Text Clustering with Extended User Feedback

Yifen Huang
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, Pennsylvania USA
hyifen@cs.cmu.edu

Tom M. Mitchell
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, Pennsylvania USA
tom.mitchell@cs.cmu.edu

ABSTRACT

Text clustering is most commonly treated as a fully automated task without user feedback. However, a variety of researchers have explored mixed-initiative clustering methods which allow a user to interact with and advise the clustering algorithm. This mixed-initiative approach is especially attractive for text clustering tasks where the user is trying to organize a corpus of documents into clusters for some particular purpose (e.g., clustering their email into folders that reflect various activities in which they are involved). This paper introduces a new approach to mixed-initiative clustering that handles several natural types of user feedback. We first introduce a new probabilistic generative model for text clustering (the SpeClustering model) and show that it outperforms the commonly used mixture of multinomials clustering model, even when used in fully autonomous mode with no user input. We then describe how to incorporate four distinct types of user feedback into the clustering algorithm, and provide experimental evidence showing substantial improvements in text clustering when this user feedback is incorporated.

Keywords

text clustering, user feedback, mixed-initiative learning

1. INTRODUCTION

We as human beings are quite familiar with clustering objects into categories based on features of these objects. For example, a computer user may sort her emails into folders that are personally meaningful because each one represents a particular activity she is involved in, or because they are emails from a particular group of people, etc. For each folder, or cluster, the user may have in mind a rich category description, but assigns objects to these categories based on their surface features (e.g., the words in the email, or recipients in the header). There are many other examples: we may informally cluster news stories into categories such as sports, politics, etc., or we may easily recognize in a supermarket what type of products a corridor belongs to.

Computer algorithms for clustering are typically cast as fully automated, unsupervised learning algorithms; that is, the algorithm is given only the collection of instances and the surface features that describe each, without any information about the nature of the clusters. Recently, however, a variety of researchers have studied ways of allowing a user to provide limited information to improve clustering quality. One approach is to allow the user to provide cluster labels for some of the instances, indicating which cluster that instance belongs to. For example, [11][2][8] use labels of this type to form initial cluster descriptions, which are then refined using both the unlabeled and labeled instances. A second type of input information consists of pair-wise constraints among instances. These constraints may assert that two documents must belong to the same cluster without indicating which one it is, or may assert that two documents must belong to different clusters. Various constraint-based methods and distance-based methods have been proposed to use this type of information. See [1] for a short survey on different approaches and also for an approach to integrating distance-based and constraint-based approaches into a probabilistic framework. A third type of additional input involves background knowledge to enrich the set of features that describe each instance. For example, [6] enriches their document representation by using an ontology (WordNet) as background knowledge. A fourth type of extra information, which we are primarily interested in, is information about the key surface features for a particular class, or cluster. For example, [9] uses a few user-supplied keywords per class and a class hierarchy to generate preliminary labels to build an initial text classifier for the class. [10] proposes an interesting technique in which they ask a user to identify interesting words among automatically selected representative words for each class of documents, and then use these user-identified words to re-train the classifier as in [9].

Researchers working on active learning have also studied using feedback about key features. For example, [5] converts a user-recommended feature into a mini-document which is used to help train an SVM classifier. An altenative approach to using this information is proposed by [12] who adjust the SVM weights associated with these key features to a predefined value in binary classification tasks.

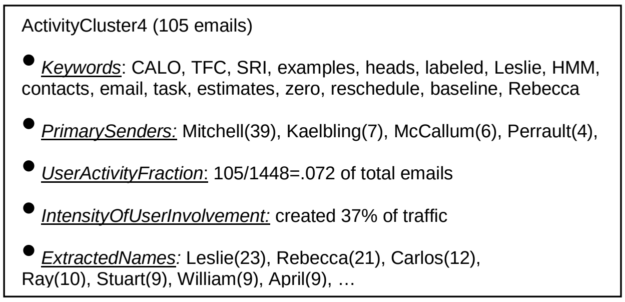
We are interested in how to best incorporate user input into automated clustering algorithms, and more generally into mixed-initiative clustering approaches that allow the user and computer to jointly arrive at coherent clusters that

ACM SIGIR '06 Seattle, Washington USA

capture the categories of interest to the user. Note this goal of discovering clusters of interest to the user is somewhat different from the objective optimized in totally unsupervised clustering algorithms that attempt instead to maximize some statistical property of the clusters (such as data likelihood, or inter-cluster distance). We are specifically interested in how to incorporate into clustering algorithms the user's emerging understanding about a **category**[1], stimulated by seeing the instances that are clustered together, and by seeing (and editing) summaries of these emerging clusters. A user's understanding about a category may be expressed in a variety of forms, such as by keywords, important person names, other types of entities, and relationships among entities. It may encapsulate a variety of types of information, and it may be difficult for a user to articulate fully their notion of the cluster.

The chief contribution of this paper is to introduce a new probabilistic model for clustering that outperforms standard unsupervised clustering in our experiments, and that also can accomodate a variety of types of user feedback to iteratively refine the clusters. We present experiments in both an email clustering domain, and in a second document clustering domain (20 Newsgroups) showing the performance of this clustering approach.

The research we report here is part of our larger research effort to build computer algorithms to automatically infer the key activities, or projects, a user is involved in, given the contents of their workstation (e.g., their emails, files, directories, calendar entries, personal contacts lists, etc.). For example, a user may be involved in activities such as teaching a particular course, participating in a particular committee, hanging out with a particular group of friends, etc. In our previous work[7], we have shown that unsupervised clustering of emails can result in useful descriptions of user activities, such as the one shown in Figure 1. The work we report in this paper is motivated in part by our interest in developing a more mixed-initiative approach to inferring such activity clusters, using both computer analysis of workstation data and user feedback based on examining proposed clusters.



ActivityCluster4 (105 emails)

- *Keywords*: CALO, TFC, SRI, examples, heads, labeled, Leslie, HMM, contacts, email, task, estimates, zero, reschedule, baseline, Rebecca
- *PrimarySenders:* Mitchell(39), Kaelbling(7), McCallum(6), Perrault(4),
- *UserActivityFraction*: 105/1448=.072 of total emails
- *IntensityOfUserInvolvement:* created 37% of traffic
- *ExtractedNames:* Leslie(23), Rebecca(21), Carlos(12), Ray(10), Stuart(9), William(9), April(9), ...

