This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

iHelp: An Intelligent Online Helpdesk System

Dingding Wang, Tao Li, Shenghuo Zhu, and Yihong Gong

***Abstract*—Due to the importance of high-quality customer service, many companies use intelligent helpdesk systems (e.g., case-based systems) to improve customer service quality. However, these systems face two challenges: 1) Case retrieval measures: most case-based systems use traditional keyword-matching-based ranking schemes for case retrieval and have difficulty to capture the semantic meanings of cases and 2) result representation: most case-based systems return a list of past cases ranked by their relevance to a new request, and customers have to go through the list and examine the cases one by one to identify their desired cases. To address these challenges, we develop iHelp, an intelligent online helpdesk system, to automatically find problem–solution patterns from the past customer–representative interactions. When a new customer request arrives, iHelp searches and ranks the past cases based on their semantic relevance to the request, groups the relevant cases into different clusters using a mixture language model and symmetric matrix factorization, and summarizes each case cluster to generate recommended solutions. Case and user studies have been conducted to show the full functionality and the effectiveness of iHelp.**

***Index Terms*—Case clustering, case summarization, intelligent helpdesk, semantic similarity.**

I. Introduction

HIGH-QUALITY customer service is extremely important for companies. It is reported that 70% of the customers hit the road not because of the price or product quality issues but because they do not like the customer service [1]. Current customer service (also called helpdesk, call center, etc.) involves a lot of manual operations, which require customer service representatives to master a large variety of malfunction issues. Moreover, it is difficult to transfer knowledge and experience between representatives. Thus, many companies attempt to build intelligent helpdesk systems to improve the quality of customer service.

Given a new customer request, one common scenario of an intelligent helpdesk system is to find whether similar requests have been processed before. Helpdesk systems usually use databases to store past interactions (e.g., descriptions of a problem and recommended solutions) between customers and companies. In this paper, we assume that the helpdesk databases are organized into a number of cases, where each case contains the interactions between a customer and the service team about a particular problem (e.g., initial information provided by the customer, clarification questions asked by the service team and answers from the customer, etc.) and the final recommendations by the service team.

There has been a rich body of work on case-based recommender systems [2]–[4] and decision guides [5], where the user provides a brief initial description of problems. The systems use the initial information to retrieve the candidate set of cases that are similar to the given problems. Recently, some work has been focused on incremental/conversational case-based reasoning [6], [7], where the system interactively asks the user questions to narrow down the problems gradually until few cases remain. However, these case-based systems face the following two challenges.

1) **Case retrieval measures:** Given a new request from a customer, most case-based systems search and rank the documents of past cases based on their relevance to the request. Many methods have been proposed to determine the relevance of past cases to requests in database [8]–[12], and to perform similarity search [13]–[15]. There has also been work on how to efficiently return the top k results rather than all results [16]–[18] and efficiently return results of skyline queries (results that are no worse than others on all dimensions). However, these methods usually use traditional keyword-matching-based ranking schemes, which have difficulty in capturing the semantic meanings of the requests and the past cases. For example, given a request "can you switch the computers?" most case-based systems would return past cases related to network switches. In addition, when the description of the cases or items becomes complicated, these case-based systems also suffer from the curse of dimensionality, and the similarity/distance between cases or items becomes difficult to measure [19]. New similarity measurements that are able to understand the semantic meanings in the requests and the past cases are thus needed.
2) **Result representation:** Most case-based systems return a list of past cases ranked by their relevance to a new request. Customers have to go through the list and examine the cases one by one to identify their desired cases. This is a time-consuming task if the list is long. Considering the aforementioned request "can you switch the computers?" the case-based systems might return tens of past cases that are related to the request, even with novel similarity measurements. A possible solution is to organize the past cases into different groups, each of which corresponds to a specific context or scenario. This would enable the customers to identify their desired contexts at a glance.

Manuscript received August 12, 2009; revised January 17, 2010 and April 3, 2010; accepted April 11, 2010 . This work was supported in part by a Florida International University Dissertation Year Fellowship and in part by National Science Foundation Grants IIS-0546280, HRD-0833093, and DMS-0915110. This paper was recommended by Associate Editor H. Wang.

D. Wang and T. Li are with the School of Computing and Information Sciences, Florida International University, Miami, FL 33199 USA (e-mail: dwang003@cs.fiu.edu; taoli@cs.fiu.edu).

S. Zhu and Y. Gong are with NEC Laboratories America, Inc., Cupertino, CA 95014 USA (e-mail: zsh@sv.nec-labs.com; ygong@sv.nec-labs.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2010.2049352

For example, a group of past cases might be related to the procedures of changing the locations of the computers, while another group is related to exchanging user accounts of the computers. It is also necessary to generate a short and concise summary for each context to improve the usability.

To address the aforementioned two challenges, we develop iHelp, an intelligent online helpdesk system, to automatically find problem–solution patterns from the past interactions between customers and representatives. First, we develop a new case-ranking method using sentence-level semantic analysis to better capture the semantic meanings of the cases. Given a new request from a customer, iHelp searches and ranks the past cases based on their relevance to the request using the new similarity measurement. Then, to improve the usability, we develop a case-clustering algorithm using a mixture language model and symmetric nonnegative matrix factorization (SNMF) to group the top-ranking cases into different categories while reducing the impact of the general and common information contained in these cases. Finally, iHelp conducts a request-based case summarization to generate a concise summary as a reference solution for each cluster of the relevant cases. In summary, there are three key features of iHelp, which are listed in the following.

1) It employs sentence-level semantic analysis to better understand the semantic meanings of the cases.
2) It utilizes a novel clustering algorithm based on a mixture language model and SNMF to capture different scenarios in the top-ranking past cases that are related to the given problems.
3) It generates a concise description for each scenario to improve the system usability.

The rest of this paper is organized as follows. Related work is discussed in Section II. Section III briefly introduces the framework and shows a sample screenshot of iHelp. Section IV presents the procedure for preprocessing the cases and performing semantic role analysis. Section V describes our proposed case-ranking approach. The case-clustering algorithm is proposed in Section VI. Section VII presents the case summarization method. Case and user studies are conducted to evaluate the effectiveness of the iHelp system in Sections VIII-B and C, respectively. Finally, Section IX concludes this paper. It should be pointed out that the techniques used in our system for case preprocessing, particularly semantic role parsing (Section IV-A), sentence-level semantic similarity calculation (Section V-A), and within-cluster sentence selection (Section VII-A), have been reported in our prior conference publication [20]. Our prior publication focuses on multidocument summarization, while this paper describes an online helpdesk system to automatically find problem–solution patterns from the past customer–representative interactions, where summarization is just one of the components for the system.

