Learning the Structures of Online Asynchronous Conversations

Jun Chen[1], Chaokun Wang[1(✉)], Heran Lin[1], Weiping Wang[2], Zhipeng Cai[3], and Jianmin Wang[1]

[1] School of Software, Tsinghua University, Beijing 100084, China
chenjun14@mails.thu.edu.cn, linhr10@gmail.com, {chaokun,jimwang}@tsinghua.edu.cn
[2] Institute of Information Engineering, CAS, Beijing 100093, China
wangweiping@iie.ac.cn
[3] Department of Computer Science, Georgia State University, Atlanta, GA 30302, USA
zcai@gsu.edu

Abstract. The online social networks have embraced huge success from the crowds in the last two decades. Now, more and more people get used to chat with friends online via instant messaging applications on personal computers or mobile devices. Since these conversations are sequentially organized, which fails to show the logical relations between messages, they are called asynchronous conversations in previous studies. Unfortunately, the sequential layouts of messages are usually not intuitive to see how the conversation evolves as time elapses. In this paper, we propose to learn the structures of online asynchronous conversations by predicting the "reply-to" relation between messages based on text similarity and latent semantic transferability. A heuristic method is also brought forward to predict the relation, and then recover the conversation structure. We demonstrate the effectiveness of the proposed method through experiments on a real-world web forum comment data set.

Keywords: Asynchronous conversations · Conversation structure · "Reply-to" relation

1 Introduction

With the blooming of Internet in the last two decades, social networks have embraced huge success from Internet users. From the early chatting room on webpage to the later instant messaging application on personal computers till nowadays' chatting APP on mobile devices, more and more people choose to chat with their friends online. The quantity of online conversations generated in a single day is very huge due to the easy access to the Internet world wide, which makes it possible for thousands of millions users to communicate with each other regardless of locations, time zones and devices. Online conversations are free-style where multiple users can be involved and multiple topics can be

S. Candan et al. (Eds.): DASFAA 2017, Part I, LNCS 10177, pp. 19–34, 2017.
DOI: 10.1007/978-3-319-55753-3_2

discussed at the same time. There have been some studies about the analysis of conversations in social networks [14,24]. In Twitter and Weibo[1], people use "@" to engage his/her friends in the conversations [2,4,9,23] where the logical structures of conversations are very clear.

Generally speaking, there will always be a *structure* in each conversation. That is, someone starts a conversation by bringing up a message of a new topic, and each later message in this conversation replies to one or more previous message(s). For example, in the social news and entertainment site Reddit[2], someone first posts a new topic, and another user can comment on the topic as well as on the previous comments by other users. The comments are structured in a *tree* layout on the webpage so that users can understand how a discussion evolves over time.

However, not all online conversations are well-structured. Instead, there are even larger volumes of free-style conversations without clear "reply-to" relations in real-world scenarios like the popular instant group chat in Tencent QQ, WhatsApp, Skype and LINE. When more than two persons are discussing together online, one user of them may reply to a previous message that (s)he is interested in, rather than the last message in the group chat history. These conversations are usually called *asynchronous conversations* [12] where the *temporal order* fails to represent the *logical order* of a message sequence. We attempt to understand the structure of online short-text conversation in this paper by predicting the "reply-to" relation between the messages in it.

The inherent value of this study is to reconstruct the logic of conversations, profile chatters' information and analyze the relations between chatters [28]. It is especially important for the third-party organizations like strategic consultant companies which cannot directly derive the conversation structure by updating the user interface, e.g. add a "reply" button to each of previous messages. Meanwhile, by recovering the conversation structure, we can visualize the conversations with hierarchical layouts like trees or graphs instead of plain message sequences. Besides, reply suggestion [13] and recommendation [3] (e.g. message/chatter recommendation) are other applications of this study.

To learn the conversation structures, we are confronted with the following challenges:

- The asynchrony of messages makes it difficult to figure out the logical relation between messages. There are multiple users engaged in the discussion and multiple topics are discussed at the same time.
- Due to privacy concerns, there is no publicly available conversation data sets before. It means we have to construct the evaluation data set from the scratch.
- Unlike formal articles, the online messages are usually informal, short and context-sensitive. Thus, the traditional natural language processing methods like Latent Dirichlet Allocation (LDA) [1] usually do not work well in dealing with online conversations.

[1] http://www.weibo.com.
[2] http://www.reddit.com/.

In this paper, we attempt to address the problem of learning online conversation structures by presenting a domain-independent framework based on text features extracted from the conversation corpus. We summarize the main contributions as follows:

- We studied the problem of asynchronous conversation structure learning based on online short-text messages. A domain-independent method was brought forward to address this problem by employing text similarity feature and latent transferability feature based on message contents.
- We proposed a heuristic method to predict the "reply-to" relations and recover the conversation structure. This method avoids yielding disconnected or cyclic structure. Besides, another graph-based method can be employed to get the optimal tree conversation structure.
- We crawled a new online short-text Chinese conversation corpus and used it to evaluate our method. The experimental results show that our method outperforms the baselines in the prediction accuracy.

The rest of the paper is organized as follows: In Sect. 2, we discuss some related work about the studies of conversations. Then, we formally define the major problem in our study in Sect. 3. In Sect. 4, the proposed method based on text similarity and latent semantic transferability is introduced in detail. We demonstrate the experimental results conducted on the new web forum data set in Sect. 5. Finally, we conclude our study and prospect our future work in Sect. 6.

