A Classification Method of Inquiry e-Mails for Describing FAQ with Automatic Setting Mechanism of Judgment Threshold Values

Yuki Tsuda[1], Masanori Akiyoshi[2], Masaki Samejima[1] and Hironori Oka[3]
[1]*Graduate School of Information Science and Technology, Osaka University, Osaka, Japan*
[2]*Faculty of Applied Information Science, Hiroshima Institute of Technology, Hiroshima, Japan*
[3]*Code Toys K.K., Osaka, Japan*

Keywords: Help Desk, FAQ, Clustering, Threshold Value.

Abstract: The authors propose a classification method of inquiry e-mails for describing FAQ (Frequently Asked Questions) and individual setting mechanism of judgment threshold values. In this method, a dictionary used for classification of inquiries is generated and updated automatically by statistical information of characteristic words in clusters, and inquiries are classified correctly to each proper cluster by using the dictionary. Threshold values are individually and automatically set by using statistical information.

1 INTRODUCTION

As web-based services such as online shopping and community management are rapidly increased, inquiry e-mails about services through the web form from users are also increased. When a user sends an inquiry e-mail to the company, an operator at the help desk in the company needs to answer the user's inquiry. In order to reduce such operators' task, the service provider sets up FAQ (Frequently Asked Questions) on the web page and expects that users read FAQ before sending inquiry e-mails. Users browse FAQ to get answers to their questions. If there are not FAQ about their questions, they send inquiry e-mails to the help desk. Here, operators mainly deal with two tasks of help desk as follows: replying to inquiries and setting up FAQ.

For reducing operators' works of replying to many inquiries, there are some researches on replying with FAQ such as automating retrieval(Sneiders, 2009). Domain ontologies based approach (Fu et al., 2009; Hsu et al., 2009; Yang, 2008), case-based approach(Hammond et al., 1995) and cluster-based approach (Kim and Seo, 2008) have been proposed for retrieving FAQ. In order to set up FAQ, operators analyze the history of both frequent inquiries and operators' replies, which takes great deal of time to read a large number of inquiries. So, the goal of our research is to generate candidates of FAQ automatically from "threads" that are pairs of an inquiry and an answer.

By hierarchical clustering (Willett, 1988) similar threads, the cluster that consists of many threads can be regarded as a candidate of FAQ. Reading major threads in each cluster, operators can set up FAQ easily. However, only by the hierarchical clustering, the cluster of candidate FAQ is not correctly generated from inquiries that have a variety of expressions and a lot of words. In order to generate the clusters of candidate FAQ correctly, we propose a stepwise clustering method to refine deciding similarities and threshold values.

2 A STEPWISE CLUSTERING METHOD FOR EXTRACTING CANDIDATE FAQ

2.1 Outline of Stepwise Clustering

The hierarchical clustering builds a tree structure of threads, cuts the tree at a given height, and generates the clusters as parts of the tree of the threads. The height is decided as a threshold value of similarities between threads, and the similarities are decided by Cosine similarity between vectors of word frequencies in threads(Sullivan, 2001). On the other hand, the similarities between clusters are defined as averages of all the similarities between threads in each cluster by using "group average method" (Willett, 1988).

Tsuda Y., Akiyoshi M., Samejima M. and Oka H..
A Classification Method of Inquiry e-Mails for Describing FAQ with Automatic Setting Mechanism of Judgment Threshold Values.
DOI: 10.5220/0003972101990205
In *Proceedings of the 14th International Conference on Enterprise Information Systems* (ICEIS-2012), pages 199-205
ISBN: 978-989-8565-12-9

Through the analysis of the clusters, we found following points on the threshold value of similarities.

- If the threshold value is high, precise but small clusters are generated.
- As the threshold value becomes low, clusters include improper threads whose contents are different from contents of the clusters.

