^{-1}(se_k.SO.class)$ ; and, finally, we can get  $se_k.Related = \{se_k.related_k \mid se_k.related_k = f^{-1}(f(se_k.type).Relation)\}$ .

For example, we can calculate the value of new sentiment dimension,  $se_5$ , which is generated by sentiment dimensions of  $se_1$  and  $se_4$  if there is a CMR rule for them in Figure 2 as

$$\begin{aligned} value(se_5) &= \frac{(level(se_1) + lay(class_2)) + (level(se_4) + lay(class_8))}{2} \\ &= \frac{(1 + 0.9) + (3 + 1)}{2} = 2.95. \end{aligned} \quad (11)$$

Then, we can get the layer value of  $se_5$  as  $layer(f(se_5.type)) = \lfloor 2.95 \rfloor = 2$ , and  $f(se_5.type) = class_6$  according to the ontology tree. Meanwhile,  $level(se_5) = value(se_5) - layer(f(se_5.type)) = 0.95$ .

Moreover, let there be a set of sentiment dimensions which satisfies rule of CMR, and they can be merged into one dimension through CMR (*mse*). The merging calculation is as follows:

$$mse = \bigodot_{i=1}^n se_i, \quad (12)$$

where  $n$  is the number of dimensions in the set.

(2) *Sentiment Migration Rule (SMR)*. Sentiment dimensions can migrate into another one if and only if there are qualitative changes under migration rules which are provided in sentiment ontology.

Sentiment migration rule is a kind of production rule. That is, SMR functions under legal reasoning rules which are provided by ontology. We also define an operator,  $\otimes$ , to calculate dimension changing based on SMR. Let there be two dimensions as  $se_i = \langle type, Related, level \rangle$  and  $se_j = \langle type, Related, level \rangle$  which satisfy SMR, and we denote the new dimension as  $se_i \otimes se_j = se_k$ . Then, the calculation of SMR is as follows:

$$\begin{aligned} se_i \otimes se_j &= se_k \\ &= \begin{cases} se_k.type = f^{-1} [r(f(se_i), f(se_j))] \\ se_k.Related = \{f^{-1}(f(se_k.type).Relation)\} \\ level(se_k) = \frac{level(se_i) + level(se_j) - 1}{2}. \end{cases} \end{aligned} \quad (13)$$

Here is an example of SMR which enables dimensions of “glad” and “satisfying” to migrate into “happiness” as follows:

$$\begin{aligned} &(\exists f(se_1.type) = glad \wedge level(se_1) \geq 0.7) \\ &\wedge (\exists f(se_2.type) = satisfying \wedge level(se_2) \geq 0.8) \\ &\wedge (r(glad, satisfying) = happiness) \\ &\longrightarrow \left( \exists (se_3.type) = happiness \wedge level(se_3) \right. \\ &= \left. \frac{level(se_1) + level(se_2) - 1}{2} \right). \end{aligned} \quad (14)$$

(3) *Time Merging Rule (TMR)*. Dimensions have their dynamic change formats if and only if they are changed with the time passing and satisfy CMR.

TMR enables sentiment dimension to be evaluated with time passing and reflects the dynamic feature of user’s sentiment. Here, we propose a dynamic iterative estimating method for describing the TMP based sentiment changing. Suppose the original sentiment dimension is  $se_i^0$  and its next value is measured to be  $chang\_se_i^1$  after a next time slice. Then, we can update the dynamic estimating result by using CMR calculation of these two sentiments; that is,

$$se_i^1 = se_i^0 \odot chang\_se_i^1. \quad (15)$$

By analogy, the dynamic estimating result of TMR can be calculated iteratively as

$$se_i^n = se_i^{n-1} \odot chang\_se_i^n. \quad (16)$$

(4) *Time Attenuation Rule (TAR)*. Level value of dimension would damp if and only if there is no new changes with the time passing.

Human’s sentiment bent would be on the wane with the time passing if there is not any new evidence which maintains this sentiment. TAR is proposed to reflect such attenuation effect. In this study, TAR only impacts the value of level for a sentiment dimension. Let there be a sentiment dimension  $se_i = \langle type, Related, level \rangle$ , and there is an attenuation

parameter  $\beta$ ; the level value of dimension can be calculated with TAR as

$$\text{level}(se_i)_n = \text{level}(se_i)_{n-1} \beta, \quad (17)$$

where  $\text{level}(se_i)_n$  is the level value of  $se_i$  after  $n$  time slices ( $\text{level}(se_i)_0$  is the original value here).

(5) *Mutually Exclusive Rule (MER)*. Dimensions cannot be calculated if and only if there is no rule which can be used for them.

MER reveals that the sentiment dimensions, which belong to different branches in ontology tree, are regarded as completely different ones unless they can be calculated under rules of CMR, SMR, and TPR.

5.2. *Sentiment Modeling*. Sentiment model describes all sentiment dimensions in a stable state. In this study, sentiment modeling method aims to normalize original sentiment dimensions through the above five rules for sentiment modeling. We propose the brief modeling process here as follows:

- (i) All related sentiment dimensions, which satisfy rule of CMR, are merged in original sentiment dimension set.
- (ii) All sentiment dimensions, which satisfy rule of SMR, are calculated for modeling new dimensions.
- (iii) All sentiment dimensions, which satisfy rules of TMR and TAR, are remodeled at every time slice.

## 6. Sentiment Delivering Estimate Scheme

6.1. *Explicit Sentiment Delivering Estimate Scheme*. In mobile social network, users' sentiments often are impacted by those who obtain users' trustworthiness. For example, user  $A$  might be very angry towards a topic if his most trusted friend  $B$  holds a very angry sentiment towards the same topic. Such sentiment delivering is shown in Figure 1(c). Therefore, we can estimate user's explicit sentiment through his/her trust chains. Here, we address an explicit sentiment delivering estimate scheme according to the trust chain.

In our consideration, people's explicit sentiment, which is expressed in text or other ways explicitly in past, would be influenced by those who he/she considers trustworthy. And then, the explicit sentiment would be influenced and changed in some degrees while the user expresses it in next time (called explicit delivering sentiment). Our explicit sentiment delivering estimate scheme aims to evaluate the user's explicit sentiment in near future based on his past explicit sentiment and his trust chains. Here, we have the following descriptions:

- (i) Source node: the user node who keeps explicit sentiment model in mobile social network.
- (ii) Target node: the user node who has expressed its explicit sentiment and would be influenced by source node to form new explicit sentiment in near future through their trust chain.
- (iii) Single explicit (implicit) sentiment: one explicit (implicit) sentiment delivered from one source node.

- (iv) Integrated explicit (implicit) sentiment: the explicit (implicit) sentiment which is composed of all single explicit (implicit) sentiments from all source nodes.

In this study, an iterative estimate evaluation method is proposed for explicit sentiment delivering estimate. The method is composed of two steps: single explicit sentiment calculation and integrated explicit sentiment evaluation. The two steps are as follows.