2 Related Work

As far as we concern, the problem studied in this paper has not been well established before. We discuss the related work on conversation disentanglement and clustering, dialogue act learning and some studies about conversation structures in this section.

2.1 Conversation Disentanglement and Clustering

Similar to the famous cocktail party problem, conversation disentanglement, a.k.a. chat disentanglement, describes the task to isolate the messages belonging to the same topic from a long conversation where multiple users are engaged and multiple topics are discussed [6,7,21,25]. Apparently, this is also a clustering problem. Based on the data like timestamp, mention, cue word and text content, the authors in [6,7] propose a maximum-entropy classifier to judge if two messages are of the same topic. They also propose an algorithm to cluster the messages on a directed weighted graph. Later in [25], the clustering performance is improved by enriching the TF-IDF feature of message m with the TF-IDF features of highly relevant messages which share similar timestamp or username with m. Then, a single-pass clustering algorithm can be used to cluster the messages of a conversation into topics.

2.2 Dialogue Act

As a specialized form of *speech act*, dialogue act [22] studies the role, e.g. *Statement*, *Question*, *Agreement* and *Disagreement*, of messages in a conversation. In [18], the authors propose the Hidden Markov Models (HMM) to study the dialogue acts in a conversation where the words are generated from the act emission distribution or the topic multinomials. In [12], the authors first find that using a graph-based model like the graph partition method [6] to deal with dialogue act annotation does not work well, and then, they use an HMM mixture model and consider the emission of dialogue act as the mixtures of multinomials that generate the words in sentences. The results of dialogue act annotation is improved using their proposed method on the Email and forum data sets.

2.3 Conversation Structure

Unlike the conversation disentanglement and the dialogue act modeling, the ultimate goal in this work is to learn the logical structures of online asynchronous conversation by predicting the "reply-to" relations between messages in a given conversation. The most related work to ours is the thread prediction problem [8,26] where it predicts how each message in a newsgroup style conversation is related to each other. However, it differs from our work in several aspects:

- We are dealing with online conversations where messages are much shorter and more informal than the newsgroup conversations in that work.
- The work in [26] only redefines the TF-IDF features and proposes some time interval constraints to predict the relations between messages without considering the message transferability like what we propose in this paper.

Thus, we are tackling a much more challenging problem here and more features of online conversations are taken into account in the proposed method.

3 Problem Definition

In this paper, we learn the online conversation structures by predicting the "reply-to" relation between messages, through which the directed transition edges can be constructed and then the conversation structure is recovered. In order to focus on online conversation structure learning, we assume that each conversation is only about one topic in this paper, and chat disentanglement could be referred in other work [7].

Definition 1 (Online Short-Text Conversation Corpus). *An online short-text conversation corpus is a set of messages* $\mathcal{M} = \{m_1, m_2, \ldots, m_{|\mathcal{M}|}\}$ *from a number of conversations. The message length, i.e. the number of words, of each* $m \in \mathcal{M}$ *is short (e.g. less than 10 words each). The words and phrases in* $\mathcal{M}$ *are usually used in an informal way (e.g. many symbols, abbreviations and Internet words).*

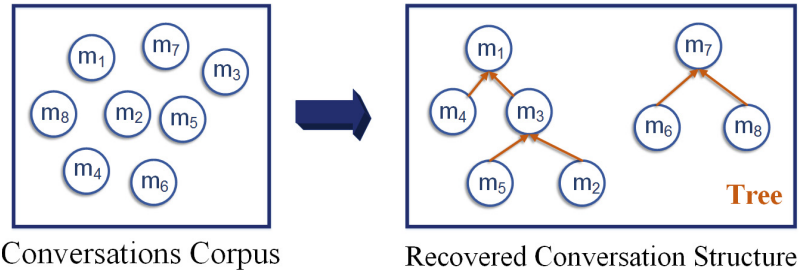

Fig. 1. An illustration of learnt conversation structure. $\mathcal{M}$ here contains two conversations. Each conversation indicates a tree structure.

Definition 2 (Online Short-Text Conversation Structure). *Online short-text conversation structure* $(\mathcal{M}, \prec)$ *is defined by a partial binary operator* $\prec$ *on an online short-text conversation corpus* $\mathcal{M}$*. For* $\forall m_i, m_j \in \mathcal{M}$ *and* $m_i \neq m_j$*, we say* $m_j \prec m_i$ *if and only if: (1)* m_i *and* m_j *are from a same conversation, (2)* m_i *is a reply to* m_j*. Thus,* $(\mathcal{M}, \prec)$ *is namely the "reply-to" structure of a conversation corpus.*

Therefore, our structure learning problem in this paper is to predict the precursor m_j for $\forall m_i \in \mathcal{M}$ based on message content, where m_i and m_j are from a same conversation. This problem is non-trivial due to the asynchrony, informality, and lack of useful cue words in short-text messages.

Figure 1 illustrates an example of the learnt conversation structure. In this example, we assume that each message can at most reply to only one precursor for simplicity, and then the conversation structure is in a tree layout, e.g. web forum conversations. Clearly, the proposed method in this paper can be adapted to deal with DAG structure learning. The study of the directed-acyclic-graph (DAG) structure learning will be our future work.

Figure 1 also shows the difference between our problem and the chat disentanglement problem [7]. The chat disentanglement problem is to cluster messages into different groups (divide the message corpus), but our problem is to predict the "reply-to" relation between messages in a given group (structure learning).