The threads in a cluster include "characteristic words" which represent a content of the cluster. However, non-characteristic words are also used for a calculation of the similarity. So, a similarity between a cluster and an improper thread to a content of the cluster may be over the threshold value, which causes that the cluster can contain the improper thread. Therefore, we propose a clustering method by reflecting characteristics of words to the similarity. The proposed method uses "category dictionary" that has values indicating how characteristic the words in each cluster are. In the dictionary, characteristic words have high values, and non-characteristic ones have small values. These are weighted to the similarity so as to reflect the characteristics. In order to generate precise clusters, the dictionary needs to have enough words and appropriate values of weights for the words. However, the construction of the dictionary is time-consuming task for operators. So, it is necessary to generate clusters and update the dictionary automatically and accurately.

Figure 1 shows the flow of extracting candidates of FAQ by clustering method that consists of the following three steps:

(1) Making Core Clusters by a High Strictly Threshold Value: In order to ensure the accuracy at the beginning of the clustering, the small but precise clusters (core clusters) are generated by hierarchical clustering with a high threshold value. And values in the dictionary are decided as *tf-idf*(term frequency inverse document frequency) : words' typical indicators for characteristics (Salton and McGill, 1983).

(2) Expanding Clusters by an Appropriately-loosened Low Threshold Value: The small cluster is not regarded as candidate FAQ, because it is thought that the content of the small cluster is not a frequent inquiry. Therefore, core clusters are expanded with a low threshold value by referring the category dictionary.

(3) Cleansing Clusters: Improper threads in a cluster are removed from the cluster.

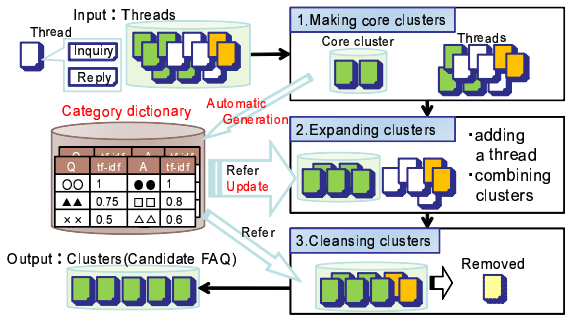

Figure 1: Overview of clustering with dictionary.

Theses three steps need thresholdvalues, which are impracticable to set appropriately by hand. Therefore we also propose an automatic setting mechanism of these threshold values.

2.2 Construction of Core Clusters

Core clusters should be constructed precisely for making the dictionary that has appropriate information of characteristics of words in order to generate correct clusters in the later steps. Therefore, core clusters have to be constructed with strictly similar threads to each other. This similarity index is used in clustering and calculated from the weighted sum of the Cosine similarity between inquiries of threads and the Cosine similarity between replies of threads.

$$Sim(Th_i,Th_j) = (1-\alpha)cosSimQ_{i,j} + \alpha cosSimA_{i,j} \quad (1)$$

$$cosSimQ_{i,j} = \frac{\vec{Q}_i \cdot \vec{Q}_j}{\|\vec{Q}_i\| \, \|\vec{Q}_j\|}, \quad cosSimA_{i,j} = \frac{\vec{A}_i \cdot \vec{A}_j}{\|\vec{A}_i\| \, \|\vec{A}_j\|}$$

Th_i is a thread of $\vec{Q}_i$ and $\vec{A}_i$, $\vec{Q}_i$ is a vector of word frequencies in an inquiry of Th_i, and $\vec{A}_i$ a vector of word frequencies in a reply of Th_i. The similarity index is derived as $Sim()$, $cosSimQ_{i,j}$ is the similarity between inquiries Q_j, Q_i, $cosSimA_{i,j}$ is the similarity between replies of A_j, A_i and $\alpha (0 < \alpha < 1)$ is a constant value to reflect which similarities can be used for the clustering. The replies are usually written by specific operators and the words used in the replies of the same content are similar. Therefore α might be larger than 0.5.

