Toward Mixed-Initiative Email Clustering

Yifen Huang
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA, 15213
hyifen@cs.cmu.edu

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA, 15213
Tom.Mitchell@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 reviews and edits the initial hierarchical clustering, providing constraints on the re-trained clustering model. 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) understanding how machine's clustering results and user feedback affect each other.

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 defined a framework for mixed-initiative clustering that combines a user and a machine interactively to solve the clustering problem. The goals of a computer system for mixed-initiative clustering are (1) to cluster the data well, and (2) to present to the user its learned clustering and the properties of its learned clustering model, and (3) to incorporate feedback from the user to produce a revised clustering. For example, properties of a learnt hierarchical clustering model may consist of a hierarchy among clusters, object-to-cluster associations, and key features and probabilistic models for each cluster. We assume an interface is used to present hypothesized properties to a user. The user can browse and modify, if needed, these hypothesized properties. These modifications, commonly called "user feedback," should also be interpretable by the computer program. 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, where each email is described by both the words appearing in the email and the set of email addresses (people) associated with it. We have studied mixed-initiative text clustering using flat, non-hierarchical user feedback (Huang and Mitchell 2006). In that work, the allowed types of flat user feedback included confirmation or removal of an entire proposed cluster (activity), confirmation or disapproval of a particular 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 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 subclusters based on sub-communities of email senders and recipients.

Hierarchical Email Clustering with Hierarchical User Feedback

In this paper, we build on this previous work by extending flat-structural email clustering to hierarchical clustering, and we explore the nature of user feedback for such hierarchical clustering problems. We consider a hierarchical email clustering approach 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 which augment the non-hierarchical feedback types used in our previous work. 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 empty cluster and place it under a reasonable parent cluster. Then a user can optionally populate this cluster by moving relevant documents into it.
- 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 not yet implemented) 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 many incorrect sibling relationships, compared to the user's desires. The benefit of using social network analysis to refine first-level clusters into subclusters is 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 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 (Huang and Mitchell 2006) 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 can accommodate 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. Our SpeClustering algorithm has an extension to assign different weightings to different feature sets, so it can handle word and person features jointly. In order to simulate the social

