Exploring Hierarchical User Feedback in Email Clustering

Yifen Huang and Tom M. Mitchell

Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15217, USA
{hyifen, tom.mitchell}@cs.cmu.edu

Abstract

Organizing data into hierarchies is natural for humans. However, there is little work in machine learning that explores human-machine mixed-initiative approaches to organizing data into hierarchical clusters. In this paper we consider mixed-initiative clustering of a user's email, in which the machine produces (initial and re-trained) hierarchical clusterings of email, and the user iteratively reviews and edits the hierarchical clustering, providing constraints on the next iteration of clustering. Key challenges include (a) determining types of feedback that users will find natural to provide, (b) developing hierarchical clustering and retraining algorithms capable of accepting these types of user feedback, (c) determining the correspondence between two hierarchical structures, and (d) understanding how user behavior changes during a single feedback session and designing machine strategies that change with the user. Preliminary experimental results of two cases shows that under ideal conditions, this mixed-initiative approach requires only 6 minutes of user effort to achieve email clusterings comparable to those requiring 13 to 15 minutes of manual editing efforts.

Introduction

It is natural for users to organize personal data using hierarchies, especially in the electronic world. An obvious example is that file systems in most workstations consist of a hierarchy of user-created directories to store documents and other files. In fact, designs of hierarchical organization prevail in computer applications such as email clients, bookmark organizers, and contact management tools. While users can spontaneously organize data into hierarchical structures, machine learning clustering algorithms are rarely used to support such applications. We believe this is because purely autonomous clustering algorithms will rarely produce exactly the hierarchy the user desires. We hypothesize that mixed-initiative human-machine approaches to hierarchical clustering hold great potential for such applications.

In this paper, we address the question, "how can autonomous clustering algorithms be extended to enable mixed-initiative clustering approaches involving an iterative sequence of computer-suggested and user-suggested revisions to converge to a useful hierarchical clustering?"

Framework for Mixed-Initiative Clustering

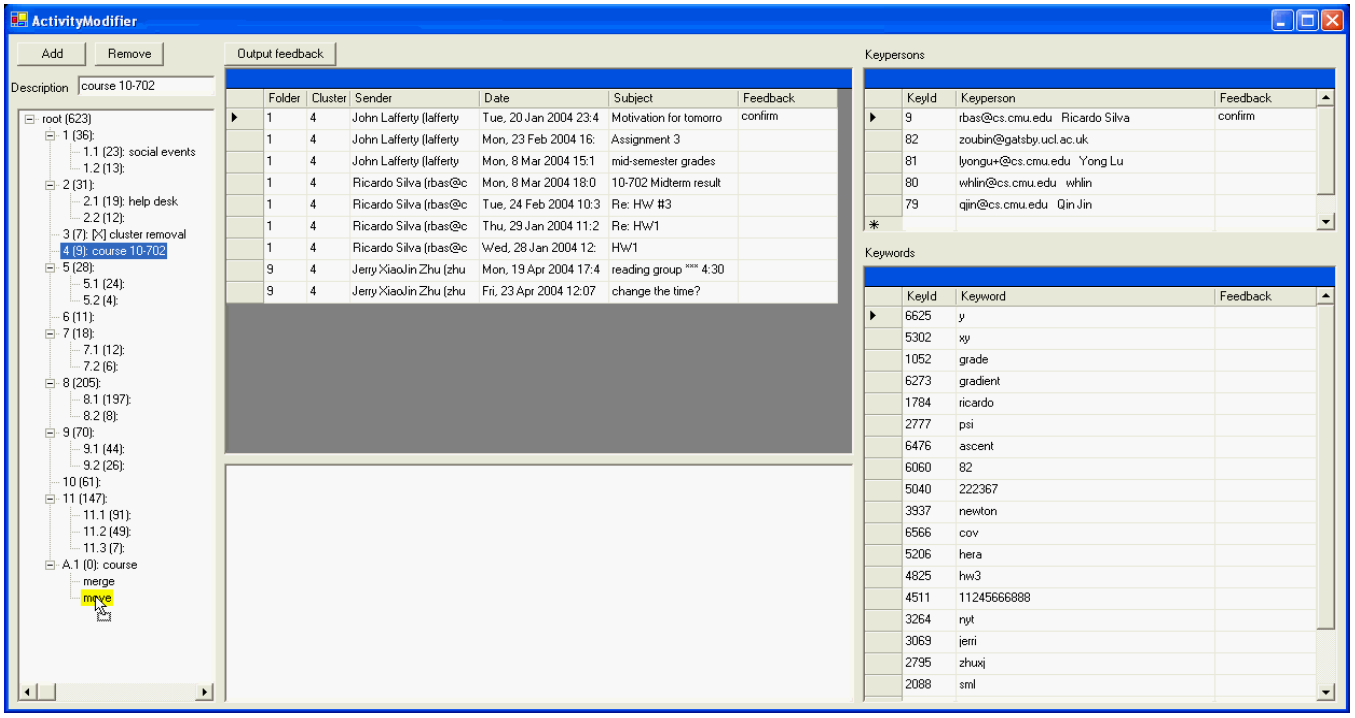
In our previous work (Huang 2007), we have defined a framework for mixed-initiative clustering that combines a user and a machine interactively to solve the clustering problem together. The goals of mixed-initiative clustering models are to perform well in clustering inference and to be capable of communicating with a user for improvements. A mixed-initiative task is similar to semi-supervised learning task, but differs in three key respects. First, a set of hypothesized properties is extracted from the trained model for the user, e.g., a hierarchical clustering of the data and key features for each cluster. Second, the user can browse and modify, if needed, hypothesized properties in an interface. These modifications, commonly called "user feedback", should be interpretable to the model. Finally, a re-training algorithm is used to revise the model consistent with the user's modifications.