After the construction of core clusters, a category dictionary is generated from the core clusters. This category dictionary is referred in the expansion and sophistication of clusters. The category dictionary keeps *tf-idf* value of each word in each cluster as a typical indicator for characteristics of each cluster. A *tf-idf* value of $Word_s$ gets a high value if the word appears frequently in the thread Th_i and the number of clusters containing the word is small.

$$tf\text{-}idf(Th_i, Word_s) = tf_{i,s} \times idf_s$$

$$tf_{i,s} = \frac{\text{Freq. of } Word_s \text{ in } Th_i}{\text{Num. of all words in } Th_i}$$

$$idf_s = \log \frac{\text{Num. of all clusters}}{\text{Num. of clusters including } Word_s}$$

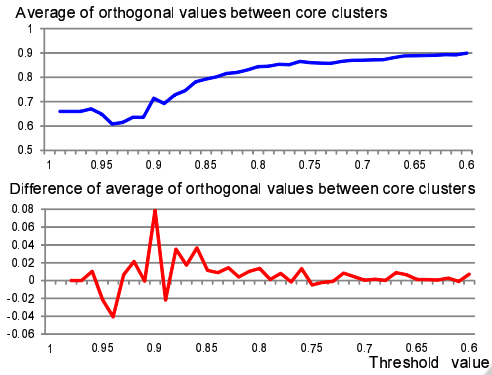

Figure 2: Change of orthogonal indices.

A help desk operator decides the threshold value in this step so as to satisfy that core clusters contain strictly similar threads to each other and contents of core clusters are exclusive. The average of similarities between threads in the cluster is useful for judging that the cluster consists of similar threads. If the average is high, the operator can grasp that a core cluster contains strictly similar threads.

In order to judge that the contents of the core clusters are exclusive, we defined the similarity subtracted from 1.0 as an orthogonal index between core clusters. Figure 2 shows a change of the average of the orthogonal indices with a change of the threshold value. Differential values of the averages gradually converge on 0, as the value of the threshold value is decreased. An operator can set the threshold value with the convergence because the convergence means that contents of core clusters get exclusive.

From these two points, an operator sets the threshold value generating core clusters which contain strictly similar threads and whose contents are exclusive to other clusters' contents.

2.3 Expansion of Clusters

In this step, core clusters constructed in the first step are expanded for extracting candidates of FAQ. The process of the cluster expansion with the dictionary is executed in the following order:

(1) Adding a thread to a cluster.

(2) Combining two clusters.

Because core clusters are constructed with strictly similar threads to each other, a lot of threads are not included in any clusters. These threads outside core clusters should be added to a similar cluster based on the category dictionary. Furthermore it is necessary to combine similar clusters. The dictionary is updated at every expansion so as to put current information in it.

2.3.1 Adding a Thread to a Cluster

Because threads in a cluster of candidate FAQ must be similar to each other, a similar thread to the cluster can be added to the cluster. A similarity between a cluster and a thread is decided by the following formula:

$$Sim(Cluster_m, Th_j) = (1-\alpha)\, cosSimQ_{m,j} + \alpha\, cosSimA_{m,j} \quad (2)$$

$$cosSimQ_{m,j} = \sum_{i=1}^{n} cosSimQ_{i,j} \,/ n$$

$$cosSimA_{m,j} = \sum_{i=1}^{n} cosSimA_{i,j} \,/ n$$

$$cosSimQ_{i,j} = \frac{tf\text{-}\vec{id}f_m(\vec{Q_i}) \cdot tf\text{-}\vec{id}f_m(\vec{Q_j})}{||tf\text{-}\vec{id}f_m(\vec{Q_i})||\,||tf\text{-}\vec{id}f_m(\vec{Q_j})||}$$

$$cosSimA_{i,j} = \frac{tf\text{-}\vec{id}f_m(\vec{A_i}) \cdot tf\text{-}\vec{id}f_m(\vec{A_j})}{||tf\text{-}\vec{id}f_m(\vec{A_i})||\,||tf\text{-}\vec{id}f_m(\vec{A_j})||}$$

where $tf\text{-}\vec{id}f_m(\vec{Q_i})$ is $\vec{Q_i}$ weighted with *tf-idf* by category dictionary of $cluster_m$ and $tf\text{-}\vec{id}f_m(\vec{A_i})$ is $\vec{A_i}$ weighted with *tf-idf* by category dictionary of $cluster_m$. When a cluster is the most similar to a thread and the similarity is over the threshold value, the thread is classified into the cluster. After this process for all threads outside clusters is done, "final clusters" are finally created.