Figure 1: An example output of activity extractor, which is extracted statistically from unsupervised clustering results.

We will describe our probabilistic model and the associated clustering algorithm in the next section. Section 3 then discusses how several types of user feedback can be incorporated into the clustering algorithm. The experimental setup and evaluation are described in section 4 and conclusions are presented in section 5.

2. SPECLUSTERING MODEL

2.1 Separating specific from general topics

We present here a clustering algorithm based on a novel probabilistic model. One commonly used probabilistic model for text clustering is the multinomial naive Bayes model described in [11], which models a document as a vector of words with each word generated independently by a multinomial probability distribution conditioned on the document's class (i.e., conditioned on which cluster it belongs to). Our SpeClustering model also assumes words are generated probabilistically, but differs in an important way from this standard model. In particular, the SpeClustering model assumes that only *some* of the words in the document are conditioned on the document's cluster, and that other words follow a more general word distribution that is independent of which cluster the document belongs to. To see the intuition behind this model, consider a cluster of emails about skiing. There will be some words (e.g., "snow") that appear in this cluster of emails because the topic is skiing, and there will be other words (e.g., "contact") that appear for reasons independent of the cluster topic. The key difference between the standard model and our SpeClustering model is that our model assumes each document is generated by a mixture of two multinomials – one associated with the document's cluster, and the other shared across all clusters. As we show below, our more elaborate SpeClustering model can lead to improved accuracy when used for automatic clustering, and it also provides a formalism that can easily accomodate several important types of user feedback to support mixed-initiative clustering.

To construct this SpeClustering model, we extend the standard multinomial model in two ways. The first modification is to add a G topic variable that is intended to capture general topics not related to the cluster. The second modification is to introduce a hidden boolean variable, X, associated with each word O in each document. If $X = 1$, the observation O is generated by the cluster-specific topic S, and if $X = 0$, the observation O is generated by a general topic G. Throughout this paper we simplify the model by assuming there is only one general topic instead of multiple topics, so the value of G is fixed at $G = g$. Figure 2 shows the graphical model representation of the model. Here the outer rectangle (or plate) is duplicated for each of the D documents, and the inner plate is duplicated for each of the N observations O and associated variables X. Note the general topic G is constant across all documents and words, whereas the cluster topic S is different for each document.

The Speclustering model θ has four sets of parameters:

$$\pi_c = P(S = c)$$

$$\xi_c = P(X = 1|S = c)$$

$$\beta_{cf} = P(O = f|S = c)$$

$$\beta_{gf} = P(O = f|G = g)$$

[1] We use the word "cluster" to indicate a set of similar instances grouped together by a clustering algorithm, and the word "category" to indicate a concept in a user's mind which may or may not be reflected by some cluster of instances.

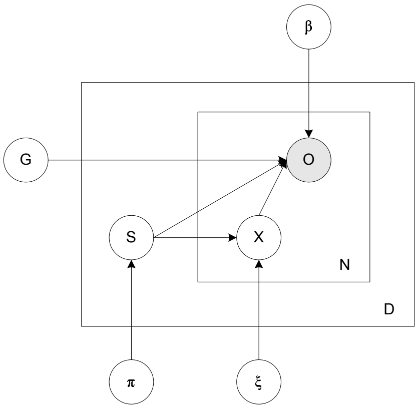


Figure 2: Graphical representation of SpeClustering model. S is a variable representing the cluster associated with a document, O represents an observed word in a document, and X is a boolean variable that indicates whether word O is generated conditioned on the cluster S or whether it is generated according to a cluster-independent general distribution of words G.

where $c \in \{1, 2, ..., |S|\}$, $g \in \{1\}$ for the simplified case and $f \in \{1, 2, ..., |O|\}$.

Given a corpus $\mathbb{C}$ that contains D instances $\mathbb{C} = \{d_1, d_2, ..., d_D\}$, and d_i is represented as a vector of observations $\{o_{ij}; j \in \{1, 2, ..., n_i\}\}$, we use the notation s_i to indicate the value of the hidden S variable for instance d_i and x_{ij} to indicate the value of the hidden X variable associated with observation o_{ij}. The corpus likelihood of $\mathbb{C}$ given θ is defined as follows:

$$\begin{aligned} P(\mathbb{C}|\theta) = & \prod_{i=1}^{D} \sum_{s_i=1}^{|S|} P(s_i) \\ & \prod_{j=1}^{n_i} [P(x_{ij} = 1|s_i)P(o_{ij}|s_i) + P(x_{ij} = 0|s_i)P(o_{ij}|g)] \end{aligned}$$

which can be written in terms of the model parameters as follows:

$$P(\mathbb{C}|\theta) = \prod_{i=1}^{D} \sum_{s_i=1}^{|S|} \pi_{s_i} \prod_{j=1}^{n_i} [\xi_{s_i} \beta_{s_i o_{ij}} + (1 - \xi_{s_i}) \beta_{g o_{ij}}]$$

Note the probability $P(X = 1|S = c, O = f; \theta)$, which can be derived from the model parameters, describes the probability that any particular feature f is generated by a particular cluster c, as opposed to the general topic g.