(1) *Single Delivering Explicit Sentiment Calculation*. In this step, we calculate all single explicit sentiments from one source node to the target node iteratively based on explicit sentiment model and their trust chains. Assume that there is a trust chain  $\Omega(N_S, N_E)$  between two user nodes, and the past source explicit sentiment model of  $N_S$  and  $N_E$  is  $SE_{N_S}$  and  $SE_{N_E}$ , respectively. Suppose that the trust chain value of  $\Omega(N_E, N_S)$  is  $T(N_S, N_E) \geq \alpha$ . And then, for each dimension in single explicit delivering sentiment  $\text{deliver\_se}_k(\Omega) \in \text{deliver\_SE}(\Omega)$ , it is generated by  $SE_{N_S}.se^i$  and  $SE_{N_E}.se^i$  and is calculated as follows:

- (i) If there is CMR for  $SE_{N_S}.se_i$  and  $SE_{N_E}.se_j$ , then the single explicit delivering sentiment dimension can be calculated as

$$\begin{aligned} (\text{deliver\_se}_k(\Omega) &= SE_{N_S}.se_i \odot SE_{N_E}.se_j) \\ \wedge (\text{value}(\text{deliver\_se}_k(\Omega)) &= \text{value}(SE_{N_S}.se_i) \\ &\odot \text{value}(SE_{N_E}.se_j) T(N_S, N_E)). \end{aligned} \quad (18)$$

- (ii) If there is SMR for  $SE_{N_S}.se_i$  and  $SE_{N_E}.se_j$ , then the single explicit delivering sentiment dimension can be calculated as

$$\begin{aligned} \text{deliver\_se}_k(\Omega) &= SE_{N_S}.se_i \otimes SE_{N_E}.se_j \times \Omega(N_E, N_S) \\ &= \begin{cases} se_k.type = f^{-1} [r(f(se_i), f(se_j))] \\ se_k.Related = \{f^{-1}(f(se_k.type).Relation)\} \\ level(se_k) = \frac{\text{level}(se_i) + \text{level}(se_j) - 1}{2} T(N_E, N_S). \end{cases} \end{aligned} \quad (19)$$

- (iii) If there is no rule for  $SE_{N_E}.se_j$ , then the single explicit delivering sentiment dimension can be calculated as

$$\text{deliver\_se}_k(\Omega) = SE_{N_E}.se_j. \quad (20)$$

- (iv) If there is no rule for  $SE_{N_S}.se_i$ , then the single explicit delivering sentiment dimension can be calculated as

$$\begin{aligned} \text{deliver\_se}_k(\Omega) &= \begin{cases} \text{deliver\_se}_k(\Omega).type = SE_{N_S}.se_i.type \\ \text{deliver\_se}_k(\Omega).Related = SE_{N_S}.se_i.Related \\ \text{deliver\_se}_k(\Omega).level = SE_{N_S}.se_i.level \times T(N_S, N_E). \end{cases} \end{aligned} \quad (21)$$

Then, single explicit sentiment model from trust chain  $\Omega(N_E, N_S)$  is described as

$$\begin{aligned} & deliver\_SE(\Omega) \\ &= \{deliver\_se_1(\Omega), deliver\_se_2(\Omega), \dots\}. \end{aligned} \quad (22)$$

(2) *Integrated Explicit Sentiment Evaluation.* In this step, all single explicit sentiment models from different source nodes are integrated into one explicit sentiment model based on proposed five rules for sentiment calculation.

Let there be a set of single explicit sentiment models  $\bigcup deliver\_SE(\Omega_j)$  from different trust chains. Then, all sentiment dimensions in  $deliver\_SE(\Omega_j)$  can be extracted and then a new integrated explicit sentiment model can be generated as

$$\begin{aligned} & deliver\_SE_{N_E} = \{deliver\_se_k(\Omega_j) \mid deliver\_se_i(\Omega_j) \\ & \in deliver\_SE(\Omega_j) \wedge deliver\_SE(\Omega_j) \\ & \in \bigcup deliver\_SE(\Omega_j)\}. \end{aligned} \quad (23)$$

Then, we use the following steps to reduce the number of dimensions in  $deliver\_SE_{N_E}$ :

- (i) All dimensions in  $deliver\_SE_{N_E}$  are calculated based on rules of CMR and SMR to reduce the number of dimensions.
- (ii) The dimensions, which satisfy the condition of  $level(deliver\_se_i) \leq \gamma$ , are regarded as noncritical dimensions and are deleted from the model.
- (iii) All dimensions, which satisfy rules of TMR and TAR, are updated at every time slice.
- (iv) The dimensions, which satisfy the condition of  $level(deliver\_se_i)_n \leq \gamma$ , are deleted as noncritical dimensions after  $n$  time slices.

**6.2. Implicit Sentiment Delivering Estimate Scheme.** Implicit sentiment is a kind of hidden emotion which reflects users' potential opinion. In our view, trust chains among users denote the probability of user's inherent attitudes based on their trust relationships. Here, we address an estimating evaluation for calculating the implicit sentiment according to the trust chain.

We consider that people's sentiment would be influenced by those who he/she trusts in. For example, a user might approve of a topic because his best friends keep positive feelings toward the topic even though he does not show any explicit evidence. Therefore, the implicit sentiment can be calculated through the trust chains. That is, implicit sentiment is transmitted through the chains. Similar to explicit sentiment delivering estimate scheme, we here propose an implicit sentiment delivering method which is composed of two steps: single implicit sentiment delivering calculation and integrated implicit delivering sentiment evaluation. The two steps are as follows.

(1) *Single Implicit Sentiment Delivering Calculation.* Assume that there is a trust chain  $\Omega(N_S, N_E)$  between two user

nodes, and the source explicit sentiment model of  $N_S$  is  $SE_{N_S}$ . Suppose that the trust chain value of  $\Omega(N_E, N_S)$  is  $T(N_S, N_E) \geq \alpha$ . And then, for each dimension in single implicit sentiment  $implicit\_se_i(\Omega) \in implicit\_SE(\Omega)$ , it is generated by  $SE_{N_S}.se^i$ , which is delivered from  $N_S$ , and is calculated as

$$\begin{aligned} & implicit\_se_i(\Omega).type = SE_{N_S}.se_i.type, \\ & implicit\_se_i(\Omega).Related = SE_{N_S}.se_i.Related, \\ & implicit\_se_i(\Omega).level = SE_{N_S}.se_i.level \times T(N_S, N_E). \end{aligned} \quad (24)$$

Then, single implicit sentiment model from trust chain  $\Omega(N_E, N_S)$  is described as

$$\begin{aligned} & implicit\_SE(\Omega) \\ &= \{implicit\_se_1(\Omega), implicit\_se_2(\Omega), \dots\}. \end{aligned} \quad (25)$$

(2) *Integrated Implicit Delivering Sentiment Evaluation.* In this step, all single implicit sentiment models from source nodes are integrated into one implicit sentiment model based on proposed five rules for sentiment calculation.

Let there be a set of single implicit sentiment models  $\bigcup implicit\_SE(\Omega_j)$  from different trust chains. Then, all sentiment dimensions in  $implicit\_SE(\Omega_j)$  can be extracted and then a new integrated implicit sentiment model can be generated as

$$\begin{aligned} & implicit\_SE_{N_E} \\ &= \{implicit\_se_i(\Omega_j) \mid implicit\_se_i(\Omega_j) \\ & \in implicit\_SE(\Omega_j) \wedge implicit\_SE(\Omega_j) \\ & \in \bigcup implicit\_SE(\Omega_j)\}. \end{aligned} \quad (26)$$

Similar to explicit sentiment, we use the following steps to reduce the number of dimensions in  $implicit\_SE_{N_E}$ :

- (i) All dimensions in  $implicit\_SE_{N_E}$  are calculated based on rules of CMR and SMR to reduce the number of dimensions.
- (ii) The dimensions, which satisfy the condition of  $level(implicit\_se_i) \leq \gamma$ , are regarded as noncritical dimensions and are deleted from the model.
- (iii) All dimensions, which satisfy rules of TMR and TAR, are updated at every time slice.
- (iv) The dimensions, which satisfy the condition of  $level(implicit\_se_i)_n \leq \gamma$ , are deleted as noncritical dimensions after  $n$  time slices.