While the final clusters should be as precise as the core clusters, it is also necessary how to decide the threshold value to ensure the precision of the expansion. The similarities between threads in the core cluster are high and the frequency distribution of the similarities is decided as shown in Figure 3. The distributions of the similarities in the final clusters can be estimated by the average μ and the standard deviation σ of similarities in core clusters. Because threads to be added are not as similar as the threads in the cluster, the frequency distribution is changed after adding a thread to the cluster. If the thread is added to the cluster correctly, similarities of the core cluster are similar to ones of the final cluster.

So when the frequency distribution of similarities changes after the expansion, the proposed method

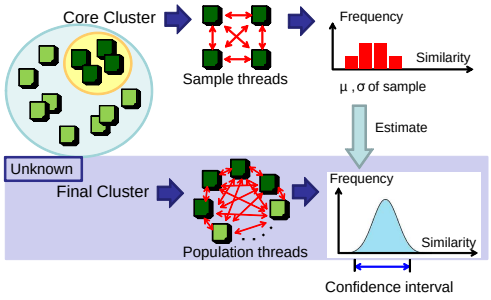

Figure 3: Estimating population from core cluster.

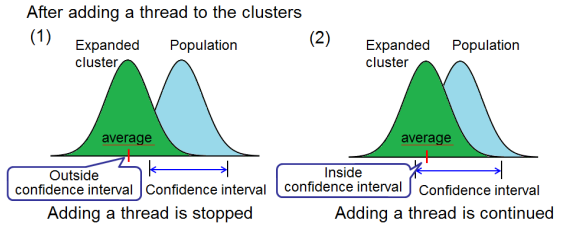

Figure 4: Judgment whether adding a thread is stopped or continued.

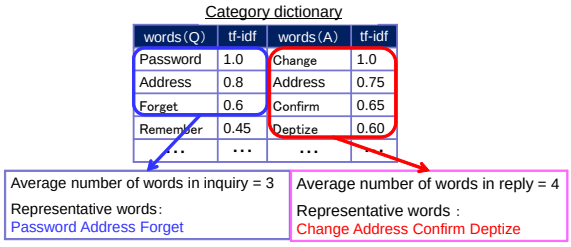

Figure 5: Representative words in a representative thread.

judges whether an added thread is correct or not by statistical testing that the clusters before and after the expansion can be regarded as the same.

For estimating the distributions of the final clusters, the average μ and the standard deviation σ are necessary. The proposed method derives μ and σ from the similarities in the $cluster_m$ by the following formula:

$$Sim_{Cluster_m}(Th_i, Th_j) = (1-\alpha)\, cosSimQ_{i,j} + \alpha\, cosSimA_{i,j} \quad (3)$$

"Confidential interval" of the average of similarities are used as the threshold values of the clustering. Figure 4 shows the judgment whether adding a thread is stopped or continued. The threshold values are decided to be the lower confidence limit. If the average of similarities in the expanded cluster is lower than the threshold value, adding a thread to the cluster is stopped. When adding to all clusters is stopped, this process is ended. Then the proposed method can set the threshold value individually and automatically for each cluster.