Thus, the main contributions of this paper are the following: 1) We search and rank the existing cases according to their relevance to users' requests in a semantic way, and 2) we provide a better result representation by grouping and summarizing the retrieved past cases to make the system fully functional and usable.

II. Related Work

In this paper, we develop an online helpdesk system that can automatically search and rank the existing cases, and group the top-ranking cases into clusters and summarize each case cluster to generate recommended solutions. Some related topics and techniques are reviewed in this section.

1) ***Case-based systems:*** Some case-based systems have been developed to interactively search the solution space by suggesting the most informative questions to ask [2]–[7]. These systems use the initial information to retrieve the first candidate set and then ask the user questions to narrow down until few cases remain or the most suitable items are found. When the description of cases or items becomes complicated, these case-based systems suffer from the curse of dimensionality, and the similarity/distance between cases or items becomes difficult to measure [19]. Furthermore, the similarity measurements used in these systems usually are based on keyword matching, which lacks the semantic analysis of customer requests and existing cases.
2) ***Database search and ranking:*** In database search, many methods have been proposed to perform similarity search and rank results of a query [8]–[15]. However, similar to the case-based systems, the similarity is measured based on keyword matching, which have difficulty to understand the text deeply.
3) ***Clustering search results:*** Since existing search engines often return a long list of search results, clustering technologies are often used in search result organization [21]–[24]. However, the existing document-clustering algorithms do not consider the impact of the general and common information contained in the documents. In our work, by filtering out this common information, the clustering quality can be improved, and better context organizations can then be obtained.
4) ***Document summarization:*** Multidocument summarization is the process of generating a summary by reducing documents in size while retaining the main characteristics of the original documents [20], [25], [26]. We utilize the idea of a request-focused multidocument summarization and propose a new summarization method to summarize each cluster of the past cases and generate reference solutions, which can better assist customers to find their desired solutions. We note that some techniques used in our system for case preprocessing, sentence-level semantic similarity calculation, and within-cluster sentence selection have been reported in our prior conference publication [20]. However, this paper develops an online helpdesk system to automatically find problem–solution patterns from the past customer–representative interactions. The online helpdesk system searches and ranks the past cases based on their semantic relevance to the new request, groups the relevant cases into different clusters using a mixture language model, and summarizes each

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

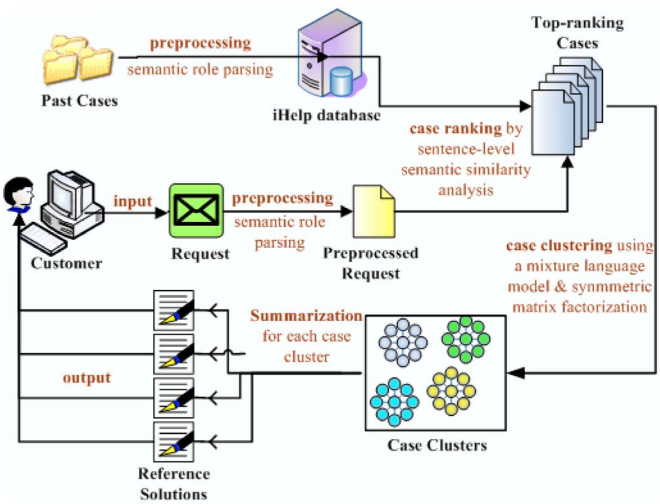

Fig. 1. Framework of the iHelp system.

case cluster to generate recommended solutions. Clearly, summarization is only one of the components in our online helpdesk system, which includes case ranking, top-ranked case clustering, and concise summary generation.

III. Framework

Fig. 1 shows the framework of iHelp. The input of the system is a request by a customer and a number of past cases. First of all, the past cases are cleaned by removing formatting characters and stopping words; then, each of the cases is trunked into sentences and passed through a semantic role parser in the preprocessing step. Then, in the case-ranking module, the past cases are ranked based on their semantic importance to the preprocessed input request. The details of the proposed ranking method are discussed in Section V. Other than searching and ranking the relevant cases, iHelp also groups the top-ranking cases into clusters using a mixture model and SNMF. Finally, a brief summary for each case cluster is generated as a reference solution to the customer. Full explanations of the proposed case-clustering and summarization approaches are presented in Sections VI and VII, respectively. Fig. 2 shows the screenshot of an example output of the iHelp system.

IV. Case Preprocessing

The past cases used in the iHelp system are over 30 000 records collected from the technical support group at a university. Each case records the interactions between the users (students and faculty members) and the technical staff and is stored as a document. First of all, we preprocess the cases by removing formatting characters. Then, in order to further analyze these documents, we trunk the cases into sentences and perform semantic role analysis at the sentence level so that the semantic meanings of the cases can be better captured.

A. Semantic Role Parsing

A semantic role is "a description of the relationship that a constituent plays with respect to the verb in the sentence" [27]. Semantic role analysis plays a very important role in semantic understanding. In iHelp, we use NEC SENNA [28] as the semantic role labeler, which is based on PropBank semantic annotation [29]. The basic idea is that each verb in a sentence is labeled with its propositional arguments, and the labeling for each particular verb is called a "frame." Therefore, for each sentence, the number of frames generated by the parser equals the number of verbs in the sentence. There is a set of abstract arguments indicating the semantic role of each term in a frame. For example, Arg0 is typically the actor, and Arg1 is the thing acted upon. The full representation of the abstract arguments [29] and an illustrative example are shown in Table I.

V. Request-Based Semantic Case Ranking

To assist users in finding answers quickly once a new request arrives, the existing cases are required to be ranked based on their semantic importance to the input request. In order to rank the cases, the similarity scores between the cases and the input request are computed. Simple word-matching similarity measurement, such as the cosine similarity, cannot faithfully capture the content similarity. Also, the sparseness of words between similar concepts makes the similarity metric uneven. Thus, we propose a method to calculate the semantic similarity between the sentences in the past cases and the request based on the semantic role analysis, as introduced in Section IV-A.

A. Sentence-Level Semantic Similarity Calculation

Given sentences S_i and S_j, we now calculate the similarity between them. Suppose that S_i and S_j are parsed into frames by the semantic role labeler, respectively. For each pair of frames $f_m \in S_i$ and $f_n \in S_j$, we discover the semantic relations of terms in the same semantic role using WordNet [30]. If two words in the same semantic role are identical or of the semantic relations such as synonym, hypernym, hyponym, meronym, and holonym, the words are considered as "related."