4 Proposed Method

Based on the problem definition, the most basic task of our problem is to identify the "reply-to" relations between messages in a given conversation. We define the "likelihood" that message m_i replies to message m_j as:

$$p_{m_j \prec m_i} = (1-\gamma)\mathcal{S}(m_i, m_j) + \gamma\mathcal{T}(\mathcal{A}(m_i), \mathcal{A}(m_j)). \quad (1)$$

This likelihood consists of two components: text similarity $\mathcal{S}(m_i, m_j)$ and latent transferability $\mathcal{T}(\mathcal{A}(m_i), \mathcal{A}(m_j))$. In this paper, latent transferability is measured by the latent dialogue act transition, which is proposed to alleviate the sparse

text feature problem induced by the short-text characteristics. Here, $\mathcal{A}(m_i)$ represents the latent dialogue act feature of m_i. $\gamma \in [0, 1]$ is a parameter balancing the relative contribution of the two components.

4.1 Measuring Text Similarity

In the literature, the content feature of message is usually represented using the bag-of-words model. We employ the widely-used TF-IDF approach [19] in this study. An alternative is to use Latent Dirichlet Allocation (LDA) [1] to preprocess the corpus. However, we found through experiments that such approach performs poorly in our data set since each message is usually very short (e.g. <10 words) and the phrases are usually informal.

The content feature of message m_i is represented by a W-dimensional column vector $\mathbf{v}_i$ where W is the vocabulary size. The w-th entry of $\mathbf{v}_i$ is the term frequency of the w-th word weighted by the inverse document frequency.

$$v_{i,w} = n_{w,i} \cdot \log \frac{1}{f_w}, \tag{2}$$

where $n_{w,i}$ is the frequency that word w appears in m_i. The document frequency f_w of word w is computed as (Laplace smoothing is applied to avoid division on zero in Eq. (2)):

$$f_w = \frac{n_w + 1}{|\mathcal{M}| + 1}, \tag{3}$$

where n_w is the number of messages which contain word w. Thus, the text similarity between two messages can be measured by their cosine similarity:

$$\mathcal{S}(m_i, m_j) = \frac{\mathbf{v}_i^\top \mathbf{v}_j}{\|\mathbf{v}_i\| \cdot \|\mathbf{v}_j\|}. \tag{4}$$

4.2 Measuring Latent Transferability

"Reply-to" relations are directed. However, the measurement of $\mathcal{S}(m_i, m_j)$ is symmetric. Therefore, we also employ the asymmetric latent transferability between messages based on latent dialogue act features to refine our model. Dialogue acts are high level features of messages. The examples of dialogue acts such as "statement", "question", "answer", or "remark" indicate the roles played by messages in conversations. However, automatic dialogue act classification requires a large amount of user annotation to perform model training [20]. Besides, the performance of these explicit dialogue act classification methods is degraded on the online short-text conversation corpus. Therefore, we propose to use unsupervised learning to get the latent dialogue act feature for each message and use it in the transferability measurement.

TF-DF Feature. Compared with the crucial role that infrequent words play in text mining and information retrieval, the functionality of frequent words is usually ignored in the literature. However, we find that frequent words usually serve as important indicators of the act that each message represents. The benefit to consider frequent words becomes more obvious when the general length of message is short. Therefore, we define the *term-frequency-document-frequency (TF-DF)* feature for each message. The TF-DF of message m_i is an F-dimensional column vector $\mathbf{x}_i$ where $F \ll W$ is the number of the most frequent words in the vocabulary. The reason why only Top-F frequent words rather than all words are used is that we need to reduce the computation cost to learn the dialogue act features without great loss of accuracy. The w-th component of $\mathbf{x}_i$ can be computed as follows:

$$\mathbf{x}_{iw} = n_{w,i} \cdot \frac{1}{1 + \mathrm{e}^{-(1+\ln f_w)}} = n_{w,i} \cdot \frac{f_w}{f_w + \mathrm{e}^{-1}}. \tag{5}$$

Note that we use the sigmoid function to rescale the document frequency. The basic idea of TF-DF is that the weights of infrequent words should be less important than those of frequent words which indicate the dialogue act features.

Since the number of existing dialogue acts is much less than the vocabulary size W[3], we need to compress TF-DF feature $\mathbf{x}_i$ into latent dialogue act feature in much lower dimensions. Suppose there are totally K distinct acts to be considered ($K \ll F \ll W$). Let $\mathbf{y}_i \in \mathbb{R}^K$ denote the latent dialogue act feature of message m_i. Then, we need a dialogue act transformation matrix $\mathbf{A} \in \mathbb{R}^{F \times K}$ such that:

$$\mathbf{x}_i = \mathbf{A}\mathbf{y}_i. \tag{6}$$

Suppose $\mathbf{A}$ is already given, we can compute the latent dialogue act feature of each message m_i as:

$$\mathcal{A}(m_i) = \mathbf{y}_i = \mathbf{A}^\dagger \mathbf{x}_i, \tag{7}$$

where $\mathbf{A}^\dagger$ is the pseudo-inverse of $\mathbf{A}$.

Latent Dialogue Act Feature. Now we focus on the estimation of matrix $\mathbf{A}$ from $\mathcal{M}$. Let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{|\mathcal{M}|})$ and $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{|\mathcal{M}|})$, then $\mathbf{X} = \mathbf{A}\mathbf{Y}$. Our aim is to estimate $\mathbf{A}$ and $\mathbf{Y}$ given the observations on $\mathbf{X}$. Although this could be considered as a non-negative matrix factorization problem [16,17,27], we choose the independent component analysis (ICA) method [10,11] instead. Because it is more likely that the latent dialogue act feature of each message is separately emitted and mixed from K independent dialogue acts (or latent independent components), but non-negative matrix factorization could not guarantee such independence.

