Topic-wise, Sentiment-wise, or Otherwise?
Identifying the Hidden Dimension for Unsupervised Text Classification

Sajib Dasgupta and **Vincent Ng**
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688
{sajib,vince}@hlt.utdallas.edu

Abstract

While traditional work on text clustering has largely focused on grouping documents by topic, it is conceivable that a user may want to cluster documents along other dimensions, such as the author's mood, gender, age, or sentiment. Without knowing the user's intention, a clustering algorithm will only group documents along the most prominent dimension, which may not be the one the user desires. To address this problem, we propose a novel way of incorporating user feedback into a clustering algorithm, which allows a user to easily specify the dimension along which she wants the data points to be clustered via inspecting only a small number of words. This distinguishes our method from existing ones, which typically require a large amount of effort on the part of humans in the form of document annotation or interactive construction of the feature space. We demonstrate the viability of our method on several challenging sentiment datasets.

1 Introduction

Text clustering is one of the most important applications in Natural Language Processing (NLP). A common approach to this problem consists of (1) computing the similarity between each pair of documents, each of which is typically represented as a bag of words; and (2) using an unsupervised clustering algorithm to partition the documents. The majority of existing work on text clustering has focused on *topic-based* clustering, where high accuracies can be achieved even for datasets with a large number of classes (e.g., 20 Newsgroups).

On the other hand, there has been relatively little work on *sentiment-based* clustering and the related task of *unsupervised polarity classification*, where the goal is to cluster (or classify) a set of documents (e.g., reviews) according to the polarity (e.g., "thumbs up" or "thumbs down") expressed by the author in an unsupervised manner. Despite the large amount of recent work on sentiment analysis and opinion mining, much of it has focused on *supervised* methods (e.g., Pang et al. (2002), Kim and Hovy (2004), Mullen and Collier (2004)). One weakness of these existing supervised polarity classification systems is that they are typically *domain-* and *language-specific*. Hence, when given a new domain or language, one needs to go through the expensive process of collecting a large amount of annotated data in order to train a high-performance polarity classifier. Some recent attempts have been made to leverage existing sentiment corpora or lexica to automatically create annotated resources for new domains or languages. However, such methods require the existence of either a parallel corpus/machine translation engine for projecting/translating annotations/lexica from a resource-rich language to the target language (Banea et al., 2008; Wan, 2008), or a domain that is "similar" enough to the target domain (Blitzer et al., 2007). When the target domain or language fails to meet this requirement, sentiment-based clustering or unsupervised polarity classification become appealing alternatives. Unfortunately, to our knowledge, these tasks are largely under-investigated in the NLP community. Turney's (2002) work is perhaps one of the most notable examples of unsupervised polarity classification. However, while his system learns the semantic orientation of the phrases in a review in an unsupervised manner, this information is used to predict the polarity of a review heuristically.

Despite its practical significance, sentiment-based clustering is a challenging task. To illustrate its difficulty, consider the task of clustering a set of movie reviews. Since each review may contain a description of the plot and the author's

sentiment, a clustering algorithm may cluster reviews along either the *plot* dimension or the *sentiment* dimension; and without knowing the user's intention, they will be clustered along the most *prominent* dimension. Assuming the usual bag-of-words representation, the most prominent dimension will more likely be *plot*, as it is not uncommon for a review to be devoted almost exclusively to the plot, with the author briefly expressing her sentiment only at the end of the review. Even if the reviews contain mostly subjective material, the most prominent dimension may still not be *sentiment*, due to the fact that many reviews are *sentimentally ambiguous*. Specifically, a reviewer may have negative opinions on the actors but at the same time talk enthusiastically about how much she enjoyed the plot. The presence of both positive and negative sentiment-bearing words in these reviews renders the sentiment dimension *hidden* (i.e., less prominent) as far as clustering is concerned. Therefore, there is no guarantee that the clustering algorithm will automatically produce a sentiment-based clustering of the reviews.

