Thread Structure Learning on Online Health Forums With Partially Labeled Data

Yunzhong Liu, Jinhe Shi, and Yi Chen

***Abstract*— Thread structures, the reply relationships between posts, in online forums are very important for readers to understand the thread content, and for improving the effectiveness of automated forum information retrieval, expert findings, and so on. However, most online forums only have partially labeled structures, which means that some reply relationships are known while the others are unknown. To address this problem, studies have been performed to learn and predict thread structures. However, existing work does not leverage the partially available thread structures to learn the complete thread structure. We have also observed that many online health forums are a type of person-centric forums, where persons are mentioned across posts, providing hints about the reply relationships between posts. In this article, we first proposed to learn the complete thread structures by leveraging the partially known structures based on a statistical machine learning model—thread conditional random fields (threadCRFs). Then, we proposed to use person resolution, the process of identifying the same person mentioned in different contexts, together with threadCRF for thread structure learning. We have empirically verified the effectiveness of the proposed approaches.**

***Index Terms*— Thread conditional random fields (threadCRF), thread structure learning.**

I. Introduction

ONLINE forums provide a convenient channel for people to share their experience and exchange ideas and have attracted more and more users. They become valuable resources for extracting useful knowledge, through the forum search, question answering, and expert finding. A typical forum thread consists of a sequence of posts, ordered according to the time when the post is submitted. Logically, a thread can be represented by a tree structure, where each post has one parent to which it replies, except the first post—the root of the tree [1]. One post can be replied by multiple posts, that is, can have many children. An example of a forum thread in tree representation is shown in Fig. 1, which is extracted from one thread in the epilepsy foundation forum (http://epilepsyfoundation.ning.com/forum). The tree structure of forum threads can save users time and effort to track and get involved in the discussion and help them to understand the interaction among forum users, such as who is following whom or who is the receiver of a suggestion. The literature also demonstrates that thread structure can boost the performance of automated forum information extraction [2], information retrieval [3], [4], clustering [5], online community search [6], topic summarization [7], and experts finding [8].

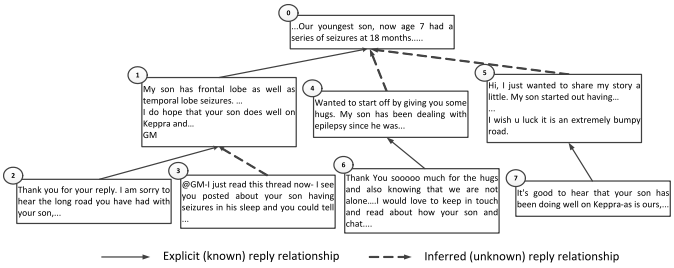


Fig. 1. Sampled thread of eight post fragments.

However, most of the Web forums do not have the complete thread structures available, which means the parents of some posts are unknown. Many forum authors just use the default mode to reply without specifying to which posts they reply, nor quoting existing posts.

There is existing work for learning complete thread structures [1], [9], [10]. For example, thread conditional random fields (threadCRFs), proposed by [1], have been shown effective in learning thread structures. They typically require training data that have complete thread structures, which is not always available and sometimes obtained through labor-intensive manual labeling. We observed two properties in online health forums that we would like to leverage to learn thread structures in a scalable way without manually labeled training data. One is the prevalently available partially labeled thread structures in online forums, and the other is the key role that person references play in person-centric forums.

A. Partially Labeled Thread Structures

In reality, online forums have an abundance of partially labeled reply structures. There are always some post authors who have a good habit of keeping an explicit reply structure. An example of such a partially labeled thread structure is shown in Fig. 1. In this forum, if one post explicitly replies to

Manuscript received February 8, 2019; revised August 23, 2019; accepted September 26, 2019. Date of publication October 29, 2019; date of current version December 9, 2019. This work was supported in part by the Leir Foundation and in part by the National Institutes of Health under Grant UL1TR003017. *(Yunzhong Liu and Jinhe Shi contributed equally to this work.) (Corresponding author: Yi Chen.)*

Y. Liu is with the Department of Computer Science, Arizona State University, Tempe, AZ 85281 USA (e-mail: liuyz@asu.edu).

J. Shi is with the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: js675@njit.edu).

Y. Chen is with the Martin Tuchman School of Management, New Jersey Institute of Technology, Newark, NJ 07103 USA, and also with the Ying Wu College of Computing, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: yi.chen@njit.edu).

Digital Object Identifier 10.1109/TCSS.2019.2946498

another post, it will quote that post. We can easily obtain some explicit reply relationships from the quotation relationship, as indicated by the solid arrow in Fig. 1. While such partially labeled thread structures are prevalent in online forums and can provide valuable information, they are not leveraged in the existing work.

B. Person-Centric Forums

We observed two types of online forums. Some forums are centered around specific questions or topics, such as most of the technical discussion forums. On the other hand, some forums are centered around persons, such as the health forums for patients and caregivers to share experiences and support each other. Typically health forum users introduce problems and make comments in a subjective way, describing personal experience and giving feedback to other users. In other words, a thread has a collection of user cases raised by some forum users and commented by others. Since a post often refers to other persons either mentioned in this post, in the parent, or ancestor post, identifying correct thread structure in person-centric forums is even more important than other forums to understand the context.

On the other hand, person-centric forums also bring opportunities. There are often people mentions in the posts. When one post replies to another, it tends to mention the person described in the parent or ancestor post. Conversely, if one post mentions a person that is described in a preceding post, then this post is likely to be a child or descendant of that preceding post. According to this observation, if we can find out the person references, we can use them to help to learn the thread reply structures. Indeed, often forums are written representation of conversations among a group of people. The references of persons provide great hints on who talks to whom in a "chat room."

C. Our Contributions

In this article, we first propose to learn complete thread structures from the partially labeled structures based on the threadCRF learning model. Then, we leverage the person reference information and combine it with threadCRF for thread structure learning. The person references can be obtained using person resolution (PR) techniques, which identify the same person mentioned in a different context. We use unsupervised PR techniques to materialize the most likely candidates for unknown thread reply structures and generate a fully labeled training data set. This data set can be considered as an approximation of the ground truth and used to bootstrap the supervised threadCRF model training. We then use the learned model to relabel the unknown thread structures with the partially known structures as constraints. In addition to being used for training data generation, the person references are also encoded as semantic features and incorporated into the learning model to further improve the thread structure learning performance. By leveraging person references information discovered in semantic analysis of posts, and combining them with the syntactic and structural features captured by threadCRF, the proposed approaches provide a unified framework for thread structure learning. We have empirically verified the effectiveness of the proposed approaches.