We need to conduct data whitening on $\mathbf{X}$ before performing ICA as discussed in [11]. According to Theorem 1, we firstly make random variable $\mathbf{x} \in \mathbf{X}$ has zero mean by subtracting its expectation (the mean in practice), $\mathbf{x} = \mathbf{x} - \mathrm{E}[\mathbf{x}]$. Then,

[3] http://en.wikipedia.org/wiki/Dialog_act.

we perform singular value decomposition (SVD) on $\mathbf{X}$, $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{F\times F}$, $\mathbf{\Sigma} \in \mathbb{R}^{F\times|\mathcal{M}|}$ and $\mathbf{V}^\top \in \mathbb{R}^{|\mathcal{M}|\times|\mathcal{M}|}$.

Theorem 1. *For* $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{|\mathcal{M}|})$*, if random variable* $\mathbf{x} \in \mathbf{X}$ *has zero mean, i.e.* $\mathrm{E}[\mathbf{x}] = \mathbf{0}$*, and* $\mathbf{X}$ *has singular value decomposition* $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$*, then let* $\mathbf{z} = \left(\frac{1}{\sqrt{|\mathcal{M}|}}\mathbf{\Sigma}\right)^{-1}\mathbf{U}^\top\mathbf{x}$*, random variable* $\mathbf{z}$ *will be whitened.*

Proof. Since $\mathrm{E}[\mathbf{x}] = \mathbf{0}$, then

$$\mathrm{E}[\mathbf{z}] = \sqrt{|\mathcal{M}|}\mathbf{\Sigma}^{-1}\mathbf{U}^\top\mathrm{E}[\mathbf{x}] = \mathbf{0}. \tag{8}$$

We also have:

$$\mathrm{E}[\mathbf{x}\mathbf{x}^\top] \approx \frac{1}{|\mathcal{M}|}\mathbf{X}\mathbf{X}^\top \tag{9}$$

$$= \frac{1}{|\mathcal{M}|}(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top)(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top)^\top \tag{10}$$

$$= \frac{1}{|\mathcal{M}|}\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^\top. \tag{11}$$

Then, we can prove that:

$$\mathrm{E}[\mathbf{z}\mathbf{z}^\top] = \mathrm{E}[\left(\frac{1}{\sqrt{|\mathcal{M}|}}\mathbf{\Sigma}\right)^{-1}\mathbf{U}^\top\mathbf{x}\mathbf{x}^\top\mathbf{U}\left(\frac{1}{\sqrt{|\mathcal{M}|}}\mathbf{\Sigma}\right)^{-1}] \tag{12}$$

$$= |\mathcal{M}|\mathbf{\Sigma}^{-1}\mathbf{U}^\top\mathrm{E}[\mathbf{x}\mathbf{x}^\top]\mathbf{U}\mathbf{\Sigma}^{-1} \tag{13}$$

$$= |\mathcal{M}|\mathbf{\Sigma}^{-1}\mathbf{U}^\top\frac{1}{|\mathcal{M}|}\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^\top\mathbf{U}\mathbf{\Sigma}^{-1} \tag{14}$$

$$= \mathbf{\Sigma}^{-1}\mathbf{U}^\top\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^\top\mathbf{U}\mathbf{\Sigma}^{-1} \tag{15}$$

$$= \mathbf{\Sigma}^{-1}\mathbf{\Sigma}^2\mathbf{\Sigma}^{-1} \tag{16}$$

$$= \mathbf{I}. \tag{17}$$

Thus, random variable $\mathbf{z}$ has zero mean and unit variance. That means $\mathbf{z}$ is whitened.

Algorithm 1 shows the procedure to estimate the matrix $\mathbf{A}$ and compute the latent dialogue act features. After performing SVD on $\mathbf{X}$, we compress TF-DF features into much lower K-dimensional space by preserving the K largest singular values and get an approximation matrix $\widetilde{\mathbf{X}} \in \mathbb{R}^{K\times|\mathcal{M}|}$ of $\mathbf{X}$ (Line 3–4). Then $\widetilde{\mathbf{x}}$ is whitened by transforming to random variable $\widetilde{\mathbf{z}}$ based on Theorem 1 (Line 5–7). Since $\widetilde{\mathbf{Z}}$ is whitened now, we can perform ICA on it, and let $\widetilde{\mathbf{Z}} = \mathbf{A}\mathbf{Y}$. From the ICA point of view, $\widetilde{\mathbf{Z}}$ is a linear mixture of some statistically independent signals. In this paper, we employ the FastICA algorithm [11][4] to get the unmixing matrix $\mathbf{W}$ (Line 8), from which we can get the inverse act transformation matrix $\mathbf{A}^\dagger$ and the dialogue act features $\mathbf{Y}$ of messages in the conversation

[4] see the Python library: http://scikit-learn.org/.

Algorithm 1. Latent Dialogue Act Feature Estimation

Input:
TF-DF features $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_{|\mathcal{M}|}\} \in \mathbb{R}^{F \times |\mathcal{M}|}$,
dimensions of dialogue act features K.