Hence, it is important for a user to provide feedback on the clustering process to ensure that the reviews are clustered along the *sentiment* dimension, possibly in an interactive manner. One way to do this would be to ask the user to annotate a small number of reviews with polarity information, possibly through an active learning procedure to minimize human intervention (Dredze and Crammer, 2008). Another way would be to have the user explicitly identify the relevant features (in our case, the sentiment-bearing words) at the beginning of the clustering process (Liu et al., 2004), or incrementally construct the set of relevant features in an interactive fashion (Bekkerman et al., 2007; Raghavan and Allan, 2007; Roth and Small, 2009). In addition, the user may supply constraints on which pairs of documents must or must not appear in the same cluster (Wagstaff et al., 2001), or simply tell the algorithm whether two clusters should be *merged* or *split* during the clustering process (Balcan and Blum, 2008). It is worth noting that many of these feedback mechanisms were developed by machine learning researchers for general clustering tasks and not for sentiment-based clustering.

Our goal in this paper is to propose a novel mechanism allowing a user to cluster a set of documents along the desired dimension, which may be a hidden dimension, with *very limited* user feedback. In comparison to the aforementioned feedback mechanisms, ours is arguably much simpler: we only require that the user *select* a dimension by examining a small number of features for each dimension, as opposed to having the user *generate* the feature space in an interactive manner or *identify* clusters that need to be merged or split. In particular, identifying clusters for merging or splitting in Balcan and Blum's algorithm may not be as easy as it appears: for each MERGE or SPLIT decision the user makes, she has to sample a large number of documents from the cluster(s), read through the documents, and base her decision on the extent to which the documents are (dis)similar to each other. Perhaps more importantly, our human experiments involving five users indicate that all of them can easily identify the sentiment dimension based on the features, thus providing suggestive evidence that our method is viable.

In sum, our contributions in this paper are three-fold. First, we propose a novel feedback mechanism for clustering allowing a user to easily specify the dimension along which she wants data points to be clustered and apply the mechanism to the challenging, yet under-investigated problem of sentiment-based clustering. Second, spectral learning, which is the core of our method, has not been applied extensively to NLP problems, and we hope that our work can increase the awareness of this powerful machine learning technique in the NLP community. Finally, we demonstrate the viability of our method not only by evaluating its performance on sentiment datasets, but also via a set of human experiments, which is typically absent in papers that involve algorithms for incorporating user feedback.

The rest of the paper is organized as follows. Section 2 presents the basics of spectral clustering, which will facilitate the discussion of our feedback mechanism in Section 3. We describe our human experiments and evaluation results on several sentiment datasets in Section 4, and present our conclusions in Section 5.

2 Spectral Clustering

When given a clustering task, an important question to ask is: which clustering algorithm should we use? A popular choice is k-means. Nevertheless, it is well-known that k-means has the major drawback of not being able to separate data points

that are not linearly separable in the given feature space (e.g., see Dhillon et al. (2004) and Cai et al. (2005)). Spectral clustering algorithms were developed in response to this problem with k-means. The central idea behind spectral clustering is to (1) construct a low-dimensional space from the original (typically high-dimensional) space while retaining as much information about the original space as possible, and (2) cluster the data points in this low-dimensional space. The rest of this section provides the details of spectral clustering.

2.1 Algorithm

Although there are several well-known spectral clustering algorithms in the literature (e.g., Weiss (1999), Shi and Malik (2000), Kannan et al. (2004)), we adopt the one proposed by Ng et al. (2002), as it is arguably the most widely-used. The algorithm takes as input a similarity matrix S created by applying a user-defined similarity function to each pair of data points. Below are the main steps of the algorithm:

1. Create the diagonal matrix D whose (i,i)-th entry is the sum of the i-th row of S, and then construct the Laplacian matrix $L = D^{-1/2}SD^{-1/2}$.
2. Find the eigenvalues and eigenvectors of L.
3. Create a new matrix from the m eigenvectors that correspond to the m largest eigenvalues.[1]
4. Each data point is now rank-reduced to a point in the m-dimensional space. Normalize each point to unit length (while retaining the sign of each value).
5. Cluster the resulting data points using k-means.