Activity Extraction from Workstation Emails

Our application is to extract a user's activities by analyzing/clustering their email. We have studied mixed-initiative text clustering using extended flat, non-hierarchical user feedback (Huang and Mitchell 2006). The types of extended user feedback include confirmation or removal of a cluster (activity), confirmation or disapproval of an email's association with an activity, and confirmation or disapproval of a keyword or key-person's association with an activity. We have improved 20% absolute accuracy in this activity extraction application by integrating the user's feedback based on their comprehension and the machine's computational power under the mixed-initiative clustering framework. In other work (Mitchell et al. 2006), we found the accuracy of clusters improves further when social network analysis is used to split clusters into sub-communities of email senders and recipients.

Hierarchical Email Clustering with Hierarchical User Feedback

In this paper, we seek to integrate the advantages achieved in the previous work, by extending flat-structural email



Description: course 10-702

```
root (623)
  1 (36):
    1.1 (23): social events
    1.2 (13):
  2 (31):
    2.1 (19): help desk
    2.2 (12):
  3 (7): [X] cluster removal
  4 (9): course 10-702
  5 (28):
    5.1 (24):
    5.2 (4):
  6 (11):
  7 (18):
    7.1 (12):
    7.2 (6):
  8 (205):
    8.1 (197):
    8.2 (8):
  9 (70):
    9.1 (44):
    9.2 (26):
  10 (61):
  11 (147):
    11.1 (91):
    11.2 (49):
    11.3 (7):
  A.1 (0): course
    merge
    move
```

Folder	Cluster	Sender	Date	Subject	Feedback
1	4	John Lafferty (lafferty	Tue, 20 Jan 2004 23:4	Motivation for tomorro	confirm
1	4	John Lafferty (lafferty	Mon, 23 Feb 2004 16:	Assignment 3	
1	4	John Lafferty (lafferty	Mon, 8 Mar 2004 15:1	mid-semester grades	
1	4	Ricardo Silva (rbas@c	Mon, 8 Mar 2004 18:0	10-702 Midterm result	
1	4	Ricardo Silva (rbas@c	Tue, 24 Feb 2004 10:3	Re: HW #3	
1	4	Ricardo Silva (rbas@c	Thu, 29 Jan 2004 11:2	Re: HW1	
1	4	Ricardo Silva (rbas@c	Wed, 28 Jan 2004 12:	HW1	
9	4	Jerry XiaoJin Zhu (zhu	Mon, 19 Apr 2004 17:4	reading group *** 4:30	
9	4	Jerry XiaoJin Zhu (zhu	Fri, 23 Apr 2004 12:07	change the time?	

