

# Answer Clustering and Fusion in a User-Interactive QA System

Feng Min, Liu Wenyin, and Wei Chen  
Department of Computer Science  
City University of Hong Kong  
83 Tat Chee Avenue  
Kowloon, Hong Kong SAR, China  
{emmwcity, csliuwy, wchen2}@cityu.edu.hk

## Abstract

*The need of answer clustering and fusion in a user-interactive question answering (QA) system is identified and its user interface and enabling technology are presented in this paper. This function aims to help a user to efficiently browse all the answers and find the correct answer to a specific question by clustering answers into groups and providing a representative (fused) answer for each group. The clustering approach proposed in this paper includes a measurement of semantic similarity between answers and an incremental soft-moVMF algorithm. An answer fusion method is proposed, which uses concept vector and authority of data sources to extract the summary for the answers in each cluster. Experiments and user studies show that the UI and the methods are effective.*

## 1 Introduction

The Web is becoming an ideal source of answers to various domain-independent questions not only because of the tremendous amount of information that is now available online, but also because there are so many people connecting to Internet, who have capabilities to provide the answers to the questions. Recently, some new-style virtual communities appear on the Web, such as Google Answers [3], Sina iAsk [4] and Baidu Zhidao [1], which provide an interactive platform for users to post questions and answers. Actually, this kind of systems can be used as a collaborative approach to knowledge acquisition [23][24]. As more knowledge is accumulated and well represented, more accurate automatic QA is possible. Although these systems serve for a new objective of user-interactive QA, their user interfaces are not suitable for browsing a large set of answers to a specific question. In these systems, there are always some questions

with so many similar or redundant answers posted that users have to spend much time to browse them and finally find it difficult to obtain a complete and correct answer especially when the answers are inconsistent.

In this paper, we design a new user interface for our user-interactive QA system, CuteAid [2], which facilitates users to efficiently browse all the answers to a specific question by clustering similar/redundant answers from different users into groups and provide a representative answer for each group. The enabling technologies are also presented, which include a semantic answer vector, an incremental soft-moVMF clustering algorithm [6] and a hybrid answer fusion approach. The incremental soft-moVMF clustering algorithm is an incremental clustering algorithm based on the soft-moVMF clustering algorithm [6]. Our fusion approach integrates the conceptually similar answers in the same cluster by extracting the summary using concept vectors [9]. During the procedure of fusion, the authority of data source is introduced to indicate the reliability of answers.

The structure of this paper is described as follows. Related work is reviewed in Section 2. In Section 3 and 4, we present the proposed clustering and fusion approaches, respectively. The proposed user interface supported by our approaches is presented in Section 5. Section 6 illustrates experimental studies. Finally, we draw the conclusions and discuss future work in Section 7.

## 2 Related work

Existing user-interactive QA systems have a common problem: their user interfaces are not suitable for browsing a large set of posts in a thread. In these systems, there are always some questions with so many similar or redundant answers posted that users have to spend much time to browse them and finally find it difficult to obtain a com-

plete and correct answer especially when the answers are inconsistent. The similar problem has already been encountered in the field of web search. To solve such kind of problem, Grouper [20], a clustering interface to the results of the HuskySearch meta-search engine, is designed to dynamically group the search results into clusters labeled by phrases extracted from the snippets. It, to some extent, makes the search engines more convenient.

The document clustering technologies have been widely used in many web-based applications to facilitate users browsing information from different sources. They have been extensively investigated as a methodology for improving document search and retrieval for a long time. Various methods have been proposed in this field, among which probabilistic model-based clustering is particularly efficient, as each iteration is linear to the size of the input. There are three major probabilistic models suitable for document clustering [22]: multivariate Bernoulli, multinomial and von Mises-Fisher (vMF), among which the vMF models [6] provide the best performance at relatively low cost. The question, how to choose the final number of clusters is an old and important problem, to which a globally optimal solution is still in demand. Banerjee and Langford (2004) proposed a new objective criterion for choosing the number of clusters based on the PAC-MDL bound [7]. Although the moVMF-based algorithms always obtain the right number of the clusters under the criterion, it only works in a semi-supervised setting.

