

Implement Helpdesk System Through Online

M.Nagendramma¹
CSE Department
B.V.S.REngg.College
Chimakurthy
A.P, India

Shaik.Ghouse Basha²
CSE Department
B.V.S.R Engg.College
Chimakurthy
A.P, India

Md.Parveen Begum³
CSE Department
B.V.S.R Engg.College
Chimakurthy
A.P, India

Abstract

Now a day's almost every kind of product or service can be bought online huge things are available to the end-user through the World Wide Web. Nevertheless, a major problem is generated by this over-abundance of products and services, i.e., how to find the "right" information that satisfies the user needs and wants. This problem is only partially addressed by search engines (e.g., Google. com, yahoo.com or MSN.com, etc.). A search engine can support only the initial stages of the search process, i.e., it can locate Web sites where relevant information is available.

But search engines are keyword-based and are not much useful within a Web site to help the user to identify his preferred service. For this purpose, a new kind 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 at terms 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

Are 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.

1. Introduction

The 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.

It is also many online intelligent systems but mainly suffer from keyword matching technologies and error-level information at the solution time. So we have *proposed a new algorithm* called *semantic role parser*, similarity score calculation. The main objectives of this automatically find the problem solution.

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 a case-based system case collects all the information provided by the user during a recommendation session such as the user's queries to the product catalogues, the selected products, and, in case the user is registered, some stable user-related preferences and demographic data. However, these case-based systems face the following two challenges.

1.1 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 [2]–[6], and to perform similarity search [7]–[9]. 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.

1.2 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.

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. It is also necessary to generate a short and concise summary for each context to improve the usability.

2. 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.

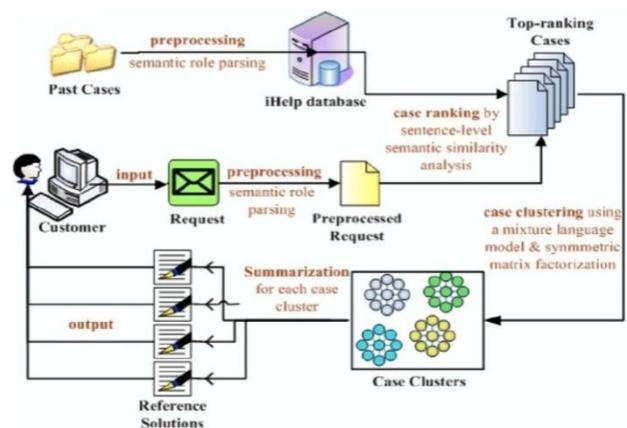


Fig1: Framework of the iHELP system

In *existing system*, a help desk is a place that a user of information technology can call to get help with a problem. In many companies, a help desk is simply one person with a phone number and a more or less organized idea of how to handle the problems that come in. In larger companies, a help desk may consist of a group of experts using software to help track the status of problems and other special software to help analyze problems (for example, the status of a company's telecommunications network).

Typically, the term is used for centralized help to users within an enterprise. A related term is call center, a place that customers call to place orders, track shipments, get help with products, and so forth. The World Wide Web offers the possibility of a new, relatively inexpensive, and effectively standard user interface to help desks (as well as to call centers) and appears to be encouraging more automation in help desk service.

Some common names for a help desk include: Computer Support Center, IT Response Center, Customer Support Center, IT Solutions Center, Resource Center, Information Center, and Technical Support Center.

The above **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 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 propose 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.

3. Request-based semantic case ranking

To assist users in finding answers quickly once a new request arrives 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.

3.1 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 $fm \in S_i$ and $fn \in S_j$, 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 $\{r_1, r_2, \dots, r_k\}$ be the set of K common semantic roles between

fm and fn , $T_m(ri)$ be the term set of fm in role ri , and $T_n(ri)$ be the term set of fn in role ri . Letting $|T_m(ri)| \leq |T_n(ri)|$, we compute the similarity between $T_m(ri)$ and $T_n(ri)$ as

$$rsim(T_m(ri), T_n(ri)) = \frac{\sum_j t_{ij}^m t_{ij}^n}{|T_n(ri)|} \quad (1)$$

Then, the similarity between f_m and f_n is

$$f_{sim}(f_m, f_n) = \frac{\sum_{i=1}^k rsim(T_m(ri), T_n(ri))}{K} \quad (2)$$

4. 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.

4.1 Semantic role parsing

Sentence-level semantic analysis can better capture the relationships between sentences, and we use it to construct the sentence similarity matrix by computing the pair wise sentence similarity.