Let $\{r_1, r_2, \ldots, r_k\}$ be the set of K common semantic roles between f_m and f_n, $T_m(r_i)$ be the term set of f_m in role r_i, and $T_n(r_i)$ be the term set of f_n in role r_i. Letting $|T_m(r_i)| \leq |T_n(r_i)|$, we compute the similarity between $T_m(r_i)$ and $T_n(r_i)$ as

$$\mathrm{rsim}\left(T_m(r_i), T_n(r_i)\right) = \frac{\sum_j \mathrm{tsim}\left(t_{ij}^m, r_i\right)}{|T_n(r_i)|} \tag{1}$$

where

$$\mathrm{tsim}\left(t_{ij}^m, r_i\right) = \begin{cases} 1, & t_{ij}^m \in T_m(r_i), \exists\, t_{ik}^n \in T_n(r_i) \\ & \text{s.t. } t_{ij}^m \text{ and } t_{ik}^n \text{ are related} \\ 0, & \text{else.} \end{cases}$$

Then, the similarity between f_m and f_n is

$$\mathrm{fsim}(f_m, f_n) = \frac{\sum_{i=1}^k \mathrm{rsim}\left(T_m(r_i), T_n(r_i)\right)}{K}. \tag{2}$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

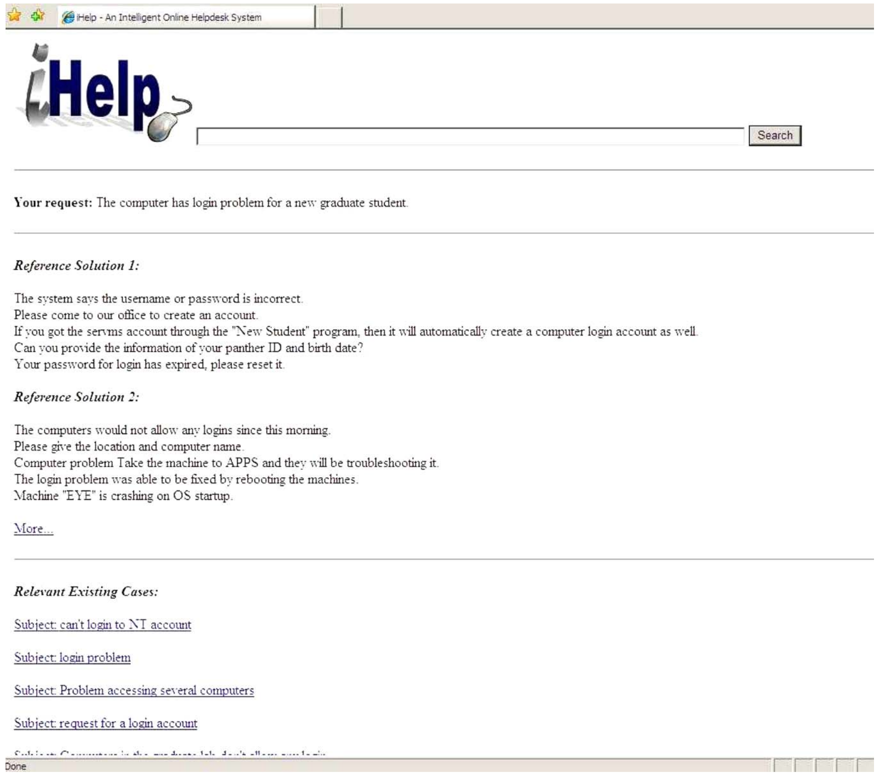

Fig. 2. Screenshot of an example output of the iHelp system.

TABLE I
REPRESENTATION OF ARGUMENTS AND AN ILLUSTRATIVE EXAMPLE

rel: the verb	Arg0: causer of motion
Arg1: thing in motion	Arg2: distance moved
Arg3: start point	Arg4: end point
Arg5: direction	ArgM-LOC: location
ArgM-EXT: extent	ArgM-TMP: time
ArgM-DIS: discourse connectives	ArgM-PNC: purpose
ArgM-ADV: general-purpose	ArgM-MNR: manner
ArbM-NEG: negation marker	ArgM-DIR: direction
ArgM-MOD: modal verb	ArgM-CAU: cause
Example: *Sentence*: The printer does not work. *Label*: The (Arg0) printer (Arg0) does (-) not (ArgM-NEG) work (rel).	

Therefore, the semantic similarity between S_i and S_j can be calculated as follows:

$$\mathrm{Sim}(S_i, S_j) = \max_{f_m \in S_i, f_n \in S_j} \mathrm{fsim}(f_m, f_n) \tag{3}$$

where each similarity score is between zero and one.

B. Case Relevance Calculation

Once there is a new request, the existing cases are ranked based on the scores calculated in the following:

$$\mathrm{Score}(d_p, request) = \max_{S_i \in d_p} \mathrm{Sim}(S_i, request) \tag{4}$$

where d_p represents the pth case in the past case collection. Moreover, the list of the ranked cases is returned to the customer as the search results.

VI. TOP-RANKING CASE CLUSTERING

To better facilitate users to find the solutions of their problems, iHelp first clusters the top-ranking cases and then generates a short summary for each case cluster. Although the top-ranking cases are all relevant to the request input by the customer, these relevant cases may actually belong to different categories. For example, if the request is "my computer does not work," the relevant cases involve various computer problems, such as system crash, hard disk failure, etc. Therefore, it is necessary to further group these cases into different contexts.

A. Mixture Model for Case Similarity Calculation

Since all of the top-ranking cases are relevant to the request, they may share some common information. To organize the cases into different contexts, the challenge is to filter out the general and common information while obtaining the informative contexts of the cases. Here, we use a mixture language model [31] to measure the similarity between documents while filtering out the general and common information from the request.

As shown in Fig. 3, each relevant case can be generated by the mixture of a general English model θ_E, a user-specific request model θ_R, and a document context model θ_C. The probabilities of a word w_i in a case generated by the language models are λ_E, λ_R, and λ_C, respectively [31]

$$\begin{aligned} &P(w_i|\theta_E, \theta_R, \theta_C, \lambda_E, \lambda_R, \lambda_C) \\ &\quad = \lambda_E P(w_i|\theta_E) + \lambda_R P(w_i|\theta_R) + \lambda_C P(w_i|\theta_C) \end{aligned} \tag{5}$$

where $\lambda_E + \lambda_R + \lambda_C = 1$.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

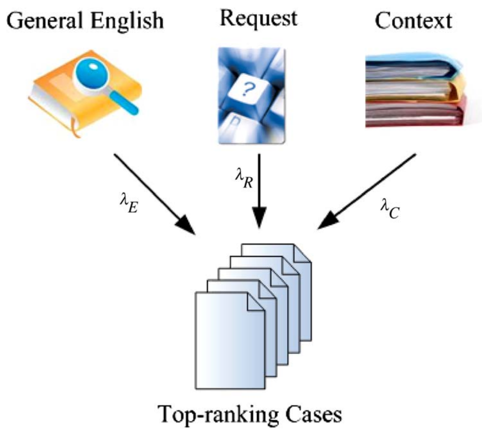

Fig. 3. Mixture model for calculating case similarities.