Definition of the similarity between the answers is a basic problem to handle before clustering. Recently, there are many literatures on measuring the similarity between long texts [5][12][14][16], which are based on word co-occurrence, corpus or descriptive features. However, since the answers are often short texts or snippets, the methods mentioned above are not suitable in answer clustering. In fact, there are relatively few publications about how to measure the similarity between short texts [15][18]. The works on the measurement of similarity between short texts can be classified into two major categories: semantic methods and web search-based methods. The semantic methods try to measure the semantic similarity between short texts by using the semantic similarity between words, which is derived by dictionary or WordNet [11], while the web search based methods first leverage web search results to provide greater context for the short texts before measuring the similarity. In this paper, a semantic method is used to measure the similarity between answers.

Most recent QA systems only provide a set of partial results, which are possibly incomplete answer related to a question with certain probability. Answer fusion can be seen as a procedure which can transform, integrate and merge diversiform answers from different data sources to generate a new answer. Although answer fusion has not

formally been a research direction in the QA research field, a few systems begin to include the answer conflict analysis process to merge the answers coming from various information sources to create a single fused answer. The method proposed by Dumais et al. follows the observation that the correct answer is always an entity with high frequency [10]. The test results in RECT12 show that the aggregation-based answer selection method in LAMP system[21] outperforms the common methods based on individual or redundancy. However, the answer selection methods can only provide partial results, too. They do not address the following two critical issues: first, these methods do not deal with the possible inconsistencies among the answers; second, users cannot participate in the answer selection process.

### 3 Answer clustering

Clustering has been widely used to discover “latent concepts” in sets of unstructured text documents, and to summarize and label such collections [9]. In this section, we present how to partition high-dimensional and sparse answer data sets into several irrelevant conceptual categories. Moreover, we study how to determine the final number of clusters and how to reduce the computation in our answer clustering approach.

#### 3.1 Semantic Answer Vector

Before presenting the clustering algorithm in our system, we first introduce the semantic vectors to represent the answers. The similarity between two answers is, thus, equivalent to the dot product of their semantic vectors. Given a question  $Q$  and its answers  $A_1, A_2, \dots, A_n$ , the joint word set of answers is defined:

$$S = \bigcup_{i=1}^n A_i - Q.$$

The joint word set contains all the distinct words from the answers excluding the words from the question and can be viewed as the semantic information for the answers. The semantic vector of each answer, denoted by  $a_i$ , is thus constructed by the use of the joint word set. The dimension of the semantic vector, denoted by  $m$ , is equivalent to the number of the words in the joint word set and each element of the semantic vector corresponds to a word in the joint word set. The value of the  $i$ -th element of the semantic vector, denoted by  $a_{i,j}$ , is not only determined by the semantic similarities between the corresponding word and the words in the answer, but also depends on the document frequencies of the words in the joint word net:

$$a_{i,j} = \log_2 \left( \frac{n}{\sum_{l=1}^m [sim(w_j, w_l) * df(w_l)]} \right)$$

**Table 1. Matrix for 6 clusters of Classic.**

cluster ID	cran	med	cacm	cisi
1	<b>1376</b>	5	103	0
2	0	0	<b>119</b>	0
3	1	<b>1008</b>	28	3
4	10	8	<b>1366</b>	52
5	2	0	<b>1122</b>	1
6	9	12	465	<b>1404</b>

$$* \sum_{w \in A_i} sim(w, w_j), \quad (1)$$

where  $df(w_i)$  indicates the number of answers containing word  $w_i$ , and  $sim(\cdot, \cdot)$  denotes the similarity between two words. If the value  $a_{i,j}$  derived from the formula is negative, we just set it into zero. In our system, edge counting-based method [17] is adopted to compute the word similarity and WordNet [11] servers as the dictionary.