Output:
inverse act transformation matrix: $\mathbf{A}^\dagger \in \mathbb{R}^{K \times F}$,
latent dialogue act features: $\mathbf{Y} \in \mathbb{R}^{K \times |\mathcal{M}|}$

$\mathbf{X} \leftarrow \mathbf{X} - \{\mathrm{E}[\mathbf{x}], ..., \mathrm{E}[\mathbf{x}]\}$
$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \mathrm{SingularValueDecomposition}(\mathbf{X})$
$\widetilde{\mathbf{U}}, \widetilde{\mathbf{\Sigma}}, \widetilde{\mathbf{V}} \leftarrow \mathrm{DimentionReduction}(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}, K)$
$\widetilde{\mathbf{X}} \leftarrow \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{V}}$
for all $\widetilde{\mathbf{x}}_i \in \widetilde{\mathbf{X}}$ **do**
  $\widetilde{\mathbf{z}}_i = \left(\frac{1}{\sqrt{|\mathcal{M}|}}\widetilde{\mathbf{\Sigma}}\right)^{-1}\widetilde{\mathbf{U}}^\top\widetilde{\mathbf{x}}_i,$
end for
$\mathbf{W} \leftarrow \mathrm{FastICA}(\widetilde{\mathbf{Z}})$ /*$\widetilde{\mathbf{Z}} = \{\widetilde{\mathbf{z}}_1, ..., \widetilde{\mathbf{z}}_{|\mathcal{M}|}\}$*/
$\mathbf{A}^\dagger \leftarrow \mathbf{W}\left(\frac{1}{\sqrt{|\mathcal{M}|}}\widetilde{\mathbf{\Sigma}}\right)^{-1}\widetilde{\mathbf{U}}^\top$
$\mathbf{Y} \leftarrow \mathbf{A}^\dagger\mathbf{X}$
return $\mathbf{A}^\dagger, \mathbf{Y}$

corpus. The result above assumes that the components of the random vector $\mathbf{y}$ is independent. This is usually not the case in reality, especially in our setting where $\mathbf{y}$ encodes the strength of different latent acts. However, previous applications of ICA show that this technique can still gain insights into the data set even if the independence assumption is violated.

Latent Transferability Measurement. We define latent transferability (likelihood of m_i replies to m_j) as below:

$$\mathcal{T}(\mathcal{A}(m_i), \mathcal{A}(m_j)) = \mathcal{T}(\mathbf{y}_i, \mathbf{y}_j) = \hat{\mathbf{y}}_i^\top \mathbf{B}\hat{\mathbf{y}}_j, \tag{18}$$

where $\hat{\mathbf{y}}_i = \frac{abs(\mathbf{y}_i)}{\|abs(\mathbf{y}_i)\|_1}$, $abs(\mathbf{y}_i)$ is the absolute value of $\mathbf{y}_i$. Please note that $\mathcal{T}$ is asymmetric. For a list of messages $m_{p_1}, m_{p_2}, \ldots, m_{p_N}$, suppose the messages they reply to are $m_{q_1}, m_{q_2}, \ldots, m_{q_N}$, respectively. Let

$$\mathbf{Y}_p = \left(\hat{\mathbf{y}}_{p_1}, \hat{\mathbf{y}}_{p_2}, \ldots, \hat{\mathbf{y}}_{p_N}\right), \tag{19}$$

$$\mathbf{Y}_q = \left(\hat{\mathbf{y}}_{q_1}, \hat{\mathbf{y}}_{q_2}, \ldots, \hat{\mathbf{y}}_{q_N}\right). \tag{20}$$

To learn the optimal transition matrix $\mathbf{B}$, we minimize the square error between $\mathbf{Y}_p^\top\mathbf{B}$ and $\mathbf{Y}_q$, as follows:

$$\widehat{\mathbf{B}} = \underset{\mathbf{B} \in \mathbb{R}^{K \times K}}{\operatorname{argmin}} \|\mathbf{Y}_p^\top\mathbf{B} - \mathbf{Y}_q\|^2. \tag{21}$$

This problem can be solved by employing the non-negative least square algorithm [15]. Thus, we can estimate the final likelihood $p_{m_j \prec m_i}$ that message m_i

replies to m_j using Eq. (1). Please note that although $\mathbf{B}$ is inferred based on the training data set, it models the transition likelihood between the latent dialogue act features, and thus, it is also generalized to the unseen message pairs in prediction.

4.3 Conversation Structure Recovery

We consider the tree structure recovery problem in this paper, and leave the DAG structure recovery problem in our future work. Actually, the difference between the tree structure recovery and the DAG structure recovery lies in the possibility that each meassage can or cannot reply to more than one previous message. A simple strategy based on this work is to predefine a threshold η of likelihood $p_{m_j \prec m_i}$ to determine the precursor(s) of each message so that n is a reply to m $\forall m, p_{m \prec n} \geq \eta$. For tree structure recovery, since each non-root node has only one parent node, we can predict that for each non-root message $m_i \in \mathcal{M}$, the one that m_i replies to should maximize the "likelihood" $p_{m_j \prec m_i}$ where m_i and m_j are from the same conversation and $m_i \neq m_j$. This strategy is straightforward and simple to implement. Unfortunately, it is also flawed since it may generate an unexpected disconnected or cyclic structure. Figure 2 illustrates a failure example using this strategy. According to the "likelihood" table in Fig. 2, message m_4 and m_5 mutually reply to each other, which makes the conversation structure disconnected and generates a cyclic sub-structure. However, the expected structure is a single rooted tree as shown in Fig. 2(c). The reason for this failure is because this strategy ignores the constraint on the global conversation structure itself, i.e. the structure connectivity and the acyclic property.