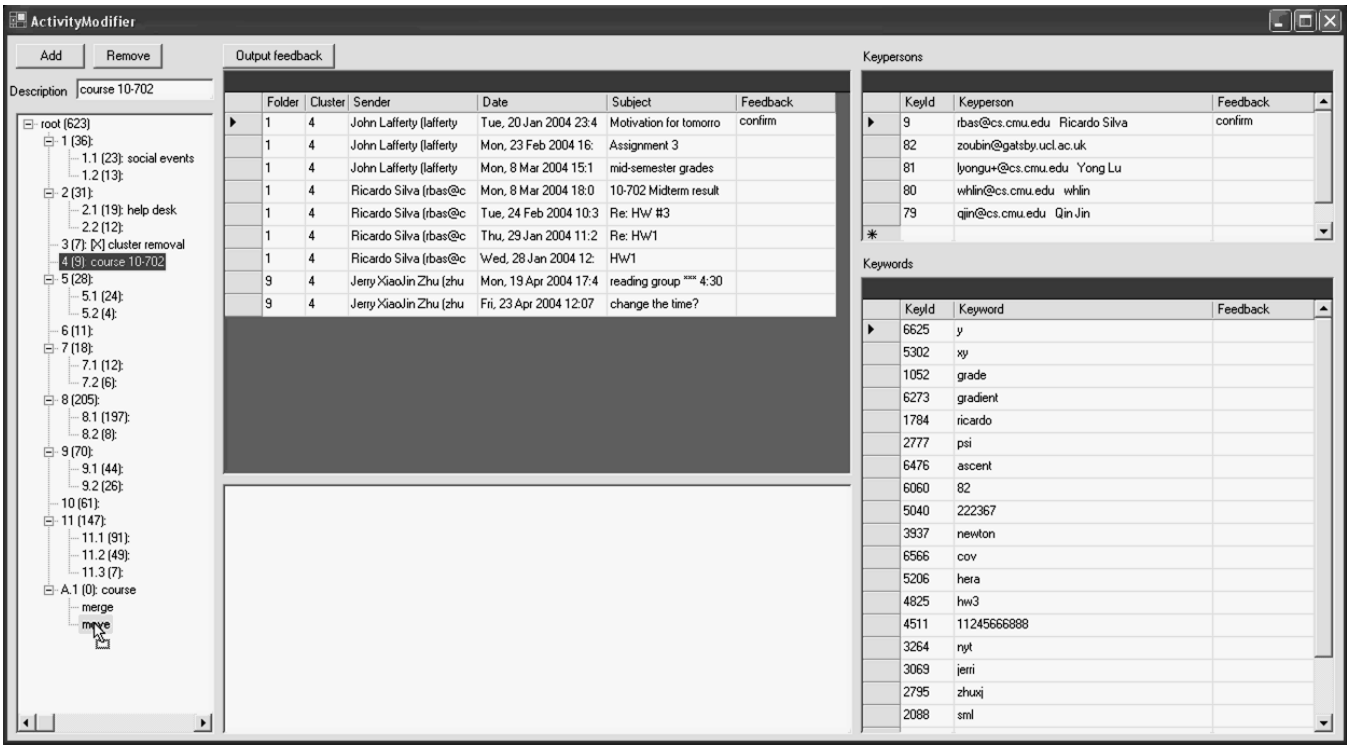


Figure 1: An interface that presents hierarchical results to a user and allows various types of hierarchical and non-hierarchical user feedback. The left panel shows the resulting hierarchy. When a user selects a cluster in the hierarchy, the middle top panel shows a list of emails in this cluster, and the middle bottom panel would show content of an email chosen by the user (blank here for privacy reasons). 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 therefore added a "course" cluster, A.1, and is moving cluster 4 underneath cluster A.1.

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, to emphasize these counts relative to word counts. Note our algorithm for retraining the hierarchical clustering in the phase of user feedback does not use social network analysis, because it is not obvious how to perform social analysis sub-clustering while respecting the constraints imposed by user feedback. This weighting of person features provides a mechanism to amplify the importance of grouping together emails involving the same people, and is thus analogous to social network analysis, but fits into our SpecClustering algorithm which can accept constraints imposed by user feedback.

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 "StructPrior" parameter that indicates the machine's belief that the user left documents at the correct locations in the hierarchy. The value of the "StructPrior" parameter is used to initialize the posterior probability distributions among sibling clusters. When the parameter is 1, the algorithm preserves these document-to-cluster assignments in model initialization. When the parameter value is lower, the re-training algorithm is more likely 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 is incompatible with our goal of efficient interaction between machine and user.

Distance Measurement between Hierarchies

One important consideration in mixed-initiative hierarchical clustering is evaluating the results (including initial hierarchical clustering results, hierarchies modified after user feedback, and re-trained hierarchical clustering results.) Our approach is to define an edit distance measure between two hierarchies, then to evaluate any proposed clustering hierarchy by its edit distance to the correct (reference) hierarchy. We obtain this reference hierarchy from the user in our experiments.

To see the difficulty in comparing two hierarchies, consider the two hierarchies, Reference and Clustering Result, in Figure 2 and the question of how to align these two hierarchies. It is not difficult to align the left-hand side subtrees. Figure 2(a) shows cluster (circle node) 2, 3, 6, and 7 can be aligned and it results in clustering errors of document (triangle node) 13, 19, and 22. However, the alignment of right-hand side subtrees is not so trivial. Figure 2(b) shows two possible mappings from Clustering Result to Reference. Alignment 1 sticks to the hierarchical constraints imposed by Clustering Result, while Alignment 2 violates the constraints but has higher precision and recall at the document level.

At first, this seems to be a 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.

