International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

Research Article

iAssist: An Intelligent Assistance System using SLSS, Mixture Model and SNMF

Khanapure V.M[Å*] and Chirchi V.R[Å]

[Å]Dept. of CNE, College of Engineering, Ambajogai, Maharashtra, India

Accepted 05 May 2014, Available online 01 June 2014, **Vol.4, No.3 (June 2014)**

Abstract

World Wide Web is the most useful source of information. A search engine can support only the initial stages of the search process. But, most of the 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, many companies use case-based systems to improve customer service quality. These systems face two challenges: 1) Case retrieval measures: 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: case-based systems return a list of past cases ranked by their relevance to a new request, and customers go through the list and examine the cases one by one to identify their desired cases. The objective of this research is to address these challenges, we develop iAssist - an Intelligent Assistance system, to automatically find problem solution patterns from the past interactions between customers and representatives.

Keywords: *Semantic similarity, Symmetric non-negative matrix factorization, Multi-document summarization.*

1. Introduction

Most of the commercial companies' uses case based helpdesk system to improve customer service quality. These systems uses keyword –matching based ranking for retrieving different cases according to user request and returns a list of past cases ranked by their relevance to new request. It also gives some irrelevant data and user has to search down the list to obtain the desired result. It is difficult to transfer knowledge and experience between customers. Many companies use intelligent assistance systems to improve the quality of customer service. Many case based systems have error level information. So that we have proposed a new algorithm called sentence level semantic similarity calculation. The main objectives of this automatically find the problem solution. The high performance of iAssist benefits from the proposed approaches of ranking, clustering using the mixture language model, symmetric matrix factorization (SNMF), and the request focused multidocument summarization.

A current case based system involves a lot of operations. It is difficult to transfer knowledge and experience between customer and representatives. Many companies use intelligent assistance systems to improve the quality of customer service. Many case based systems mainly suffer from keyword matching technologies and error-level information at the solution time. So we have proposed a new algorithm. The main objectives of this automatically find the problem solution. Given a new customer request, one common scenario of an intelligent assistance system is to find whether similar requests have been processed before. Assistance systems usually use databases to store past interactions (e.g., descriptions of a problem and recommended solutions) between customers and companies.

2. Related Work

2.1 Case-based systems: It is based on keyword matching. This case based system lacks the semantic analysis of customer requests and existing cases (D. Radev et al,2002).Thus new similarity measurement are needed that are able to understand the semantic meaning in the request & past cases (S. Agrawal et al,2003; D. W. Aha et al,2005).

These systems use to retrieve the initial information from the first candidate set and then ask the user to narrow down until few cases remain or the 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 become difficult to measure.

2.2 Database search and ranking: Similarity is measure based on Keyword matching, which have difficulty to understand text deeply (A. Leuski et al,2000). For finding answers quickly once a new request arrives, cases are rank based on semantic importance (D.Wang et al, 2008).In database search, many methods have been proposed to perform similarity search and rank results of a query. However, similar to the case based systems, the similarity is measured based on keyword matching, which have difficulty to understand the text deeply.

*Corresponding author **Khanapure V.M** is a PG student and **Chirchi V.R** is working as Asst Prof.

2.3 Clustering search results: Search results are long list, so it is time consuming process (X. Liu et al,2002).To find the better solution for problem, Online Helpdesk System first cluster the top ranking cases (K. Beyer et al, 1999; D. W. Aha et al,2005).

Since existing search engines often return a long list of search results, clustering technologies are often used in search result organization. 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.

2.4 Document summarization: Information contain in different document often overlap with each other (R. Collobert et al, 2007). Therefore, it is necessary to find an effective way to merge the document while recognizing and removing redundancy (K. Beyer et al, 1999).

Multidocument summarization is the process of generating a summary by reducing documents in size while retaining the main characteristics of the original documents . 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.