2.3.2 Combining Two Clusters

Because the threshold value in the step of constructing core clusters is a high value, a lot of small clusters can be constructed. Similar clusters have to be combined for acquiring large clusters as candidates of FAQ. A similarity between clusters is calculated by using *tf-idf* in the category dictionary. There are non-characteristic words that have small *tf-idf*, which makes similarities higher even if the contents are not similar. So, words in the top k of *tf-idf* are used for deciding similarities as the following formula:

$$Sim(Cluster_m, Cluster_n) = \frac{tf\text{-}id\vec{f}_{Q_m}[k]\cdot tf\text{-}id\vec{f}_{Q_n}[k] + tf\text{-}id\vec{f}_{A_m}[k]\cdot tf\text{-}id\vec{f}_{A_n}[k]}{2} \quad (4)$$

where $tf\text{-}id\vec{f}_{Q_m}[k]$ and $tf\text{-}id\vec{f}_{A_m}[k]$ are vectors having upper k elements of inquiry and reply in category dictionary of $cluster_m$ respectively. And k is decided as follows. Firstly, the accumulated average of the top i of *tf-idf* in category dictionary of the $cluster_m$ is calculated. Because *tf-idf* of non-characteristic words are small and not so different, the accumulated average converges to 0 as i is increased. So, the proposed method calculates the second difference of the accumulated average and selects k when the second difference converges on 0.

If the highest similarity is more than the threshold value given in advance, the two clusters are combined.

2.4 Sophistication of Clusters

The pre-process may add threads to a improper cluster because *tf-idf* values of characteristic words in the dictionary are not completely calculated. After the construction of the dictionary is completed, the proposed method can judge whether an added thread is proper or improper to the cluster. So, the proposed method removes threads that do not include the characteristic words of the cluster.

As a criteria to judge whether a thread include the characteristic words or not, the method generates a virtual thread called "representative thread" that includes just all characteristic words in the cluster. In order to generate a representative thread, the upper m words on *tf-idf* in the category dictionary are chosen as Figure 5 shows. Then m is decided as an average number of words in threads in a cluster. The method decides whether threads in a cluster should be removed by Cosine similarity with the representative thread. If the similarity is lower than a threshold value, the thread is removed.

To set the threshold value in cleansing clusters individually and automatically, we use the average of similarity with the representative thread. The average of similarities with the representative thread has a relation to the similarities between the representative thread and the threads in a cluster. If the cluster is not precise, the similarities with the representative thread are low. So, the threshold value of $cluster_m$ is the value which is the standard deviation (σ_m) of similarity in each cluster subtracted from the average of similarity (μ_m):

$$Threshold_m = \mu_m - \sigma_m$$

3 EXPERIMENT

3.1 Results of the Clustering

In the experiment, threads of inquiries and replies about the web site for a sport membership administration are used. The number of threads is 1318 and these threads are written in Japanese. Inquiries have 16.8 words and reply have 34.6 words on average. A constant value (α) in expressions in former sections is 0.7 for weighting replies because replies are probably written by particular operators and they use same words in the replies having same contents. The confidence coefficient in adding a thread is 99% and the threshold value in combining clusters is 0.30.

The generated clusters must reflect frequencies of inquiries in input data, and contents of them must be read easily by operators. So, we set the criteria of evaluation as follows:

- **Cluster Size:** The cluster size is defined as the number of threads in the cluster. By comparing cluster sizes each other, we can judge how high the frequency of inquiries is in each cluster, and evaluate which clusters reflect precisely frequencies of inquiries in input data.
- **Precision of Clustering:** The precision of clustering is the rate of threads classified correctly in a cluster. If it is high, operators can read easily a content of a cluster without reading wrong threads.

We compared results of clustering by the proposed method to the conventional hierarchical clustering by Cosine similarity. This conventional method uses the same similarity and clustering method as the proposed method in section 2.2, and a constant value (α) in expressions is also 0.7. The threshold value of the hierarchical clustering is 0.47 that is adjusted to get the best precision manually. We generated clusters by hand and defined the clusters having over 50 threads as candidates of FAQ. Table 1 shows the candidate FAQ and the numbers of threads that have the content of candidate FAQ.