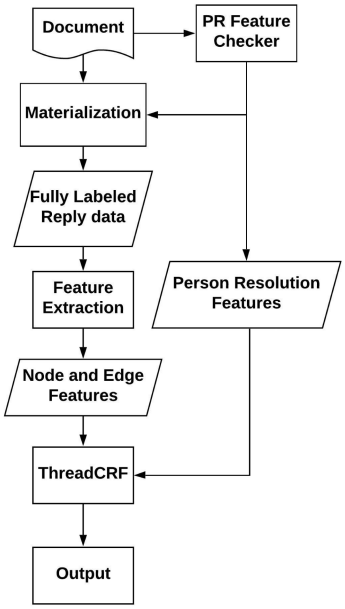


Fig. 2. System architecture.

II. Leveraging Partially Labeled Data for Thread Structure Learning

In this section, we first define the problem and introduce the threadCRFs model. Then, we introduce how to train and learn with the partially labeled thread structures.

A. Problem Definition

Given a thread X_n with a sequence of m posts $\{p_0, p_1, \ldots, p_{m-1}\}$, we need to find the parent post for each post in X_n, denoted by $Y_n = \{y_1, y_2, \ldots, y_{m-1}\}$, where y_i is the known or predicted parent for p_i. Note that we only need to predict y_i for $i > 1$, since the first post has no parent and the second post's parent is always the first post ($y_1 = p_0$).

B. System Architecture

Fig. 2 shows the system architecture. The input data is partially labeled forum threads. The PR feature checker (to be discussed in Section III-A) extracts PR features from partially labeled data. Utilizing the PR features, the materialization module converts a partially labeled thread to a fully labeled thread (Sections II-D and III-B). Then, the feature extraction module extracts node and edge features from the fully labeled thread. These features, together with PR features, are subsequently fed into the threadCRF model (Sections II-C and III-C) to learn the parent label of each post (if unknown) in the thread and output the thread with completely labeled reply relationships. Next, we discuss each module in the system.

C. Thread Conditional Random Fields

The (*threadCRF*) model, proposed by [1], is shown effective in learning thread structures. It is a supervised learning

approach that requires a fully labeled data set for training. In threadCRF, given the post sequence in X_n and the model parameter set $\Theta = \{\lambda_k\}_{k=1}^{K}$, the conditional distribution of Y_n is defined as follows:

$$p(Y_n|X_n, \Theta) \propto \exp\left(\sum_{k=1}^{K} \lambda_k f_k(X_n, Y_n)\right) \quad (1)$$

where $\{f_k(X_n, Y_n)\}_{k=1}^{K}$ is the set of features for the post sequence in X_n and the parent labeling sequence Y_n, and $\{\lambda_k\}_{k=1}^{K}$ are the weights for those corresponding features. The thread structure learning task is formulated as a maximum *a posteriori* (MAP) inference problem to find the optimal reply structure Y^*

$$Y^* = \arg\max_{Y \in \mathbf{\Psi}} \; p(Y|X_n, \Theta) \quad (2)$$

where $\mathbf{\Psi}$ is the set of all possible reply structures for thread X_n.

The key to the above-mentioned threadCRF framework is to define a set of features to capture the interdependency among the posts in terms of the reply structure. Thirteen features are used, including six node features and seven edge features. A node feature only depends on a pair of posts, say p_i and p_j with $i > j$, to determine how likely p_i replies to p_j. For example, *content similarity* is one of such node features—if the content of p_i is similar to that of p_j, p_i is likely replying to p_j. An edge feature captures the dependence between two pairs of reply relationships. For example, one edge feature is *repeat reply*—if we know that Alice has replied to Bob, then the following post written by Bob is likely replying to Alice.

To handle the complexity associated with the edge features, which capture the long-distance dependence among posts, an approximate MAP inference is used for (2) to learn the model parameters from the training data set. Given a training set $T = \{X_1, X_2, \ldots, X_N\}$, with the ground-truth parent labels $R = \{Y_1, Y_2, \ldots, Y_N\}$, it estimates the optimal model parameters $\Theta = \{\lambda_k\}_{k=1}^{K}$ by maximizing the following log-likelihood function:

$$\begin{aligned} L_\Theta &= \sum_{n=1}^{N} \log p(Y_n|X_n, \Theta) \\ &= \sum_{n=1}^{N} [\Theta^T F(X_n, Y_n) - \log Z_\Theta(X_n)] \end{aligned} \quad (3)$$

where $F(X_n, Y_n)$ are the accumulated feature values for one thread in the training set and $\log Z_\Theta(X_n) = \sum_Y \exp(\Theta^T F(X_n, Y))$. The Limited-memory Broyden–Fletecher–Goldfarb–Shanno (L-BFGS) algorithm is used to optimize the object function in (3). The gradient is derived by taking the derivative of the object function

$$\nabla L_\Theta = \sum_{n=1}^{N} [F(X_n, Y_n) - E_{p_\Theta(Y|X_n)} F(X_n, Y)] - \frac{\lambda}{\sigma^2} \quad (4)$$

where $E_{p_\Theta(Y|X_n)} F(X_n, Y)$ is the model expectation of the features' occurrences for the given training thread and (λ/σ^2) is the regularization term.

D. Training Set Generation With Partially Labeled Data

Note that the threadCRF is a supervised learning model, which requires a completely labeled data set for model training. In this section, we propose to generate a fully labeled training set given the partially labeled data.

We materialize all the possible thread reply structures given the partially labeled reply structures. Specifically, if a post has explicitly specified its parent, then we use this information directly. Otherwise, we consider each preceding post as a possible parent of the post and generate multiple possible training instances with each containing a possible thread reply structure. In this way, the obtained training data sets are fully labeled. We call this process of converting a partially labeled data set to a fully labeled one as *materialization*.