3. Proposed Implementation

3.1 Existing 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.

Existing customer service or helpdesk systems were dealing with keyword-matching based ranking scheme for case retrieval and results will be in a list format i.e. these case based systems faces two main challenges :

1) Case retrieval measures: most case-based systems use traditional keyword-matching-basedranking schemes for case retrieval and have difficulty to capturethe semantic meanings of cases and

2) Result representation: mostcase-based systems return a list of past cases ranked by theirrelevance to a new request, and customers have to go through thelist and examine the cases one by one to identify their desired cases**.** Example: Apache Lucene keyword based text ranking engine.

3.2 Proposed System

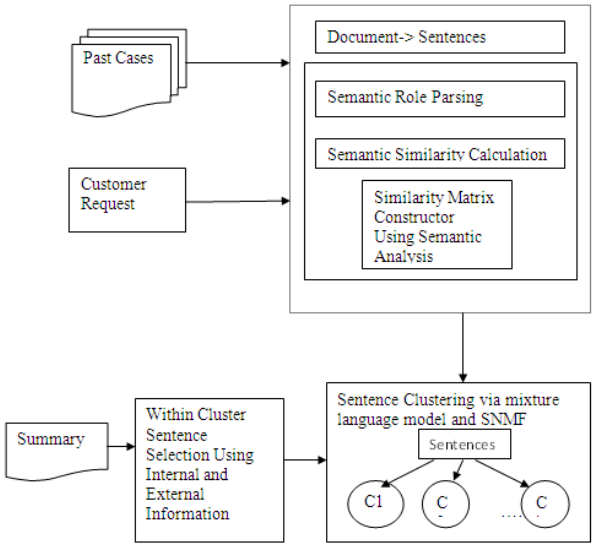

Figure 1: Proposed Intelligent Assistance system.

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. In iAssist, we use NEC SENNA as the semantic role labeler, which is based on PropBank semantic annotation. It is fast Semantic Role Labeling (SRL) tool. SENNA is software distributed under a non-commercial license, which outputs a host of Natural Language Processing (NLP) predictions: part-of-speech (POS) tags, chunking (CHK), name entity recognition (NER), semantic role labeling (SRL) and syntactic parsing (PSG).

SENNA is fast because it uses a simple architecture, self-contained because it does not rely on the output of existing NLP system, and accurate because it offers state-of-the-art or near state-of-the-art performance. Semantic role labeling, sometimes also called shallow semantic parsing, is a task in natural language processing consisting of the detection of the semantic arguments associated with the predicate or verb of a sentence and their classification into their specific role.

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.

We discover the semantic relations of terms in the same semantic role using WordNet. WordNet is lexical database for the English language. It group English words into sets of synonyms. 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." Then, in the case-ranking module, the past cases are ranked based on their semantic importance to the preprocessed input request. Other than searching and ranking the relevant cases, iAssist 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.

4. Semantic Similarity Matrix Construction

After removing stemming and stopping words, we trunk the documents in the same topic into sentences. Simple Word-matching types of similarity such as cosine can not faithfully capture the content similarity. Also the sparseness of words between similar concepts make the similarity metric uneven. Thus, we perform semantic role analysis on sentences and propose a method to calculate the semantic similarity between any pair of sentences.

4.1 Sentence-level semantic analysis (SLSS)

A semantic role is defined as "a description of the relationship that plays with respect to the verb in the sentence". Each verb in the sentences is labelled with Argument and the verb which is labelled is called "frame". Input to the SLSS algorithm is sentences Si and Sj. Assign labels to each verb in the sentences using Semantic role labler.After assigning label calculate the common semantic roles WordNet. Then to find role similarity between Tm(ri) and Tn(ri) as

$$rsim(Tm(ri), Tn(ri)) = \frac{\sum_j tsim(t_{ij}^m, ri)}{|Tn(ri)|} \quad (1)$$

Then, the similarity between *fm* and *fn* is

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

Therefore, the semantic similarity between Si and Sj can be calculated As follows:

$$Sim(S_{i,} S_j) = \max_{f_m \in s_{i,} f_n \in s} fsim(f_{m,} f_n) \quad (3)$$

where each similarity score is between zero and one.