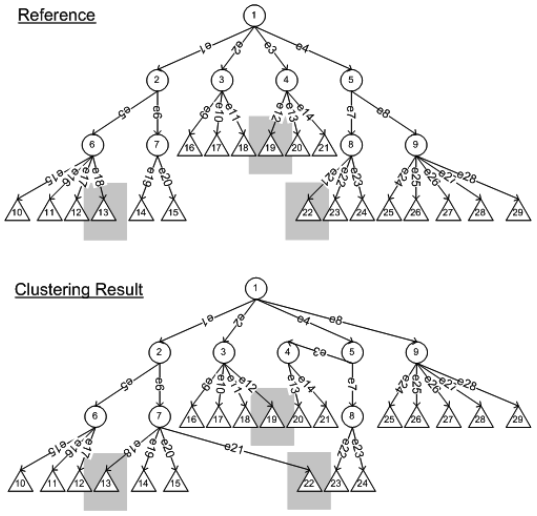
Instead of calculating hierarchical similarities, the distance between two hierarchies can be measured by the number of editing operations needed to change one hierarchy into the other. This measurement is more suitable for our mixed-initiative scenario because the set of hierarchical feedback types can be transformed naturally into a set of editing operations. For example, "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.

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 Clustering Result hierarchy 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.

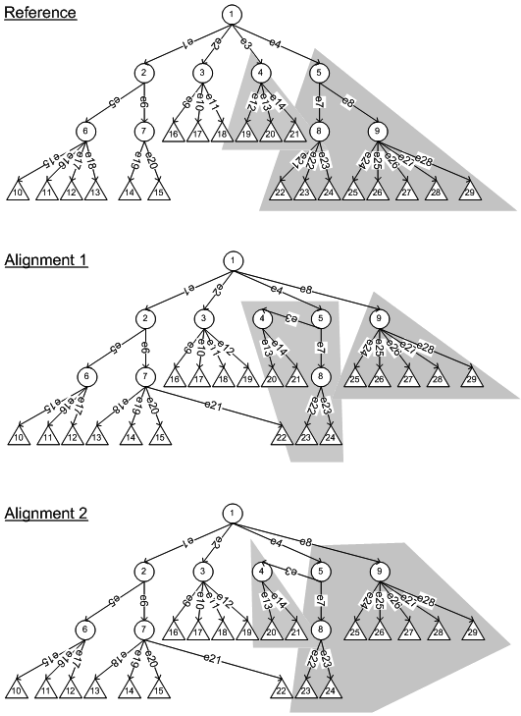
The concept of "edge modification ratio" is very similar to "tree edit distance" (Bille 2003) where different feedback types are mapped into different operations and the cost function is uniform.

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 had been manually organized as a hierarchy prior to our experi-



(a) The 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 the right-hand side sub-tree of a clustering result to a reference.

Figure 2: Hierarchical structure comparison

ments. We use this as the reference hierarchy. It consists of 15 nodes including a root, 11 leaf folders, and 3 intermediate nodes. It contains 6684 unique words and 135 individual people (email addresses).

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 Figure 1 to browse and give feedback.

Our algorithm for producing initial clusters is non-deterministic. We picked five initial first-level clustering results with the highest likelihood values among fifty results. Each of these five single-level clustering results were then extended to two-level hierarchical clusterings by applying social network analysis. For each of these five initial hierarchical clusterings, the email owner performed a diligent feedback session and a lazy feedback session on different days. In the diligent session, the user examines keywords and keypersons in detail, and often checks document assignments to clusters. In the lazy session, the user may select a few keywords and keypersons, and skims through or skips documents.

The first row in Figure 3 shows an example of the mixed-initiative interactions, derived from a single initial clustering result plus its "diligent" or "lazy" feedback sessions. The horizontal axis in each plot corresponds to minutes of user feedback, whereas the vertical axis gives the quality of the hierarchical clustering at that point in time, measured by its edge modification ratio relative to the reference (ideal) clustering. The histogram below each plot shows the cumulative count of user feedback, with colors indicating the different types of feedback. Notice the diligent feedback sessions are longer and involve larger total counts of feedback from the user.