### 3.2 Soft-moVMF Clustering

Document clustering has been developed for many years, and a large number of text clustering algorithms have been proposed. Although the graph partitioning algorithm [13] performs the best in document clustering [22], it is also the most computationally expensive, and can only deal with the hard assignment. Hence, to fulfill the *overlap* and *speed* requirement and acquire high quality results, we adopt the soft-moVMF algorithm [6] in our system, which is not presented here due to the limited space.

### 3.3 Estimation of the Number of Clusters

Although the soft-moVMF algorithm seems quite straightforward, there is still one critical issue to be dealt with in practice: How to choose the final number of the answer clusters? Most existing partitioning algorithms also face the same problem.

Table 1 shows the resulting matrix from 6 clusters of the Classic corpus [13], which actually contains 4 natural groups of documents. Because the number of clusters is a bit larger than the number of natural groups, the group CACM is splitted into several similar clusters, the centroid vectors of which are very close in cosine similarity. This is a common result when the algorithm is required to produce a greater than the natural number of clusters.

Hence, our system initially generates a larger number of clusters than natural. After clustering, the two closest clusters are merged to reduce the cluster number. Repeat clustering and merging until the inter-cluster distances are large enough. The detailed algorithm combined with incremental clustering algorithm is described in the next section.

### 3.4 Incremental Soft-moVMF Clustering

For an online system, *speed* is one of the most important factors. The time complexity of soft-moVMF algorithm is  $O(knm)$ , where  $m$  is the number of iterations used in the clustering process and could be very large if we use our estimation algorithm to determine the final number of clusters. Such complexity is unacceptable for an online system. Therefore, we must reduce the number of iterations.

Generally, the times of a question being browsed is always much more than its number of answers. Hence, our system does clustering when a new answer comes up and saves the results into the database. Since the clusters of answers grow gradually, we propose the incremental soft-moVMF algorithm to reduce the iterations of each re-clustering. The algorithm is described as follows:

- Step 1. Load the parameters  $\{c_j\}_{j=1}^k$  and  $\{\kappa_j\}_{j=1}^k$  of existing clusters  $\{\pi_j\}_{j=1}^k$ .
- Step 2. Compute  $\max_{1 \leq j \leq k} (a_{new}^T c_j)$ , where  $a_{new}$  is the vector of the new answer.
- Step 3. If  $\max(a_{new}^T c_j) < \theta$ , then add a new cluster  $\pi_{new}$  into existing clusters, and set its centroid vector  $c_{new} = a_{new}$ ; else, just keep the existing clusters.
- Step 4. Re-cluster the answers using the soft-moVMF algorithm.
- Step 5. Calculate  $(j, l) = \arg \max(c_j^T c_l), 1 \leq j, l \leq k$ .
- Step 6. If  $c_j^T c_l \geq \theta$ , combine  $\pi_j$  and  $\pi_l$  into the new cluster  $\pi_{new}$ , decrease  $k$  by 1, and go to step 4.  $\theta$  is a threshold value between 0 and 1.
- Step 7. Return  $\{\pi_j\}_{j=1}^k$ .

Because a single new answer will not affect too much on the structure of the existing answer set, the number of iterations in each re-clustering is supposed to be small, and the time of each re-clustering is accordingly reduced. Moreover, the incremental algorithm assures that the initial number of clusters is larger than the natural number. This is essential for our method to achieve a correct final number of clusters.

## 4 Answer fusion

In order to facilitate a user determining which answer cluster interests him, a summary must be created for each cluster. In this section, we present our fusion approach, which provides concise and accurate descriptions of the clusters by extracting the summary using concept vectors [9].

## 4.1 Authority of Data Source

Authority of a data source is usually an important attribute of the data from different sources, and implies the reliability of the data. Our system considers the authority of a user as a factor in the evaluation of his answers, based on the hypothesis that the user who has higher authority will provide more complete, accurate and correct answers. In our system, the user logs, which contains the questions and answers posted by each user and the inter-actions among each others are recorded to compute the user authority [8].