One possible approach is to consider all the materialized instances from the same partially labeled thread equally possible, denoted *M*aterialization with *E*qual *P*robabilities (MEP). Assume that the nth partially labeled training thread can be materialized into M_n completely labeled instances. With the materialized training instances, we have the following equation for the derivative of the threadCRF object function, which is modified from (4):

$$\nabla L_\Theta = \sum_{n=1}^{N} \left[\frac{\sum_{i=1}^{M_n} F(X_n, Y_{n_i})}{M_n} - E_{p_\Theta(Y|X_n)} F(X_n, Y) \right] - \frac{\lambda}{\sigma^2} \quad (5)$$

where X_n is the post sequence of the nth training thread and $Y_{n_i}, 1 \le i \le M_n$ is one possible parent labeling sequence for the nth thread. $((\sum_{i=1}^{M_n} F(X_n, Y_{n_i}))/M_n)$ is the accumulated empirical feature value for X_n.

This materialization approach considers all the possible reply structures in a thread equally important, which may not be accurate. Furthermore, a huge amount of materialized training instances will be generated, which will lead to a dramatical increase in the time and space complexity for the factor graph generation and marginal probability inference during the threadCRF model learning. For example, if there are T posts, denoted as $p_{u_1}, p_{u_2}, \ldots, p_{u_T}$, with unknown parents in a thread, where p_{u_i} has u_i candidate parents and $u_i \ge 2$, then the total number of materialized instances will be $u_1 \cdot u_2 \ldots \cdot u_T \ge T$, which is a huge number if T is relatively large. A more effective and efficient materialization process will be introduced later.

E. Constrained ThreadCRF for Partially Labeled Data

When applying the trained model to learn a complete reply structure for a given thread, threadCRF predicts the parents for all the posts despite the fact that some of them are already known. We propose to use the existing partially known structure as constraints. We denote this approach as **constrained threadCRF**. We not only want to preserve the existing reply structures in the final output of the complete reply structures but also want the existing structures to help infer the unknown structures by encoding them into the model. In order to do that, we add one parent feature into the original

threadCRF model, which is defined as follows:

$$\text{Parent}(y_i = j) = \begin{cases} 1, & \text{if } y_i = j \text{ is known} \\ -1, & \text{if } y_i \neq j \text{ is known} \\ \frac{1}{i}, & \text{if } y_i \text{ is unknown.} \end{cases} \quad (6)$$

Here, y_i is the parent post ID of the ith post, and j is a parent post ID, where $i \geq 1$, $j \geq 0$, and $i > j$. For the ith post, if its parent is unknown, all the i candidates are assigned with the same feature value $(1/i)$.

III. Leveraging Person Resolution for Thread Structure Learning

As discussed in Section I, PR can be useful for thread structure learning. In this section, we discuss how to leverage PR for thread structure learning on online health forums. We first introduce our PR system, and then, we proposed to use PR to generate thread structures. At last, we introduce how to combine PR with threadCRF for thread structure learning.

A. Person Resolution

PR is the process of identifying the same person mentioned in different contexts. Usually, some general coreference resolution, anaphora resolution, or pronoun resolution systems can be used for PR [11]–[14]. In terms of the scope, there are three types of PR in a forum—intrapost, interpost, and interthread PR. The intrapost PR confines the PR within a post. The interpost PR considers the PR between posts but within a thread, while the interthread PR considers the PR between threads.

We mainly use the interpost PR for thread structure learning, since the thread reply structures focus on the relationships between posts within a thread. The intuition behind that is as follows: if one post replies to another post, it tends to mention the same person who has been mentioned in its parent post. Conversely, if one post contains some person mentions that refer to the same person mentioned in a preceding post, then the post is likely to be a child or a descendant of that preceding post.

We design our own interpost PR system for thread structure learning since there are no publicly available systems for interpost PR. We observed that a forum thread could be considered as a multiperson dialog, where a post author is like a speaker, and the post content is analogous to the utterance, though a post can be very long. We designed several types of PR features. Each feature type has a different priority. We manually review some posts during the feature generation. We ranked the features based on our review These feature types are arranged in the descending order of priority, which serve as a multipass sieve with the first pass (type) having the highest priority. Specifically, our current PR system for thread structure learning includes the following four types of features in the order of descending priority.

1) PR Feature—Address and Signature Matching: Matching between the address in the current post content and the signature in the parent post content. Usually, the address appears at the beginning or follows some tokens such as "hi," "hello," and so on, and the signature appears at the end of a post following some tokens such as "thanks," "regards," and so on. We consider a person name recognized by a name entity recognition system in the Stanford Core NLP tools right after one of these tokens as an address or a signature. Furthermore, we also identify nicknames, acronyms, authorID, and so on, using patterns expressed by regular expressions. For example, acronyms typically consist of uppercase letters and end with "." and authorID is a combination of letters and numbers.

2) PR Feature—Role Matching: Matching between the same role related to the same person. First, our system identifies all the role mentions, such as "son," "daughter," "sister," and so on, using the family group semantic type in metamap [15]. Second, we combine the identified role with the first or second personal pronouns like "our," "my," and "your" for matching pairs, where the pronouns are identified by the Part-of-Speech (POS) module in the Stanford Core NLP tools. Finally, the word "my/our" followed by a role (such as daughter) in a preceding post can match "your" followed by the same role in the current post.

3) PR Feature—First-Person Pronoun Matching: Matching between the first-person pronouns, such as "I," "we," and "my," in the candidate parent post and the second-person pronouns, such as "you" and "your," in the current post that tend to refer to the same person. Semantic role labeling (SRL) [16] and WordNet [17] are used for checking if they tend to refer to the same person. First, we identify the first-person pronouns in the parent post and the second-person pronouns in the current post. Then, we use SRL for finding the associated verbs with the pronouns in a sentence. If the verb associated with the first-person pronoun is a synonym of the verb associated with the second-person pronoun according to WordNet, then we consider them as a matching. For example, "I" in "I got fever" can be matched to "you" in "The symptom you had. . .," as "get" and "have" are synonyms in WordNet.

4) PR Feature—Third-Person Pronoun Matching: Matching between the third-person pronoun in the current post and the person name or role in the parent post that are consistent in gender. Note that we also ensure that the third-person pronoun has not been resolved to a person's name or role appearing in the same post, and this person is a different one from that in the parent post. The Stanford coreference resolution system is used for checking the PR within a post. For example, consider the first sentence of a post to be "she should. . .," the pronoun "she" can refer to "my daughter" mentioned in a proceeding post. However, "she" in a post such as "My friend Mary . . . she . . ." is more likely to refer to "Mary," instead of referring to "my daughter" in a proceeding post.