KeyId	Keyperson	Feedback
9	rbas@cs.cmu.edu Ricardo Silva	confirm
82	zoubin@gatsby.ucl.ac.uk	
81	lyongu+@cs.cmu.edu Yong Lu	
80	whlin@cs.cmu.edu whlin	
79	qjin@cs.cmu.edu Qin Jin	

KeyId	Keyword	Feedback
6625	y	
5302	xy	
1052	grade	
6273	gradient	
1784	ricardo	
2777	psi	
6476	ascent	
6060	82	
5040	222367	
3937	newton	
6566	cov	
5206	hera	
4825	hw3	
4511	11245666888	
3264	nyt	
3069	jerri	
2795	zhuxj	
2088	sml	

Figure 1: An interface that presents hierarchical results to a user and allows various types of hierarchical and non-hierarchical user feedback. The left panel shows the resulting hierarchy. When a user selects a cluster in the hierarchy, the middle top panel shows a list of emails in this cluster, and the middle bottom panel would show content of an email chosen by the user. The right panels show key-persons and keywords associated with this cluster. In this example, the user thinks cluster 4 is related to a specific course (the confirmed key-person is the TA), which should be under a general course cluster. The user has added a "course" cluster, A.1, and is moving cluster 4 underneath cluster A.1.

clustering to hierarchical clustering, and to explore the nature of user feedback for such hierarchical clustering problems. We build a hierarchical email clustering task composed of the following steps: (1) generating initial hierarchical clusters of depth two by using a generative clustering model in the first level and social network analysis for the second, (2) presenting the hierarchical clustering results in a user interface and recording users' modifications of the hierarchical clustering with time stamps, and (3) re-training the hierarchical clustering model according to this hierarchical user feedback.

Figure 1 shows our design of a user interface that can accept various types of hierarchical and non-hierarchical user feedback.

Types of Hierarchical User Feedback

We can categorize several types of hierarchical user feedback. Five of them relate to modifying parent-child or sibling relationships in a cluster hierarchy. The sixth type relates to moving a document to another cluster. These feedback types are supported by the user interface shown in Figure 1.

- Cluster-Remove: when a cluster is too noisy to be understood or a user doesn't think the idea conveyed by the cluster is significant, the user may remove this cluster and its descendants.
- Cluster-Add: when there is no cluster that represents a certain idea a user has in mind, the user can create a new cluster and place it under a reasonable parent cluster.
- Cluster-Split: when a cluster is noisy and a user thinks that it mixes up different ideas and still wants to keep it, a user may request that the computer split this cluster into smaller clusters.
- Cluster-Move: a user can drag and drop this cluster under a more reasonable parent cluster.
- Cluster-Merge: when a user thinks a cluster contains a repetitive idea that has been represented in another cluster, the user can merge the two clusters.
- Document-Move: a user can drag and drop a document from its current cluster to another cluster.

Table 1: Feedback types currently in our system: (*) are new feedback types added to accommodate hierarchical clustering.

Type \ Level	Cluster	Document	Feature
Non-hierarchical feedback	Confirm (Remove)	Confirm Remove	Confirm Remove
Hierarchical feedback	Remove* Add* Move* Merge*	Move*	

We have integrated five of these six feedback types (cluster-splitting feedback is skipped) into our first mixed-initiative hierarchical clustering system. In addition to hierarchical user feedback, we still keep all non-hierarchical user feedback types we have studied in our previous work. These previous feedback types consist of positive and negative feedback on clusters, documents, and features within individual clusters. Table 1 shows a list of feedback types currently in our system.

We define "complete hierarchical user feedback" as a modification from an imperfect hierarchy, which contains undesirable parent-child and sibling relationships (from the user's perspective), to a reference (target) hierarchy. Since it is not practical to expect complete hierarchical feedback from a user, a user can quit whenever they want, and leave the machine to retrain the hierarchical model starting from the user-modified hierarchy and subject to their non-hierarchical feedback.

Retraining the Hierarchical Model

As mentioned above, we generate initial hierarchical clusters of depth two by using a generative clustering model in the first level and applying social network analysis for the second level to produce purer sub-clusters. The results from this approach often contain many errors in parent-child relationships and almost no correct sibling relationships. The benefit of using social network analysis is not to generate good siblings in the hierarchy, but to create separate social cliques (purer sub-clusters) so a user can understand and manipulate them more easily. After a user feedback session on this initial hierarchical result, we retrain the hierarchical model based on the user modified hierarchy and feedback.