5. Mixture Language Model and 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.

5.1 Mixture Language Model

Mixture language model is used to measure the similarity between documents while filtering out the general and common information from the request. Mixture model measure is based on a novel view of how relevant documents are generated. We can also view it as a language model with a smoothing algorithm designed specifically for our task.

Algorithm Mixture Model( )

1. Input : number of data points n. n*n similarity matrix w
2. Initialization: double r,prob,x,y=0
3. Compute thetaE,thetaT,thetaD
if(synsets.length > 0 ||GEWords.contains(alphaStr))
thetadE.add(alphaStr);
else if(query.contains(alphaStr))
thetadT.add(alphaStr);
else
thetadD.add(alphaStr);
4. Calculate Probability prob
prob = (lmdaE*(tfwiE/tfwjE)) + (lmdaT*(tfwiT/tfwjT)) + (lmdaD*(tfwiD/tfwjD));
5. Compute relevance r
r-=x*Math.log(y/x);
6. Output: r

5.2 SNMF

We propose a new multi-document summarization framework based on sentence-level semantic analysis (SLSS) and symmetric non-negative matrix factorization(SNMF). SLSS is able to capture the semantic relationshipsbetween sentences and SNMF can divide the sentences into groups for extraction. It has been shown that SNMF is equivalent to kernel K-means clustering and is a special case of trifactor NMF. 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. These results demonstrate the clustering ability of SNMF.

Algorithm SNMF( )

1. Input Sentence pair wise similarity matrix W
2. Initialize H,H=1
3. Compute the norm of Matrix
$$\min_{H \geq 0} J = \| W - HH^T \|^2$$
4. Check the KKT condition
$$\text{If}((-4WH + 4HH^T H)_{ij} H_{ij} = 0)$$

$$H_{ij} \leftarrow H_{ij} - \in_{ij}$$

Else

$$H_{ij} \leftarrow \frac{1}{2}[H_{ij}(1+\frac{(WH)_{ij}}{(HH^{T}H)_{ij}}]$$

5. Output H

5.3 Multidocument Summarization

After grouping the sentences into clusters by the SNMF algorithm, in each cluster, we rank the sentences based on the sentence score calculation . The score of a sentence measures how important a sentence is to be included in the summary.

Algorithm Multidocument_Summarization( )

1. Input : Cluster document
2. Initialize : lmd=0.7
3. Compute f1sim
F1sim=f1sim+snmf.w[x][y];
F1sim=f1sim /double(k-1);
4. Compute f2sim
F2sim=Sim(Si,request)
5. Calculate Score
Score=(lmd*f1sim)+((1-lmd)*f2sim)

6. Results and Discussions

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

6.1 Comparison of cases

In this set of experiments, we randomly select five questions from different categories and manually label the related cases for each question. Then, we examine the top 10 retrieved cases by keyword-based Lucerne and our iAssist system, respectively.

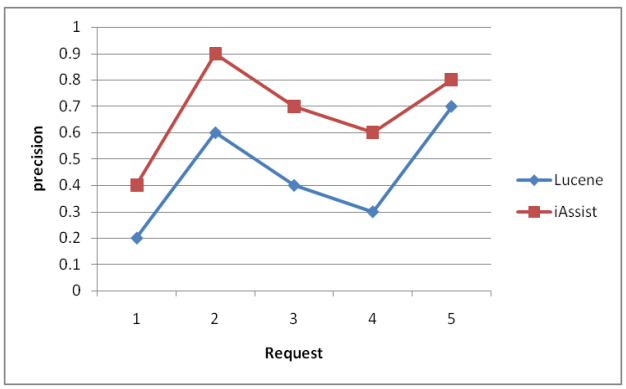

Figure 2: Precision of the retrieved cases.

Figure 2 and 3 show the average precision and recall of the two methods. The high precision of iAssist demonstrates that the semantic similarity calculation can better capture the meanings of the requests and case documents. Since we only look at the top 10 retrieved cases while some of the cases may have more than 20 relevant cases, the recall is also reasonable and acceptable.