Since the Stanford Core NLP tools do not output confidence level for PR, we set the priorities of the PR features based on data characteristics. Address and signature matching has the highest priority as it gives direct signals about the reply relationships based on PR. The other three PR features give less direct signals about reply relationships and are ranked based on expected accuracy. Role matching has the second-highest priority since PR is more likely to be accurate when role mentions are matched. The next priorities go to first-person

Algorithm 1: Multipass Candidate Thread Structure Selection

```
Input: A thread with a partially known structure.
Output: A complete thread structure.
for each post do
    if parent is unknown then
        Put all the preceding posts in the candidate set;
        for PR features (descending order of priority) do
            for each post in the candidate set do
                Mark the post if this type of features can be matched;
            end
            if Only one candidate is marked then
                Output that candidate as the labeled parent;
            else if More than one candidates are marked then
                Remove unmarked candidates;
        end
        if More than one candidate remains then
            Use the rule in Section III-B to find the winner
    else
        Output the predicted parent.
end
```

pronoun matching, and then, third-person pronoun matching, where accurate PR is more challenging.

Here, we only define pairwise PR features, which means each feature only involves a pair of posts. When applying PR for the unsupervised thread structure generation, we predict the unknown thread structure by evaluating each individual reply relationship independently. Note that we can also define the PR features involving multiple pairs of posts corresponding to multiple reply relationships, and learn the entire thread structure tree by considering all these PR features at the same time. In this way, we learn a globally optimal thread structure in terms of PR. However, such global optimization will be highly expensive in computation.

B. Leveraging Person Resolution for Thread Structure Labeling

Given the above-mentioned different types of PR features, we design the multipass candidate structure selection algorithm as shown in Algorithm 1, which takes a thread with a partially known structure as input and outputs a complete thread structure. We compare the priority of feature types matched by each candidate. In particular, each type of feature, in the descending order of priority, is used to filter out those less likely candidates. For each feature type, we divide the candidates into two subsets—matched and unmatched. We then remove the unmatched from the candidate set. We continue in this way until all the types (passes) of features have been checked.

We break a tie when there are multiple candidates left at the end using the following two rules. First, we observed that the forum users tend to reply to the thread initiator (the author of the first post), except for the initiator himself/herself. If a post that is not authored by the thread initiator has a set of candidate parents, one of which is from the thread initiator, then we output the post from thread initiator as the labeled parent. Second, we observed that forum users tend to reply to more recent post given other factors the same. If two candidate parents are both from the thread initiator or neither of them is from the thread initiator, then the more recent one will be output as the labeled parent.

C. Combining PR With ThreadCRF For Thread Structure Learning

In this section, we introduce how to combine PR with threadCRF, including how to materialize training instances with PR evaluation, and how to encode PR into threadCRF.

1) Materialization With PR Evaluation: As we have discussed earlier, materializing partially labeled data to fully labeled one with equal probabilities will result in an exponential increase in the data size, and thus, the computational time and space in model training process. Furthermore, not all preceding posts are equally likely to be the parent of one post. To improve the model accuracy and efficiency, we propose to materialize fully labeled instances considering the probability of each post being a candidate parent. The challenge is how to estimate the probability of a post being the parent of another post.

As person references give hints on the parent–child relationships between posts, we propose to use the PR techniques to evaluate the likelihood of all possible candidate parents and only materialize the most likely candidates. In this way, we expect the materialized thread structures are more accurate and help learn a more accurate threadCRF model. When there is no clear PR indication for some posts, we use three unsupervised rule-based baseline approaches—reply to the first post, reply to the last post, and reply to the post with the highest content similarity, referred *FIRST*, *LAST*, and *SIM*, respectively. In our implementation, we use term frequency–inverse document frequency (TF-IDF) weighted cosine similarity to measure content similarity, the same as the ones used in threadCRF [1], in order to make fair performance comparisons. Here, IDF is calculated based on the number of posts containing the word in the given thread. Other content similarity methods, including combining multiple methods, can also be used in the system.

The whole materialization process is shown in Algorithm 2.

2) Encoding PR Into ThreadCRF: The PR model can not only help generate fully labeled training data set to train the threadCRF model, but it can also be incorporated as part of the threadCRF model. Recall that the threadCRF model includes a set of node and edge features. We encode PR as a node feature. Specifically, we define the PR feature value $\mathrm{PR}(y_i = j)$ between post i and its candidate parent j, where $i \geq 1$, $j \geq 0$, and $i > j$. We assign a weight for each PR feature type based on its priority. Assume there are L types arranged in the descending order of priority. For the kth type, $1 \leq k \leq L$, its weight is assigned as $\exp(L-k)$. Suppose we are evaluating the PR feature value for the ith post to reply to the jth post, represented as $\mathrm{PR}(y_i = j)$, we have

$$\mathrm{PR}(y_i = j) = \sum_{k=1}^{L} \exp(L-k) \cdot \delta_{ijk} \tag{7}$$

Algorithm 2: Materialization With PR Evaluation

Input: One thread with a partially known reply structure.
Output: Multiple thread instances each with a complete reply structure.

```
for each post (starting from the third post) do
    if its parent post is unknown then
        List all the preceding posts as the candidate parents;
        Use the pseudo codes from Line 4 to Line 12 in Algorithm 1 to shrink the candidate set;
        if the candidate set includes the first, last, and the post with the highest content similarity then
            Further shrink the candidate set by only retaining the first, last, and the most similar post as the candidates;
    else
        List the known parent as the only candidate.
end
Generate all the possible thread structures by picking one candidate for each post (starting from the third post).
```

where

$$\begin{cases} 1, & \text{if the } k\text{th type is matched for post } i \\ & \text{and its parent } j \\ 0, & \text{Otherwise.} \end{cases} \tag{8}$$

IV. Performance Evaluation

In this section, we present the data sets, comparison methods, evaluation metrics, results, and analysis of the experimental evaluation of the proposed methods.

A. Data Sets and Training Data

As discussed earlier, our goal is to leverage partially labeled thread structures to learn complete thread structures. This is important as most of the forums do not have completely labeled thread structure. On the other hand, this also poses challenges since the training set is not fully labeled, making the model training difficult. To address this challenge, we use a small part of partially labeled data to train the model. Instead of randomly selecting threads as a training set, we select threads with large percent of reply relationships labeled as the training set in order to boost model training. Then, the model predicts unknown reply relationship in the whole data set. This approach is applicable to any forum data.