The dot-marked (black) lines in Figure 3 show how user feedback modifies the quality (the edge modification ratio) of the initial hierarchical clustering over time. Lower edge modification means a user can achieve the reference hierarchy using fewer remaining feedback steps. The other lines show edge modification ratios for hierarchical results after re-training. Each marked point represents a retrained model learnt from the user feedback on the initial 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. We can also compare a circle-marked (red) line with the square-marked (cyan) line in a same plot. The StructProb parameter can be interpreted as the level of trust placed upon the user modified hierarchy where a higher value means more trust. The circle-marked lines show the re-trained results with a low trust value, and square-marked lines show results with a high trust value.

For this specific initial clustering result, a diligent user can benefit from the machine's retraining when she provides less than 7 minutes' feedback. After the 8th minute, the re-trained result only maintains performance similar to the user's manual efforts. This is because if the user has meticulously corrected all document-to-cluster errors occurred in the initial hierarchy, the re-trained model can predict at best what a user has manually done. On the other hand, the re-trained results from a lazy user's feedback gets better after the user completes the cluster-level corrections, e.g., there is no cluster-level feedback after the 4th minute. In terms of comparing the performances between different user behaviors, in this specific case, a diligent user needs to spend 11 minutes correcting the cluster hierarchy if they work alone, whereas a lazy user with the machine's assistance achieved an equivalent performance in four minutes.

Figure 3(c) and 3(d) show the aggregated re-trained results from four out of five initial hierarchies with respect to two different feedback sessions. The general trends hold that a diligent user gains marginally and a lazy user gains more from model re-training.

Notice the edge modification ratio of the black line (user feedback) at the zeroth minute in Figure 3 gives the performance of the two-step approach (flat clustering plus social network analysis) for generating initial hierarchical results. We have learnt that this approach can integrate different aspects of email content to produce user comprehensible subclusters successfully. The reason we switch to the Cascading SpeClustering model for model retraining is because the two-step approach cannot utilize various types of user feedback. However, given no user feedback and without weighting the person corpus heavily, e.g., PersonWeight equals one, the edge modification ratio is much worse than the two-step approach. This confirms our previous study that social network analysis helps generate more user-comprehensible clusters (activities) and also shows the Cascading SpeClustering model cannot beat the good heuristics without user feedback. With PersonWeight set to 100, the Cascading SpeClustering model is more capable of achieving results similar to social network analysis. Given the same StructPrior value, weighting the person corpus heavily results in a lower edge modification ratio in the early stage and using no special person weighting is better in the later stage. It shows that when the user's feedback on a hierarchy is partial, the background knowledge of social cliques is informative and when the user's feedback has fixed the hierarchical structure, the textual content is more helpful.

The StructProb parameter can be interpreted as the level of trust placed upon the user modified hierarchy where a higher value means more trust. The aggregated results in Figure 3(c) and 3(d) show that it is better to assume the document-to-cluster assignments in the user modified hierarchy are correct and initiate the re-trained model accordingly.

However, the performances of re-trained results vary greatly depending on the quality of initial clustering results and the personal style of giving feedback. The re-trained results using the 5th initial result is shown in Figure 4. These results are a lot worse and spikier than the other four pairs of re-trained results. This initial result has two big clusters and each of these two clusters has documents belonged to two or three reference clusters. With our current feedback types, the user must resort to a sub-optimal feedback strategy like confirming it as one reference folder and moving the other

half of documents to a newly created cluster. When a lazy user doesn't provide enough user feedback to teach a machine, the re-trained model may converge incorrectly as in Figure 4(b). A better solution is to implement the cluster-split feedback type, which we plan in future work. In addition, we will also study re-training a subset of the hierarchy instead of re-training the entire hierarchy.

Conclusions

In this paper, we propose an approach to mixed-initiative hierarchical clustering, and apply it to hierarchical clustering of email. We previously observed that hierarchical clustering helps a user understand the results better than flat clustering. In this study, we found that hierarchical feedback types such as merging clusters and adding clusters helps a machine learn better hierarchical models.