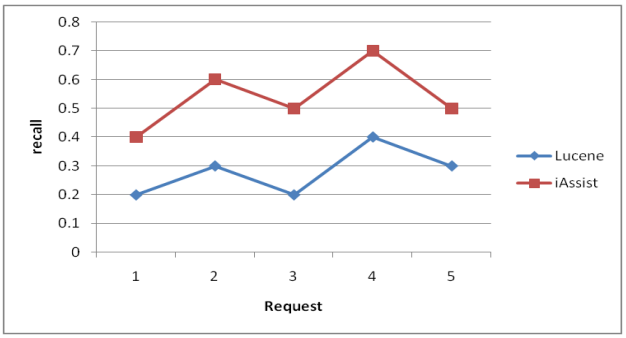

Figure 3: Recall of the retrieved cases.

6.2 Comparative Study

6.2.1 Sentence-level semantic similarity and the traditional keyword-based similarity calculation

For better evaluating our proposed method, we implement alternative solutions for each phase of the summarization procedure as listed in Table 1.

Table1: Different methods implemented in each phase.

Phase	Proposed Method	Alternative 1	Alternative 2
Similarity Measurement	Semantic Similarity (SLSS)	Keyword based Similarity	Keyword based Similarity
Clustering Algorithm	SNMF	K-means (KM)	NMF
Within-Cluster Sentence Ranking	$M_p=\lambda F_1(S_i)+(1-\lambda) F_2(S_i)$	$M_1=F_1(S_i)$	$M_2=F_2(S_i)$

In Table 1, the keyword-based similarity between any pair of sentences is calculated as the cosine similarity. The parameter λ in Mp is set to 0.7 empirically.

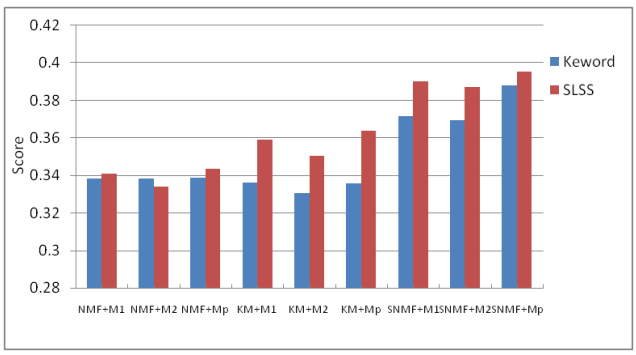

Figure 4: Methods comparison in similarity matrix construction phase.

We compare the proposed sentence-level semantic similarity with the traditional keyword-based similarity

calculation. In order to better understand the results, we use Figure 4 to visually illustrate the comparison. The results clearly show that no matter which methods are used in other phases, SLSS outperforms keyword-based similarity. This is due to the fact that SLSS better captures the semantic relationships between sentences.

6.2.2 Comparing redundancy measure

We proposed five measures for assessing the redundancies of a new document with respect to a previously seen stream of documents. A redundancy score was calculated for each relevant document dt, based on the relevant documents di that preceded it in the document stream. The results are shown in Figures 5 in the form of average Recall-Precision graphs over the set of redundant documents. The mixture model approach was consistently more accurate than the other two smoothing algorithms on both corpora. It was also about as effective as the cosine similarity measure.

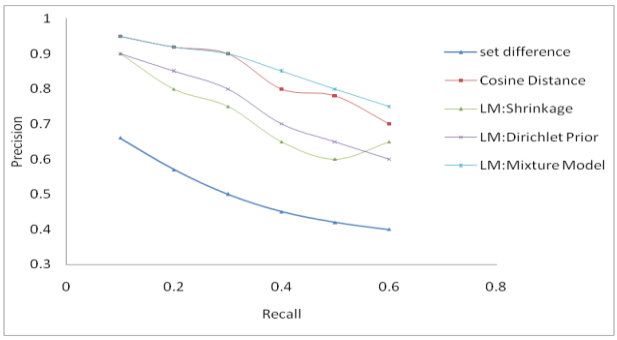

Figure 5: Comparing redundancy measures.

6.2.3 Different Clustering Algorithms

Now we compare different clustering algorithms in Figure 6. We observe that our proposed SNMF algorithm achieves the best results.