We evaluated the proposed methods with two different health forum data sets.

1) Patients Forum Data Set: Although most of the health forums only have partially labeled structures, we managed to find one forum, the patients' forum on tumors of the parotid gland (http://patientsforum.com), that has fully labeled thread structures with each represented in a hierarchical tree view, representing a good test data set. We collected all 23 842 posts in 2646 threads. In our experiments, we randomly removed a known reply relationship with probability 0.3 and got 5561 unknown reply relationships. We selected 1105 threads with more labeled relationships to compose the training set and used the proposed methods to predict the removed reply relationships in all 2646 threads. We use the original data (without reply relationship removal) as the ground truth for performance evaluation.

2) Epilepsy Forum Data Set: We collected 9210 posts in 911 threads (topics) published on the "patient help patient" subforum in the previous mentioned epilepsy foundation discussion forum (http://epilepsyfoundation.ning.com/forum). In this forum, some posts explicitly reply to a preceding post by quoting that post; while others have unknown reply relationships. We selected a subset of 200 threads that have more known reply relationships to train the threadCRF model. As it is very expensive to manually label ground truth, we only obtained all the 468 unknown reply relationships in the selected 200 threads for evaluation.

B. Comparison Methods

We tested our proposed methods and compared them with existing methods. All the tested methods are divided into three categories—rule-based, *CRF* [1], and *CRF + PR*. Table I explains all those methods. For comparison methods, *FIRST*, *LAST*, and *SIM* have been used as comparison methods by [1]. For *SIM*, we used the standard TF-IDF weighted cosine similarity, where IDF is calculated based on the number of posts containing the word in the given thread. *MEP* is the direct adaption of threadCRF for our proposed application scenario.

C. Evaluation Metrics

We followed the same evaluation method as the one used by [1] as presented below. The first category is about the accuracy of individual parent labels or paths from the node to the root in the thread structure tree. The **accuracy of individual labels**, denoted as Acc_{edge}, is defined as the proportion of correct labels in the whole set of predicted labels. Let U denote the set of posts with unknown parent labels, $\overline{y_i}$ denote the ground-truth label for $p_i \in U$, and $\hat{y}_i$ denote the predicted label for p_i. We define

$$\text{Acc}_{\text{edge}} = \frac{\sum_{p_i \in U} \delta[\overline{y}(i) = \hat{y}(i)]}{|U|} \tag{9}$$

where $|U|$ is the size of set U. $\delta[\overline{y}(i) = \hat{y}(i)] = 1$ if the two labels are the same. Otherwise, it is zero.

We also defined the **path accuracy**, denoted by Acc_{path}, as the proportion of correct paths from each node to the root in the thread structure tree

$$\text{Acc}_{\text{path}} = \frac{\sum_{p_i \in U} \delta[\overline{\text{path}(i)} = \widehat{\text{path}(i)}]}{|U|} \tag{10}$$

where $\overline{\text{path}(i)}$ and $\widehat{\text{path}(i)}$ are the set of nodes (posts) in the path from the node i (post p_i) to the root node in the ground-truth path and the predicted path, respectively. $\delta[\overline{\text{path}(i)} = \widehat{\text{path}(i)}] = 1$ if the two paths are identical. Otherwise, it is zero. Note that the path-based metrics emphasize that correct prediction of the labels for those nodes with more descendants is more important.

TABLE I
COMPARISON METHODS (OUR PROPOSED METHODS ARE MARKED IN A BOLD FONT)

Rule-based	FIRST	Reply to the first post.
	LAST	Reply to the last post.
	SIM	Reply to the post with the highest content similarity.
	PR	Reply to the post selected according to Algorithm 1.
CRF	MEP	Materialize all the possible training instances with equal probabilities, and then train and test threadCRF with the known structures as constraints.
CRF + PR	**MPR**	Materialize the training set using Algorithm 2, and train and test threadCRF with the known structures as constraints.
	EPR	Materialize the training set using Algorithm 2, and train and test threadCRF with the known structures as constraints plus the PR feature.

TABLE II
PERFORMANCE COMPARISON ON THE PATIENTS FORUM DATA SET

		Rule-based				**CRF**	**CRF + PR**	
		FIRST	LAST	SIM	PR	MEP	MPR	EPR
Acc_{edge}	*thread*	0.427	0.362	0.376	0.651	0.638	0.643	**0.693**
	corpus	0.360	0.363	0.336	0.606	0.583	0.591	**0.659**
Acc_{path}	*thread*	0.427	0.119	0.276	0.504	0.606	0.610	**0.657**
	corpus	0.360	0.086	0.217	0.404	0.525	0.533	**0.593**
P_{path}	*thread*	**1.000**	0.119	0.535	0.674	0.870	0.886	0.887
	corpus	**1.000**	0.086	0.461	0.565	0.819	0.838	0.846
R_{path}	*thread*	0.427	**1.000**	0.520	0.691	0.659	0.649	0.708
	corpus	0.360	**1.000**	0.457	0.627	0.584	0.579	0.650
$F1_{path}$	*thread*	0.599	0.212	0.528	0.682	0.750	0.750	**0.787**
	corpus	0.529	0.158	0.459	0.594	0.682	0.685	**0.735**

In the second category, we defined the **path-based precision and recall**, which are a relaxation of the accurate path matching, as in (10). The precision is the proportion of the predicted paths that are part of the ground-truth paths in all the predicted paths. The recall is the proportion of the ground-truth paths that are part of the predicted paths in all the ground-truth paths. They are mathematically defined as follows:

$$P_{\text{path}} = \frac{\sum_{p_i \in U} \delta[\widehat{\text{path}(i)} \subseteq \overline{\text{path}(i)}]}{|U|} \tag{11}$$

$$R_{\text{path}} = \frac{\sum_{p_i \in U} \delta[\overline{\text{path}(i)} \subseteq \widehat{\text{path}(i)}]}{|U|} \tag{12}$$

where $\delta[\widehat{\text{path}(i)} \subseteq \overline{\text{path}(i)}] = 1$ if $\widehat{\text{path}(i)}$ is a subset of $\overline{\text{path}(i)}$. Otherwise, it is zero. We also define $F1_{\text{path}}$ as the harmonic mean of P_{path} and R_{path}.