2.2 Learning Clusters with the SpeClustering Model

In the most general case we are interested in unsupervised clustering of documents given just the observed features O of a set of documents, where the values for the S and X variables are unobserved. Because of the existence of unobserved variables, we use an EM process [3] for parameter estimation. The EM algorithm is commonly applied to find a (locally) maximum-likelihood estimate of the parameters in situations when the observable data is incomplete and the model depends on unobserved latent variables. Given $\mathcal{X}$ and $\mathcal{Y}$ as the incomplete and complete data, the algorithm iterates through two steps: in the E step, we evaluate $Q(\theta|\theta^t) = E[\log P(\mathcal{Y}|\theta)|\mathcal{X}, \theta^t)]$, and in M step, we obtain new estimation of parameters $\theta^{t+1} = \arg\max_\theta Q(\theta|\theta^t)$. In our SpeClustering model, the incomplete data is $\mathcal{X} = \{o_{ij}\ \forall i \in \{1, ..., D\}\ j \in \{1, ..., n_i\}\}$ and complete data is $\mathcal{Y} = \{s_i, x_{ij}, o_{ij}\ \forall i \in \{1, ..., D\}\ j \in \{1, ..., n_i\}\}$. The exact estimation for each parameter in M step is listed below.

$$\pi_c^{t+1} = \frac{\sum_{i=1}^{D} \phi_i^t(c)}{D}$$

$$\xi_c^{t+1} = \frac{\sum_{i=1}^{D} \phi_i^t(c) \sum_{j=1}^{n_i} \psi_{ij}^t(c)}{\sum_{i=1}^{D} \phi_i^t(c) \cdot n_i}$$

$$\beta_{cv}^{t+1} = \frac{\sum_{i=1}^{D} \phi_i^t(c) \sum_{j=1}^{n_i} \delta(o_{ij} = v) \psi_{ij}^t(c)}{\sum_{i=1}^{D} \phi_i^t(c) \sum_{j=1}^{n_i} \psi_{ij}^t(c)}$$

$$\beta_{gv}^{t+1} = \frac{\sum_{i=1}^{D} \sum_{k=1}^{|S|} \phi_i^t(k) \sum_{j=1}^{n_i} \delta(o_{ij} = v) \psi_{ij}^t(k)}{\sum_{i=1}^{D} \sum_{k=1}^{|S|} \phi_i^t(k) \sum_{j=1}^{n_i} \psi_{ij}^t(k)}$$

where the following quantities are computed in the E step:

$$\begin{aligned} \phi_i^t(c) &\equiv P(s_i = c|d_i; \theta^t) \\ &= \frac{\pi_c^t \prod_{j=1}^{n_i} [\xi_c^t \beta_{co_{ij}}^t + (1 - \xi_c^t) \beta_{go_{ij}}^t]}{\sum_{k=1}^{|S|} \pi_k^t \prod_{j=1}^{n_i} [\xi_k^t \beta_{ko_{ij}}^t + (1 - \xi_k^t) \beta_{go_{ij}}^t]} \qquad (1) \end{aligned}$$

$$\begin{aligned} \psi_{ij}^t(c) &\equiv P(x_{ij} = 1|s_i = c, o_{ij}; \theta^t) \\ &= \frac{\xi_c^t \beta_{co_{ij}}^t}{\xi_c^t \beta_{co_{ij}}^t + (1 - \xi_c^t) \beta_{go_{ij}}^t} \qquad (2) \end{aligned}$$

By iterating through E step and M step, the likelihood will converge to a (local) maximum and values of parameters will be stabilized.

2.3 Extension to multiple types of features

In some cases instances may be described by multiple types of features. For example, when clustering emails we might describe each email by the set of words in its body, plus the set of email addresses the email is sent to. If there are multiple types of features in an instance, we can extend the SpeClustering model. Figure 3 shows the extended model with two feature types. The model adds one new block $\{Y, Q\}$ for the introduction of a new feature type. $\{Y, Q\}$ is identical and parallel to $\{X, O\}$. In the activity-discovery-via-emails task, we can apply this model to represent an activity in terms of both its key words and the primary participants of the activity.

Parameter estimation in the extended SpeClustering model is nearly identical to that described in section 2.2. The only exception is a change to the posterior probability estimate in Eq 1. The new posterior probability estimate in the extended model combines generative probabilities from multiple feature types. Eq 3 shows the estimate from two different feature types.

$$\phi_i^t(c) \equiv P(s_i = c|d_i; \theta^t)$$

$$= \frac{\pi_c^t \prod_{j=1}^{n_i} [\xi_c^t \beta_{co_{ij}}^t + (1 - \xi_c^t) \beta_{go_{ij}}^t] \prod_{h=1}^{m_i} [\eta_c^t \gamma_{cq_{ih}}^t + (1 - \eta_c^t) \gamma_{gq_{ih}}^t]}{\sum_{k=1}^{|S|} \pi_k^t \prod_{j=1}^{n_k} [\xi_k^t \beta_{ko_{kj}}^t + (1 - \xi_k^t) \beta_{go_{kj}}^t] \prod_{h=1}^{m_k} [\eta_c^t \gamma_{cq_{kh}}^t + (1 - \eta_c^t) \gamma_{gq_{kh}}^t]} \qquad (3)$$

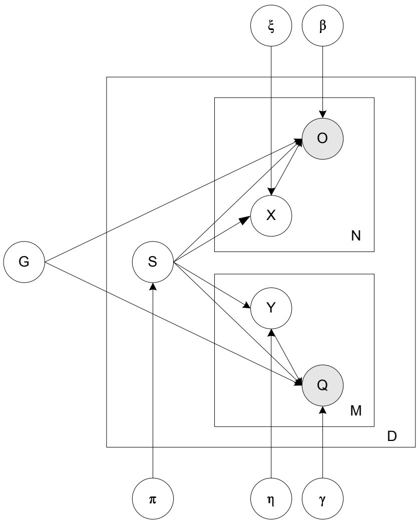


Figure 3: Graphical representation of the SpeClustering model with two feature types, where O and Q are observations with different feature types and X and Y are boolean variables deciding whether their respective observation is generated from the specific topic S or the general topic G

3. MODEL ADAPTATION ACCORDING TO USER'S FEEDBACK

As discussed earlier, we are particularly interested in allowing extended forms of user feedback to help direct the clustering process. In this section we discuss how several types of user feedback are incorporated to guide the clustering algorithm. We describe each feedback type in terms of the task of clustering emails to discover descriptions of a user's activities. The types of user feedback allowed are:

1. Remove activity cluster S
2. Email E belongs (or does not) to activity cluster S
3. Keyword W belongs (or does not) to activity cluster S
4. Person H belongs (or does not) to activity cluster S
5. A short text description for activity cluster S is Y

There are two posterior probabilities in the SpeClustering model that turn out to be highly related to the above types of feedback. To be more specific, type 1 and 2 feedback are related to Eq. 3 and type 3, 4, and 5 feedback are related to Eq. 2.