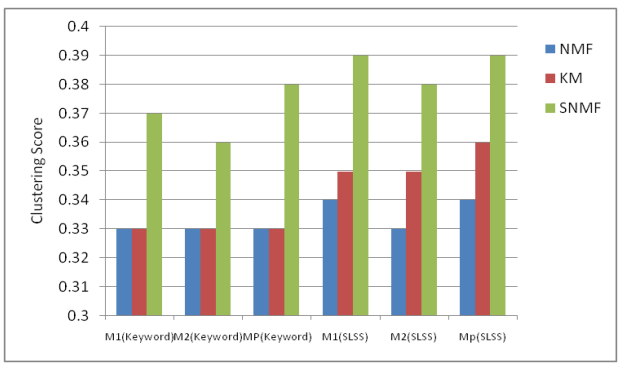

Figure 6: Different clustering algorithms.

K-means and NMF methods are generally designed to deal with a rectangular data matrix and they are not suitable for clustering the similarity matrix. Our SNMF method, which has been shown to be equivalent normalized spectral clustering, can generate more meaningful clustering results based on the input similarity matrix.

6.2.4 Discussion on Parameter

Figure 7 demonstrate the influence of the weight parameter λ in the within-cluster sentence selection phase. When λ = 1 (it is actually method M1), only internal information counts, i.e. the similarity between sentences. And λ= 0 represents that only the similarity between the sentence and the given topic is considered (method M2). We gradually adjust the value of λ, and the results show that combining both internal and external information leads to better performance.

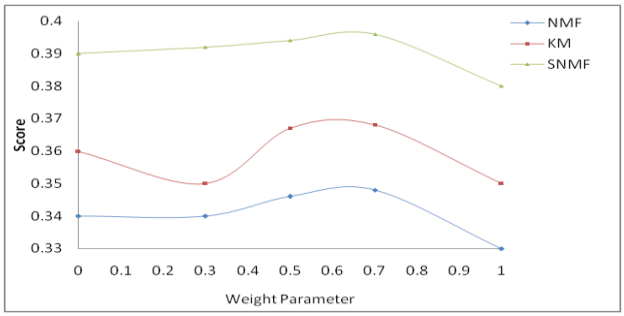

Figure 7: Study of Weight Parameter λ.

6.2.5 Lucene and iAssist

Example1. Can I update my iPod music collection from more than one computer: The full representation of the abstract arguments of an illustrative example is shown in Table 2. Table 3 shows the top-ranking case samples retrieved by Lucene and Multidocument summarization. For ranking results, we find that Lucene takes the word "iPod", "Computer" as the keyword and return many cases related to them as the search result in list format. Obviously they are not what the customer want.

Table 2: Representation of arguments of an illustrative example

Can - S-AM-MOD I - S-A0 update - S-V my - B-A iPod - I-A1 music - I-A1 collection - E-A1	from - B-AM-MNR more - I-AM-MNR than - I-AM-MNR one - I-AM-MNR computer - E-AM-MNR
Example: Sentence: Can I update my iPod music collection from more than one computer Label: Can[S-AM-MOD] I[S-A0] update[S-V] my[B-A1] iPod [I-A1] music[I-A1] collection [E-A1] from[B-AM-MNR] more[I-AM-MNR] than[I-AM-MNR] one[I-AM-MNR] computer[E-AM-MNR]	

Table 3: Top ranking case samples by lucene and iAssist

Request	Can I update my iPod music collection from more than one computer
Lucene	Top Ranking Cases iPod is compatible with computers running on Mac OS X and PCs running on Windows 2000 or Windows XP

Multidocument summarization	Top Ranking Cases Yes. When you first connect iPod to your computer, iPod find that computer as its "home" computer. Each time you connect, iPod downloads the music library stored on it. This means that you cannot transfer music, automatically or manully, from your iPod to computer, and you cannot use iPod to copy a music library from one computer to another.