For each defined metric, there are two levels of evaluation—thread level and corpus level. In the thread level, these metrics are first measured for each thread, and then, they are averaged through all the threads in the test set. It emphasizes the thread structure learning performance for each thread. In the corpus level, these metrics are directly evaluated for the whole test set without the thread-level evaluation and aggregation process.

D. Results and Analysis

In this section, we show and analyze the experimental results. Tables II and III show the thread structure learning performance on the two data sets. We underline the numbers in each row if they are highest among the rule-based methods or highest among the CRF-based methods. The numbers in bold font represent the best performance among all the methods. Figs. 3 and 4 show the impact of training set size.

1) Comparison Among Rule-Based Methods: Tables II and III show that, among the rule-based methods, *PR* achieved the best performance for most of the evaluation metrics. In Table II, *PR* achieved the best performance for Acc_{edge}, Acc_{path}, and $F1_{\text{path}}$. In Table III, *PR* achieved the best performance for Acc_{edge} and $F1_{\text{path}}$. Note that *FIRST* has perfect P_{path} since all the ground-truth paths have to contain the first post and itself, which are the only two posts in the predicted paths. In other words, all the predicted paths are part of the ground-truth paths, which leads to a perfect P_{path}. The similar reason explains *LAST*'s R_{path} performance. However, their $F1$ performances are worse than the *PR* method.

2) Performance of CRF-Based Methods: For the *CRF*-based methods, we evaluated the performance of *MEP*, *MPR*, and *EPR* (see Table I for description of methods). First, we found that in many cases, *MEP* is not as good as *PR*. In Table II, *MEP* outperformed *PR* in Acc_{path}, P_{path}, and $F1_{\text{path}}$, but achieved a slightly worse performance in Acc_{edge} and R_{path}. In Table III, *MEP* only outperformed *PR* in R_{path}. Now, we analyze why *MEP* has inferior performance. *MEP* is unaware of person reference relationships in thread structures and assumes that all possible reply structures are equally likely,

TABLE III
PERFORMANCE COMPARISON ON THE EPILEPSY FORUM DATA SET

		Rule-based				CRF	CRF + PR	
		FIRST	LAST	SIM	PR	MEP	MPR	EPR
Acc_{edge}	*thread*	0.528	0.383	0.405	0.647	0.478	0.668	**0.688**
	corpus	0.444	0.429	0.361	0.583	0.48	0.635	**0.650**
Acc_{path}	*thread*	0.528	0.233	0.345	0.508	0.385	0.608	**0.646**
	corpus	0.444	0.203	0.274	0.391	0.335	0.526	**0.568**
P_{path}	*thread*	**1.000**	0.233	0.675	0.833	0.488	0.791	0.832
	corpus	**1.000**	0.203	0.647	0.771	0.466	0.750	0.810
R_{path}	*thread*	0.528	**1.000**	0.608	0.623	0.812	0.779	0.776
	corpus	0.444	**1.000**	0.521	0.517	0.750	0.720	0.705
$F1_{path}$	*thread*	0.691	0.377	0.639	0.713	0.609	0.785	**0.803**
	corpus	0.615	0.337	0.578	0.619	0.575	0.735	**0.754**

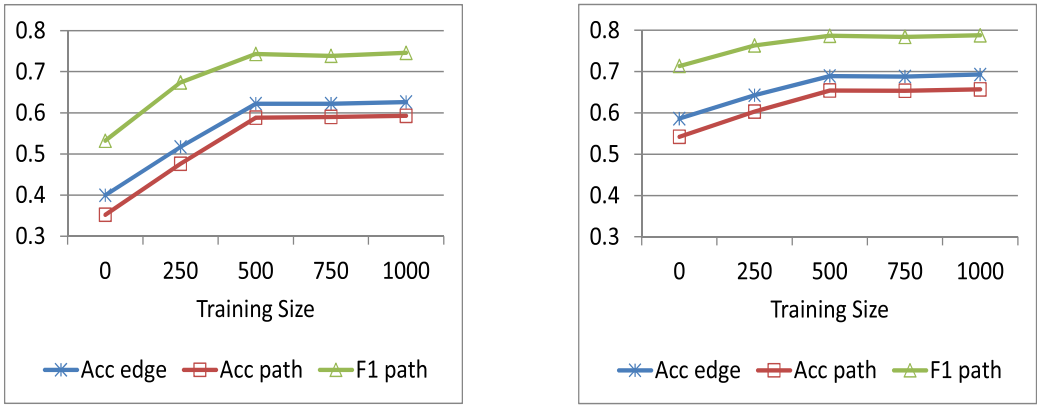


Fig. 3. Performance on different training sizes (patients forum data set). (a) MPR. (b) EPR.

which introduces lots of incorrect labels into the training data. The materialized fully labeled training set is thus not a good approximation of the ground truth, which leads to a less accurate trained model, and consequently lower prediction quality.

When we combined PR with threadCRF, we can see that the performance was significantly improved. In terms of Acc_{edge}, Acc_{path}, and $F1_{\text{path}}$, *MPR* outperformed *MEP*, and *EPR* outperformed *MPR* for both data sets. For *MPR*, as we have used PR to select the more likely candidate parent posts during the training set generation, the training set is more similar to the ground truth, which helps to learn a more accurate model. That explains why *MPR* outperformed *MEP*. Comparing *MPR* with *EPR*, we can clearly see that the threadCRF model with the additional *PR* feature has a better prediction performance. *EPR* consistently outperformed all the other methods in Acc_{edge}, Acc_{path}, and $F1_{\text{path}}$.

3) Impact of Training Set Size: We also analyzed the impact of the training set size on the prediction performance of two best methods—*MPR* and *EPR*. We randomly selected a set of threads as the training set and tested on all the threads in the data set. When the training set size is zero, it means that there is no training process, and the feature weights are all set to 1.0. Figs. 3 and 4 show the thread-level performance. Note that the trend of the corpus-level performance is similar and not shown here. It shows that the increase in training size can improve performance at the beginning. In Fig. 3, the performance of both *MPR* and *EPR* has stopped improving after the training size is larger than 500. In fact, [1] also observed that with a small training set, threadCRF can achieve an encouraging performance compared with a larger training set. Such findings show that, in order to bootstrap our training process and predict all the unknown thread structures, we only need a small set of threads that have a majority of labeled reply relationships. In Fig. 4, the performance for *MPR* does not continue improving with the training size increasing from 150 to 200, while the performance for *EPR* keeps increasing. It suggests that with more features, more training instances are needed.