There are two methods to initialize the SpeClustering model with user feedback. The simple method inherits previous clustering results which the user gives her feedback upon and for removed clusters, we reset the initial value of $P(s_i|d_i;\theta^t)$ by distributing the probability mass uniformly among all clusters and halving the probability for the cluster S. The joint method uses feedback jointly to initialize the model. We first select several documents that have the highest cosine similarity with confirmed documents and keywords (where we treat keywords as a mini-documents) associated with current clusters. We then search for a small set of similar documents that maximize inter-cluster distances and replace any cluster that is removed in the feedback.

We perform type 2 to 4 adjustments during each EM iteration while training the SpeClustering model. For type 2 feedback, we adjust the value of $P(s_i = c|d_i;\theta^t)$ to be one if the email-to-cluster bound is confirmed by the user or set it to zero if the bound is disapproved by the user. Proper adjustment to normalize posterior probabilities of $\{P(s_i = c'|d_i;\theta^t)\ \forall c' \neq c\}$ is also required in this case. For type 3 and 4 feedback, we adjust the value of $P(x_{ij} = 1|s_i = c, o_{ij} = v;\theta^t)$ to be one if the keyword/person-to-activity bound is confirmed by the user or set it to zero if the bound is disapproved by the user. For type 5 feedback, we tokenize the description Y and make each token of Y a confirmed keyword as in type 3 feedback.

Figure 4 summarizes this Mixed-Initiative-Clustering process which integrates user feedback into the clustering process.

```
Algorithm: Mixed-Initiative-Clustering
Input: Corpus $\mathcal{C}$ with D instances.
Output: A list of activity clusters $\mathcal{A}$, where each activity
cluster is described by its top K features for each feature
type.
Method:

  1. Generate initial model $\theta^{ini}$ and summarization of
     clusters $\mathcal{A}^{ini}$ with top K features of each feature
     type. $\theta^t = \theta^{ini},\ \mathcal{A}^t = \mathcal{A}^{ini},\ \mathcal{F} = \{\}$.
  2. Add user's feedback regarding $\mathcal{A}^t$ into $\mathcal{F}$.
  3. $(\theta^{t+1}, \mathcal{A}^{t+1})$ =Speclustering-with-Feedback($\mathcal{C}, \theta^t, \mathcal{F}$).
  4. $\theta^t = \theta^{t+1},\ \mathcal{A}^t = \mathcal{A}^{t+1}$
  5. repeat step 2 to 4 until user's satisfaction.

Algorithm: Speclustering-with-Feedback
Input: Corpus $\mathcal{C}$ with D instances. $\theta^t$ as the current
model. $\mathcal{F}$ as the collection of user's feedback.
Output: $\theta^{t+1}$ as the model after adaption according to
user's feedback. $\mathcal{A}^{t+1}$ as the new summarization of clusters
according to $\theta^{t+1}$.
Method:

  1. Estimate posterior probabilities $\mathcal{P}^t$ of Eq 3 and Eq 2
     given $\mathcal{C}$ and $\theta^t$.
  2. Adjust $\mathcal{P}^t$ according to $\mathcal{F}$ to obtain $\mathcal{P}^t_{adj}$.
  3. Re-estimate model parameters using $\mathcal{P}^t_{adj}$ to obtain
     $\theta^t_{adj}$.
  4. $\theta^t = \theta^t_{adj}$; repeat step 1 to 3 until the model
     converges.
  5. $\theta^{t+1} = \theta^t_{adj}$. Generate $\mathcal{A}^{t+1}$ according to $\theta^{t+1}$ and $\mathcal{F}$.
```

Figure 4: The algorithm for mixed-initiative clustering.

3.1 Applying to Supervised Classification

We have described details of the SpeClustering model. However, the model is not restricted to clustering. It can also be applied to supervised classification tasks. The difference in classification is that the topic variable S is no longer a hidden variable. We can treat the classification tasks as knowing all the type 2 user feedback and replace the estimate of posterior probabilities $P(s_i = c|d_i; \theta^t)$ with the true value specified by the instance label.

4. EXPERIMENTS

4.1 Datasets

To test the SpeClustering algorithm we used two data sets. The first is an email dataset ($EmailYH$) from one of the authors that contains 623 emails. This dataset had previously been sorted into 11 folders and contains 6684 unique words and 135 individual people after pre-processing[2]. The second data set is the publicly available 20-Newsgroups collection. This data set contains text messages from 20 different Usenet newsgroups, with 1000 messages harvested from each newsgroup. We derived three datasets according to [1]. The first, *News-Similar-3*, consists of messages from 3 similar newsgroups (comp.graphics, comp.os.ms-windows.misc, comp.windows.x) where cross-posting occurs often between these three newsgroups. *News-Related-3* consists of messages from 3 related newsgroups (talk.politics.misc, talk.politics.guns and talk.politics.mideast). *News-Different-3* contains 3 newsgroups of quite different topics (alt.atheism, rec.sport.baseball, and sci.space).

We use only the text part of messages in the three newsgroup datasets because a reviewer won't have the knowledge needed to decide which author is the key-person with regard to which newsgroup. For the text part, we applied the same pre-processing we used in ($EmailYH$). There are 3000 messages in these datasets. *News-Different-3* contains 8465 unique words, *News-Related-3* contains 9998 unique words and *News-Similar-3* has 10037 unique words.

4.2 Measurement for Cluster Evaluation

We use two measurements to estimate cluster quality: folder-reconstruction accuracy, and normalized mutual information (NMI) [4].