In essence, each dimension in the reduced space is defined by exactly one eigenvector. The reason why eigenvectors with large eigenvalues are used is that they capture the largest variance in the data. As a result, each of them can be thought of as revealing an important dimension of the data.

2.2 Clustering with Eigenvectors

As Ng et al. (2002) point out, "different authors still disagree on which eigenvectors to use, and how to derive clusters from them". There are two common methods for deriving clusters using the eigenvectors. These methods will serve as our baselines in our evaluation.

Method 1: Using the second eigenvector only
The first method is to use only the second eigenvector, $\mathbf{e}_2$, to partition the points. Besides revealing one of the most important dimensions of the data, this eigenvector induces an intuitively ideal partition of the data — the partition induced by the minimum normalized cut of the similarity graph[2], where the nodes are the data points and the edge weights are the pairwise similarity values of the points (Shi and Malik, 2000). Clustering in a one-dimensional space is trivial: since we have a linearization of the points, all we need to do is to determine a threshold for partitioning the points. However, we follow Ng et al. (2002) and cluster using 2-means in this one-dimensional space.

Method 2: Using m eigenvectors
Recall from Section 2.1 that after eigen-decomposing the Laplacian matrix, each data point is represented by m co-ordinates. In the second method, we simply use 2-means to cluster the data points in this m-dimensional space, effectively exploiting all of the m eigenvectors.

3 Our Approach

As mentioned before, sentiment-based clustering is challenging, in part due to the fact that the reviews can be clustered along more than one dimension. In this section, we propose and incorporate a user feedback mechanism into a spectral clustering algorithm, which makes it easy for a user to specify the dimension along which she wants to cluster the data points.

Recall that our method first applies spectral clustering to reveal the most important dimensions of the data, and then lets the user select the desired dimension. To motivate the importance of user feedback, it helps to understand why the two baseline clustering algorithms described in Section 2.2, which are also based on spectral methods but do not rely on user feedback, may not always yield a sentiment-based clustering. To begin with, consider the first method, where only the second eigenvector is used to induce the partition. Recall that the second eigenvector reveals the most prominent dimension of the data. Hence, if sentiment is not the most prominent dimension (which can happen if the non-sentiment-bearing

---

[1]For brevity, we will refer to the eigenvector with the n-th largest eigenvalue simply as the n-th eigenvector.

[2]Using the normalized cut (as opposed to the usual cut) ensures that the size of the two clusters are relatively balanced, avoiding trivial cuts where one cluster is empty and the other is full. See Shi and Malik (2000) for details.

words outnumber the sentiment-bearing words in the bag-of-words representation of a review), then the resulting clustering of the reviews may not be sentiment-oriented. A similar line of reasoning can be used to explain why the second baseline clustering algorithm, which clusters based on all of the eigenvectors in the low-dimensional space, may not always work well. Since each eigenvector corresponds to a different dimension (and, in particular, some of them correspond to non-sentiment dimensions), using all of them to represent a review may hamper the accurate computation of the similarity of two reviews as far as clustering along the sentiment dimension is concerned. In the rest of this section, we discuss the major steps of our user-feedback mechanism in detail.

Step 1: Identify the important dimensions
To identify the important dimensions of the given reviews, we take the top eigenvectors computed from the eigen-decomposition of the Laplacian matrix, which is in turn formed from the input similarity matrix. We compute the similarity between two reviews by taking the dot product of their feature vectors (see Section 4.1 for details on feature vector generation). Following Ng et al., we set the diagonal entries of the similarity matrix to 0.