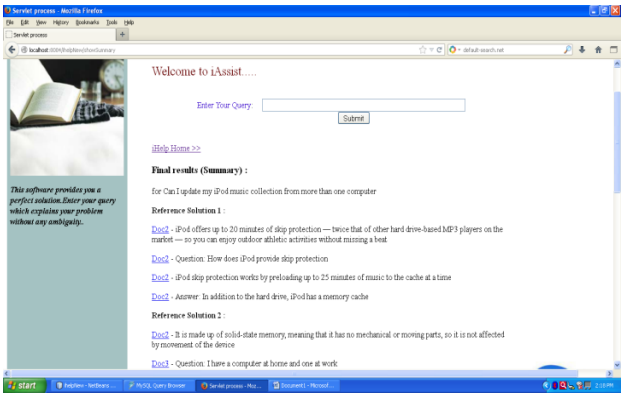

Figure 8: Screenshot of iAssist.

In our proposed system, iAssist provides the semantic meaning of request. We first calculate sentence-sentence similarities using semantic psychoanalysis and construct the similarity matrix. Then mixture language model and symmetric matrix factorization is used to group sentences into clusters for extraction. Finally, the informative sentences are selected from each group to form the summary.

Conclusions

To improve the usability of the system, we perform multidocument summarization to generate a brief summary for each case cluster. In this paper we search and rank the existing cases according to their relevance to users' requests in a semantic way and we provide a better result representation by grouping and summarizing the retrieved past cases to make the system fully functional and usable. The high performance of iAssist based on cluster using sentence-level semantic analysis (SLSS), mixture model and symmetric non-negative matrix factorization (SNMF).

References

S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis (2003), Automate ranking of database query results,*CIDR*, pp. 888–899.

D.Wang, S. Zhu, T. Li, and C. Ding (2008), Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization, *SIGIR*, pp. 307–314.

K. Beyer, J. Goldstein, R. Ramakrishna, and U. Shaft (1999), When is nearest neighbor meaningful, *ICDT*, pp. 217–235.

D. Radev, E. Hovy, and K. McKeown (2002), Introduction to the special issue on summarization, *Comput. Linguist.*, vol. 28, no. 4, pp. 399–408.

D. W. Aha, D. Mcsherry, and Q. Yang (2005), Advances in conversational case-based reasoning, *Knowl. Eng. Rev.*, vol. 20, no. 3, pp. 247–254.

R. Agrawal, R. Rantzau, and E. Terzi (2006), Context-sensitive ranking, *SIGMOD*,pp. 383–394.

A. Leuski and J. Allan (2000), Improving interactive retrieval by combining ranked list and clustering, *RIAO*, pp. 665–681.

X. Liu, Y. Gong, W. Xu, and S. Zhu (2002), Document clustering with cluster refinement and model selection capabilities, *SIGIR*, pp. 191–198.

R. Collobert and J.Weston (2007), Fast semantic extraction using a novel neural network architecture, *ACL*, pp. 560–567

M. Palmer, P. Kingsbury, and D. Gildea (2005), The proposition bank: An annotated corpus of semantic roles, *Comput. Linguist.*, vol. 31, no. 1, pp. 71–106

C. Fellbaum (1998), WordNet: An Electronic Lexical Database. *Cambridge, MA: MIT* Press.

X. Liu, Y. Gong, W. Xu, and S. Zhu (2002), Document clustering with cluster refinement and model selection capabilities, *SIGIR*, pp. 191–198

D. Radev, E. Hovy, and K. McKeown (2002), Introduction to the special issue on summarization, *Comput. Linguist*., vol. 28, no. 4, pp. 399–408

Shi and J. Malik (2000), Normalized cuts and image segmentation, *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 22, no. 8, Aug. 2000

D.W.Aha, D. Mcsherry, and Q. Yang (2005), Advances in conversational case-based reasoning, *Knowl. Eng*

D. Bridge, M. H. Goker, L. Mcginty, and B. Smyth (2005), Case-based recommender systems, *Knowl. Eng. Rev.*, vol. 20, no. 3, pp. 315–320

Dingding Wang, Tao Li, Shenghuo Zhu, and Yihong Gong (2011), iHelp: An Intelligent Online Helpdesk System, *IEEE TransactionS* 2011