In order to calculate the folder-reconstruction accuracy, we search through all possible alignments of cluster indices $\mathcal{I}_c$, to folder indices $\mathcal{I}_f$ in order to find the alignment resulting in optimal accuracy, then report the accuracy under this optimal alignment:

$$Acc = max_A \frac{\sum_{i=1}^{D} \delta(A(s_i) = f_i)}{D} \qquad A \in \{Map(\mathcal{I}_c) \overset{1-to-1}{\longrightarrow} \mathcal{I}_f\} \tag{4}$$

The normalized mutual information measurement is defined as Eq. 5, where $I(S;F)$ is the mutual information between cluster assignment S and folder labels F, $H(S)$ is the entropy of S and $H(F)$ is the entropy of F. It measures the shared information between S and F.

$$NMI = \frac{I(S;F)}{(H(S) + H(F))/2} \tag{5}$$

These two measurements are correlated but show different aspects of clustering performance. Accuracy calculates the ratio between major chunks of clusters to its reference. NMI measures the similarity between cluster partitions and reference partitions.

4.3 Results and Discussion

To experimentally study the SpeClustering model and algorithms, we consider three distinct algorithms. First, we consider the standard multinomial naive Bayes text clustering[11] algorithm as a baseline approach representing a typical probabilistic approach to text clustering. We modified this baseline approach by allowing it to search for a good cluster initialization and to avoid situations in which one cluster gets eliminated during the EM iterations[7]. Two versions of SpeClustering algorithm are tested. The fist version is the original SpeClustering algorithm as described in Section 2. The second version, *SpeClustering-bound*, adds range constraints on parameter values ξ: for word features, the range is $[0.1, 0.4]$ and for person features, the range is $[0.6, 0.9]$. The reason for introducing these range constraints is to avoid situations where some values of ξ converge to 1 or 0. This is undesirable because the value of ξ reflects the percentage of specific features ($X = 1$) occuring over all observations. Both SpeClustering algorithms are initialized using the output from the baseline naive Bayes clustering.

4.3.1 Autonomous Clustering

First we compared our SpeClustering approach to the Naive Bayes baseline in fully autonomous clustering without user feedback. We made 50 individual runs on $EmailYH$ dataset and 20 runs each on *News-Similar-3*, *News-Related-3*, and *News-Different-3*. Table 1 shows the average accuracy and NMI results of different datasets and the three clustering algorithms. Notice in all datasets, the SpeClustering algorithm performs better than the naive Bayes algorithm, and the SpeClustering-bound model performs better than SpeClustering. The naive Bayes clustering results are used to initialize its associated SpeClustering and SpeClustering-bound runs, so the performance gain are directly due to the difference between the SpeClustering probabilistic model and the and naive Bayes model. When we examined the details of individual runs, we found that every one of the runs resulted in SpeClustering-bound outperforming Naive Bayes in terms of the NMI measure, and that in the vast majority of these runs it also outperformed Naive Bayes in terms of the accuracy measure.

4.3.2 Clustering with User Feedback

We next studied the impact of user feedback on the bounded SpeClustering model. In particular, we chose 5 clustering results using the multinomial naive Bayes model with the best log-likelihood among 50 runs on $EmailYH$ and presented each of these to the user. We also chose one best run from 20 runs on *News-Different-3*, *News-Related-3*, and *News-Similar-3*. The user gave feedback using the interface shown in Fig 5. The top left panel shows a list of documents that are clustered into the selected cluster label, the top right panel shows 5 key-persons of the cluster and the bottom left panel shows 20 keywords of the cluster. The keywords and key-persons of the cluster are selected using a Chi-squared measurement [13]. When a user clicks on a document in the document list, the content of the document

[2] The pre-processing for words includes stemming, stop word removal and removal of words that appear only once in the dataset. The pre-processing for people contains reference-reconciliation over email senders and recipients, and removal of people that are involved in only one email.

dateset	method	Accuracy (%)	NMI (%)
Email-YH	naive Bayes	48.44 ± 7.01	48.02 ± 3.93
	SpeCluster	52.28 ± 8.61	53.25 ± 5.65
	SpeC-bound	$\mathbf{53.98 \pm 8.04}$	$\mathbf{56.25 \pm 4.90}$
News-Sim-3	naive Bayes	46.31 ± 7.21	9.86 ± 7.34
	SpeCluster	51.38 ± 6.33	15.80 ± 6.82
	SpeC-bound	$\mathbf{51.98 \pm 5.91}$	$\mathbf{16.46 \pm 6.27}$
News-Rel-3	naive Bayes	60.18 ± 10.64	34.36 ± 10.58
	SpeCluster	60.61 ± 11.08	36.06 ± 10.71
	SpeC-bound	$\mathbf{61.14 \pm 11.41}$	$\mathbf{36.92 \pm 11.04}$
News-Diff-3	naive Bayes	91.24 ± 13.45	79.76 ± 14.56
	SpeCluster	93.80 ± 11.49	83.57 ± 14.27
	SpeC-bound	$\mathbf{96.52 \pm 6.47}$	$\mathbf{87.79 \pm 11.56}$

Table 1: Clustering results of different datasets and different clustering algorithms. SpeCluster and Spec-bound are short-hands of the SpeClustering model with unbounded and bounded parameter values. Both versions of the SpeClustering model out-perform the multinomial naive Bayes model and the bounded SpeClustering model achieves the best performance.

shows in the bottom left panel. The user can give various types of feedback described in Section 3 and the interface displays feedback the user has entered so far. The user can also go back and forth to correct conflict assumptions she has made to achieve consistent cluster interpretations.

An interesting observation we found is that displaying keywords and key-persons tremendously helps the user make judgements about a cluster. In fact, to decide the meaning of a large cluster based only on examining the documents is extremely difficult. A reviewer would tend to decide based on the first several documents she goes through even when the cluster contains more than hundreds of documents, and the biased decision often causes conflicts with later clusters. The reviewer usually chooses to remove a cluster, if the keywords and key-persons don't show any consistency and are not meaningful to the user, or if documents in the cluster are a hodgepodge from several categories. If the keywords or key-persons make sense to the user, the user gives feedback about document-cluster associations according to these meanings. We don't put constraints on how the reviewer does the feedback, so the reviewer can make decisions freely based on how she perceives the clustering results, and gives feedback using her own interpretation of the results.

This way of collecting feedback may result in a situation where the meaning in the reviewer's mind doesn't match the majority of documents associated with the cluster because the reviewer rationalizes clusters mostly according to key features. Keyword selection favors words that occur in the cluster and don't appear in other clusters, so if a category contains many documents and gets spread out to several clusters, even the majority of documents in the cluster belong to that category, the keyword selection may give low scores to words belong to that category because those words appear in other clusters.