The re-training algorithm re-uses the user-modified hierarchy but adjusts the document-to-cluster assignments. It adopts the "Pachinko-machine" concept described in (Koller and Sahami 1997) and trains a SpeClustering model as the classifier for the root node and intermediate nodes. Each document is distributed to one sub-cluster for further training. The distribution is based on the document's posterior probabilities given the model of the parent node. The SpeClustering model is a probabilistic clustering model we developed in (Huang and Mitchell 2006). It stands for "Specific Clustering", and refers to the fact that the probabilistic model estimates a latent variable for each feature (word or person) to determine whether it is relevant to or independent of a specific cluster. Put another way, the SpeClustering algorithm assumes that each document is a mixture of two distributions of features – one distribution that is specific to the cluster, and one distribution that is independent of the cluster. Since we train separate SpeClustering classifiers for root and intermediate nodes, this is similar to performing different soft feature selections at each node. The SpeClustering algorithm supports feedback adaptation for all non-hierarchical user feedback types shown in Table 1. We will refer to this hierarchical model as the "Cascading SpeClustering model" in the rest of the paper.

For this email clustering task, we extract both word features and person features from the email corpus. The SpeClustering algorithm has an extension to multiply probabilities of different feature sets, so it can handle word and person features jointly. In order to simulate the social network analysis that is used to obtain the initial hierarchical clustering result, we add a "PersonWeight" parameter for the second and deeper levels in the hierarchy. The value of PersonWeight multiplies counts in the person corpus. This produces a double counting effect in the SpeClustering model so the final classifier will be biased towards the person corpus like what social network analysis does.

The user's hierarchical feedback is embedded in the user-modified hierarchy, used for model retraining. However, the user modification is most likely not complete. Therefore, we add a "PriorProb" parameter that indicates the machine's belief that the user left documents at the correct locations in the hierarchy. When the parameter is 1, the algorithm preserves these document-to-cluster assignments. When the parameter is lower, the re-trained model has more freedom to re-assign documents to other clusters within the hierarchy.

For each intermediate node, the SpeClustering classifier is trained using the relevant feedback entries. We extract feedback entries relevant to this node and all its descendant nodes so descendants' feedback entries can propagate to their parent and ancestor nodes. We need to convert hierarchical feedback entries to positive/negative feedback because the SpeClustering model accepts only non-hierarchical feedback. For example, a document-move feedback entry can be converted to a negative feedback entry for the original cluster and a positive feedback entry for the target cluster.

Alternative hierarchical models like the shrinkage model (McCallum et al. 1998) or Hierarchical Latent Dirichlet Allocation (Blei et al. 2003) are both possible. For the shrinkage model, we need to design model adaptation heuristics for feature feedback types. Hierarchical Latent Dirichlet Allocation was our first choice but the Markov chain processes' slow convergence contradicts our goal of efficient interaction between machine and user.

Distance Measurement between Hierarchies

Another important problem in mixed-initiative hierarchical clustering is evaluating the performances of hierarchical results (including initial hierarchical results, hierarchies modified after user feedback, and re-trained hierarchical clustering results.) The solution of this problem requires a distance measurement between the resulting hierarchy and the reference hierarchy while these two hierarchies don't share a common skeleton.

At first, this seems to be a very difficult problem. As (Sun and Lim 2001) pointed out in hierarchical classification, flat precision/recall measurements do not consider the hierarchical structure. They proposed heuristic measurements that consider degree of misclassification.