## 7. Experiment and Analysis

In this section, we proposed examinations to explain the efficiency of our proposed method. In our scenario of examinations, the data comes from the *Tencent Weibo* which is a popular mobile social network platform in China. We collect information manually from <http://sina.com/> microblog within a

TABLE 1: Characteristics of five communities in examination prototype.

Community	Social life	Finance	Sporting	Entertainment	Technology	Total
Number of IDs	913	792	1,183	1,054	854	3,016
Number of posts	3,858	2,693	4,692	3,013	3,065	17,321
Number of comments	89,076	86,871	90,562	49,065	69,651	385,225

span of three months through the API interface which is provided by <http://sina.com/>. Our data included about 3,016 IDs (some IDs located in two or more communities) and more than 400,000 records (including posts and comments) from October 2013 to February 2014. In the examination prototype, there is a one-way direct link from a user towards another one if he/she followed the user in <http://sina.com/microblog>. All direct links (atomic trust chain) are generated from initial data set and are fixed and invariable in prototype. All these data are used to calculate the initial trust and form network topology based on real-world source.

Based on data, we develop a prototype for monitoring and evaluating the effectiveness of our proposed method in this study. In our prototype, nodes can be controlled by experimenters for examining requirements. For initial setting, there are 5 communities, that is, social life, finance, sporting, entertainment, and technology. We classify all IDs into 5 communities according to their verified identifications (one ID might locate in two or more communities). Detailed characteristics of five communities in the prototype are shown in Table 1. There are four kinds of behaviors in the prototype: posting, transmitting, judging, and accepting/rejecting. Meanwhile, atomic trust chains between IDs which have direct interaction, such as publishing judgments, transferring, or following, are set initially by calculating their past interaction according to the collected data. In addition, the network topology of our prototype is generated according to trust chain relationships and average out-degree of a node is 7.

In addition, we construct sentiment ontology in our examination for sentiment delivering evaluation. In this ontology, we collected 107 sentiment classes for sentiment semantic description. There are 1,091 relations and 923 rules in ontology for sentiment modeling.

*7.1. Examination for Trust Chain Calculation.* The first examination aims to testify the effectiveness of trust chain. We verify the effectiveness of trust chain by adding about 800 additional nodes which can be set as honest or malicious nodes in the prototype. All additional nodes are deployed randomly. And then, we testify the trust chain calculation method under no malicious node and malicious nodes. In the examination, we define that a trust chain is accurate if the trust chain value toward an honest node is larger than 0.5 or the trust chain value toward a malicious node is lower than 0.4.

*7.1.1. Accuracy of Four Types of Trust Chain.* In this examination, we compare the calculation methods of the four types of trust chain, that is, atom trust chain (ATC), serial trust chain (STC), parallel trust chain (PTC), and combined trust chain (CTC), and record their accuracies. We test 10,000 times of interaction under no malicious node and 15% malicious

nodes. The results are shown in Figures 4(a) and 4(b). We can see that the average accuracy of atom trust chain obtains the best performance in the examination. Meanwhile, combined trust chain and parallel trust chain have close accuracies which reach up to 90.4%, 86.1%, 89.6%, and 83.8% under no malicious node and 15% malicious nodes. We think that the reasons are as follows: (1) the long distance of path is punished in BF and our methods; (2) confidence value is impacted by the reputation weight in our proposed method so that the confidence would be attenuated if there were poor reputation nodes in the path; and (3) parallel path confidence is a comprehensive evaluation result based on all single paths in the parallel path, which leads to better effects than single path confidence.

In addition, we verify the impacts of parameter  $\eta$  for trust chain calculation. We test the different values of  $\eta$  for parallel trust chain and combined trust chain, respectively. In Figure 4(c), we can see that the accuracy is low while the value is  $\eta$  too low or too high. On the basis of test validation, a threshold around 0.6–0.7 is often a reasonable compromise. In our consideration, the reasons are as follows: (1) some untrustworthy neighbors are taken into account if the value of threshold  $\eta$  is set too low, which would decrease the accuracy of parallel and combined trust chain and (2) reasonable neighbors and their following trust chains are neglected if the value of threshold  $\eta$  is set too high, which leads to the accuracy of parallel and combined trust chain decrease.

*7.1.2. Performance Comparison of Proposed Trust Chain Calculation.* In this examination, we first revealed the effectiveness of our proposed trust chain calculation method by comparing with other existing methods. Here, we selected nodes which located in our prototype at random and then calculated their trust chain values through our proposed methods (TC). For comparison, we set the other two methods: trust with Bellman-Ford algorithm (BF) [15] and average method of trust aggregation (AT) [6]. The results are shown in Figures 5(a) and 5(b). After 10,000 times of comparison, average accuracy of our proposed parallel confidence method is better than other methods under no malicious node and 15% malicious nodes environments, respectively. In our analysis, the reasons are as follows: (1) the long distance of path is punished in BF and our methods; (2) trust value is impacted by the reputation weight in our proposed method so that the trust value would be attenuated if there were poor reputation nodes in the path; and (3) trust chain value is a comprehensive evaluation result based on all composition paths in the chain, which leads to better effects than other methods.

Moreover, we verified the impacts of the numbers of paths, communities, and depths. In Figure 5(c), average accuracies of trust calculation are decreased with the number