A semantic role is a description of the relationship that a constituent plays with respect to the verb in the sentence"[10]. Semantic role analysis plays a very important role in semantic understanding. In iHelp, we use NEC SENNA [11] as the semantic role labeler, which is based on Prop Bank semantic annotation [12]. 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 [12] and an illustrative example are shown in Table 1.

Table1: Representation of arguments and 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: extnt	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:	
<i>Sentence:</i> The printer does not work.	
<i>Label:</i> The (Arg0) printer (Arg0) does (-) not (ArgM-NEG) work (rel).	

4.2 Symmetric nonnegative matrix factorization

Once we obtain the similarity matrix of the relevant cases, clustering algorithms need to be performed to group these cases into clusters. In our work, *we propose the SNMF algorithm* to conduct the clustering. It has been shown that SNMF is equivalent to kernel *K-means clustering*.

SNMF is based on a similarity measure between data points, and factorizes a symmetric matrix containing pair wise similarity values (not necessarily nonnegative). The experiment results shows the substantially enhanced clustering quality of SNMF over spectral clustering and NMF. Therefore, SNMF is able to achieve better clustering results on both linear and nonlinear manifolds, and serves as a potential basis for many extensions and applications. we present the formulation of SNMF and argue that it has additional clustering capability compared to NMF and meanwhile has good interpretability offered by nonnegative.

Multi-document summarization aims to create a compressed summary while retaining the main characteristics of the original set of documents. In this paper, we propose a new multidocument Summarization framework based on sentence-level semantic analysis and symmetric non-negative matrix factorization. We first calculate sentence-sentence similarities using semantic analysis and construct the similarity matrix.

Then symmetric matrix factorization, which has been shown to be equivalent to normalized spectral clustering, is used to group sentences into clusters. Finally, the most informative sentences are selected from each group to form the summary.

Algorithm 1 Framework of algorithms for SNMF: $min_{H \geq 0} f(x) = ||A - HH^T||^2_F$

```

1: Input: number of data points n, number of clusters k, n x n similarity matrix A, reduction factor 0 < beta < 1, acceptance parameter 0 < sigma < 1, and tolerance parameter 0 < mu << 1
2: Initialize x, x^(0) ← x
3: repeat
4:   Compute scaling matrix S
5:   Step size alpha = 1
6:   while true do
7:     x_new = [x - alpha Srf(x)]+
8:     if f(x_new) - f(x) <= sigma f(x)^T (x_new - x) then
9:       break
10:    end if
11:    alpha ← beta alpha
12:  end while
13:  x ← x_new
14: until kr^P f(x)k <= mu kr^P f(x^(0))k [15]
15: Output: x

```

In the above Algorithm First, we introduce several notations for clarity. Let $H = [h_1, \dots, h_k]$. A vector x of length nk is used to represent the factorization of H by column, i.e. $x = \text{vec}(H) = [h_1^T, \dots, h_k^T]^T$. For simplicity, functions applied on x have the same notation with functions applied on H , i.e. $f(x) = f(H)$. $[\cdot]^+$ denotes the projection to the nonnegative orthant, i.e. changing any negative element of a vector to be 0. Superscripts denote iteration indices, e.g. $x(t) = \text{vec}(H(t))$ is the iterate of x in the t -th iteration. For a vector v , v_i denotes its i -th element. For a matrix M , M_{ij} denotes its (i, j) -th entry; and $M[i][j]$ denotes its (i, j) -th $n \times n$ block, assuming both the numbers of rows and columns of M are multiples of n . $M \succeq 0$ refers to positive definiteness of M .

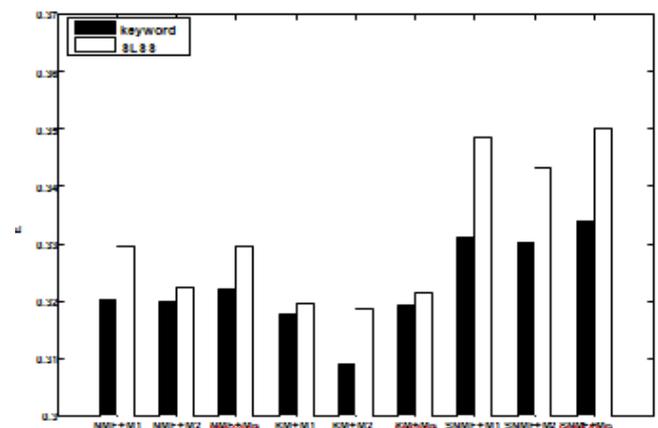


Figure 1: Methods comparison in similarity matrix construction phase.