In summary, the proposed CRF-based approach *EPR* achieved the best performance in Acc_{edge}, Acc_{path}, and $F1_{\text{path}}$ for both data sets. A relatively small training set can achieve an encouraging performance.

V. RELATED WORK

Recently, some research has been performed on learning or predicting thread structures for online forums, blogs, or news websites. Reference [18] incorporated topic modeling

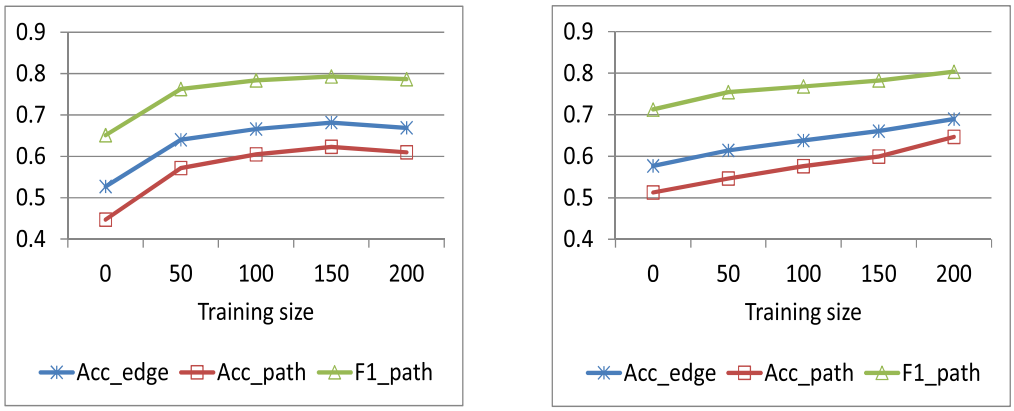


Fig. 4. Performance on different training sizes (epilepsy forum data set). (a) MPR. (b) EPR.

and temporal dependence between posts in a sparse coding approach. In particular, one post is represented as a linear combination of all the preceding posts in the latent semantic space. The structure information is embedded by adding the constraints that the topics of each post can only be sampled from the topics of those preceding posts. The sparse coding approach can be used for reply relationship reconstruction, junk post detection, and expert finding. Reference [19] used an extended block hidden Markov model, which combines the functionality of topic models with Markovian sequence models for unsupervised conversation structure modeling. While sequence dependence modeling has been captured by the CRF-based model, topic information can also be encoded as a feature into our model to learn from the partially labeled data under the supervised learning framework. Reference [10] proposed an unsupervised approach to predicting the thread reply structures, which utilizes the lexical chains between word tokens within a discourse to recover the interpost links. Their work is also orthogonal to ours, which can be combined with our PR techniques and integrated into our learning framework. Reference [9] used a classification approach to reconstructing the reply structures based on a set of simple features, such as time difference, content similarity, and quotation relationship. Reference [10] used a joint classification approach with a linear-chain CRF or dependence parsing for predicting thread structures, which considers both the link relationships between posts and the dialog acts assigned to each link. The dialog acts are made from the five categories—question, answer, resolution, production, and others. Reference [20] proposed a supervised approach based on the ranking-SVM model to reconstruct the thread structures in blogs, online news agencies, and news websites, which are slightly different from online forums. Compared with those supervised approaches, threadCRF [1] is more effective by introducing edge features to capture the long-distance dependency. In this article, we extend the threadCRF model by considering the partially known structures and the abundant person reference information available in person-centric forums. Our contributions, learning from the partially labeled data and leveraging person reference relationships, have not been exploited by prior work. They are orthogonal to the contributions of some related work and can be combined with them to boost performance.

To address the partially labeled data problem in a text mining application, a semisupervised training procedure for CRFs has been proposed by [21], which can be used with a combination of labeled and unlabeled training data. However, instead of having some instances fully labeled and some unlabeled, each training instance in our setting, which is a thread, is partially labeled. Therefore, the above-mentioned semisupervised training procedure is not applicable to our problem. Reference [22] proposed a training procedure with incomplete annotated sentence instances for the Japanese word segmentation and POS tagging tasks. Inspired by their work, we materialize thread structures to train the threadCRF model for our thread structure learning task.

In addition, there are research works related to PR on forums. First, some general coreference resolution or anaphora resolution systems can be used for PR [11]–[13]. However, although these general resolution systems are good at resolving mentions within a post, they are not suitable for coreference resolution across posts. The only coreference resolution system related to forums is introduced by [23], which focuses on the coreference resolution on blogs and commented news in Dutch. Blogs and commented news in Dutch are different from health forums in English, and their system is also not publicly available. Some coreference resolution or pronoun resolution systems for dialogs, such as those proposed by [14] and [24], focus more on short-text dialog in spoken language, while our article addresses PR in forums.

Our preliminary findings have been presented in a short workshop article [25]. In this article, we systematically investigate the theories and algorithms in this problem. Two additional methods, MEP and EPR, are introduced and evaluated in this article. We also used more metrics to evaluate the proposed methods, together with a new larger data set, the patients' forum data set, which has the original labels as the ground truth.

VI. Conclusion and Future Work

In this article, we proposed to learn the complete thread structures on online health forms with partially labeled data.

We first leverage the partially labeled structures that are prevalent in Web forums to learn the complete thread structures based on a statistical machine learning model: threadCRF. We then exploit the abundant person reference information in person-centric forums, together with threadCRF, for thread structure learning. Experimental evaluation demonstrates the effectiveness of our proposed methods. In the future, we will explore if other CRF models can be adapted to address this problem. We also plan to generalize our approaches for other types of forums. For example, we may use entity resolution instead of PR to leverage the interactions of entities mentioned in the posts for thread structure learning for all forums.

REFERENCES

[1] H. Wang, C. Wang, C. Zhai, and J. Han, "Learning online discussion structures by conditional random fields," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2011, pp. 435–444.

[2] Y. Liu and Y. Chen, "Patient-centered information extraction for effective search on healthcare forum," in *Proc. Int. Conf. Social Comput., Behav.-Cultural Modeling, Predict.* Berlin, Germany: Springer, 2013, pp. 175–183.