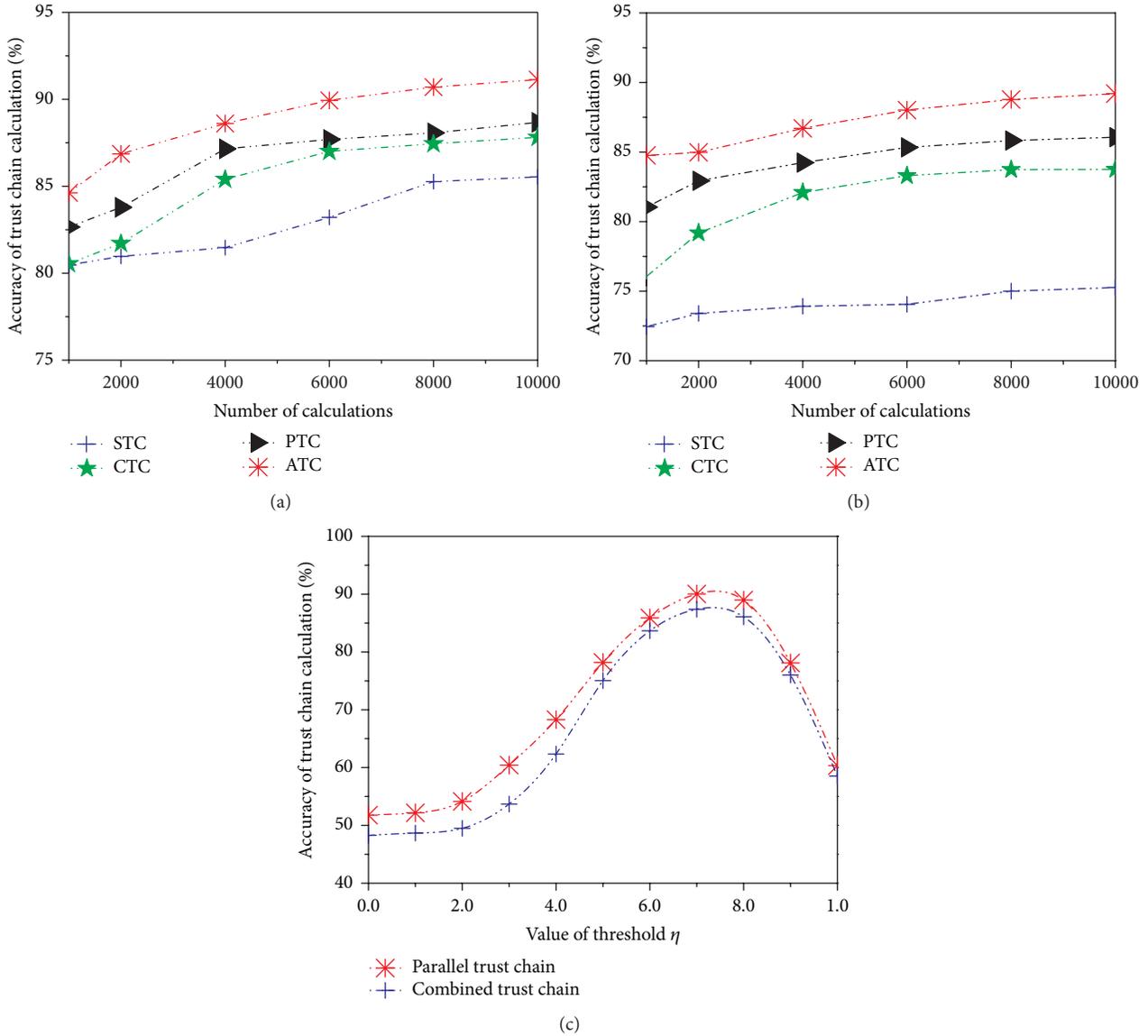


FIGURE 4: Effects of trust chain calculation.

of paths increasing in BF and AT, while the accuracy of proposed trust chain kept increasing. That is because the shortest path is admitted for trust computation in BF and all paths are treated equally in AT, which results in the ignorance of other significant paths in BF and mistakenly adoption of malicious paths or nodes in AT. In our method, all trustable paths (impacted by parameter  $\eta$ ) were computed comprehensively and all nodes are impacted by their reputation based on related rules. Figure 5(d) shows that with the numbers of paths (trust chain) and depths increasing, the accuracies were decreasing. In BF method, accuracy was nearly stable because long paths are deemed untrustworthy. In our method, trust chain values are attenuated with the depth increasing according to depth attenuation impacts in (7). From this point, a larger value of depth in trust chain also implies that the trust chain would be an untrustworthy one. Figure 5(e) shows the accuracies of trust calculation under

different numbers of communities which trust chain paths passed through. We can see that the accuracy of our proposed trust chain was higher than the other two methods with the number of communities increasing.

**7.2. Performance Evaluation of Sentiment Modeling.** In this examination, we aim to verify the effectiveness and feasibility of sentiment modeling method based on proposed rules in this study. The initial setting of sentiment ontology is shown as the beginning of this section. Firstly, we testify the feasibility of our proposed rules for sentiment modeling. We tracked the sentiments, which were expressed by users towards specific topics, and recorded the average accuracies of sentiment modeling under rules of CMR, SMR, TAR, and TMR. We selected 300 users at random and modeled their sentiments in their posts and comments. In Figure 6(a), results show that the accuracies of our proposed four rules

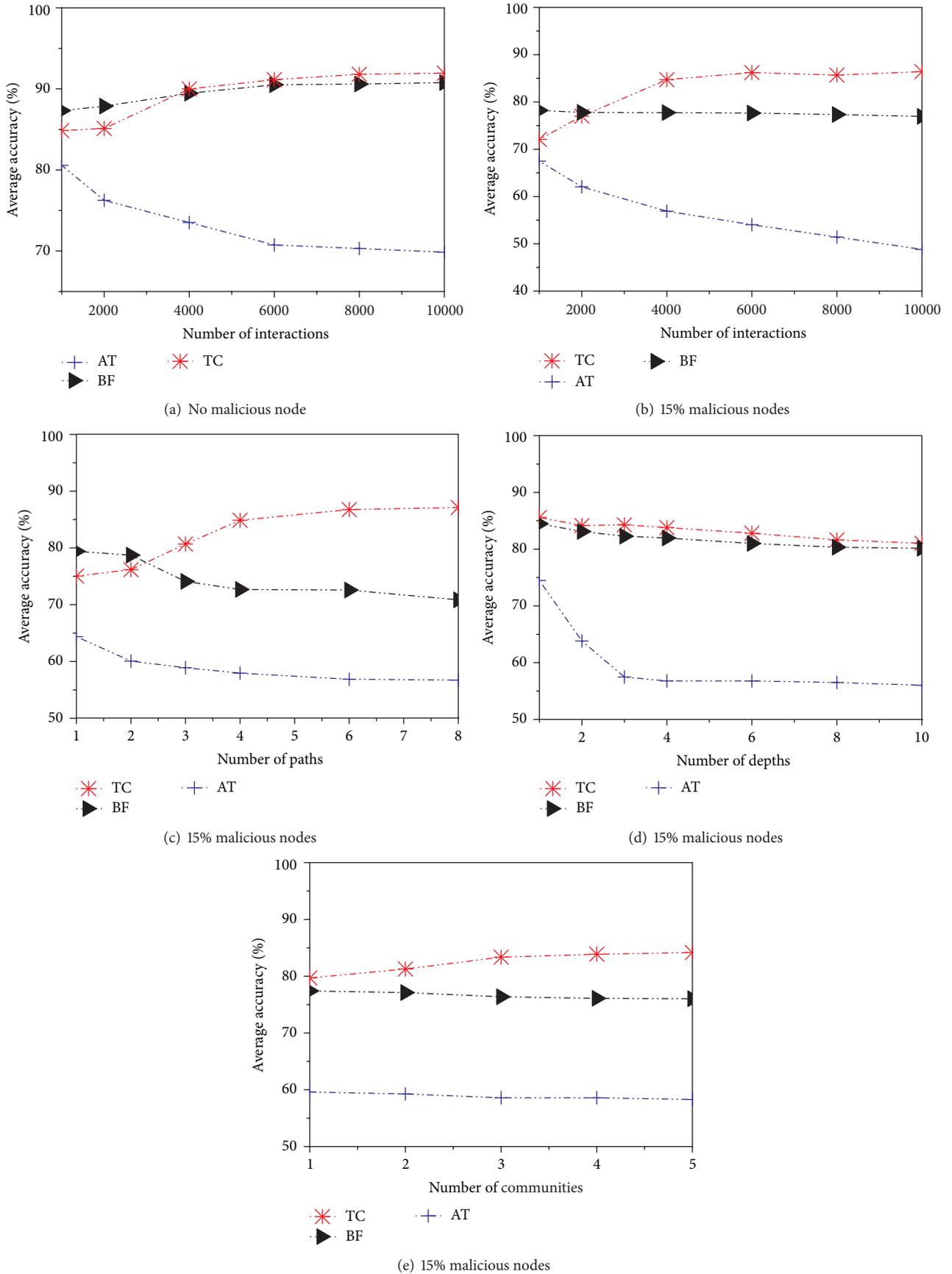


FIGURE 5: Performance comparison of proposed trust chain calculation.