## 4.2 Answer Summarization

Answer summarization in our system is based on the concept vector [9], which is localized, sparse, and tends to be orthogonal. The concept vector in our system is defined as follows:

$$s_j = \frac{\sum a_i}{\|\sum a_i\|}, P(\pi_j|a_i) > \omega, 1 \leq i \leq n, 1 \leq j \leq k, \quad (2)$$

where  $\omega$  is the threshold to determine whether  $a_i$  belongs to  $\pi_j$ . Each element in the concept vector can be considered as the significance factor of the corresponding word. Therefore, the significance factor of a sentence in cluster  $\pi_j$  can be calculated by:

$$SF = authority * \sum_{w_l \in sentence} s_{j,l}, \quad (3)$$

where *authority* is the normalized authority [8] of the user who posted the sentence,  $s_{j,l}$  is the  $l$ -th element in concept vector  $s_j$ , and  $w_l$  is the word indexed by  $l$ . Our system selects 10% (at most 5) most significant sentences as the summary for each cluster.

## 5 USER INTERFACE

We implement the proposed answer clustering and fusion approach in our user-interactive QA system-CuteAid [2]. CuteAid is a special kind of Web community for users to interactively post and browse questions/answers. The proposed techniques are to provide the correct and concise answers directly by eliminating redundant and inconsistent content in answers. Supported by this approach, we also implement a UI in our QA system to facilitate users' browsing and understanding of large sets of answers.

The developed UI presents a novel layout of answers to a specific question in our QA system, in which a user can quickly finish browsing all possible answers and find the correct answer. By default, a user browses a thread of answers with the normal view, in which the answers are simply sorted by their submission time. Each answer is labeled

with the number of its similar answers, which is equal to the number of answers in its related cluster. If the number of the answers about a question is so large that he cannot quickly finish browsing the answers, he may choose the cluster view, which displays the answers in clusters. Each cluster is summarized by the number of answers it contains and a answer summary based on concept vector. A cluster can be expanded to show all detailed answers in the cluster. If the cluster summary satisfies the user's requirement, he does not need to spend more time to browse all the answers in this cluster. Otherwise, he can expand the related cluster to see every answer in the cluster. Our novel UI provides great convenience to users and help them quickly browse and understand the entire set of answers, as we can see from the evaluation in the next section.

## 6 Experiments and user studies

We present an empirical study of the proposed incremental soft-moVMF algorithm on several benchmark datasets and several user studies on our clustering-based UI.

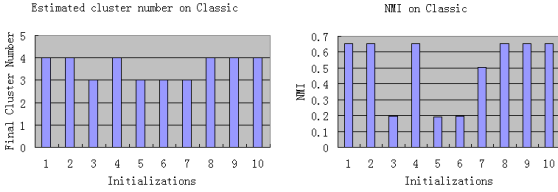
### 6.1 Clustering Evaluation

The datasets that we use for the clustering experiments are carefully selected to represent some typical clustering problems. We also create various subsets of the datasets to model some particular clustering scenarios which often occur in an online user-interactive system. We draw our data from CMU 20-newsgroups and the CLUTO test data [13]. The description of test data is omitted due to the limited space.

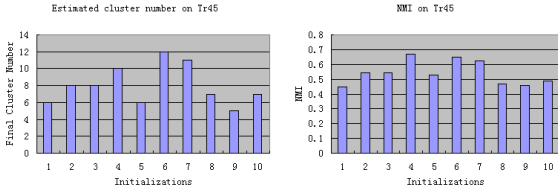
In our experiments, we use normalized mutual information (NMI) as the evaluation criterion [19], which gives the amount of statistical similarity between the clusters and class labels. All results reported here are averaged over 10 runs. All algorithms are started with the same random initialization to ensure fairness of comparison.