Step 2: Identify the relevant features
Given the eigen-decomposition from Step 1, we first obtain the second through the fifth eigenvectors[3], which as mentioned above, correspond to the most important dimensions of the data. Then, we ask the user to select one of the four dimensions defined by these eigenvectors according to their relevance to sentiment. One way to do this is to (1) induce one partition of the reviews from each of the four eigenvectors, using a procedure identical to Method 1 in Section 2.2, and (2) have the user inspect the four partitions and decide which corresponds most closely to a sentiment-based clustering. The main drawback associated with this kind of user feedback is that the user may have to read a large number of reviews in order to make a decision. Hence, to reduce human effort, we employ an alternative procedure: we (1) identify the most informative features for characterizing each partition, and (2) have the user inspect just the features rather than the reviews.

While traditional feature selection techniques such as log-likelihood ratio and information

gain can be applied to identify these informative features (see Yang and Pedersen (1997) for an overview), we employ a more sophisticated feature-ranking method that we call *maximum margin feature ranking* (MMFR). Recall that a maximum margin classifier (e.g., a support vector machine) separates data points from two classes while maximizing the margin of separation. Specifically, a maximum margin hyperplane is defined by $\mathbf{w} \cdot \mathbf{x} - b = 0$, where $\mathbf{x}$ is a feature vector representing an arbitrary data point, and $\mathbf{w}$ (a weight vector) and b (a scalar) are parameters that are learned by solving the following constrained optimization problem:

$$\arg\min \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i$$

subject to

$$c_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 - \xi_i, \quad 1 \leq i \leq n,$$

where $c_i \in \{+1, -1\}$ is the class of the i-th training point $\mathbf{x}_i$, ξ_i is the degree of misclassification of $\mathbf{x}_i$, and C is a regularization parameter that balances training error and model complexity.

We use $\mathbf{w}$ to identify the most informative features for a partition. Note that a feature with a large positive weight is strongly indicative of the positive class, whereas a feature with a large negative weight is strongly indicative of the negative class. In other words, the most informative features are those with large absolute weight values. We exploit this observation and identify the most informative features for a partition by (1) training an SVM classifier[4] on the partition, where data points in the same cluster belong to the same class; (2) sorting the features according to the SVM-learned feature weights; and (3) generating two ranked lists of informative features using the top and bottom 100 features, respectively.

Given the ranked lists generated for each of the four partitions, the user will select one of the partitions/dimensions as most relevant to sentiment by inspecting as many features in the ranked lists as needed. After picking the most relevant dimension, the user will label one of the two feature lists associated with this dimension as POSITIVE and the other as NEGATIVE. Since each feature list represents one of the clusters, the cluster associated with the positive list is labeled POSITIVE and

---

[3]The first eigenvector is not used because it is a constant vector, meaning that it cannot be used to partition the data.

[4]All the SVM classifiers in this paper are trained using the SVM^{light} package (Joachims, 1999), with the learning parameters set to their default values.

the cluster associated with the negative list is labeled NEGATIVE.

In comparison to existing user feedback mechanisms for assisting a clustering algorithm, ours requires comparatively little human intervention: we only require that the user select a dimension by examining a small number of features, as opposed to having the user construct the feature space or identify clusters that need to be merged or split as is required with other methods.

Step 3: Identify the unambiguous reviews
There is a caveat, however. As mentioned in the introduction, many reviews contain both positive and negative sentiment-bearing words. These ambiguous reviews are more likely to be clustered incorrectly than their unambiguous counterparts. Now, since the ranked lists of features are derived from the partition, the presence of these ambiguous reviews can adversely affect the identification of informative features using MMFR. As a result, we remove the ambiguous reviews before deriving informative features from a partition.

We employ a simple method for identifying unambiguous reviews. In the computation of eigenvalues, each data point factors out the orthogonal projections of each of the other data points with which they have an affinity. Ambiguous data points receive the orthogonal projections from both the positive and negative data points, and hence they have near zero values in the pivot eigenvectors. We exploit this important information. The basic idea is that the data points with near zero values in the eigenvectors are more ambiguous than those with large absolute values. As a result, we posit 250 reviews from each cluster whose corresponding values in the eigenvector are farthest away from zero as unambiguous, and induce the ranked list of features only from the resulting 500 unambiguous reviews.[5]

Step 4: Cluster along the selected dimension
Finally, we employ the 2-means algorithm to cluster all the reviews along the dimension (i.e., the eigenvector) selected by the user, regardless of whether a review is ambiguous or not.