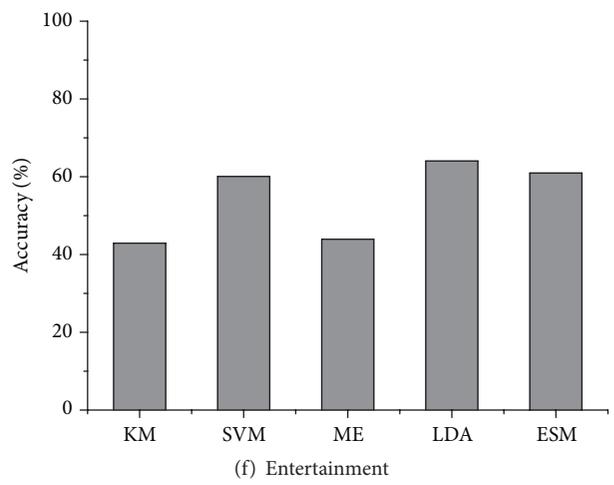
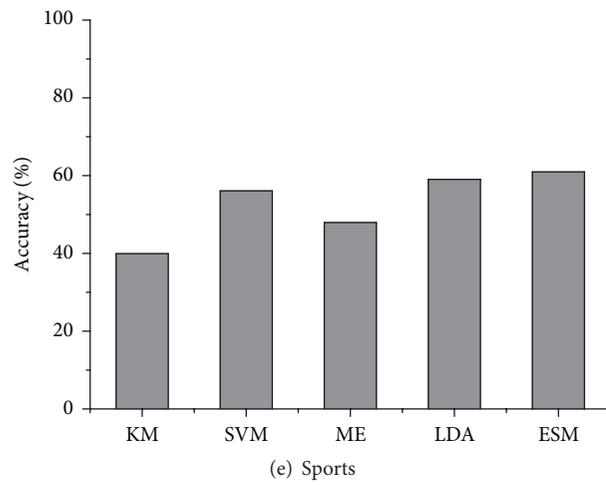
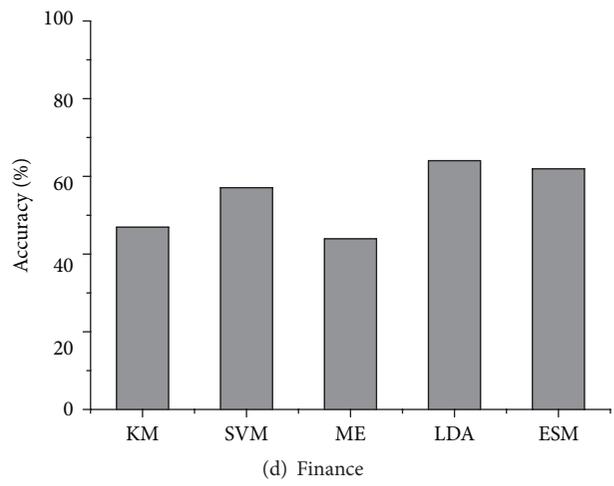
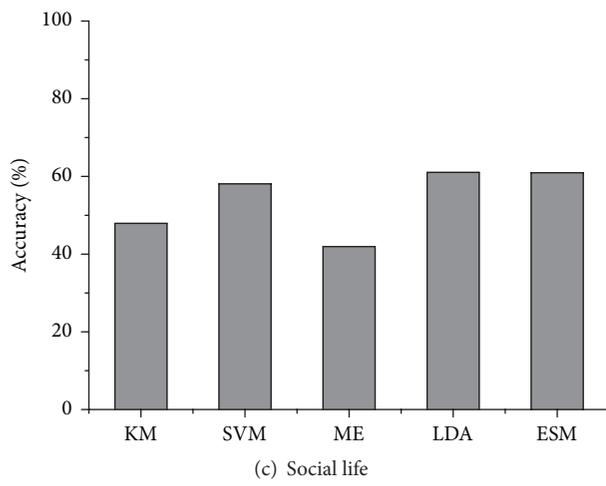
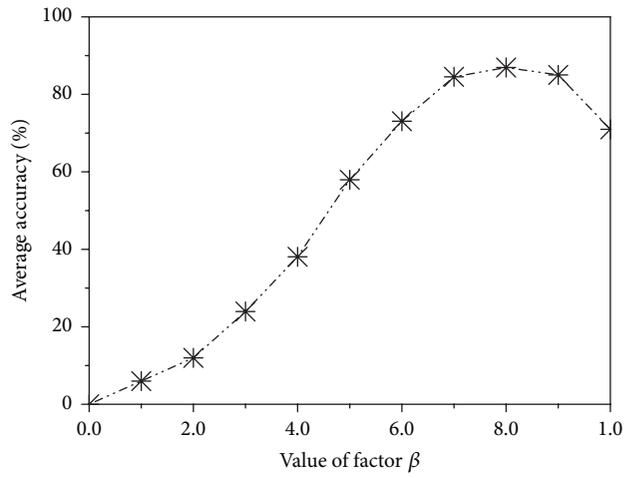
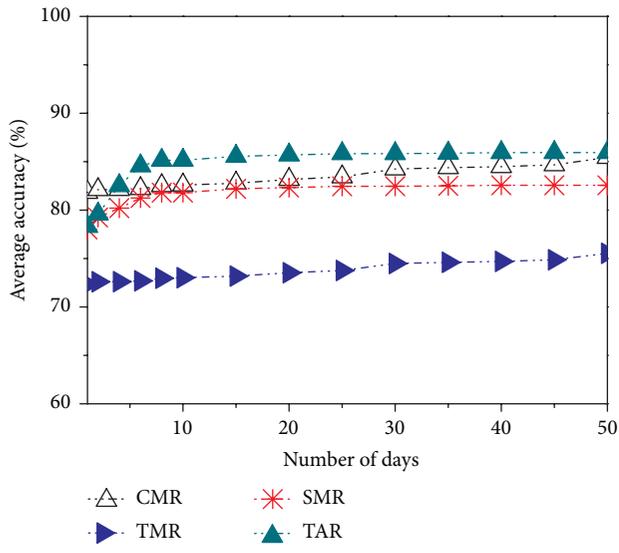


FIGURE 6: Continued.

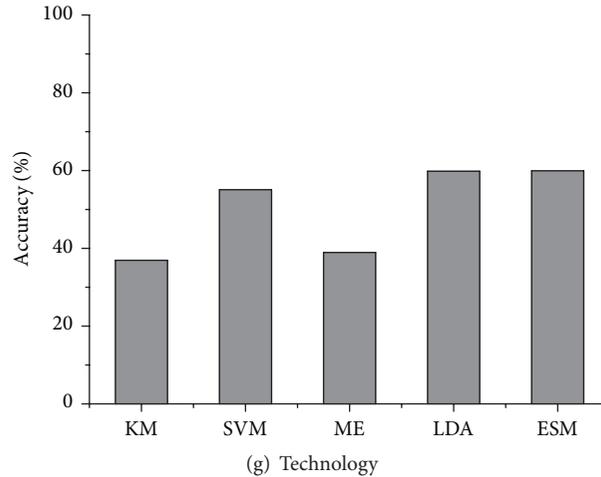


FIGURE 6: Performance of explicit sentiment modeling.

are acceptable after 50 days of tracking. In addition, we verify the impact of factor  $\beta$  in rule TAR. Figure 6(b) shows that the accuracies of sentiment evaluation are low while the value of factor  $\beta$  is too low or too high. We can see that a threshold around 0.7–0.9 is often a reasonable compromise and the accuracy is around 86%–89%.