In Figure 1 and Figure 2, we present the estimated number of clusters and NMIs for our estimation method for the number of clusters on **Classic** and **Tr45** for 10 runs with different initializations. As a simple heuristic, we set the threshold  $\theta = 0.33$ . Since **Classic** is an easy dataset of 4 well-separated groups of documents, the result of estimated number of clusters is rather close to the natural number of clusters. Over the 10 runs, our method achieves the final number of clusters 4 among 6 runs and 3 among 4 runs. **Tr45** is a relatively difficult, unbalanced dataset of 10 groups of documents. The average of estimated number of clusters achieved by our method is 8, and the standard deviation is 2.31.

Table 2 shows the average performance over 10 runs for the incremental soft-moVMF algorithm and the soft-



**Figure 1. Estimated number of clusters and NMIs on Classic: 10 runs with different initializations**



**Figure 2. Estimated number of clusters and NMIs on Tr45: 10 runs with different initializations**

moVMF algorithm on 7 different datasets, where the time in the incremental soft-moVMF column is the average time of each re-clustering run. All the runtime results are recorded on a 3.0 GHz PC running Windows XP with 1 GB memory and reflect only the clustering time. **CompNews** and **Hitech** are very difficult to separate, on which both algorithms achieve poor results. **Classic**, **Hitech**, and **K1b** are so huge that both algorithms spend a lot of time. However, in a user-interactive QA system, the number of answers to a specific question is usually less than one hundred, and hence the clustering time will be very short. Overall, our incremental algorithm requires just 35.94% the time required by the soft-moVMF algorithm while it loses little NMI in most datasets.

## 6.2 Answer Summarization Evaluation

In this section, we present an interesting user study to evaluate the answer summarization. We prepare ten clusters of short passages extracted from the Web, ten passages in each cluster. There are averagely 5.7 sentences in each passage. Each cluster is summarized by selecting 5 most significant sentences using our summarization method. 10 participants are also required to select five most significant sentences that they think from each cluster. We compare the results from humans and our method using the follow-

**Table 2. NMI and run time of the incremental soft-moVMF algorithm and the soft-moVMF algorithm on CompNews, SciNews, DiffNews, Classic, Hitech, K1b, and Tr45.**

Dataset	Incremental		Soft-moVMF	
	NMI	Time(s)	NMI	Time(s)
CompNews	0.17	1.10	0.24	2.78
SciNews	0.40	1.16	0.43	3.54
DiffNews	0.38	0.58	0.36	1.62
Classic	0.51	39.00	0.55	64.33
Hitech	0.10	22.34	0.27	81.33
K1b	0.65	24.71	0.64	63.23
Tr45	0.55	2.12	0.59	13.01

ing formula:

$$sim = \frac{1}{n_p} \frac{1}{n_c} \sum_{i=1}^{n_p} \sum_{j=1}^{n_c} \frac{|S_{i,j} \cap S_j^c|}{|S_{i,j} \cup S_j^c|}, \quad (4)$$

where  $n_c$  is the number of cluster,  $n_p$  is the number of participants,  $S_{i,j}$  is the significant sentence set selected by the  $i$ -th participant from the  $j$ -th cluster, and  $S_j^c$  is the significant sentence set selected by our summarization method from the  $j$ -th cluster.

The similarity (defined in Eq(4)) between the results from human and our method is 0.375, which means average 2.73 common sentences between the results from humans and our method for each cluster.

## 6.3 User Study on Clustering-based UI

In this section, we compare our clustering/fusion based UI with traditional list-based UI. We design an experiment to study the effect and efficiency of users' browsing.

We prepare four threads of questions and answers and their corresponding comprehension tests (full mark is 10 points) for the user study, each thread having both normal version and clustered version. We randomly distribute these threads associated with their corresponding comprehension tests among these participants, each of whom is given only one version of a thread and required to search the answers of the comprehension questions in the given thread. The average cost time and the comprehension result is also shown in Table 3. From the result, we can see that our new UI saves users' time by 22.3% for browsing all the answers and also increase their comprehension of these answers by 26.3% in the same time. Furthermore, the comprehension per unit time is greatly increased by 62.3%, which means our new UI can significantly improve the efficiency of web information browsing.