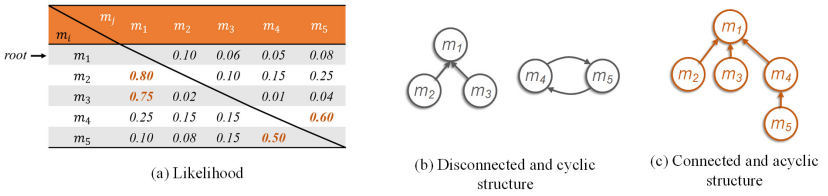

Fig. 2. An example of failure of simple prediction method.

To tackle this problem, we propose a heuristic method for fast computation. Alternatively, we can also use a less-efficient graph-based method to get the optimal results. For the heuristic method, we initialize two sets: $\mathbf{D}$ as empty set, $\mathbf{M}$ as the set containing all messages in a given conversation. We iteratively move one message from $\mathbf{M}$ to $\mathbf{D}$ until $\mathbf{M}$ becomes empty. Each time, we move $m \in \mathbf{M}$ so that:

$$\left(\underset{m_i \in \mathbf{M} \cup \mathbf{D}}{\operatorname{argmax}} \, p_{m_i \prec m} \right) \in \mathbf{D}. \tag{22}$$

Algorithm 2. Conversation Structure Recovery

Input: "reply-to" likelihood table $p_{\prec}$, message set $\mathbf{M}$, total message set $\mathbf{N}$.
Output: conversation structure $\mathcal{G}$.

Initialize $\mathcal{G} \leftarrow \emptyset, \mathbf{D} \leftarrow \emptyset$.
Identify conversation root message $r \in \mathbf{M}$.
$\mathbf{D} \leftarrow \mathbf{D} \cup \{r\}, \mathbf{M} \leftarrow \mathbf{M} \setminus \{r\}$.
while $\mathbf{M} \neq \emptyset$ **do**
  while $m, n^* \leftarrow$ NextToMove($p_{\prec}$, $\mathbf{N}$, $\mathbf{M}$, $\mathbf{D}$) $\neq$ NULL **do**
    $\mathbf{D} \leftarrow \mathbf{D} \cup \{m\}, \mathbf{M} \leftarrow \mathbf{M} \setminus \{m\}$
    $\mathcal{G} \leftarrow \mathcal{G} \cup \{n^* \prec m\}$
  end while
  if $\mathbf{M} \neq \emptyset$ **then**
    $n^* \prec m^* \leftarrow \text{argmax}_{m \in \mathbf{M}} \left(\max_{n \in \mathbf{D}} p_{n \prec m} \right)$
    $\mathbf{D} \leftarrow \mathbf{D} \cup \{m^*\}, \mathbf{M} \leftarrow \mathbf{M} \setminus \{m^*\}$
    $\mathcal{G} \leftarrow \mathcal{G} \cup \{n^* \prec m^*\}$
  end if
end while
return $\mathcal{G}$

Algorithm 3. NextToMove

Input: "reply-to" likelihood table $p_{\prec}$, total message set $\mathbf{N}$, unvisited message set $\mathbf{M}$, visited message set $\mathbf{D}$.
Output: next movable candidate m and its precursor n^*.

for $m \in \mathbf{M}$ **do**
  $n^* \leftarrow \text{argmax}_{n \in \mathbf{N}} p_{n \prec m}$
  if $n^* \in \mathbf{D}$ **then**
    return m, n^*
  end if
end for
return NULL

It means that for any message in $\mathbf{M}$, the maximum "reply-to" likelihood should be associated with a message in $\mathbf{D}$. If such an m cannot be found in $\mathbf{M}$, we move the following message,

$$\underset{m \in \mathbf{M}}{\text{argmax}} \left(\max_{m_i \in \mathbf{D}} p_{m_i \prec m} \right). \tag{23}$$

After each move, we create a "reply-to" relation from m to $\text{argmax}_{m_i \in \mathbf{D}} p_{m_i \prec m}$. It is apparent that the heuristic method generates a connected and acyclic tree structure. The pseudo code of the heuristic method is show in Algorithm 2. The root message is always chosen as the topic itself in our experiments on the web forum data set.

The heuristic method is fast but also sub-optimal. To get the optimal tree structure, we can consider the messages as nodes in a directed weighted graph and the likelihood $p_{m_i \prec m_j}$ as edge weights. Then, the optimal tree structure can be obtained by applying the Edmond's algorithm [5] to find the maximum spanning arborescence.

Auxiliary Filters. The proposed method is solely based on message contents. However, we can further improve it by employing auxiliary filters.

Time Filter: It is obvious that each message can only reply to the earlier posted message(s). If time information or posted order of messages is available, we can apply this filter in the recovery process.

User Filter: Generally, a chatter does not reply to himself in online conversations. This filter removes the candidates of self-replies in the recovery process, but it works if user identity of posted messages is known.

Both filters are applied in later experiments.

5 Experiments

In this section, we first introduce the new data set we collected. Then, the experimental results on this new data set are demonstrated and discussed.

5.1 Data Set

We investigated Douban Group[5], a popular Chinese web forum. In Douban Groups, users can publish topics for discussion. When someone replies to a comment c under a conversation, the content of c is automatically quoted by the new comment. This makes it possible to reconstruct conversations by tracing the quoting relations among comments. This is how we obtain the ground-truth of "reply-to" relations in our experiments. Please note that we choose web forum chats for evaluation since the ground-truth can be obtained, but we aim to solve the conversation structure learning problem for those unstructured chats, e.g. online group chat. We crawled 10,425 conversations on Douban Group in August, 2013, containing 137,980 messages in total. Each conversation in this data set has a tree structure ground-truth since one user posts a comment (a.k.a. reply) by quoting only one previous comment or the topic. After performing Chinese word cut, each message has about 12 words on average, which is very short.