For example, if the request is “create an account, in a relevant past case, words such as “an and “hi probably come from the general English model. Since the words of “create and “account appear in the request sentence, we know that they may have high probability from the request model. For the content of “a new student wants to create an e-mail account, in a case, words such as “new, “student, and “e-mail more likely come from the context model.

The similarity of two cases d_i and d_j can be computed as

$$S(d_i, d_j) = \frac{KL\left(\theta_C(d_i), \theta_C(d_j)\right) + KL\left(\theta_C(d_j), \theta_C(d_i)\right)}{2} \tag{6}$$

where KL is the Kullback–Leibler divergence (which is a distributional similarity measure [32]) and $\theta_C(d_i)$ is the context model for d_i. θ_E, θ_R, and θ_C can be trained using the expectation–maximization algorithm, which is widely used to fit language models [31], [33], [34]. By using the mixture model, the effect of the words that occur frequently in the request or in general English on the similarity calculation is naturally reduced.

B. SNMF

Once we obtain the similarity matrix of the relevant cases, clustering algorithms need to be performed to group these cases into clusters. Most document-clustering algorithms deal with a rectangular data matrix (e.g., document-term matrix and sentence-term matrix), and they are not suitable for clustering a pairwise similarity matrix. In our work, we propose the SNMF algorithm to conduct the clustering.

Problem Formulation and Algorithm Procedure: Given a matrix of pairwise similarity W, we want to find H such that

$$\min_{\mathbf{H} \geq 0} \mathcal{J} = \|\mathbf{W} - \mathbf{H}\mathbf{H}^T\|^2 \tag{7}$$

where the matrix norm $\|\mathbf{X}\|^2 = \sum_{ij} \mathbf{X}_{ij}^2$ is the Frobenius norm.

Theorem 1: The loss $\|\mathbf{W} - \mathbf{H}\mathbf{H}^\top\|^2$ is nonincreasing under the update rule

$$\mathbf{H}_{ik} \leftarrow \mathbf{H}_{ik} \sqrt[4]{\frac{[\mathbf{W}\mathbf{H}]_{ik}}{[\mathbf{H}\mathbf{H}^\top\mathbf{H}]_{ik}}}. \tag{8}$$

The loss is invariant under these updates if and only if $\mathbf{H}$ is at a stationary point of the loss with the constraints.

The proof is given in the Appendix. Hence, the algorithm procedure for solving SNMF is as follows: Given an initial guess of $\mathbf{H}$, iteratively update $\mathbf{H}$ using (8) until convergence. This method will converge to a local minima of the problem.

It has been shown that SNMF is equivalent to kernel K-means clustering and is a special case of trifactor NMF [35]–[37]. Another important property is that the simple SNMF is equivalent to the sophisticated normalized cut spectral clustering. Spectral clustering is a principled and effective approach for solving normalized cuts [38]. These results demonstrate the clustering ability of SNMF.

To determine the number of clusters, we use the model selection method described in [39]. For each number of clusters, we cluster cases many times with random initial seeds. The average of mutual information between different clustering results is used as the criterion for determining the goodness of the number of clusters. The small average of the mutual information implies that the corresponding number of clusters produces stable clustering results.

VII. Multidocument Summarization for Each Case Cluster

To improve the usability of the system, we perform multidocument summarization to generate a brief summary for each case cluster. The general issues for multidocument summarization are as follows: First of all, the information contained in different documents often overlaps with each other; therefore, it is necessary to find an effective way to merge the documents while recognizing and removing redundancy. Another issue is identifying important difference between documents and covering the informative content as much as possible [40].

Our proposed sentence-level semantic analysis approach (see Section V-A) and SNMF clustering algorithm (see Section VI-B) can be naturally applied to the summarization task to address the aforementioned issues. Sentence-level semantic analysis can better capture the relationships between sentences, and we use it to construct the sentence similarity matrix by computing the pairwise sentence similarity as described in Section V-A. Based on the similarity matrix, we use the SNMF algorithm described in Section VI-B to cluster the sentences. Note that the standard NMF deals with a rectangular matrix and is thus not appropriate here. Then, in each sentence cluster, we perform within-cluster sentence selection to identify the most semantically important sentence using a measure combining the internal information (e.g., the computed similarity between sentences in the case cluster) and the external information (e.g., the request input by customers).

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

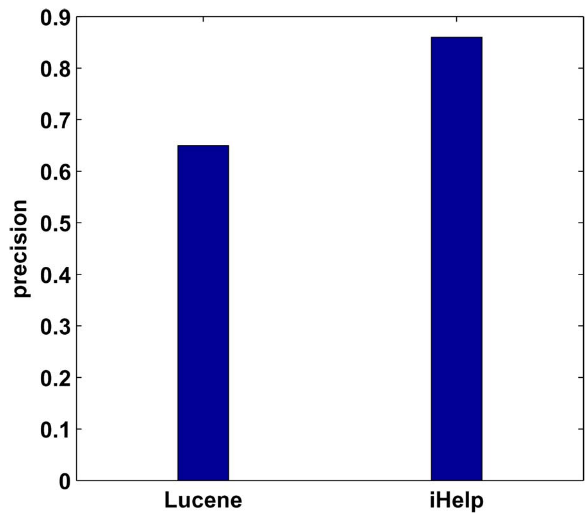

Fig. 4. Precision of the retrieved cases.

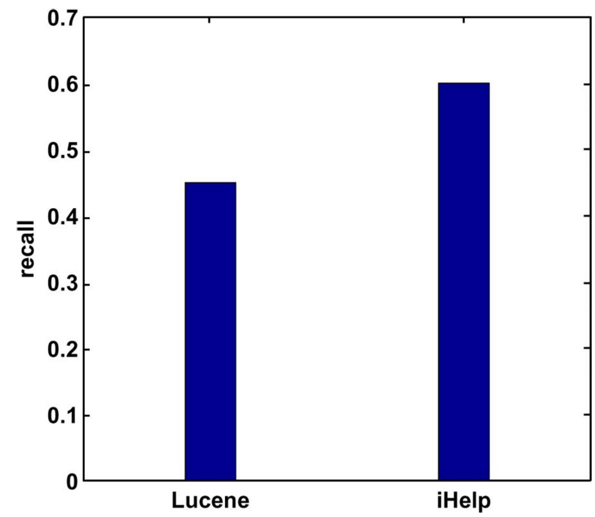

Fig. 5. Recall of the retrieved cases.