Figure 6 and Table 2 show results of the experiment. They show cluster sizes and precisions of generated clusters having contents of candidate FAQ. In Figure 6, the sizes of FAQ2 and FAQ3 clusters by

Table 1: Examples of candidate FAQ.

	Content	Number of threads
FAQ1	Forgetting my password	210
FAQ2	Correcting my date of birth	123
FAQ3	Altering to player from staff	61

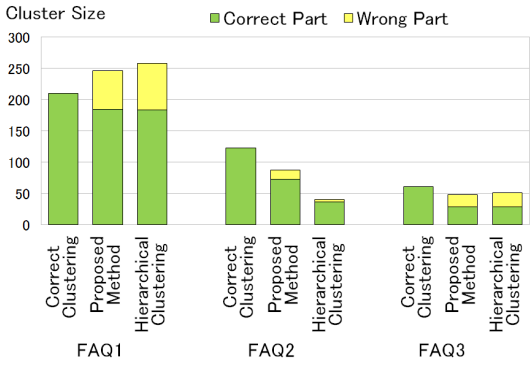

Figure 6: Result of cluster size.

Table 2: Result of precision.

Cluster	FAQ1	FAQ2	FAQ3
Proposed method	75%	84%	58%
Hierarchical clustering	71%	90%	55%

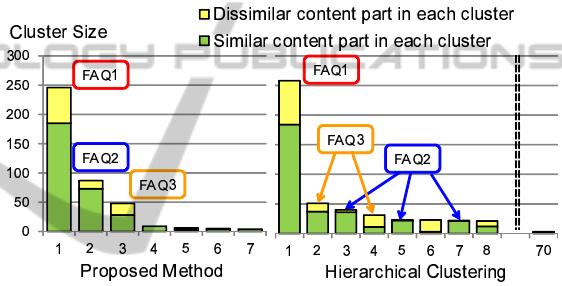

Figure 7: Clusters generated by each method.

hierarchical clustering are almost the same. These clusters do not reflect frequencies of inquiries in input data, which makes it difficult for help desk operators to grasp which content of the cluster is more frequently inquired. On the other hand, the proposed method generates clusters reflecting frequencies of inquiries although there are gaps between cluster sizes of clusters generated by the proposed method and correct clustering. The precisions of FAQ1 and FAQ2 clusters generated by the proposed method are over 70% in Table 2. The precision of FAQ3 cluster by the proposed method is also higher than one by hierarchical clustering. Therefore help desk operators can grasp the contents of the cluster more easily by the proposed method.

Figure 7 shows a result of all clusters generated by each method. In Figure 7, the generated clusters are placed in order of the cluster size. Operators extract major clusters as a candidate FAQ by reading all threads in each cluster. And, they judged whether the threads are a similar content part and a dissimilar content part to major threads. The hierarchical clustering generates different clusters even if they have same

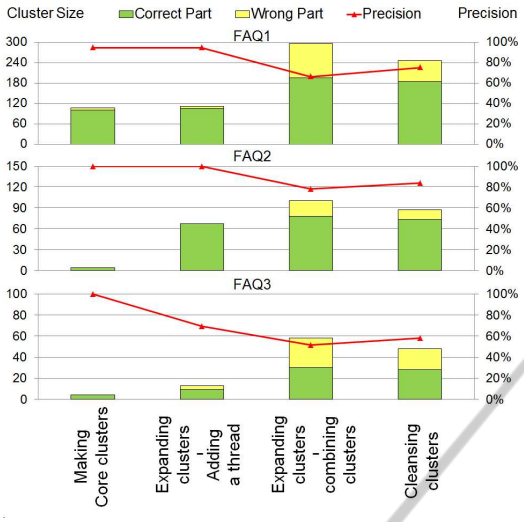

Figure 8: Candidate FAQ clusters in each step.

contents: clusters for FAQ2 and FAQ3 are separated to some clusters. On the other hand, the proposed method does not generate such scattered clusters and generates much less clusters than hierarchical clustering does. So, help desk operators can find candidate FAQ more efficiently by the proposed method.