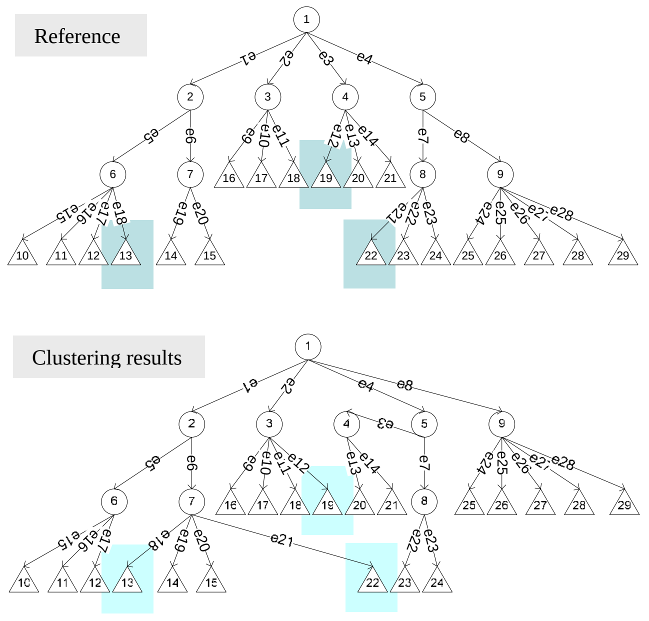
Instead of calculating hierarchical similarities, differences between two hierarchies can be measured by the number of editing operations needed to change one hierarchy into the other. This measurement is more suitable for our mixed-initiative scenario which allows hierarchical user feedback. The set of editing operations are defined by the set of hierarchical feedback types. We will explain this idea in further detail using the following example.

In our previous flat clustering task, we found a one-to-one accuracy-maximizing cluster alignment between a clustering result and its reference. The left-hand side subtree in Figure 2(a) is such an example where cluster (circle node) 2, 3, 6, and 7 are aligned between two hierarchies and results in clustering errors of document (triangle node) 13, 19, and 22. However, the one-to-one correspondence may achieve a sub-optimal result if we take hierarchical constraints imposed by the clustering result into account. Figure 2(b) Alignment 1 shows a possible mapping if we stick with the hierarchical constraints. On the other hand, Alignment 2 in Figure 2(b) shows a mapping that has higher precision and recall at the document level but violates the hierarchical constraints. Considering which alignment is more likely to be identified by a human user when the whole hierarchy is shown, we suspect that Alignment 2 is more likely than Alignment 1 because recognizing similar small clusters is easier for users than generalizing heterogeneous clusters.

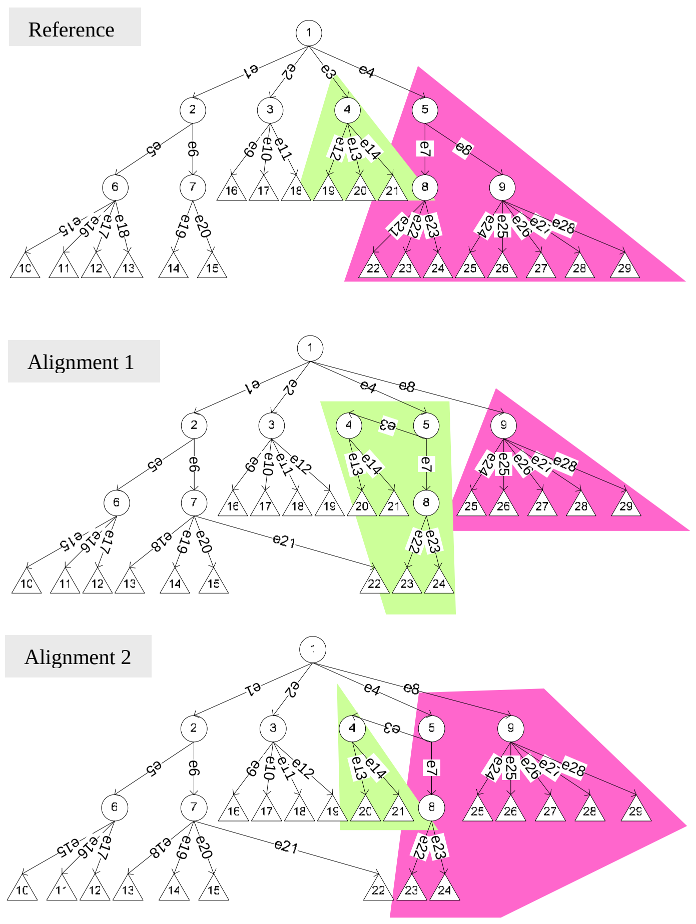
Furthermore, considering the optimal feedback steps a user may give to correct the resulting hierarchy to the reference hierarchy, each new feedback type can be interpreted as a set of edge modifications. "Cluster-move" feedback is equivalent to modifying the parent orientation in the moved cluster's parent-child edge. Similarly, "document-move" feedback is changing the parent end of parent-child edge for the moved document. "Cluster-add" feedback is equivalent to creating a new node and a new edge pointing to its specified parent. "Cluster-remove" feedback is equivalent to removing a node and every edge linked to it. "Cluster-merge" is slightly different; it can be treated as performing "document-move" for every document associated with this cluster to the target cluster and removing the merged cluster.