A. Within-Cluster Sentence Selection

After grouping the sentences into clusters by the SNMF algorithm, first of all, we remove the noisy clusters if the cluster of sentences contains less than three sentences. Then, in each sentence cluster, we rank the sentences based on the sentence score calculation, as shown in (9)–(11). The score of a sentence measures the importance of a sentence to be included in the summary

$$\mathrm{Score}(S_i) = \lambda F_1(S_i) + (1-\lambda) F_2(S_i) \tag{9}$$

$$F_1(S_i) = \frac{1}{N-1} \sum_{S_j \in C_k - S_i} \mathrm{Sim}(S_i, S_j) \tag{10}$$

$$F_2(S_i) = \mathrm{Sim}(S_i, request) \tag{11}$$

where $F_1(S_i)$ measures the average similarity score between sentence S_i and all the other sentences in cluster C_k, and N is the number of sentences in C_k. $F_2(S_i)$ represents the similarity between sentence S_i and input request $request$. λ is the weight parameter, which is set to 0.7 empirically.

VIII. Experiments

A. Case Retrieval Comparison

In this set of experiments, we randomly select ten questions from different categories and manually label the related cases for each question. Then, we examine the top 20 retrieved cases by keyword-based Lucene and our iHelp system, respectively. Figs. 4 and 5 show the average precision and recall of the two methods.

The high precision of iHelp demonstrates that the semantic similarity calculation can better capture the meanings of the requests and case documents. Since we only look at the top 20 retrieved cases while some of the cases may have more than 40 relevant cases, the recall is also reasonable and acceptable.

B. Case Study

In this section, we present some illustrative scenarios, in which we compare our proposed request-focused case-ranking results and Apache Lucene, which is one of the most popular keyword-based text-ranking engines. The summarized solutions for each case cluster by iHelp from top 20 cases are also presented.

Scenario 1: Can you switch those two computers?: Table II shows the top-ranking case samples retrieved by Lucene and iHelp, respectively. While looking at the ranking results, we find that Lucene takes the word "switch" as the keyword; thus, some high-ranking past cases actually discuss about the equipment switch. However, these cases are not semantically related to the user's request at all. For iHelp, the word "switch" is a verb, and the corresponding semantic role is "rel." Therefore, the cases related to the equipment switch are unlikely to be ranked high. In addition, iHelp ranking is able to find some informative content that Lucene cannot. For example, some cases related to "move machines" ("machines" and "computers" are related words) and "exchange computers" are highly ranked in iHelp, although the word of "switch" or "computer" does not appear in the text itself.

After clustering the top-ranking cases, we find that the request of switching computers can be either changing the location of two computers or changing the users and accounts for the computers. Sample summaries by iHelp are shown in Table III.

Scenario 2: The computer in the printing room needs to add memory: In Scenario 2, Lucene will take "printing" as the keyword and return many cases related to printing or printers as the search results. Obviously, they are not what the customer wants. In iHelp ranking, the semantic role of the word "printing" is the location tag, which decides that the cases related to "printing" will not be retrieved. Table IV compares the highly ranked case samples by Lucene and iHelp. Table V shows the summarized reference solutions generated by iHelp.

Scenario 3: The computer has log-in problem for a new graduate student: In this scenario, Lucene takes "new," "graduate," "student," "computer," and "log-in" as the keywords,

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE II
TOP-RANKING CASE SAMPLES BY LUCENE AND iHELP IN SCENARIO 1

Request	Can you switch those two computers?
Lucene	*Top-ranking case sample 1:* Subject: KVM ***switches***/extra cables order request We would like to purchase following ***switches*** and cables: - 1 eight port KVM switch w/corresponding 8 VGA/(ps/2) ***switch*** cable sets - 6 VGA/(ps/2) additional sets for our current CYBEX ***switches*** Could we get quotes please ?
	Top-ranking case sample 2: Subject: research 2-port usb ***switch*** Steve: When we deploy Tim's new machine in his office, we will need another 2-port KVM ***switch***, like the one we purchased for the ILab. (W will need to order cables, this time....)
iHelp	*Sample top-ranking case 1:* Subject: ***Moving*** people and ***computers*** from ECS 257 The following moves have been approved. move n94 on desk D-322 to 238 also remove D-322. Deploy 1 new Dell to 238. If you need help identifying these machines/desks, Andriy or I can help you.
	Top-ranking case sample 2: Subject: my computer I have an approval from Becky to ***exchange computers***, and I need some time to make a backup of important information. Then I will bring both computers on Tuesday.

TABLE III
SAMPLE SUMMARIES GENERATED BY iHELP IN SCENARIO 1

Case Cluster	Summarization for Each Case Cluster
Exchange the location of the computers	I have an approval from Becky to exchange computers, and I need some time to make a backup of important information. Which computers? Please note that I will be coming in today at about 3pm to swap the machines. It is desirable to have this done using one of the recently purchased computers. The speakers connected to the computer won't be moved.
Re-assign computers to different users	The machines in ECS 238 are re-assigned and need to be set up. Please make sure you backup the information and sign off the computer ASAP. Username and password to use the computer have been created. One is in the second row from the front of the classroom on the left-side, and the other is in the middle of the row behind. The computers are deployed and ready for use now.

TABLE IV
TOP-RANKING CASE SAMPLES BY LUCENE AND iHELP IN SCENARIO 2

Request	The computer in the printing room needs to add memory.
Lucene	*Top-ranking case sample 1:* Subject: request I would like to know for ***printing*** a document if I have to add/set the type of the ***printer*** that is being used in this section myself or to wait for assistance on this respect from your section? Besides I will appreciate if you could arrange to send a pack of paper to this section.
	Top-ranking case sample 2: Subject: ***Printing*** to np243 The ***printer*** np243 has not been responding from MSword.
iHelp	*Top-ranking case sample 1:* Subject: Re: ***upgrade memory*** The machines F12 F13 and F14 have been ***upgraded*** with 128 MB of ***RAM***.
	Top-ranking case sample 2: Subject: ***upgrade memory*** If memory ok, please schedule the following desktop memory upgrades. You can ***deploy*** grad lab ***memory*** at your discretion.

TABLE V
SAMPLE SUMMARIES GENERATED BY iHELP IN SCENARIO 2

Case Cluster	Summarization for Each Case Cluster
Memory update	Your computer is going to be upgraded with 128 MB of RAM. When would be a good time to install a memory upgrade for your computer? Please assign staff to investigate if there is an available memory slot available. We will obtain pricing and quotations to purchase memory. Can you please indicate the memory total you would like to see installed in N94.
Alternative solutions instead of updating memory	The matlab code needs more memory. What is the error message for out of memory. Many programs are available in unix server which has more memory than your local PC. Please explain the reason for purchasing additional memory. Running several programs at the same time may cause lacking of memory.