3.2 Evaluation of Each Step in Clustering

In order to verify the effectiveness of each step in stepwise clustering, Figure 8 shows how three candidate FAQ clusters shown in the former section change in each step. Evaluation criteria are the cluster size and the precision as well as in the former section.

As for the result of cluster sizes, few threads are added to cluster of FAQ1 but the cluster size of FAQ1 is increased more than doubled by combining clusters. This is because there are many inquiries related to FAQ1 and the inquiries compose large core clusters. Regarding FAQ2 and FAQ3, cluster sizes get more than ten times through adding a thread and combining clusters. The step of adding a thread works well for FAQ2 and the step of combining clusters works well FAQ1 and FAQ3. So, the step of expanding clusters is effective for generating large clusters. As for the result of precisions, in each FAQ, core clusters are generated with high precisions. The precisions of these clusters are decreased through the step of expanding clusters and increased by about 10% through the step of cleansing clusters. From these results, we verify that three steps contribute to generating candidate FAQ clusters in stepwise clustering method.

Table 3: The numbers of threads added to/removed from cluster and their precisions.

	Number of threads		Precision	
	adding	removing	adding	removing
automatic thresholds	230	80	63%	70%
manual thresholds	225	73	48%	74%

3.3 Results of Setting Thresholds

We compared results in case of using threshold values set manually and automatically by the proposed method in adding a thread and cleansing clusters.

Table 3 shows the numbers of threads added to and removed from clusters, and the precisions. The numbers of the threads added to clusters are almost same between the results by the manual setting and the automatic setting. However the precision in the result by automatic setting is better than one by manual setting. This is why the proposed method can set the appropriate threshold value to each cluster automatically. On the other hand, a unique threshold value is given to all the clusters by manual setting, which is not appropriate for some of clusters. Therefore adding a thread works effectively by automatic setting of threshold values. Additionally, the number of threads removed from clusters in the result by automatic setting is about 10% more than one by manual setting, but the precision is not better on the contrary. Automatic setting works as well as manual setting. So, it is effective for reducing operators' time spent on setting the threshold values.

4 CONCLUSIONS

We proposed an effective clustering method that consists of three steps of clustering; making core clusters, expanding clusters and cleansing clusters. And we introduced a similarity index between clusters and threads in each step respectively. The threshold values are set individually and automatically to each cluster. The experiment shows that the proposed method could generate more useful clusters for help desk operators than the conventional method. In future works, we will propose the method for setting thresholds in whole steps, and improve the precision.

REFERENCES

Fu, J., Xu, J., and Jia, K. (2009). Domain ontology based automatic question answering. *ICCET '08. Interna-*

tional Conference on, 2:346–349.

Hammond, K., Burke, R., Martin, C., and Lytinen, S. (1995). Faq finder: a case-based approach to knowledge navigation. *11th Conference on Artificial Intelligence for Applications*, pages 80–86.

Hsu, C.-H., Guo, S., Chen, R.-C., and Dai, S.-K. (2009). Using domain ontology to implement a frequently asked questions system. *World Congress on Computer Science and Information Engineering*, 4:714–718.

Kim, H. and Seo, J. (2008). Cluster-based faq retrieval using latent term weights. *IEEE Intelligent Systems*, 23(2):58–65.

Salton, G. and McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill.

Sneiders, E. (2009). Automated faq answering with question-specific knowledge representation for web self-service. *2nd Conference on Human System Interactions(HSI'09)*, pages 298–305.

Sullivan, D. (2001). *Document Warehousing and Text Mining: Techniques for Improving Business Operations, Marketing, and Sales*. John Wiley and Sons In.

Willett, P. (1988). Recent trends in hierarchic document clustering: A critical review. *Information Processing and Management*, 24(5):577–597.

Yang, S.-Y. (2008). Developing an ontological faq system with faq processing and ranking techniques for ubiquitous services. *First IEEE International Conference on Ubi-Media Computing*, pages 541–546.