We define "**edge modification ratio**" as the minimum number of feedback steps required for complete hierarchical user feedback, divided by the total number of edges in the reference hierarchy. The tree labeled "clustering results" in Figure 2 needs five feedback steps that modify edge e3, e8, e12, e18, and e21 accordingly in order to match the reference hierarchy. There are 28 edges in the reference hierarchy, so the edge modification ratio is 0.18 (5/28) for this clustering result.

The concept of "edge modification ratio" is very similar to "tree edit distance" (Bille 2003) where different feedback types are mapped into different operations and the cost function is uniform.



(a) Left-hand side sub-tree (node 2, 3, 6, and 7) can be aligned between two hierarchies and results in 3 document clustering errors.



(b) Two different alignments from a clustering result to a reference.

Figure 2: Hierarchical structure comparison

Experimental Results

We use an email dataset (EmailYH) that consists of emails from one author for the hierarchical mixed-initiative email clustering task. There are 623 emails in this dataset that have been manually organized as a hierarchy that consists of 15 cluster nodes including a root and 11 leaf folders. It contains 6684 unique words and 135 individual people.

In our experiments, a mixed-initiative process consists of the generation of an initial hierarchical clustering with depth two, a user feedback session, and model retraining. In the feedback session, the initial clustering is presented to the email owner in the user interface we introduced in Section 2 to browse and give feedback. The feedback session lasts about 20 minutes and each feedback entry is recorded with its timestamp. Then, the feedback record is used in model retraining and feedback adaptation. Figure 3 shows results of two mixed-initiative processes: (a)-(c) are based on the same initial hierarchical result and (d) (e) are based on the other initial hierarchical result.

We can observe the user behavior in the feedback session by calculating the edge modification ratio of the user-modified hierarchy over time. The dot-marked (black) lines in Figure 3 show the user behavior from two feedback sessions which have different initial hierarchical results. We interpret the user's manual adjustment as having two phases: a comprehension phase to understand the structure and a completion phase where error correction is accelerated and more complicated feedback is given. The vertical dashed line in 3(a) indicates the hypothesized boundary between these two phases.

The other lines in Figure 3 show edge modification ratios for retraining hierarchical results. Each marked point represents a result inferred by a retrained model that uses the user-modified hierarchy up to that time. The cross-marked (green) lines show the edge modification ratio of results with no special weighting on the person corpus, whereas the circle-marked (red) lines show results that give the person corpus a high weight. Weighting the person corpus heavily results in a lower edge modification ratio than when using no special weighting in Figure 3(a) and in the early stage of 3(d). Lower edge modification means a user can achieve the reference hierarchy using fewer feedback steps. The result confirms our previous study that social network analysis helps generate more user-comprehensible clusters (activities), and indicates that the Cascading SpeClustering model is capable of achieving results similar to social network analysis. At the late stage in 3(d), where the edge modification ratio is lower than most results in 3(a), heavily weighting the person corpus does not work as well as focusing on the word corpus. This indicates that our strategy of producing purer sub-clusters and exploiting user's comprehension has certain limits.

The dashed horizontal line in 3(a) shows that with 6-minute user feedback, the re-trained hierarchical clustering can achieve the same performance as 13 minutes of user effort when using completely manual adjustment. The same situation happens in 3(d) where the hierarchical model adaptation with 6 minutes' user feedback achieves results comparable to applying 15 minutes of manual user efforts. The manual effort savings are 7 minutes and 9 minutes correspondingly. However, when the user moves into the completion phase, the Cascading SpeClustering model seems not sophisticated enough to adapt towards complicated and subtle user feedback.

The PriorProb parameter can be interpreted as the level of trust placed upon the user modified hierarchy where a higher value means more trust. Experimental results using different trust values are shown in Figures 3(b) and 3(d). Results are heterogeneous for different settings where sometimes high trust yields better performance but sometimes the opposite. It is unlikely that a fixed optimal strategy exists for this parameter.