We use the following notation to indicate various feedback types:

- CR: cluster removal
- PP: document-to-cluster association

run	doc #	CR	PP	WX	HX
Email1	623	3	99	37	30
Email2	623	3	73	35	31
Email3	623	4	92	48	26
Email4	623	7	32	28	15
Email5	623	4	91	43	28
Sim1	3000	2	39	9	-
Rel1	3000	1	29	20	-
Diff1	3000	0	16	39	-

Table 2: Entry numbers of different feedback types for 5 selected naive Bayes runs.

- WX: keyword-to-cluster association
- HX: keyperson-to-cluster association

Table 2 shows how many entries of different feedback types the reviewer enters for each selected run. The reviewer spends about 15 mins to finish one run from *EmailYH* dataset and 5-10 mins to finish one run from newsgroup datasets.

We ran the SpeClustering-bound algorithm with user feedback and compared the results to the naive Bayes baseline and the SpeClustering-bound algorithm without feedback. The difference between SpeClustering with and without feedback is the parameter adjustment described in Section 3.

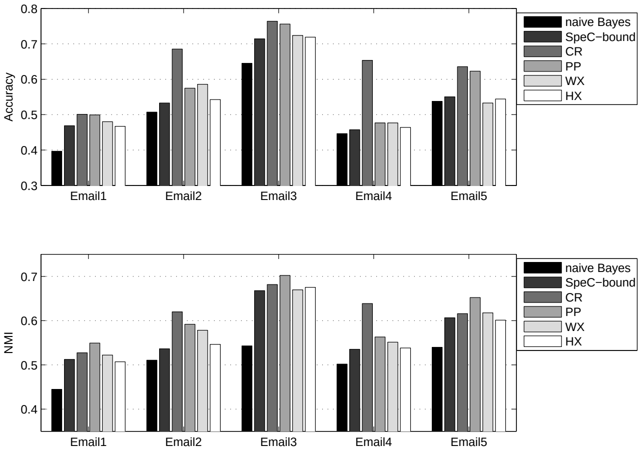
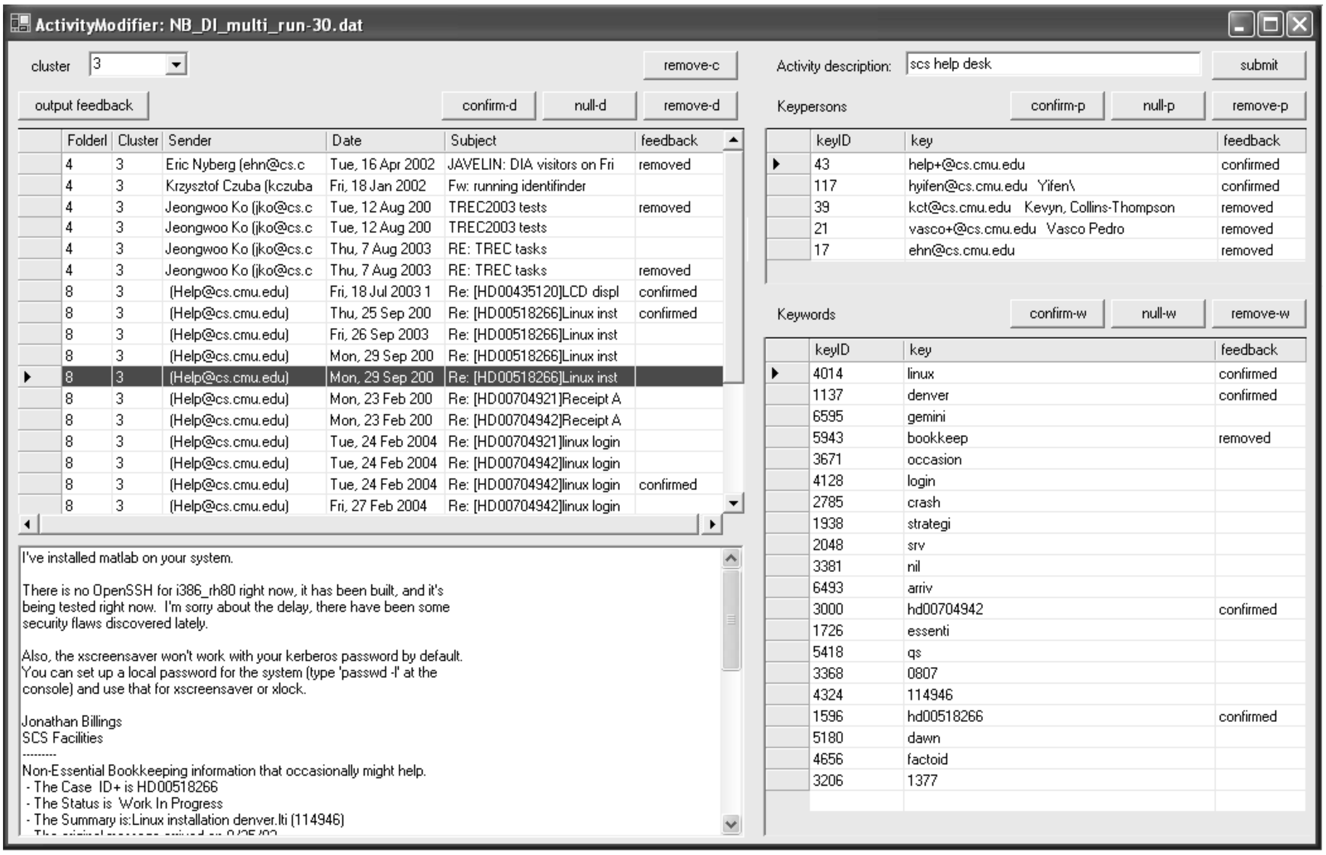


Figure 6: Performance of using single feedback types (CR, PP, WX and HX) on the *EmailYH* dataset. SpeC-bound is the SpeClustering-bound model without feedback. The SpeClustering-bound model with one type of feedback out-performs naive Bayes and SpeClustering-bound without feedback in 17 out of 20 runs.