In order to evaluate hierarchical clustering quality, we defined "edge modification ratio" to compare hierarchical clusterings against a reference hierarchy. This measurement computes the ratio of edited edges in the sequence of optimal complete user feedback, to the total number of edges in the reference hierarchy.

We have applied a simple hierarchical clustering model, Cascading SpeClustering, to the mixed-initiative email clustering task. The experimental results show that the joint efforts of a machine and a user can usually achieve better edge modification ratio, or equivalently, save a user's time compared to manually editing the initial clustering. We also learned that a user's feedback style matters. A lazy user can gain more benefits from machine's retraining than a diligent user. From the worst case of re-trained results, it seems safer to re-train a specified subset of a hierarchy instead of the whole hierarchy. We will further study this issue.

Acknowledgments

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

References

Bille, P. 2003. Tree edit distance, alignment distance and inclusion. Technical report, IT University.

Blei, D. M.; Griffiths, T. L.; Jordan, M. I.; and Tenenbaum, J. B. 2003. Hierarchical topic models and the nested chinese restaurant process. In *NIPS 16*.

Huang, Y., and Mitchell, T. 2006. Text clustering with extended user feedback. In *SIGIR*.

Huang, Y. 2007. Mixed-initiative clustering and its application to workstation activity extraction. Thesis Proposal.

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

McCallum, A. K.; Rosenfeld, R.; Mitchell, T. M.; and Ng, A. Y. 1998. Improving text classification by shrinkage in a hierarchy of classes. In *ICML*.

Mitchell, T. M.; Wang, S. H.; Huang, Y.; and Cheyer, A. 2006. Extracting knowledge about users' activities from raw workstation contents. In *AAAI-06*.

Sun, A., and Lim, E.-P. 2001. Hierarchical text classification and evaluation. In *ICDM*.