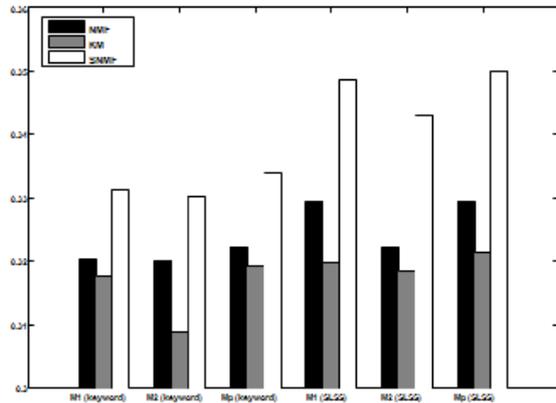


Figure 3: Deferent clustering algorithms

The above fig2 and 3 shows the efficiency of the algorithms Case clustering and SNMF. The results clearly show that no matter which methods are used in other phases, Semantic Role Parser outperforms keyword-based similarity. This is due to the fact that Semantic Role Parser better captures the semantic relationships between sentences.

5. Experimental results

To improve the usability of the system, we proposed sentence-level semantic analysis approach and SNMF clustering algorithm can be naturally applied to the summarization task to address the aforementioned issues.

5.1 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 Lucerne 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.

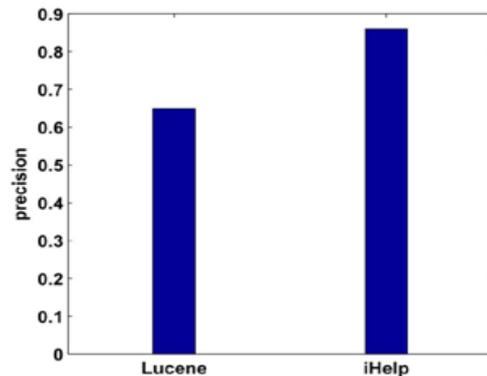


Fig 4. Precision of the retrieved cases.

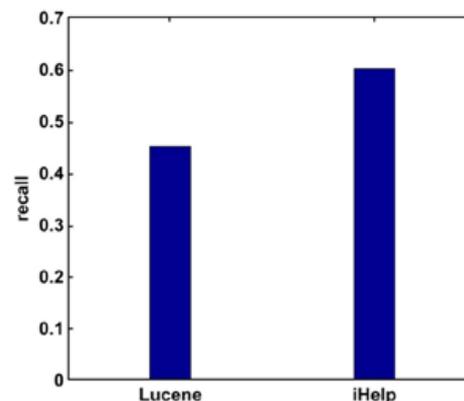


Fig 5. Recall of the retrieved cases.

5.2 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.

5.2.1 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 II shows the average scores of Lucene and iHelp for each request.

Table 2
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

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.

5.2.2 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.
- 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 3
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

Table 3 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. 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.

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. returned to the user.

6. 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.

7. References

- [1] B. K. Giamanco, Customer service: The importance of quality customer service. [Online]. Available: <http://www.ustomerservicetrainingcenter.com>
- [2] R. Agrawal, R. Rantau, and E. Terzi, "Context-sensitive ranking," in *Proc. SIGMOD*, 2006, pp. 383–394.
- [3] S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis, "Automated ranking of database query results," in *Proc. CIDR*, 2003, pp. 888–899.
- [4] K. Chakrabarti, V. Ganti, J. Han, and D. Xin, "Ranking objects based on relationships," in *Proc. SIGMOD*, 2006, pp. 371–382.
- [5] S. Chaudhuri, G. Das, V. Hristidis, and G. Weikum,

- “Probabilistic ranking of database query results,” in *Proc. VLDB*, 2004, pp. 888–899.
- [6]G. Das, V. Hristidis, N. Kapoor, and S. Sudarshan, “Ordering the attributes of query results,” in *Proc. SIGMOD*, 2006, pp. 395–406.
- [7]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.
- [8]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.
- [9]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.
- [10]D. Arnold, L. Balkan, S. Meijer, R. Humphreys, and L. Sadler, *Machine Translation: An Introductory Guide*. London, U.K.: Blackwells-NCC, 1994.
- [11]R. Collobert and J. Weston, “Fast semantic extraction using a novel neural network architecture,” in *Proc. ACL*, 2007, pp. 560–567.
- [12]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