TABLE VI
TOP-RANKING CASE SAMPLES BY LUCENE AND iHELP IN SCENARIO 3

Request	The computer has login problem for a new graduate student.
Lucene	*Top-ranking case sample 1:* Subject: Weasel ***login*** I'm a ***graduate student***, and I was trying to telnet to "Weasel" and my username and password is rejected. Currently I can ***login*** without any problem to "Goliath" and "Leopard", but I will like to ***login*** to a third machine in order to test my distributed program. Can you give me a list of host that I can use to connect from outside (SSH)?
	Top-ranking case sample 2: Subject: How to send emails under unix Since you are a ***new student*** in our department, please come see me tomorrow in ECS258 after 10AM, and I can show you how to send E-mails with arbitrary From: headers using Pine.
iHelp	*Top-ranking case sample 1:* Subject: can't ***login*** to NT account I can not ***login*** to my NT account. It takes a long time to load my personal settings....and finally gives an error message. Could you please solve this?
	Top-ranking case sample 2: Subject: Problem ***accessing*** several ***computers*** I can not ***login*** to "Camille", "David" or "Bertha". The error message is "Windows can not ***log*** you on because your profile can not be loaded."

TABLE VII
SAMPLE SUMMARIES GENERATED BY iHELP IN SCENARIO 3

Case Cluster Sample	Summarization for Each Case Cluster
Account problem	The system says the username or password is incorrect. Please come to our office to create an account. If you got the servms account through the "New Student" program, then it will automatically create a computer login account as well. Can you provide the information of your panther ID and birth date? Your password for login has expired, please reset it.
Computer problem	The computers would not allow any logins since this morning. Please give the location and computer name. Take the machine to APPS and they will be troubleshooting it. The login problem was able to be fixed by rebooting the machines. Machine "EYE" is crashing on OS startup.

and various log-in problems and problems encountered by new students are retrieved, while results by iHelp focus on the computer log-in problems. Table VI lists the top-ranking case samples obtained by Lucene and iHelp, respectively. Moreover, after clustering the top-ranking cases, the relevant cases are either related to the account or computer problems, as shown in Table VII.

The aforementioned case studies clearly demonstrate the effectiveness of iHelp in semantic text understanding, case clustering, and case summarization. The better performance of iHelp benefits from the sentence-level semantic analysis, the use of mixture language model, and the SNMF clustering algorithm.

C. User Study

To better evaluate the ranking and summarization results of iHelp, we conduct two surveys. The subjects of the survey are 16 students at different levels and from various majors of a university. We randomly choose five requests from the following different categories: 1) opening accounts; 2) installing software; 3) printing problems; 4) ordering new equipments; and 5) networking connection problems. In the first survey, the participants are asked to evaluate the ranking quality of Apache Lucene and iHelp, and in the second survey, the participants need to compare the summaries generated by iHelp with several alternative solutions. In both surveys, each participant is asked to assign a score of 1 to 5, according to their satisfaction of the ranking or summarization results for a request. The higher the score, the better the ranking or summarization quality.

TABLE VIII
SURVEY 1: RANKING COMPARISON

Request	1	2	3	4	5
Lucene	3.0000	3.4375	3.0000	2.8750	2.9375
iHelp	4.1250	4.5625	4.0625	3.9375	4.0000

1) Case Ranking Comparison: In this survey, we compare the case-ranking results by Lucene and iHelp for the randomly selected five requests in different categories. The orders of the results of Lucene and iHelp rankings are randomly permuted for each user. The participants are asked to rate these two approaches based on the relevance of top five cases retrieved by them. Table VIII shows the average scores of Lucene and iHelp for each request.

The results of the survey show the superiority of the iHelp ranking method. Our ranking approach in iHelp utilizes sentence-level semantic analysis to better understand the contexts of the cases, which leads to the higher user satisfaction than the traditional keyword-based ranking.

2) Case Clustering and Summarization Comparison: This survey compares the summaries generated by iHelp for each case cluster with four alternative clustering and summarization methods as follows.

1) *Method 1*—No summarization: Only ranking results are returned to the user.

TABLE IX
SURVEY 2: CASE CLUSTERING AND SUMMARIZATION COMPARISON

Request	1	2	3	4	5
Method 1	1.5000	1.5000	1.5000	1.5000	1.5000
Method 2	3.0000	2.9375	2.5000	2.4375	2.4375
Method 3	2.0625	2.1875	2.0625	1.3750	1.8750
Method 4	3.0625	3.0625	2.6875	2.0625	2.5000
iHelp	4.3125	4.3750	4.0625	3.875	3.9375

2) *Method 2*—Summarization without clustering: The top-ranking cases are not clustered, so the summary is generated based on all the top 20 relevant cases.
3) *Method 3*—Case clustering using NMF algorithm: The top-ranking cases are filtered out by the mixture language model and clustered by the standard NMF algorithm.
4) *Method 4*—Case clustering without the mixture model: The top-ranking cases are clustered based on all the contents contained in the cases without filtering out the general and common information. The clustering algorithm and summarization method for each cluster are the same as those developed in iHelp.

Table IX shows the ratings that the participants assign to each method for each request. Since Method 1 does not provide clustering and summarization functions, we set it to be the baseline method with the score 1.5000 for all the requests. We set Method 1 a relatively low score of 1.5 (but not the lowest score, for example, 1) for the following reasons: 1) We believe that the case grouping and summarization can help users to capture the ideas embedded in the search results, and 2) it is possible that some methods may have lower scores than 1.5, so we leave some space for a poorer method to have the lowest scores (between 1 and 1.5). Moreover, the participants need to compare all the other methods and rate them with reference to the baseline method. A rank of 5 (or 1) indicates that the summary is most (or least) descriptive and helpful.

Comparing Method 1 with other methods, we observe that the user satisfaction is improved along with the recommending reference solutions from past cases at most circumstances, which proves the necessity of summarization. From the ratings of the last four methods, we confirm that combining the mixture language model that filters out the general and common information and the SNMF clustering algorithm can help users to easily find their desired solutions. However, if an inappropriate clustering algorithm or insufficient language model is performed, the results may be poorly organized. For example, in Method 3, the traditional NMF algorithm is used to cluster cases, and we observe that the ratings of Method 3 are even lower than the ratings of Method 2 in which the summarization results are displayed without case clustering.

IX. CONCLUSION

Helpdesk is critical to every enterprise's IT service delivery. In this paper, we have proposed iHelp, an intelligent online helpdesk system, to automatically find problem–solution patterns given a new request from a customer by ranking, clustering, and summarizing the past interactions between customers and representatives. Case and user studies have been conducted to show the full functionality and effectiveness of iHelp. The high performance of iHelp benefits from the proposed approaches of semantic case ranking, case clustering using the mixture language model and symmetric matrix factorization, and the request-focused multidocument summarization.