5.2 Results and Discussion

In the experiments, the method using the conventional text similarity (TF-IDF) only is denoted by "TEXT". The two methods in [26] which redefines the TF-IDF feature and makes some constraints on the time interval between two potentially related messages are denoted by "FIXED" and "TIMED", respectively. The former reduces the number of candidate messages by setting a fixed time intervals, while the latter decreases the importance of candidate messages as the time interval increases. The method only based on latent act transferability is named as ACT. The proposed method is denoted by TACT. We use -H and -E to denote methods using the heuristic and the Edmond's algorithms as the structure recovery strategies, respectively.

[5] http://www.douban.com/group.

In the experiment, we randomly choose 80% conversations from our Douban Group data set as the training set (e.g. learn the matrices **A** and **B** in the proposed method), and leave the rest as the testing set. The numbers we reported are the averages after running experiments for 5 times. We use "reply-to" relation prediction accuracy as the major measurement. The accuracy is computed as:

$$\frac{\#\text{correctly predicted reply-to relations}}{\#\text{total reply-to relations}}.$$

Only one precursor is predicted for each message, i.e. Top-1 prediction.

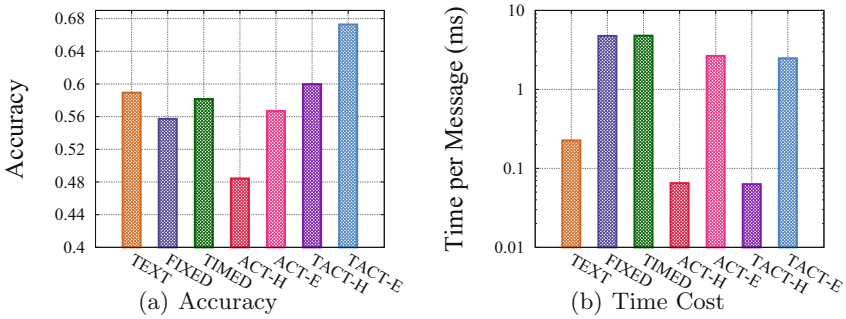

Fig. 3. Comparisons of the accuracy and the efficiency performance.

Predict "Reply-To" Relations. Figure 3(a) shows the results of accuracy performance in the experiments. According to the results, we can see: (1) The proposed methods (TACT-H and TACT-E) generally have the best accuracy performance compared with other baselines. The best accuracy is achieved by TACT-E method at around 67.5%. Considering that the accuracy is obtained on Top-1 precursor prediction, the proposed method is very effective on this data set; (2) Obviously, Edmond's algorithm is always better in accuracy performance than the heuristic method; (3) Both "FIXED" and "TIMED" perform slightly worse than "TEXT", which may result from the redefinition of its TF-IDF features that changes the important signals in representing the short and informal messages; besides, the interval constraint in "FIXED" may also leave the real precursor messages out of consideration and lead to poor performance.

As for the efficiency evaluation, we use the average time cost to predict a single "reply-to" relation as the metric, i.e.

$$\frac{\text{Time to recover a conversation}}{\#\text{ messages in a conversation}}.$$

The results are shown in Fig. 3(b). Apparently, the heuristic method is much efficient than the Edmond's algorithm as well as the other baselines. Both of FIXED and TIMED work very slow in the experiments, while TEXT has the medium efficiency.

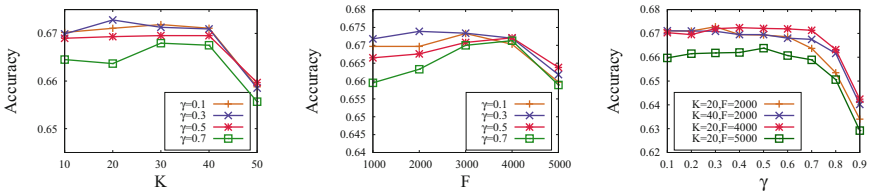

Fig. 4. Accuracy performance under different settings of parameters.

The comparisons show that TACT-H and TACT-E are both effective and efficient in the structure recovery problem. Meanwhile, TACT-H and TACT-E have the advantages of efficiency and effectiveness over the baselines, respectively.

Sensitivity of Parameters. We also analyzed the sensitivity of parameters in TACT, i.e. the dimensions of latent dialogue act feature K, the dimensions of TF-DF feature F, as well as the balancing parameter γ. If not explicitly specified, the default settings of parameters are $K = 20$, $F = 2000$ and $\gamma = 0.5$ in the experiments.

Figure 4(a) shows the performance with different K values. A larger value of K indicates a larger number of latent dialogue acts to consider, but also a larger cost to learn the transition matrix $\mathbf{B}$ and a higher probability to incorporate redundant latent dialogue acts. From the results, we can see that the optimal setting of K value should be around 20.

Figure 4(b) illustrates the performance by changing the dimensions of TF-DF features, i.e., the number of frequent words. The value of F determines the size of matrix $\mathbf{X}$, which means a larger value of F leads to a larger cost to factorize $\mathbf{X}$ with SVD. According to the results, we can see that $F = 2000$ is a good choice in our experiment.

Lastly, the performance with different γ values is shown in Fig. 4(c). It is obvious from the results that the accuracy is very similar when $\gamma \leq 0.7$, and there is an accuracy drop when γ gets closer to 1.0. But the overall performance of the proposed method is stable in the experiments.