**Table 3. Comparison of the two UIs.**

	Score	Time(min)	Score/Time
Traditional	7.27	13.72	0.53
Clustering-based	9.18	10.65	0.86

## 7 Conclusion

In the paper, we have proposed a new UI and its supporting method to cluster and fuse answers in our user-interactive QA system to facilitate users' reading of the answers. The incremental soft-moVMF clustering algorithm proves to be effective and efficient in an online system. Our experiments also show that, answer fusion is very effective to enhance answer completeness and readability. In the future, we will conduct further study on how to take advantage of the structural relationship between the question and its answers to improve answer clustering and fusion. We believe that our idea can also be easily applied to other virtual communities and QA systems, and we need to test it in various cases.

## 8 Acknowledgement

The work described in this paper was supported by the China Semantic Grid Research Plan (National Grand Fundamental Research 973 Program, Project No. 2003CB317002).

## References

- [1] Baidu Zhidao. <http://zhidao.baidu.com/>.
- [2] CuteAid. <http://www.cs.cityu.edu.hk/QA/>.
- [3] Google Answer. <http://answers.google.com/answers/>.
- [4] Sina iAsk. <http://iask.sina.com.cn/>.
- [5] J. Allen. *Natural language understanding (2nd ed.)*. Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 1995.
- [6] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6:1345–1382, 2005.
- [7] A. Banerjee and J. Langford. An objective evaluation criterion for clustering. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 515–520, Seattle, WA, USA, 2004. ACM Press.
- [8] W. Chen, Q. Zeng, and W. Liu. A user reputation model for a user-interactive question answering system. Accepted by SKG'06 workshop.
- [9] I. Dhillon and D. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1-2):143–175, 2001.
- [10] S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng. Web question answering: is more always better? In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 291–298, Tampere, Finland, 2002. ACM Press.
- [11] C. Fellbaum. *WordNet: A lexical database for English*. MIT Press, Cambridge, MA, 1998.
- [12] V. Hatzivassiloglou, J. Klavans, and E. Eskin. Detecting text similarity over short passages: exploring linguistic feature combinations via machine learning. In *EMNLP '99: Proceedings of empirical methods in natural language processing and very large corpora*, MD, USA, 1999.
- [13] G. Karypis. CLUTO - A clustering toolkit. Technical report, Dept of Computer Science, University of Minnesota, 2002. <http://www-users.cs.umn.edu/karypis/cluto/>.
- [14] T. Landauer, D. Laham, and P. Foltz. How well can passage meaning be derived without using word order? a comparison of latent semantic analysis and humans. In *Proceedings of 19th Annual Meeting of the Cognitive Science Society*, pages 412–417, 1997.
- [15] Y. Li, D. McLean, Z. Bandar, J. O'Shea, and K. Crockett. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1138–1150, 2006.
- [16] C. Meadow, D. Kraft, and B. Boyce. *Text Information Retrieval Systems*. Academic Press, Inc., Orlando, FL, USA, 1999.
- [17] M. Rodriguez and M. Egenhofer. Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 15:442–456, 2003.
- [18] M. Sahami and T. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th International Conference on World Wide Web*, pages 377–386, Edinburgh, Scotland, 2006. ACM Press.
- [19] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2003.
- [20] O. Zamir and O. Etzioni. Grouper: a dynamic clustering interface to web search results. *Computer Networks*, 31(11-16):1361–1374, 1999.
- [21] D. Zhang. Web based question answering with aggregation strategy. In *Proceedings of the 6th Asia Pacific Web Conference (APWEB2004)*, Hangzhou, China, 2004.
- [22] S. Zhong and J. Ghosh. Generative model-based document clustering: a comparative study. *Knowledge and Information Systems*, 8(3):374–384, 2005.
- [23] H. Zhuge. China's e-science knowledge grid environment. *IEEE Intelligent Systems*, 19:13–17, 2004.
- [24] H. Zhuge. *The knowledge grid*. World Scientific Publishing Co., Singapore, 2004.