APPENDIX
PROOF OF THEOREM 1

If $\mathbf{H}_{ik}$ is zero, $\mathbf{H}_{ik}$ is a fixed point and satisfies the Karush–Kuhn–Tucker (KKT) condition. Now, we assume that all $\mathbf{H}_{ikm}$'s are positive. Then, $[\mathbf{H}^\top\mathbf{H}]_{kl}$ is positive for any l. The following inequality holds for any $\widetilde{\mathbf{H}}$:

$$\begin{aligned}\mathcal{L}(\mathbf{H}) &\equiv \|\mathbf{W}-\mathbf{H}\mathbf{H}^\top\|^2\\ &= \mathrm{tr}(\mathbf{H}\mathbf{H}^\top\mathbf{H}\mathbf{H}^\top) - 2\mathrm{tr}(\mathbf{W}^\top\mathbf{H}\mathbf{H}^\top) + \mathrm{tr}(\mathbf{W}^\top\mathbf{W})\\ &\le \frac{1}{2}\mathrm{tr}(\mathbf{P}\widetilde{\mathbf{H}}^\top\widetilde{\mathbf{H}}) + \frac{1}{2}\mathrm{tr}(\mathbf{P}\widetilde{\mathbf{H}}^\top\widetilde{\mathbf{H}})\\ &\quad - 2\mathrm{tr}(\mathbf{W}^\top\mathbf{H}\mathbf{H}^\top) + \mathrm{tr}(\mathbf{W}^\top\mathbf{W})\\ &\le \frac{1}{2}\mathrm{tr}(\mathbf{R}\widetilde{\mathbf{H}}^\top\widetilde{\mathbf{H}}\widetilde{\mathbf{H}}^\top) + \frac{1}{2}\mathrm{tr}(\mathbf{R}\widetilde{\mathbf{H}}^\top\widetilde{\mathbf{H}}\widetilde{\mathbf{H}}^\top)\\ &\quad - 2\mathrm{tr}(\mathbf{W}^\top\widetilde{\mathbf{H}}\mathbf{Z}^\top) - 2\mathrm{tr}(\mathbf{W}^\top\mathbf{Z}\widetilde{\mathbf{H}}^\top)\\ &\quad - 2\mathrm{tr}(\mathbf{W}^\top\widetilde{\mathbf{H}}\widetilde{\mathbf{H}}^\top) + \mathrm{tr}(\mathbf{W}^\top\mathbf{W})\\ &\equiv \mathcal{Q}(\mathbf{H},\widetilde{\mathbf{H}})\end{aligned}$$

where $\mathbf{P}_{kl} = [\mathbf{H}^\top\mathbf{H}]_{kl}^2/[\widetilde{\mathbf{H}}^\top\widetilde{\mathbf{H}}]_{kl}$, $\mathbf{R}_{ik} = [\mathbf{H}]_{ik}^4/[\widetilde{\mathbf{H}}]_{kl}^3$, and $\mathbf{Z}_{ij} = \widetilde{\mathbf{H}}_{ij}\ln(\mathbf{H}_{ij}/\widetilde{\mathbf{H}}_{ij})$. It is easy to check that the equality holds when $\mathbf{H}=\widetilde{\mathbf{H}}$, i.e., $L(\widetilde{\mathbf{H}}) = \mathcal{Q}(\widetilde{\mathbf{H}},\widetilde{\mathbf{H}})$. If $\mathbf{H}$ minimizes $\mathcal{Q}(\mathbf{H},\widetilde{\mathbf{H}})$, we have $\mathcal{L}(\mathbf{H}) \le \mathcal{Q}(\mathbf{H},\widetilde{\mathbf{H}}) \le \mathcal{Q}(\widetilde{\mathbf{H}},\widetilde{\mathbf{H}}) = \mathcal{L}(\mathbf{H})$, i.e., reducing the loss. To find the minimum point of $\mathcal{Q}(\mathbf{H},\widetilde{\mathbf{H}})$, the KKT condition is

$$\frac{\partial \mathcal{Q}}{\partial \mathbf{H}_{ik}} = 2\frac{\mathbf{H}_{ik}^3}{\widetilde{\mathbf{H}}_{ik}^3}[\widetilde{\mathbf{H}}\widetilde{\mathbf{H}}^\top\widetilde{\mathbf{H}} + \widetilde{\mathbf{H}}\widetilde{\mathbf{H}}^\top\widetilde{\mathbf{H}}]_{ik} - 2\frac{\widetilde{\mathbf{H}}_{ik}}{\mathbf{H}_{ik}}[\mathbf{W}^\top\widetilde{\mathbf{H}} + \mathbf{W}\widetilde{\mathbf{H}}]_{ik} = 0.$$

We have the update rule for $\mathbf{H}$ as (8), since $\mathbf{W}$ is symmetric. ■

REFERENCES

[1] B. K. Giamanco, Customer service: The importance of quality customer service. [Online]. Available: http://www.customerservicetrainingcenter.com
[2] D. Bridge, M. H. Goker, L. Mcginty, and B. Smyth, "Case-based recommender systems," *Knowl. Eng. Rev.*, vol. 20, no. 3, pp. 315–320, Sep. 2005.
[3] A. A. Kamis and E. A. Stohr, "Parametric search engines: What makes them effective when shopping online for differentiated products?" *Inf. Manage.*, vol. 43, no. 7, pp. 907–918, Oct. 2006.
[4] N. Mirzadeh, F. Ricci, and M. Bansal, "Feature selection methods for conversational recommender systems," in *Proc. IEEE Int. Conf. e-Technol., e-Commerce e-Service*, 2005, pp. 772–777.
[5] M. Doyle and P. Cunningham, "A dynamic approach to reducing dialog in on-line decision guides," in *Proc. EWCBR*, 2000, pp. 49–60.
[6] D. W. Aha, D. Mcsherry, and Q. Yang, "Advances in conversational case-based reasoning," *Knowl. Eng. Rev.*, vol. 20, no. 3, pp. 247–254, Sep. 2005.
[7] P. Cunningham and B. Smyth, "A comparison of model-based and incremental case-based approaches to electronic fault diagnosis," Dept. Comput. Sci., Trinity College Dublin, Dublin, Ireland, Tech. Rep. TCD-CS-94-21, 1994.
[8] R. Agrawal, R. Rantzau, and E. Terzi, "Context-sensitive ranking," in *Proc. SIGMOD*, 2006, pp. 383–394.