6 Conclusion

We investigate the problem of recovering the structure of online short-text conversations. A novel framework combining text similarity and latent semantic transferability between messages is brought forward, and a heuristic method as well as a graph-based one are also presented to recover the conversation structure. The evaluation on the new data set we collected shows the effectiveness and the efficiency of the proposed method. In the future, we are considering to incorporate more linguistic features like syntactic feature and word embeddings in the framework to get more accurate in exploring the relations between messages.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China (No. 61373023, No. 61133002, No. 61502116), the China National Arts Fund (No. 20164129), and the National Science Foundation (NSF) under grant No. CNS-1252292.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
2. Boyd, D., Golder, S., Lotan, G.: Tweet, tweet, retweet: conversational aspects of retweeting on twitter. In: Proceedings of the 43rd Hawaii International Conference on System Sciences, pp. 1–10. IEEE (2010)
3. Chen, J., Wang, C., Wang, J.: A personalized interest-forgetting markov model for recommendations. In: AAAI, pp. 16–22 (2015)
4. Cook, J., Kenthapadi, K., Mishra, N.: Group chats on twitter. In: WWW, pp. 225–236 (2013)
5. Edmonds, J.: Optimum branchings. J. Res. Natl. Bur. Stand. B. Math. Math. Phys. **71B**(4), 233–240 (1967)
6. Elsner, M., Charniak, E.: You talking to me? a corpus and algorithm for conversation disentanglement. In: ACL, pp. 834–842 (2008)
7. Elsner, M., Charniak, E.: Disentangling chat. Comput. Linguist. **36**(3), 389–409 (2010)
8. Gandhi, S., Jones, A.R., Nesbitt, P.A., Seacat, L.A.: Instant conversation in a thread of an online discussion forum, November 2015. http://www.freepatentsonline.com/9177284.html
9. Honey, C., Herring, S.C.: Beyond microblogging: conversation and collaboration via twitter. In: Proceedings of the 42nd Hawaii International Conference on System Sciences, pp. 1–10. IEEE (2009)
10. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis. IEEE Trans. Neural Netw. **10**(3), 626–634 (1999)
11. Hyvärinen, A., Oja, E.: Independent component analysis: algorithms and applications. Neural Netw. **13**(4), 411–430 (2000)
12. Joty, S., Carenini, G., Lin, C.Y.: Unsupervised modeling of dialog acts in asynchronous conversations. In: IJCAI, pp. 1807–1813 (2011)
13. Kannan, A., Kurach, K., Ravi, S., Kaufmann, T., Tomkins, A., Miklos, B., Corrado, G., Lukacs, L., Ganea, M., Young, P., Ramavajjala, V.: Smart reply: automated response suggestion for email. In: KDD, pp. 955–964 (2016)
14. Kumar, R., Mahdian, M., McGlohon, M.: Dynamics of conversations. In: KDD, pp. 553–561 (2010)
15. Lawson, C.L., Hanson, R.J.: Solving Least Squares Problems, vol. 161. Prentice-Hall, Englewood Cliffs (1974)
16. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. NIPS **13**, 556–562 (2000)
17. Lin, C.J.: Projected gradient methods for nonnegative matrix factorization. Neural Comput. **19**(10), 2756–2779 (2007)
18. Ritter, A., Cherry, C., Dolan, B.: Unsupervised modeling of twitter conversations. In: NAACL, pp. 172–180 (2010)
19. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. **24**(5), 513–523 (1988)

20. Serafin, R., Eugenio, B.D.: FLSA: extending latent semantic analysis with features for dialogue act classification. In: ACL (2004). No. 692
21. Shen, D., Yang, Q., Sun, J.T., Chen, Z.: Thread detection in dynamic text message streams. In: SIGIR, pp. 35–42 (2006)
22. Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Ess-Dykema, C.V., Meteer, M.: Dialogue act modeling for automatic tagging and recognition of conversational speech. Comput. Linguist. **26**(3), 339–373 (2000)
23. Uthus, D.C., Aha, D.W.: The Ubuntu chat corpus for multiparticipant chat analysis. In: Proceedings of the AAAI Spring Symposium (2013)
24. Wang, C., Ye, M., Huberman, B.A.: From user comments to on-line conversations. In: KDD, pp. 244–252 (2012)
25. Wang, L., Oard, D.W.: Context-based message expansion for disentanglement of interleaved text conversations. In: NAACL, pp. 200–208 (2009)
26. Wang, Y.C., Joshi, M., Cohen, W.W., Rosé, C.P.: Recovering implicit thread structure in newsgroup style conversations. In: Proceedings of the 2nd International Conference on Weblogs and Social Media (2008)
27. Wang, Y.X., Zhang, Y.J.: Nonnegative matrix factorization: a comprehensive review. TKDE **25**(6), 1336–1353 (2013)
28. Zhang, J., Wang, C., Wang, J., Yu, J.X., Chen, J., Wang, C.: Inferring directions of undirected social ties. TKDE **28**(12), 3276–3292 (2016)

http://www.springer.com/978-3-319-55752-6

Database Systems for Advanced Applications
22nd International Conference, DASFAA 2017, Suzhou, China, March 27-30, 2017, Proceedings, Part I
Candan, S.; Chen, L.; Pedersen, T.B.; Chang, L.; Hua, W. (Eds.)

2017, XXIII, 688 p. 228 illus., Softcover
ISBN: 978-3-319-55752-6