We use the simple initialization method on *EmailYH* dataset in order to break down feedback to single types. Figure 6 shows the results using just one type of feedback on 5 selected runs from *EmailYH* dataset. The CR feedback is independent from other types of feedback and all other types involve feedback only from clusters that are not removed. All 5 runs with CR or PP feedback, 4 runs with WX feedback and 3 runs with HX feedback outperform both naive Bayes baseline and SpeClustering-bound without feedback. Figure 7 shows the results using combina-



ActivityModifier: NB_DI_multi_run-30.dat

cluster: 3

Activity description: scs help desk

FolderI	Cluster	Sender	Date	Subject	feedback
4	3	Eric Nyberg (ehn@cs.c	Tue, 16 Apr 2002	JAVELIN: DIA visitors on Fri	removed
4	3	Krzysztof Czuba (kczuba	Fri, 18 Jan 2002	Fw: running identifinder	
4	3	Jeongwoo Ko (jko@cs.c	Tue, 12 Aug 200	TREC2003 tests	removed
4	3	Jeongwoo Ko (jko@cs.c	Tue, 12 Aug 200	TREC2003 tests	
4	3	Jeongwoo Ko (jko@cs.c	Thu, 7 Aug 2003	RE: TREC tasks	
4	3	Jeongwoo Ko (jko@cs.c	Thu, 7 Aug 2003	RE: TREC tasks	removed
8	3	(Help@cs.cmu.edu)	Fri, 18 Jul 2003 1	Re: [HD00435120]LCD displ	confirmed
8	3	(Help@cs.cmu.edu)	Thu, 25 Sep 200	Re: [HD00518266]Linux inst	confirmed
8	3	(Help@cs.cmu.edu)	Fri, 26 Sep 2003	Re: [HD00518266]Linux inst	
8	3	(Help@cs.cmu.edu)	Mon, 29 Sep 200	Re: [HD00518266]Linux inst	
8	3	(Help@cs.cmu.edu)	Mon, 29 Sep 200	Re: [HD00518266]Linux inst	
8	3	(Help@cs.cmu.edu)	Mon, 23 Feb 200	Re: [HD00704921]Receipt A	
8	3	(Help@cs.cmu.edu)	Mon, 23 Feb 200	Re: [HD00704942]Receipt A	
8	3	(Help@cs.cmu.edu)	Tue, 24 Feb 2004	Re: [HD00704921]linux login	
8	3	(Help@cs.cmu.edu)	Tue, 24 Feb 2004	Re: [HD00704942]linux login	
8	3	(Help@cs.cmu.edu)	Tue, 24 Feb 2004	Re: [HD00704942]linux login	confirmed
8	3	(Help@cs.cmu.edu)	Fri, 27 Feb 2004	Re: [HD00704942]linux login	

I've installed matlab on your system.

There is no OpenSSH for i386_rh80 right now, it has been built, and it's
being tested right now. I'm sorry about the delay, there have been some
security flaws discovered lately.

Also, the xscreensaver won't work with your kerberos password by default.
You can set up a local password for the system (type 'passwd -l' at the
console) and use that for xscreensaver or xlock.

Jonathan Billings
SCS Facilities

Non-Essential Bookkeeping information that occasionally might help.
- The Case ID+ is HD00518266
- The Status is Work In Progress
- The Summary is:Linux installation denver.lti (114946)

Keypersons

keyID	key	feedback
43	help+@cs.cmu.edu	confirmed
117	hyifen@cs.cmu.edu Yifen\	confirmed
39	kct@cs.cmu.edu Kevyn, Collins-Thompson	removed
21	vasco+@cs.cmu.edu Vasco Pedro	removed
17	ehn@cs.cmu.edu	removed

Keywords

keyID	key	feedback
4014	linux	confirmed
1137	denver	confirmed
6595	gemini	
5943	bookkeep	removed
3671	occasion	
4128	login	
2785	crash	
1938	strategi	
2048	srv	
3381	nil	
6493	arriv	
3000	hd00704942	confirmed
1726	essenti	
5418	qs	
3368	0807	
4324	114946	
1596	hd00518266	confirmed
5180	dawn	
4656	factoid	
3206	1377	

Figure 5: The user interface for feedback gathering. It displays a list of documents, keywords and key-persons for a selected cluster. A reviewer can decide (1) to keep the cluster or not, (2) confirm or remove keywords or key-persons (3) confirm or remove documents, (4) give a short description about the cluster. The reviewer can also go back and forth between clusters to make her feedback consistent.

tion of feedback types. User's feedback gives huge improvements in all runs (19.55% average accuracy improvements from naive Bayes results to SpeClustering-bound with full feedback). SpeClustering-bound with full feedback performs best in 4 out of 5 runs. In the remaining one run, CR+PP feedback performs best. The quantity of PP feedback is about 1/7th to 1/9th to the whole dataset and even higher if we exclude documents in removed clusters. The number of WX+HX feedback are fewer than PP feedback in these runs. However, CR+WX+HX performs better than CR+PP in 2 runs, which shows that meanings of clusters gives comparable information like document-cluster association. More compellingly, it is also much easier to get CR+WX+HX feedback than CR+PP in terms of time efficiency. In [12], they measure the time spend on labeling a document or a feature, and they find a person only need 1/5th of time to label a feature compared to the time to label a document.

For the 3 newsgroup datasets, the ratio of feedback number to the corpus size is very small. The inheritance of old results in the simple initialization overwhelms the training process so we use the joint initialization method to remedy the problem.