[9] S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis, “Automated ranking of database query results,” in *Proc. CIDR*, 2003, pp. 888–899.
[10] K. Chakrabarti, V. Ganti, J. Han, and D. Xin, “Ranking objects based on relationships,” in *Proc. SIGMOD*, 2006, pp. 371–382.
[11] S. Chaudhuri, G. Das, V. Hristidis, and G. Weikum, “Probabilistic ranking of database query results,” in *Proc. VLDB*, 2004, pp. 888–899.
[12] G. Das, V. Hristidis, N. Kapoor, and S. Sudarshan, “Ordering the attributes of query results,” in *Proc. SIGMOD*, 2006, pp. 395–406.
[13] C. C. Aggarwal and P. S. Yu, “The IGrid index: Reversing the dimensionality curse for similarity indexing in high dimensional space,” in *Proc. KDD*, 2000, pp. 119–129.
[14] S. Berchtold, B. Ertl, D. A. Keim, H.-P. Kriegel, and T. Seidl, “Fast nearest neighbor search in high-dimensional space,” in *Proc. ICDE*, 1998, pp. 209–218.
[15] H. V. Jagadish, B. C. Ooi, K.-L. Tan, C. Yu, and R. Zhang, “idistance: An adaptive b^+-tree based indexing method for nearest neighbor search,” *ACM Trans. Database Syst.*, vol. 30, no. 2, pp. 364–397, Jun. 2005.
[16] S. Chaudhuri and L. Gravano, “Evaluating top-k selection queries,” in *Proc. VLDB*, M. P. Atkinson, M. E. Orlowska, P. Valduriez, S. B. Zdonik, and M. L. Brodie, Eds., 1999, pp. 397– 410.
[17] R. Fagin, A. Lotem, and M. Naor, “Optimal aggregation algorithms for middleware,” in *Proc. PODS*, 2001, pp. 102–113.
[18] Y. Luo, X. Lin, W. Wang, and X. Zhou, “Spark: Top-k keyword query in relational databases,” in *Proc. SIGMOD Conf.*, 2007, pp. 115–126.
[19] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, “When is nearest neighbor meaningful?” in *Proc. ICDT*, 1999, pp. 217–235.
[20] D. Wang, S. Zhu, T. Li, and C. Ding, “Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization,” in *Proc. SIGIR*, 2008, pp. 307–314.
[21] M. Hearst and J. Pedersen, “Reexamining the cluster hypothesis: Scatter/gather on retrieval results,” in *Proc. SIGIR*, 1996, pp. 76–84.
[22] A. Leuski and J. Allan, “Improving interactive retrieval by combining ranked list and clustering,” in *Proc. RIAO*, 2000, pp. 665–681.
[23] O. Zamir and O. Etzioni, “A dynamic clustering interface to web search results,” in *Proc. WWW*, 1999, pp. 1361–1374.
[24] O. Zamir and O. Etzioni, “Web document clustering: A feasibility demonstration,” in *Proc. SIGIR*, 1998, pp. 46–54.
[25] I. Mani, *Automatic Summarization*. Amsterdam, The Netherlands: John Benjamins Publishing Company, 2001.
[26] B. Ricardo and R. Berthier, *Modern Information Retrieval*. New York: ACM Press, 1999.
[27] D. Arnold, L. Balkan, S. Meijer, R. Humphreys, and L. Sadler, *Machine Translation: An Introductory Guide*. London, U.K.: Blackwells-NCC, 1994.
[28] R. Collobert and J. Weston, “Fast semantic extraction using a novel neural network architecture,” in *Proc. ACL*, 2007, pp. 560–567.
[29] M. Palmer, P. Kingsbury, and D. Gildea, “The proposition bank: An annotated corpus of semantic roles,” *Comput. Linguist.*, vol. 31, no. 1, pp. 71–106, Mar. 2005.
[30] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998.
[31] Y. Zhang, J. Callan, and T. Minka, “Novelty and redundancy detection in adaptive filtering,” in *Proc. SIGIR*, 2002, pp. 81–88.
[32] S. Kullback and R. Leibler, “On information and sufficiency,” *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, Mar. 1951.
[33] W. Kraaij, R. Pohlmann, and D. Hiemstra, “Twenty-one at TREC-8: Using language technology for information retrieval,” in *Proc. TREC-8*, 1999, pp. 285–900.
[34] C. Zhai and J. Lafferty, “Model-based feedback in the language modeling approach to information retrieval,” in *Proc. CIKM*, 2001, pp. 403–410.
[35] H. Zha, X. He, C. Ding, M. Gu, and H. Simon, “Bipartite graph partitioning and data clustering,” in *Proc. CIKM*, 2001, pp. 25–32.
[36] C. Ding, T. Li, W. Peng, and H. Park, “Orthogonal nonnegative matrix t-factorizations for clustering,” in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 126–135.
[37] T. Li and C. Ding, “The relationships among various nonnegative matrix factorization methods for clustering,” in *Proc. 6th ICDM*, 2006, pp. 362–371.
[38] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[39] X. Liu, Y. Gong, W. Xu, and S. Zhu, “Document clustering with cluster refinement and model selection capabilities,” in *Proc. SIGIR*, 2002, pp. 191–198.
[40] D. Radev, E. Hovy, and K. McKeown, “Introduction to the special issue on summarization,” *Comput. Linguist.*, vol. 28, no. 4, pp. 399–408, Dec. 2002.

Dingding Wang received the B.S. degree from the Department of Computer Science, University of Science and Technology of China, Hefei, China, in 2003. She is currently working toward the Ph.D. degree in the School of Computing and Information Sciences, Florida International University, Miami.

Her research interests are data mining, information retrieval, and machine learning.

Tao Li received the Ph.D. degree in computer science from the Department of Computer Science, University of Rochester, Rochester, NY, in 2004.

He is currently an Associate Professor with the School of Computing and Information Sciences, Florida International University, Miami. His research interests are data mining, machine learning, information retrieval, and bioinformatics.

Shenghuo Zhu received the B.E. degree from Zhejiang University, Hangzhou, China, in 1994, the B.E. degree from Tsinghua University, Beijing, China, in 1997, and the Ph.D. degree in computer science from the University of Rochester, Rochester, NY, in 2003.

He is currently a Research Staff Member with NEC Laboratories America, Inc., Cupertino, CA. His primary research interests include information retrieval, probabilistic modeling, machine learning, and data mining.

Yihong Gong received the B.S., M.S., and Ph.D. degrees in electronic engineering from the University of Tokyo, Tokyo, Japan, in 1987, 1989, and 1992, respectively.

In 1992, he joined Nanyang Technological University, Singapore, Singapore, where he was an Assistant Professor with the School of Electrical and Electronic Engineering for four years. From 1996 to 1998, he was a Project Scientist with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. Since 1999, he has been leading the Multimedia Processing Group, NEC Laboratories America, Inc., Cupertino, CA, where he became the Site Manager to lead the Cupertino branch of the laboratory in 2006. His research interests include multimedia content analysis and machine learning applications. The major research achievements from his group include news video summarization, sports highlight detection, data clustering, and SmartCatch video surveillance that led to a successful spinoff.

Dr. Gong was a Principal Investigator for both the Informedia Digital Video Library project and the Experience-On-Demand project funded in multimillion dollars by the National Science Foundation, the Defense Advanced Research Projects Agency, the National Aeronautics and Space Administration, and other government agencies.