---

[5]Note that 500 is a somewhat arbitrary choice. Underlying this choice is our assumption that a fraction of the reviews is unambiguous. As we will see in the evaluation section, these 500 reviews can be classified with a high accuracy; consequently, the features induced from the resulting clusters are also of high quality. Additional experiments reveal that the list of top-ranking features does not change significantly when induced from a smaller number of unambiguous reviews.

4 Evaluation

4.1 Experimental Setup

Datasets. We use five sentiment classification datasets, including the widely-used movie review dataset [MOV] (Pang et al., 2002) as well as four datasets containing reviews of four different types of products from Amazon [books (BOO), DVDs (DVD), electronics (ELE), and kitchen appliances (KIT)] (Blitzer et al., 2007). Each dataset has 2000 labeled reviews (1000 positives and 1000 negatives). To illustrate the difference between topic-based clustering and sentiment-based clustering, we will also show topic-based clustering results on POL, a dataset created by taking all the documents from two sections of 20 Newsgroups, namely, `sci.crypt` and `talks.politics`.

To preprocess a document, we first tokenize and downcase it, and then represent it as a vector of unigrams, using frequency as presence. In addition, we remove from the vector punctuation, numbers, words of length one, and words that occur in only a single review. Following the common practice in the information retrieval community, we also exclude words with high document frequency, many of which are stopwords or domain-specific general-purpose words (e.g., "movies" in the movie domain). A preliminary examination of our evaluation datasets reveals that these words typically comprise 1–2% of a vocabulary. The decision of exactly how many terms to remove from each dataset is subjective: a large corpus typically requires more removals than a small corpus. To be consistent, we simply sort the vocabulary by document frequency and remove the top 1.5%.

Evaluation metrics. We employ two evaluation metrics. First, we report results in terms of the accuracy achieved on the 2000 labeled reviews for each dataset. Second, following Kamvar et al. (2003), we evaluate the clusters produced by our approach against the gold-standard clusters using the Adjusted Rand Index (ARI). ARI ranges from –1 to 1; better clusterings have higher ARI values.

4.2 Baseline Systems

Clustering using the second eigenvector only. As our first baseline, we adopt Shi and Malik's approach and cluster the reviews using only the second eigenvector, $\mathbf{e}_2$, as described in Section 2.2. Results on POL and the five sentiment datasets are

System Variation	Accuracy						Adjusted Rand Index					
	POL	MOV	KIT	BOO	DVD	ELE	POL	MOV	KIT	BOO	DVD	ELE
Baseline: 2nd eigenvector	93.7	70.9	69.7	58.9	55.3	50.8	0.76	0.17	0.15	0.03	0.01	0.01
Baseline: m eigenvectors	95.9	59.3	63.2	60.1	62.5	63.8	0.84	0.03	0.07	0.04	0.06	0.08
Our approach	93.7	70.9	69.7	69.5	70.8	65.8	0.76	0.17	0.15	0.15	0.17	0.10

Table 1: Results in terms of accuracy and Adjusted Rand Index for the six datasets.

shown in row 1 of Table 1.[6] As we can see, this baseline achieves an accuracy of 90% on POL, but a much lower accuracy (of 50–70%) on the sentiment datasets. The same performance trend can be observed with ARI. These results provide support for the claim that sentiment-based clustering is more difficult than topic-based clustering.