The user feedback is quite different across these three runs. For the selected run of *news-similar-3*, the naive Bayes results are extremely noisy and the cluster summarization is hardly recognizable by the reviewer. It turns out the feedback contains the removal of two out of three clusters and the reason that one is kept is because some keywords weakly indicates the meaning of one newsgroup, but the documents in the remaining cluster contain huge chunks from each newsgroup. For the selected run of *news-related-3*, talk.politics.guns and talk.politics.mideast are referred to two remaining clusters while talk.politics.misc has no reference due to the removal of the last cluster, which the reviewer cannot figure out its meaning. The cluster summarization is noisy but comprehensible, so the reviewer can make positive and negative feedback easily. The baseline accuracy of *news-different-3* is very high so most feedback is positive about the automatically generated summarization

Figure 8 shows experimental results from user feedback on one selected run from each newsgroup dataset. It is difficult to improve on the already accurate *news-different-3* run. Incorporating feedback gives no significant improvement on the selected *news-similar-3* runs whose feedback is based on extremely noisy clusters and a user is barely able to associate meaningful criterion to any cluster. However, one sees huge improvement from using feedback on the noisy but still meaningful cluster results. The accuracy of the selected *news-related-3* run jumps from 63.23% to 81.07%.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we focus on the problem of *how to cluster*

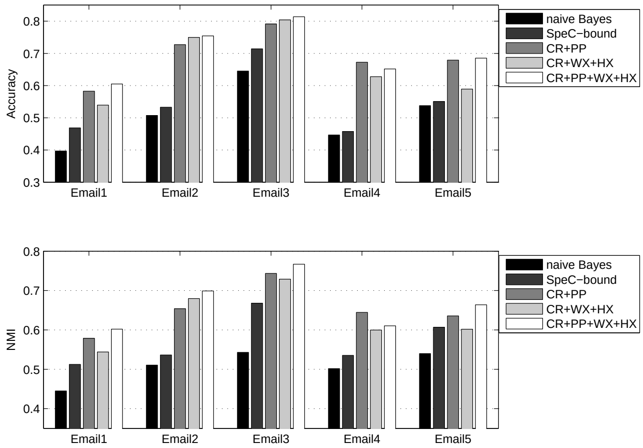


Figure 7: Performance of using combination of feedback types on the *EmailYH* dataset. SpeC-bound is the SpeClustering-bound model without feedback. User feedback gives huge improvements in all runs.

text documents based on the meanings of categories a user understands or wants. Often the meanings of clusters become clear to a user only after examining their descriptions and providing feedback to explore the space of possible clusters.

Our solution to this problem involves three components. First, we propose a new SpeClustering model that separates the features of a document that are specific to a cluster from other general features that are unrelated to the cluster's semantics. The second component is a method to collect user feedback about the meanings of the clusters. We present an interface that enables a user to browse through cluster results and provide several types of feedback. The process requires the user's understanding of desired categories, and her judgement about which cluster is associated with the meaning of which category. The third component is an algorithm for integrating user feedback with the SpeClustering model. The structure of the SpeClustering model provides a natural way to adjust parameters according to a variety of types of user feedback.

Our experimental results show our unsupervised SpeClustering algorithm outperforms the commonly used multinomial naive Bayes clustering algorithm for both of the text data sets we considered. Furthermore, when provided with user feedback, the SpeClustering model gains significant improvement in a personal email dataset and in the newsgroup dataset when the clustering results is noisy but meaningful. Our approach combines the advantage of the machine's computational power to analyze large data sets, with the advantages of a human's understanding of categories of interest. The results indicate that cooperation between computers and human beings is a promising direction for future work. There are many future challenges, such as using active learning principles to optimize the summarization of a cluster, and building more sophisticated models to allow more natural types of user feedback.

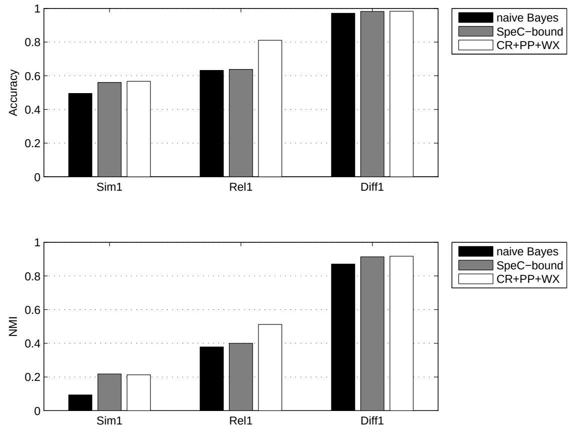


Figure 8: Experiments results of SpeClustering with user feedback on the newsgroup datasets. SpeCluster-bound is the model without feedback and CR+PP+WX is the SpeClustering-bound model with full user feedback. Incorporating feedback gives significant improvement on the selected *news-related-3* run, whose feedback is harvested from noisy but still meaningful clustering results.

Acknowledgments We thank Sophie Wang for useful discussions and contributions to earlier versions of the clustering email algorithm. This work was supported in part by Darpa contract NBCD030010, as part of the Personal Assistants that Learn project.

6. REFERENCES

[1] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *KDD-04*, 2004.

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 1998 Conference on Computational Learning Theory*, 1998.

[3] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. In *Journal of the Royal Statistical Society*, volume 39 of *B*, pages 1–38, 1977.

[4] B. Dom. An information-theoretic external cluster-validity measure. Technical Report RJ 10219, IBM, 2001.

[5] S. Godbole, A. Harpale, S. Sarawagi, and S. Chakrabarti. Document classification through interactive supervision of document and term labels. In *PKDD-04*, 2004.

[6] A. Hotho, S. Staab, and G. Stumme. Text clustering based on background knowledge. Technical Report 425, University of Karlsruhe, Institute AIFB, 2003.

[7] Y. Huang, D. Govindaraju, T. Mitchell, V. R. Carvalho, and W. Cohen. Inferring ongoing activities of workstation users by clustering email. In *First Conference on Email and Spam*, 2004.

[8] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML-99*, 1999.

[9] R. Jones, A. McCallum, K. Nigam, and E. Riloff. Bootstrapping for text learning tasks. In *IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications*, 1999.

[10] B. Liu, X. Li, W. S. Lee, and P. S. Yu. Text classification by labeling words. In *AAAI-04*, 2004.

[11] K. Nigam, A. K. McCallum, S. Thrun, and T. M. Mitchell. Learning to classify text from labeled and unlabeled documents. In *AAAI-98*, 1998.

[12] H. Raghavan, O. Madani, and R. Jones. Interactive feature selection. In *IJCAI-05*, 2005.

[13] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML-97*, 1997.