To counter this issue, we developed a dynamic strategy that utilizes the existence of confirmation feedback for each cluster. If the cluster is confirmed by the user, the trust level is set high and vice versa. Figure 3(c) and 3(e) show the results of applying the dynamic strategy, and duplicate the fixed trust level results with the heavy person weighting from 3(b) and 3(d) as reference. For example, Figure 3(e) shows that different trust levels are better in different phases: fixed low trust in the comprehension phase outperforms high trust and fixed high trust in the completion phase out-performs low trust. In contrast to fixed strategies, the dynamic strategy is able to not only set trust levels automatically but also achieve good performances in both phases.

Conclusions

In this paper, we propose an approach to mixed-initiative hierarchical clustering, and apply it to hierarchical clustering of email. We notice that hierarchical clustering helps a user understand the results better than flat clustering. Also, hierarchical feedback lets a user modify results more efficiently.

In order to evaluate hierarchical clustering quality, we define "edge modification ratio" to compare resulting hierarchies against a reference hierarchy. This measurement computes the ratio of edited edges in optimal complete user feedback sequences over the edge numbers in the reference.

We have applied a simple hierarchical clustering model, Cascading SpeClustering, to the mixed-initiative email clustering task and achieved improved performances from the joint efforts of a machine and a user. We also learned that a good mixed-initiative system should consider the sequential differences in a feedback session to develop timely or dynamic strategies for retraining models based on multiple types of user feedback.

Acknowledgements

This research was supported by Darpa contract NBCD030010, as part of the Personal Assistants that Learn project.

References

Huang, Y. 2007. Mixed-Initiative Clustering and Its Application to Workstation Activity Extraction. Thesis proposal.

Huang, Y. and Mitchell, T. 2006. Text Clustering with Extended User Feedback, SIGIR.

Mitchell, T., Wang, S., and Huang, Y. 2006. Extracting Knowledge about Users' Activities from Raw Workstation Contents, AAAI.

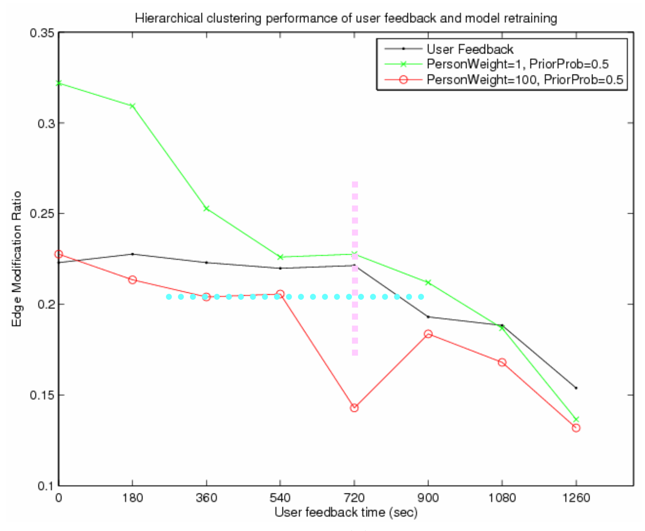
Sun, A. and Lim E. 2001. Hierarchical Text Classification and Evaluation, ICDM.

Bille, P. 2003. Tree Edit Distance, Alignment Distance and Inclusion, IT University Technical Report.

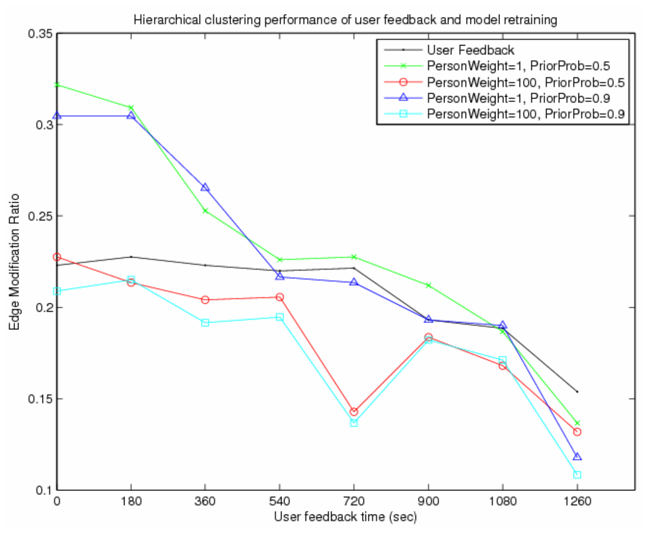
Koller, D. and Sahami, M. 1997. Hierarchically classifying documents using very few words. ICML.