In addition, it is worth noting that the baseline achieves much lower accuracies and ARI values on BOO, DVD, and ELE than on the remaining two sentiment datasets. Since $\mathbf{e}_2$ captures the most prominent dimension, these results suggest that sentiment dimension is not the most prominent dimension in these three datasets. In fact, this is intuitively plausible. For instance, in the book domain, positive book reviews typically contain a short description of the content, with the reviewer only briefly expressing her sentiment somewhere in the review. Similarly for the electronics domain: electronic product reviews are typically aspect-oriented, with the reviewer talking about the pros and cons of each aspect of the product (e.g., battery, durability). Since the reviews are likely to contain both positive and negative sentiment-bearing words, the sentiment-based clustering is unlikely to be captured by $\mathbf{e}_2$.

Clustering using top five eigenvectors. As our second baseline, we represent each data point using the top five eigenvectors (i.e., $\mathbf{e}_1$ through $\mathbf{e}_5$), and cluster them using 2-means in this 5-dimensional space, as described in Section 2.2. Hence, this can be thought of as an "ensemble" approach, where the clustering decision is collectively made by the five eigenvectors.

Results are shown in row 2 of Table 1. In comparison to the first baseline, we see improvements in accuracy and ARI for the three datasets on which the first baseline performs poorly (i.e., BOO, DVD, and ELE), with the most drastic improvement observed on ELE. On the other hand, performance on the remaining two senti-

ment datasets deteriorates. These results can be attributed to the fact that for BOO, DVD, and ELE, $\mathbf{e}_2$ does not capture the sentiment dimension, but since some other eigenvector in the ensemble does, we see improvements. On the other hand, $\mathbf{e}_2$ has already captured the sentiment dimension in MOV and KIT; as a result, employing additional dimensions, which may not be sentiment-related, may only introduce noise into the computation of the similarities between the reviews.

4.3 Our Approach

Human experiments. Unlike the two baselines, our approach requires users to specify which of the four dimensions (defined by the second through fifth eigenvectors) are most closely related to sentiment by inspecting a set of features derived from the unambiguous reviews for each dimension using MMFR. To better understand how easy it is for a human to select the desired dimension given the features, we performed the experiment independently with five humans (all of whom are computer science graduate students not affiliated with this research) and computed the agreement rate.

More specifically, for each dataset, we showed each human judge the top 100 features for each cluster according to MMFR (see Tables 4–6 for a snippet). In addition, we informed them of the intended dimension: for example, for POL, the judge was told that the intended clustering is Politics vs. Science. Also, if she determined that more than one dimension was relevant to the intended clustering, she was instructed to rank these dimensions in terms of their degree of relevance, where the most relevant one would appear first in the list.

The dimensions (expressed in terms of the IDs of the eigenvectors) selected by each of the five judges for each dataset are shown in Table 2. The agreement rate (shown in the last row of the table) was computed based on only the highest-ranked dimension selected by each judge. As we can see, perfect agreement is achieved for four of the five sentiment datasets, and for the remaining two datasets, near-perfect agreement is achieved.

---

[6]Owing to the randomness in the choice of seeds for 2-means, these and all other experimental results involving 2-means are averaged over ten independent runs.

Judge	POL	MOV	KIT	BOO	DVD	ELE
1	2,3,4	2	2	4	3	3
2	2,4	2	2	4	3	3
3	4	2,4	4	4	3	3
4	2,3	2	2	4	3	3,4
5	2	2	2	4	3	3
Agr	80%	100%	80%	100%	100%	100%

Table 2: Human agreement rate.

	POL	MOV	KIT	BOO	DVD	ELE
Acc	99.8	87.0	87.6	86.2	87.4	77.6

Table 3: Accuracies on unambiguous documents.

These results together with the fact that it took 5–6 minutes to identify the relevant dimension, indicate that asking a human to determine the intended dimension based on solely the "informative" features is a viable task.

Clustering results. Next, we cluster all 2000 documents for each dataset using the dimension selected by the majority of the human judges. The clustering results are shown in row 3 of Table 1. In comparison to the better baseline for each dataset, we see that our approach performs substantially better on BOO, DVD and ELE, at almost the same level on MOV and KIT, but slightly worse on POL. Note that the improvements observed for BOO, DVD and ELE can be attributed to the failure of e_2 to capture the sentiment dimension. Perhaps most importantly, by exploiting human feedback, our approach has achieved more stable performance across the datasets than the baselines, with accuracies ranging from 65.8% to 93.7% and ARI ranging from 0.10 to 0.76.