[3] H. Duan and C. Zhai, "Exploiting thread structures to improve smoothing of language models for forum post retrieval," in *Proc. Eur. Conf. Inf. Retr.* Berlin, Germany: Springer, 2011, pp. 350–361.

[4] L. Wang, S. N. Kim, and T. Baldwin, "The utility of discourse structure in forum thread retrieval," in *Proc. Asia Inf. Retr. Symp.* Berlin, Germany: Springer, 2013, pp. 284–295.

[5] K. Pattabiraman, P. Sondhi, and C. Zhai, "Exploiting forum thread structures to improve thread clustering," in *Proc. Conf. Theory Inf. Retr.*, 2013, p. 15.

[6] J. Seo, W. B. Croft, and D. A. Smith, "Online community search using thread structure," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 1907–1910.

[7] A. P. Louis and S. B. Cohen, "Conversation trees: A grammar model for topic structure in forums," in *Proc. Assoc. Comput. Linguistics*, 2015, pp. 1543–1553.

[8] J. Zhang, M. S. Ackerman, and L. Adamic, "Expertise networks in online communities: Structure and algorithms," in *Proc. 16th Int. Conf. World Wide Web*, 2007, pp. 221–230.

[9] E. Aumayr, J. Chan, and C. Hayes, "Reconstruction of threaded conversations in online discussion forums," in *Proc. ICWSM*, vol. 11, 2011, pp. 26–33.

[10] L. Wang, M. Lui, S. N. Kim, J. Nivre, and T. Baldwin, "Predicting thread discourse structure over technical Web forums," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 13–25.

[11] H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky, "Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task," in *Proc. 15th Conf. Comput. Natural Lang. Learn., Shared Task*, 2011, pp. 28–34.

[12] H. Lee, A. Chang, Y. Peirsman, N. Chambers, M. Surdeanu, and D. Jurafsky, "Deterministic coreference resolution based on entity-centric, precision-ranked rules," *Comput. Linguistics*, vol. 39, no. 4, pp. 885–916, 2013.

[13] J. Steinberger, M. Poesio, M. A. Kabadjov, and K. Ježek, "Two uses of anaphora resolution in summarization," *Inf. Process. Manage.*, vol. 43, no. 6, pp. 1663–1680, 2007.

[14] A. J. Stent and S. Bangalore, "Interaction between dialog structure and coreference resolution," in *Proc. IEEE Spoken Lang. Technol. Workshop*, Dec. 2010, pp. 342–347.

[15] A. R. Aronson, "MetaMap: Mapping text to the UMLS metathesaurus," NLM, NIH, DHHS, Bethesda, MD, USA, Tech. Rep., 2006, pp. 1–26.

[16] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Aug. 2011.

[17] G. Miller, *WordNet: An Electronic Lexical Database*. Cambridge, MA, USA: MIT Press, 1998.

[18] C. Lin, J.-M. Yang, R. Cai, X.-J. Wang, and W. Wang, "Simultaneously modeling semantics and structure of threaded discussions: A sparse coding approach and its applications," in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2009, pp. 131–138.

[19] M. J. Paul, "Mixed membership Markov models for unsupervised conversation modeling," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, 2012, pp. 94–104.

[20] A. Balali, H. Faili, and M. Asadpour, "A supervised approach to predict the hierarchical structure of conversation threads for comments," *Sci. World J.*, vol. 2014, Feb. 2014, Art. no. 479746.

[21] F. Jiao, S. Wang, C.-H. Lee, R. Greiner, and D. Schuurmans, "Semi-supervised conditional random fields for improved sequence segmentation and labeling," in *Proc. 21st Int. Conf. Comput. Linguistics 44th Annu. Meeting Assoc. Comput. Linguistics*, 2006, pp. 209–216.

[22] Y. Tsuboi, H. Kashima, H. Oda, S. Mori, and Y. Matsumoto, "Training conditional random fields using incomplete annotations," in *Proc. 22nd Int. Conf. Comput. Linguistics*, vol. 1, 2008, pp. 897–904.

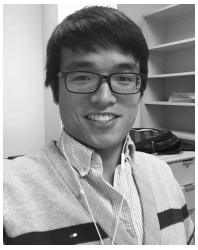
[23] I. Hendrickx and V. Hoste, "Coreference resolution on Blogs and commented news," in *Discourse Anaphora and Anaphor Resolution Colloquium*. Berlin, Germany: Springer, 2009, pp. 43–53.

[24] X. Luo, R. Florian, and T. Ward, "Improving coreference resolution by using conversational metadata," in *Proc. Hum. Lang. Technol., Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Companion Volume: Short Papers*, 2009, pp. 201–204.

[25] Y. Liu, F. Chen, and Y. Chen, "Learning thread reply structure on patient forums," in *Proc. Int. Workshop Data Manage. Anal. Healthcare*, 2013, pp. 1–4.

Yunzhong Liu received the Ph.D. degree in computer science from Arizona State University, Tempe, AZ, USA.

His current research interests include data mining and machine learning.



Jinhe Shi received the bachelor's degree from Shanghai University, Shanghai, China, in 2012, and the master's degree from Rutgers University, Newark, NJ, USA, in 2015. He is currently pursuing the Ph.D. degree in computer science with the Ying Wu College of Computing, New Jersey Institute of Technology, Newark.

His current research interests include developing machine learning and deep learning techniques for healthcare applications

Yi Chen received the B.S. degree from Central South University, Changsha, China, in 1999, and the Ph.D. degree in computer science from the University of Pennsylvania, Philadelphia, PA, USA, in 2005.

She is currently a Full Professor and the Henry J. Leir Chair with the Martin Tuchman School of Management with a joint appointment with the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA, where she also serves as the Co-Director of the Big Data Center. Her current research interests include many aspects of data management and data mining.

Dr. Chen has served in the organization and program committees for various conferences, including The ACM Special Interest Group on Management of Data, the International Conference on Very Large Data Bases, the IEEE International Conference on Data Engineering, and the International World Wide Web Conference. She served as the General Chair for SIGMOD'2012. She served as an Associate Editor for the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, *Distributed and Parallel Databases*, the *International Journal of Communication*, and the *Proceedings of the VLDB Endowment*.