Further, we reveal the performance of proposed sentiment modeling method by comparing existing methods. In our examination, we set three methods for comparison as: sentiment analysis based on keyword matching (KM), sentiment classification based on support vector machines (SVM), max entropy method (ME), LDA model (LDA), and our proposed sentiment modeling (ESM) method. The results are shown in Figures 6(c)–6(g), which reveal the accuracies of sentiment analysis in our initial setting communities. We can see that the LDA method had the best performances, while the accuracies of our proposed sentiment modeling were almost equal to LDA method. That means the proposed method is feasible and effective in sentiment analysis.

**7.3. Performance Evaluation of Explicit Sentiment Delivering Estimate Scheme.** This examination reveals the effectiveness of explicit sentiment delivering estimate scheme. We selected about 1,000 topics in our dataset for evaluating explicit sentiment delivering. In the test, we recorded the accuracy of sentiment estimate based on past explicit sentiment records. For verifying the effects on sentiment estimate, we set data from September 2013 to December 2013 as past data for priori probability training preparation. Then, we testified the accuracies of results with the microblog data from January 2014 to February 2014. For comparison, we set four groups as: maximum likelihood estimate method (MLE), the priori estimate method (PE), Bayesian method (BM), and our proposed explicit sentiment delivering method (ESD). The results are shown in Figures 7(a) and 7(b). In Figure 7(a), we recorded the accuracies of sentiment delivering for extreme sentiment, including anger, blame, guilt, excitement, and happiness, in different groups, while accuracies in Figure 7(b) were recorded for normal sentiment. As a result of our

analysis, the performances of probability-based estimate methods rely on the training dataset in past. Therefore, with the number of estimate times increasing, the accuracies were increasing in groups of MLE, PE, and BM. Compared with the three methods, our method mainly relies on the effects of trust chain evaluation and sentiment ontology. Then, we can see that our method got the best performances which were around 74% and 79%, respectively. We also notice that the accuracies were higher than other methods from the beginning. In our consideration, the reasons are as follows: (1) the sentiment ontology is constructed in advance so that the knowledge and rules of explicit sentiment delivering are accurate at the beginning; (2) the trust chains among users are established in advance and the sentiment delivering is based on accurate trust chains; (3) trust chains can be renewed according to results of sentiment delivering to ensure the high accuracy of trust chain.

Moreover, we verify the impacts of parameter  $\alpha$  for explicit sentiment delivering. We test the different values of  $\alpha$  for accuracy of sentiment delivering estimate. In Figure 7(c), we can see that the average accuracy is lower while the value of  $\alpha$  is lower than 0.6. In addition, we notice that the accuracy is decreased while the value of  $\alpha$  is larger than 0.8. From our perspective, that is because many valuable trust chains, which may impact the explicit sentiment delivering result, were ignored in such case. On the basis of test validation, a value around 0.6–0.8 is often a reasonable compromise. In our other examinations, its default value is set as 0.7.

**7.4. Performance Evaluation of Implicit Sentiment Delivering Estimate Scheme.** In this examination, we aim to testify the effectiveness and feasibility of proposed implicit sentiment delivering estimate scheme. The examination setting is similar to examinations in Section 7.4. We evaluated implicit sentiments of users which were never expressed in their past posts and comments explicitly if they had transferring, following, and similar behaviors toward specific topics in the dataset from September 2013 to December 2013. And then, we verified the users' sentiments towards the topics if they

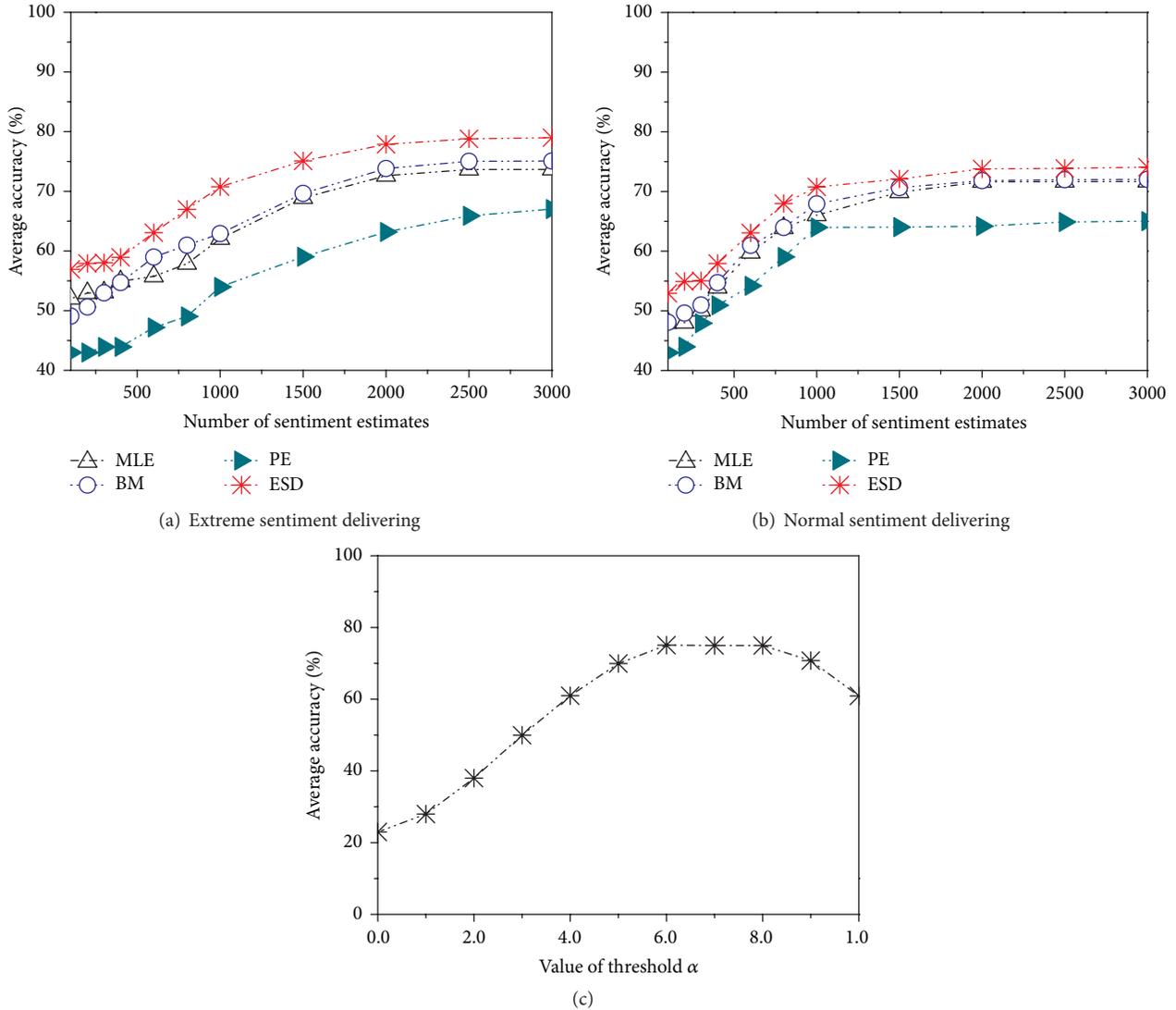


FIGURE 7: Examinations of explicit sentiment delivering estimate.

expressed their sentiments explicitly in the last two months. For comparison, we set two groups as user relationship based implicit sentiment estimate scheme (URM) in which delivering factor is set as 1 and our proposed trust chain based method (TCM). The results are shown in Figures 8(a) and 8(b). We recorded the accuracies of the above two methods in collected five communities, that is, social life (SL), finance (FI), sports (SP), entertainment (EN), and technology (TE). Figure 8(a) shows that the accuracies of extreme sentiments delivering estimate was around 54.7%, while the accuracies of normal sentiments delivering estimate was around 45.8%. In our consideration, extreme sentiment would have more probabilities to be delivered than normal sentiment through users' trustworthiness in trust chain, which resulted in that the accuracy of extreme sentiment delivering estimate was higher than accuracy of normal sentiment delivering estimate.