Role of unambiguous documents. Recall that the features with the largest MMFR were computed from the unambiguous documents only. To get an intuitive understanding of the role of unambiguous documents in our approach, we show in Table 3 the accuracy when the unambiguous documents in each dataset were clustered using the eigenvector selected by the majority of the judges. As we can see, the accuracy of each dataset is higher than the corresponding accuracy shown in row 3 of Table 1. In fact, an accuracy of more than 85% was achieved on all but one dataset. This suggests that our method of identifying unambiguous documents is useful.

Note that it is crucial to be able to achieve a high accuracy on the unambiguous documents: if clustering accuracy is low, the features induced from

the clusters may not be an accurate representation of the corresponding dimension, and the human judge may have a difficult time identifying the intended dimension. In fact, some human judges reported difficulty in identifying the correct dimension for the ELE dataset, and this can be attributed in part to the low accuracy achieved on the unambiguous documents.

Features as summary. Recall that the method we proposed represents each dimension with a small number of features and asks a user to select the desired dimension by inspecting the corresponding feature lists. In other words, each feature list serves as a "summary" of its corresponding dimension, and inspecting the features induced for each dimension can give us insights into the different dimensions of a dataset. Hence, if a user is not sure how she wants the data points to be clustered (due to lack of knowledge of the data, for instance), our automatically induced features may serve as an overview of the different dimensions of the data. To better understand whether these features can indeed provide a user with additional useful information about a dataset, we show in Tables 4–6 the top ten features induced for each cluster and each dimension for the six datasets. As an example, consider the MOV dataset. Inspecting the induced features, we can determine that it has a sentiment dimension (e_2), as well as a humor vs. thriller dimension (e_4). In other words, if we cluster along e_2, we get a sentiment-based clustering; and if we cluster along e_4, we obtain a genre-based (humor vs. thriller) clustering.

User feedback vs. labeled data. Recall that our two baselines are unsupervised, whereas our approach can be characterized as semi-supervised, as it relies on user feedback to select the intended dimension. Hence, it should not be surprising to see that the average clustering performance of our approach is better than that of the baselines.

To do a fairer comparison, we conduct another experiment in which we compare our approach against a semi-supervised sentiment classification system, which uses transductive SVM as the underlying semi-supervised learner. More specifically, the goal of this experiment is to determine how many labeled documents are needed in order for the transductive learner to achieve the same level of performance as our approach. To answer this question, we first give the transductive learner access to the 2000 documents for each dataset as

POL				MOV			
$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$
$\mathbf{C}_1$	$\mathbf{C}_1$	$\mathbf{C}_1$	$\mathbf{C}_1$	$\mathbf{C}_1$	$\mathbf{C}_1$	$\mathbf{C}_1$	$\mathbf{C}_1$
serder	beyer	serbs	escrow	relationship	production	jokes	starts
armenian	arabs	palestinians	serial	son	earth	kids	person
turkey	andi	muslims	algorithm	tale	sequences	live	saw
armenians	research	wrong	chips	husband	aliens	animation	feeling
muslims	israelis	department	ensure	perfect	war	disney	lives
sdpa	tim	bosnia	care	drama	crew	animated	told
argic	uci	live	strong	focus	alien	laughs	happen
davidian	ab	matter	police	strong	planet	production	am
dbd@ura	z@virginia	freedom	omissions	beautiful	horror	voice	felt
troops	holocaust	politics	excepted	nature	evil	hilarious	happened
$\mathbf{C}_2$	$\mathbf{C}_2$	$\mathbf{C}_2$	$\mathbf{C}_2$	$\mathbf{C}_2$	$\mathbf{C}_2$	$\mathbf{C}_2$	$\mathbf{C}_2$
sternlight	escrow	standard	internet	worst	sex	thriller	comic
wouldn	sternlight	sternlight	uucp	stupid	romantic	killer	sequences
pgp	algorithm	des	uk	waste	school	murder	michael
crypto	access	escrow	net	bunch	relationship	crime	supporting
algorithm	net	employer	quote	wasn	friends	police	career
isn	des	net	ac	video	jokes	car	production
likely	privacy	york	co	worse	laughs	dead	peter
access	uk	jake	didn	boring	sexual	killed	style
idea	systems	code	ai	guess	cute	starts	latest
cryptograph	pgp	algorithm	mit	anyway	mother	violence	entertaining