McCallum, A., Rosenfeld, R., Mitchell, T., and Ng, A.Y. 1998. Improving Text Classification by Shrinkage in a Hierarchy of Classes. ICML.

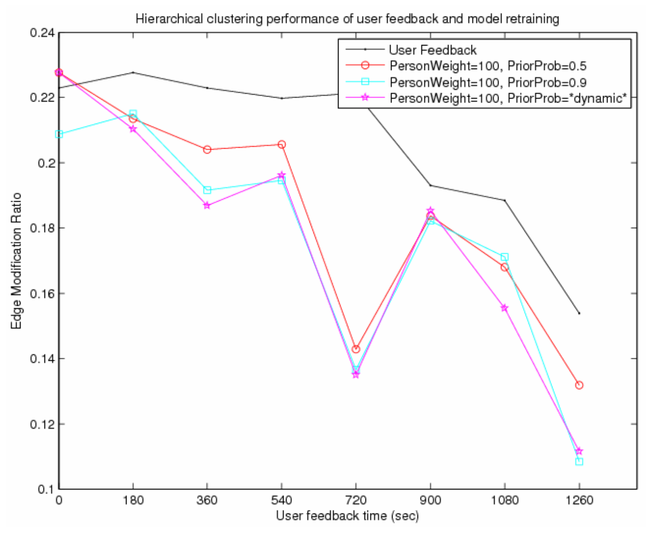
D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. 2003. Hierarchical topic models and the nested Chinese restaurant process. NIPS.



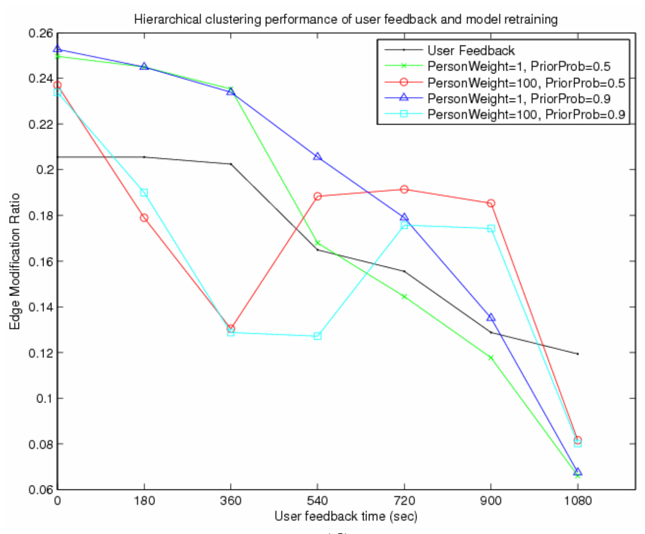
(a)



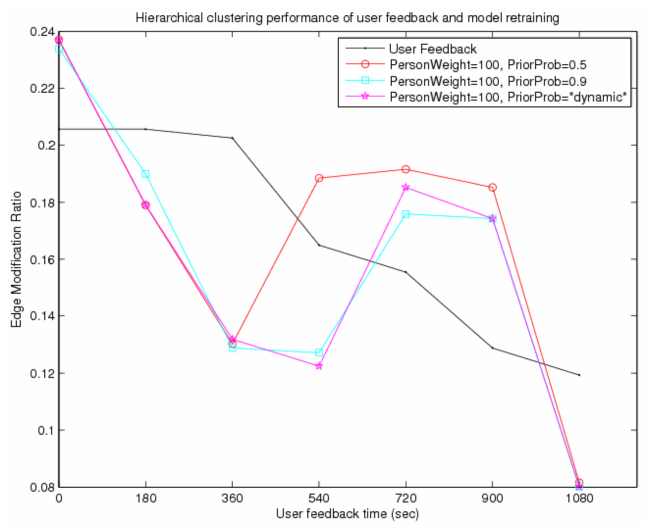
(b)



(c)



(d)



(e)

Figure 3: Experimental results of two mixed-initiative processes where (a)-(c) belongs to the same process and (d) (e) belongs to the other process.