Likewise, we verify the impacts of parameter  $\gamma$  for implicit sentiment delivering estimate. We test the different values of  $\gamma$  for accuracy. In Figure 8(c), we can see that the average

accuracy is lower while the value of  $\gamma$  is lower than 0.2 or larger than 0.4. And also we can get that a value around 0.2–0.4 is often a reasonable compromise. In our other examinations, its default value is set as 0.3.

## 8. Conclusion

Sentiment analysis is essential for mobile social network because of its convenient communication pattern and the nature of gathering mass of public opinions. Sentiments in such relationship oriented platform are delivered through users' trustworthy relationships and then are impacted mutually. As a result, sentiment evaluation in mobile social network should be considered from a delivering view. Most existing researches indicate that mining explicit sentiment is feasible and effective. However, few of them take the sentiment delivering estimate into consideration. In this study, sentiment, which includes explicit sentiment and implicit sentiment, is analyzed based on its delivering nature

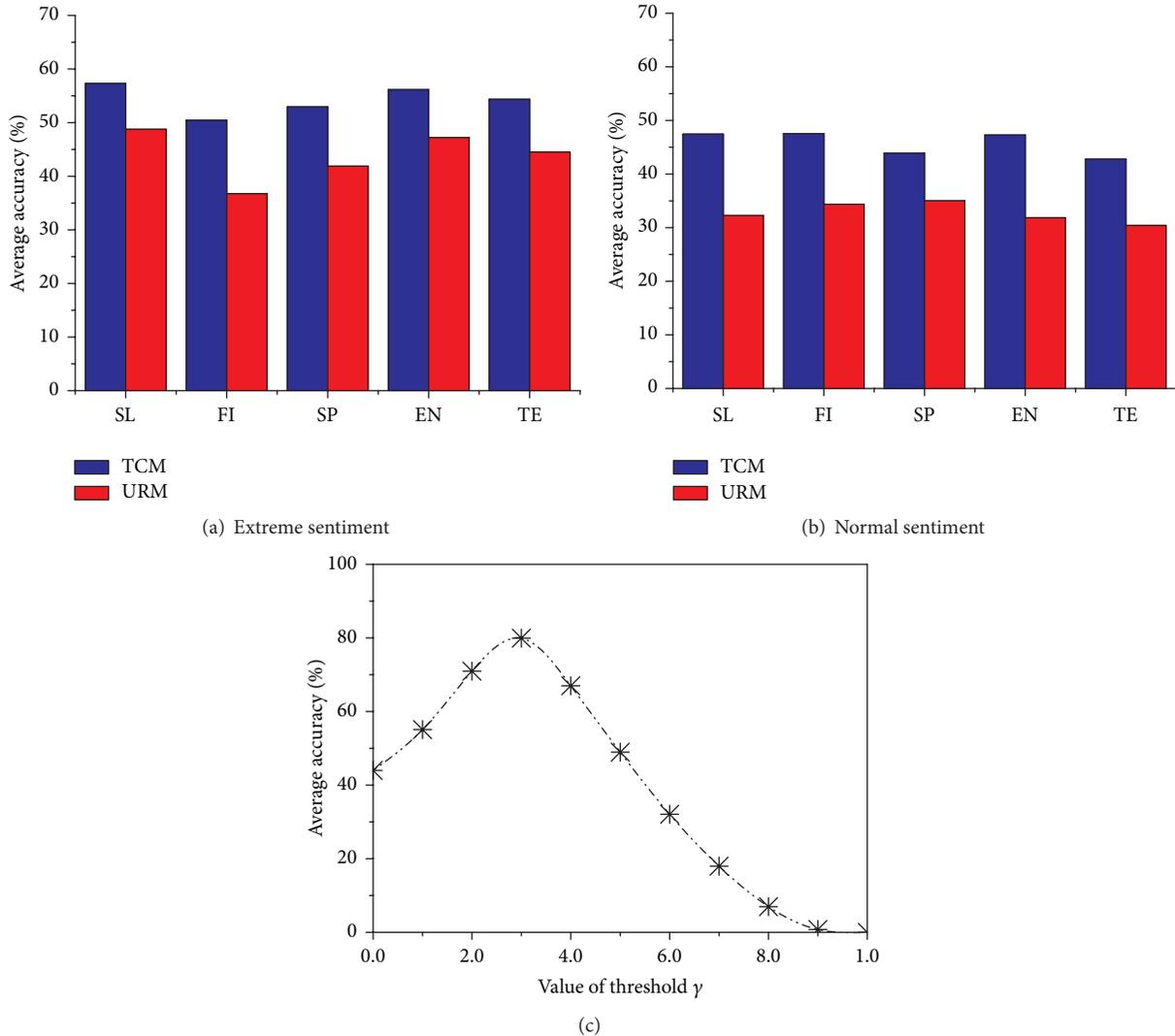


FIGURE 8: Examinations of implicit sentiment delivering estimate.

and the users' trustworthy relationships. In our work, trust chain, which is the model of describing and evaluating trustworthiness among users, is defined and its calculation method is addressed according to its path composition as atomic path, serial path, parallel path, and combined path. Sentiment in this work is described in a formal semantic method based on sentiment ontology. Then, the method of explicit sentiment modeling is proposed through a set of sentiment modeling rules. Further, delivering estimate scheme for explicit sentiment and implicit sentiment is proposed based on trust chain and sentiment modeling. In future studies, we aim to identify a mechanism of group sentiment delivering mining for user community or group in mobile social network.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### Acknowledgments

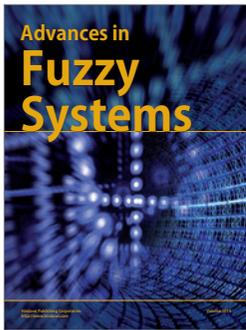
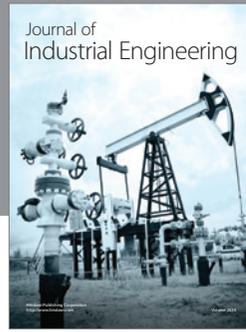
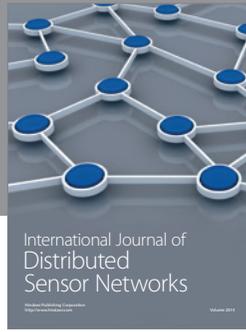
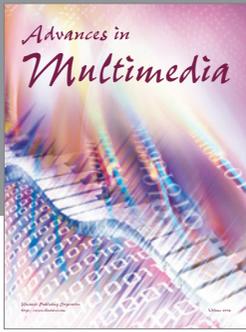
This work is funded by National Basic Research Program of China (2014CB340404), National Natural Science Foundation of China (61572326, 61103069, 71171148, and 61272268), Innovation Program of Shanghai Municipal Education Commission (13YZ052), the Program of Shanghai Normal University (DCL201302), and the Shanghai Rising-Star Program (no. 15QA1403900).

### References

- [1] P. Groenewegen and C. Moser, "Online communities: challenges and opportunities for mobile social network research," *Research in the Sociology of Organizations*, vol. 40, pp. 463–477, 2014.
- [2] D. Boyd, S. Golder, and G. Lotan, "Tweet, tweet, retweet: conversational aspects of retweeting on twitter," in *Proceedings of the 43rd Annual Hawaii International Conference on System Sciences (HICSS '10)*, pp. 1–10, IEEE, Honolulu, Hawaii, USA, January 2010.