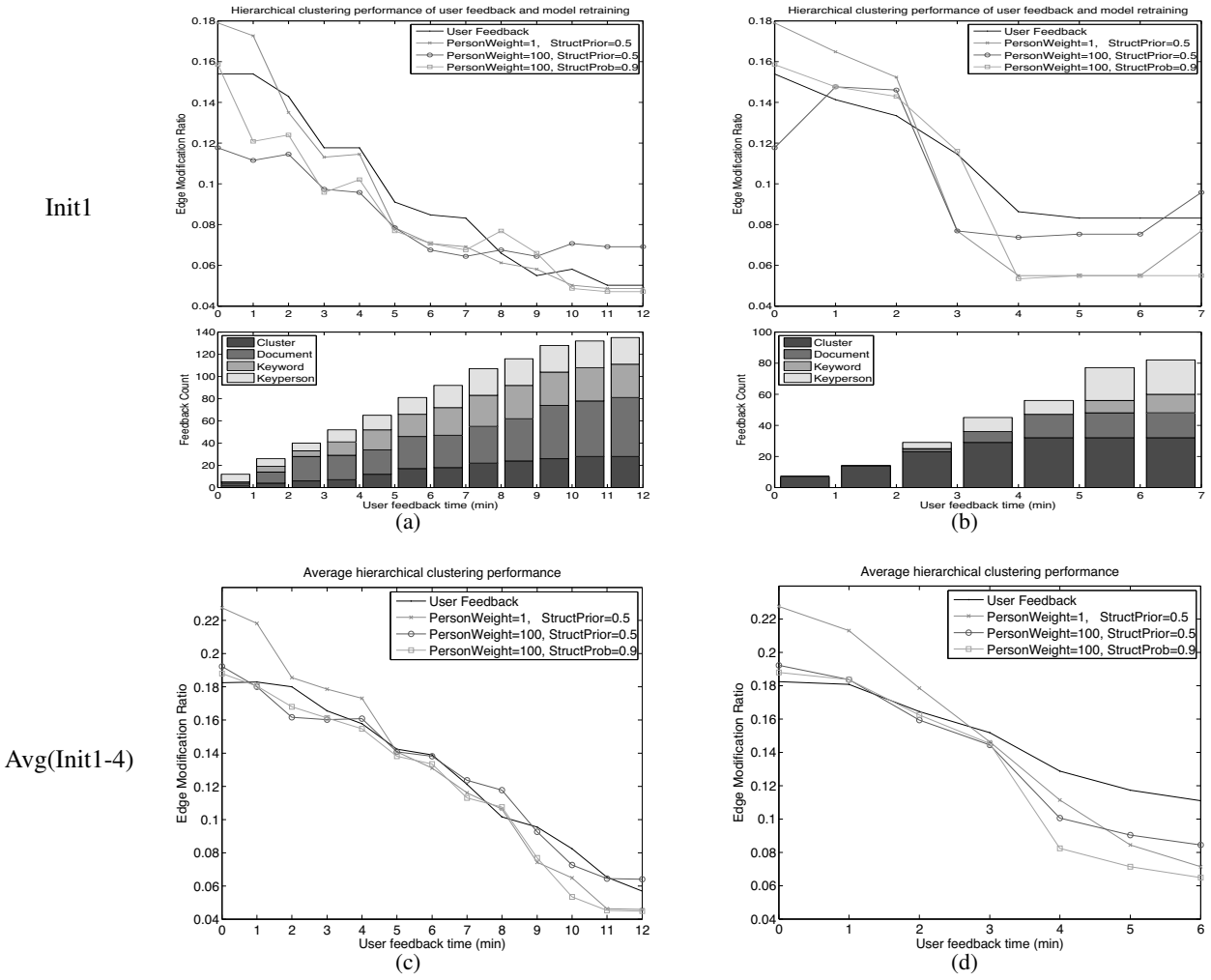


Figure 3: (a)(b): An example pair of re-trained results derived from a specific initial result and feedback counts of two feedback sessions on this initial result. (c)(d): Averages of 4 re-trained results in different feedback sessions.

Diligent Session

Lazy Session

Init5

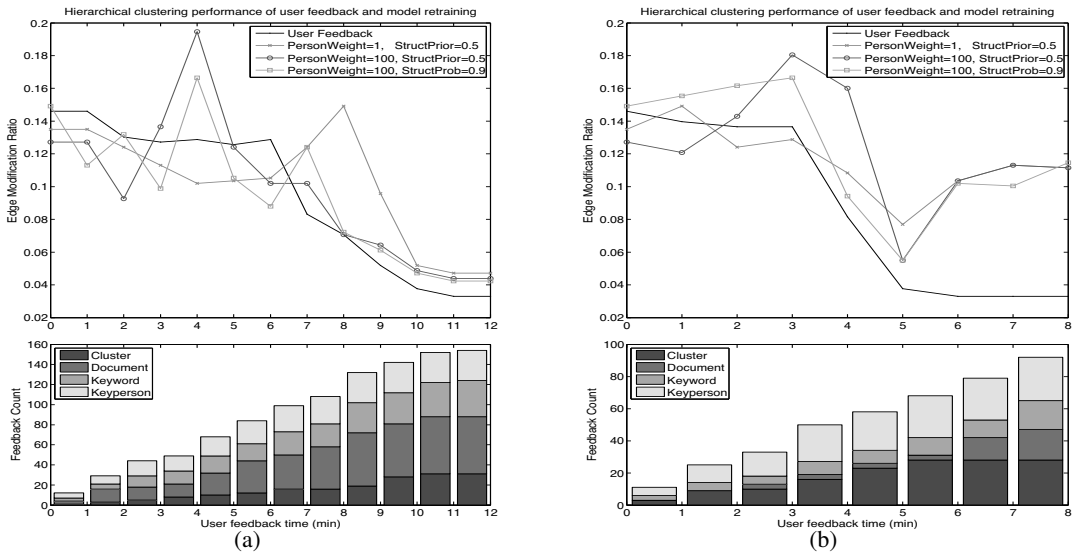


Figure 4: This pair of re-trained results shows that our current model re-training method doesn't always help the user.