- [3] H. He, "Sentiment analysis of Sina Weibo based on semantic sentiment space model," in *Proceedings of the 20th International Conference on Management Science and Engineering (ICMSE '13)*, pp. 206–211, IEEE, Harbin, China, July 2013.
- [4] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.
- [5] M. M. Mostafa, "More than words: social networks' text mining for consumer brand sentiments," *Expert Systems with Applications*, vol. 40, no. 10, pp. 4241–4251, 2013.
- [6] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.
- [7] E. W. K. See-To and K. K. W. Ho, "Value co-creation and purchase intention in mobile social network sites: the role of electronic Word-of-Mouth and trust—a theoretical analysis," *Computers in Human Behavior*, vol. 31, pp. 182–189, 2014.
- [8] J. Wu and F. Chiclana, "A social network analysis trust-consensus based approach to group decision-making problems with interval-valued fuzzy reciprocal preference relations," *Knowledge-Based Systems*, vol. 59, pp. 97–107, 2014.
- [9] G. Wang and X.-L. Gui, "Selecting and trust computing for transaction nodes in online social networks," *Jisuanji Xuebao*, vol. 36, no. 2, pp. 368–383, 2013.
- [10] D. Li, Q. Lv, X. Xie et al., "Interest-based real-time content recommendation in online social communities," *Knowledge-Based Systems*, vol. 28, pp. 1–12, 2012.
- [11] Z. Peiyun, C. Enhong, and L. Bo, "Web services trust computation based on mobile social network dynamic feedback," *Pattern Recognition and Artificial Intelligence*, vol. 26, no. 4, pp. 337–343, 2013.
- [12] X.-Q. Qiao, C. Yang, X.-F. Li, and J.-L. Chen, "A trust calculating algorithm based on social networking service users' context," *Jisuanji Xuebao*, vol. 34, no. 12, pp. 2403–2413, 2011.
- [13] F. Javier Ortega, J. A. Troyano, F. L. Cruz, C. G. Vallejo, and F. Enríquez, "Propagation of trust and distrust for the detection of trolls in a social network," *Computer Networks*, vol. 56, no. 12, pp. 2884–2895, 2012.
- [14] B. Qureshi, G. Min, and D. Kouvatsos, "Trusted information exchange in peer-to-peer mobile social networks," *Concurrency Computation: Practice and Experience*, vol. 24, no. 17, pp. 2055–2068, 2012.
- [15] H. Zhao and X. Li, "VectorTrust: trust vector aggregation scheme for trust management in peer-to-peer networks," in *Proceedings of the 18th International Conference on Computer Communications and Networks (ICCCN '09)*, pp. 1–6, IEEE, San Francisco, Calif, USA, August 2009.
- [16] J. Golbeck, "Trust and nuanced profile similarity in online social networks," *ACM Transactions on the Web*, vol. 3, no. 4, article 12, 2009.
- [17] N. Li and D. D. Wu, "Using text mining and sentiment analysis for online forums hotspot detection and forecast," *Decision Support Systems*, vol. 48, no. 2, pp. 354–368, 2010.
- [18] S. Tan, X. Cheng, Y. Wang, and H. Xu, "Adapting naive bayes to domain adaptation for sentiment analysis," in *Advances in Information Retrieval*, vol. 5478 of *Lecture Notes in Computer Science*, pp. 337–349, Springer, Berlin, Germany, 2009.
- [19] J. Boyd-Graber and P. Resnik, "Holistic sentiment analysis across languages: multilingual supervised latent Dirichlet allocation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '10)*, pp. 45–55, Association for Computational Linguistics, October 2010.
- [20] M. Thelwall and K. Buckley, "Topic-based sentiment analysis for the social web: the role of mood and issue-related words," *Journal of the American Society for Information Science and Technology*, vol. 64, no. 8, pp. 1608–1617, 2013.
- [21] E. Cambria, B. Schuller, B. Liu, H. Wang, and C. Havasi, "Knowledge-based approaches to concept-level sentiment analysis," *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 12–14, 2013.
- [22] F. Greaves, D. Ramirez-Cano, C. Millett, A. Darzi, and L. Donaldson, "Use of sentiment analysis for capturing patient experience from free-text comments posted online," *Journal of Medical Internet Research*, vol. 15, no. 11, article e239, 2013.
- [23] E. Kontopoulos, C. Berberidis, T. Dergiades, and N. Bassiliades, "Ontology-based sentiment analysis of twitter posts," *Expert Systems with Applications*, vol. 40, no. 10, pp. 4065–4074, 2013.
- [24] T. Nasukawa and J. Yi, "Sentiment analysis: capturing favorability using natural language processing," in *Proceedings of the 2nd International Conference on Knowledge Capture (K-CAP '03)*, pp. 70–77, ACM, Sanibel Island, Fla, USA, October 2003.
- [25] P. Gonçalves, M. Araújo, F. Benevenuto, and M. Cha, "Comparing and combining sentiment analysis methods," in *Proceedings of the 1st ACM Conference on Online Social Networks (COSN '13)*, pp. 27–37, ACM, Boston, Mass, USA, October 2013.
- [26] Q. Su, X. Xu, H. Guo et al., "Hidden sentiment association in Chinese web opinion mining," in *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*, pp. 959–968, ACM, Beijing, China, April 2008.
- [27] A. Athar and S. Teufel, "Detection of implicit citations for sentiment detection," in *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse (DSSD '12)*, pp. 18–26, Association for Computational Linguistics, July 2012.
- [28] A. Balahur, J. M. Hermida, and A. Montoyo, "Detecting implicit expressions of sentiment in text based on commonsense knowledge," in *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA '11)*, Association for Computational Linguistics, Portland, Ore, USA, June 2011.
- [29] A. Balahur, J. M. Hermida, and A. Montoyo, "Detecting implicit expressions of emotion in text: a comparative analysis," *Decision Support Systems*, vol. 53, no. 4, pp. 742–753, 2012.
- [30] X. Zhou, X. Tao, J. Yong, and Z. Yang, "Sentiment analysis on tweets for social events," in *Proceedings of the IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD '13)*, pp. 557–562, Whistler, Canada, June 2013.
- [31] T. Dergiades, C. Milas, and T. Panagiotidis, "Tweets, Google trends, and sovereign spreads in the GIIPS," *Oxford Economic Papers*, vol. 67, no. 2, pp. 406–432, 2015.
- [32] M. Thelwall, D. Wilkinson, and S. Uppal, "Data mining emotion in social network communication: gender differences in MySpace," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 1, pp. 190–199, 2010.
- [33] T. H. Soliman, M. A. Elmasry, A. Hedar, and M. M. Doss, "Sentiment analysis of Arabic slang comments on facebook," *International Journal of Computers & Technology*, vol. 12, no. 5, pp. 3470–3478, 2014.
- [34] A. Severyn, A. Moschitti, O. Uryupina, B. Plank, and K. Filippova, "Multi-lingual opinion mining on YouTube," *Information Processing & Management*, 2015.
- [35] C. Baecchi, T. Uricchio, M. Bertini, and A. Del Bimbo, "A multimodal feature learning approach for sentiment analysis of social network multimedia," *Multimedia Tools and Applications*, 2015.

- [36] C. Clavel and Z. Callejas, "Sentiment analysis: from opinion mining to human-agent interaction," *IEEE Transactions on Affective Computing*, 2016.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