Table 4: Top ten features induced for each dimension for the POL and MOV domains. The shaded columns correspond to the dimensions selected by the human judges. $\mathbf{e}_2, \ldots, \mathbf{e}_5$ are the top eigenvectors; $\mathbf{C}_1$ and $\mathbf{C}_2$ are the clusters.

BOO				ELE			
$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$
$\mathbf{C}_1$	$\mathbf{C}_1$	$\mathbf{C}_1$	$\mathbf{C}_1$	$\mathbf{C}_1$	$\mathbf{C}_1$	$\mathbf{C}_1$	$\mathbf{C}_1$
history	series	loved	must	mouse	music	easy	amazon
must	man	highly	wonderful	cable	really	used	cable
modern	history	easy	old	cables	ipod	card	card
important	character	enjoyed	feel	case	too	fine	recommend
text	death	children	away	red	little	using	dvd
reference	between	again	children	monster	headphones	problems	camera
excellent	war	although	year	picture	hard	fine	fast
provides	seems	excellent	someone	kit	excellent	drive	far
business	political	understand	man	overall	need	computer	printer
both	american	three	made	paid	fit	install	picture
$\mathbf{C}_2$	$\mathbf{C}_2$	$\mathbf{C}_2$	$\mathbf{C}_2$	$\mathbf{C}_2$	$\mathbf{C}_2$	$\mathbf{C}_2$	$\mathbf{C}_2$
plot	buy	money	boring	working	worked	money	phone
didn	bought	bad	series	never	problem	worth	off
thought	information	nothing	history	before	never	amazon	worked
boring	easy	waste	pages	phone	item	over	power
got	money	buy	information	days	amazon	return	battery
character	recipes	anything	between	headset	working	years	unit
couldn	pictures	doesn	highly	money	support	much	set
ll	look	already	page	months	months	headphones	phones
ending	waste	instead	excellent	return	returned	sony	range
fan	copy	seems	couldn	second	another	received	little

Table 5: Top ten features induced for each dimension for the BOO and ELE domains. The shaded columns correspond to the dimensions selected by the human judges. $\mathbf{e}_2, \ldots, \mathbf{e}_5$ are the top eigenvectors; $\mathbf{C}_1$ and $\mathbf{C}_2$ are the clusters.

unlabeled data. Next, we randomly sample 50 unlabeled documents and assign them the true label. We then re-train the classifier and compute its accuracy on the 2000 documents. We keep adding more labeled data (50 in each iteration) until it reaches the accuracy achieved by our system. Results of this experiment are shown in Table 7. Owing in the randomness involved in the selection of unlabeled documents, these results are averaged over ten independent runs. As we can see, our

KIT				DVD			
e_2	e_3	e_4	e_5	e_2	e_3	e_4	e_5
C_1	**C_1**	**C_1**	**C_1**	**C_1**	**C_1**	**C_1**	**C_1**
love	works	really	pan	worth	music	video	money
clean	water	nice	oven	bought	collection	music	quality
nice	clean	works	cooking	series	excellent	found	video
size	work	too	made	money	wonderful	feel	worth
set	ice	quality	pans	season	must	bought	found
kitchen	makes	small	better	fan	loved	workout	version
easily	thing	sturdy	heat	collection	perfect	daughter	picture
sturdy	need	little	cook	music	highly	recommend	waste
recommend	keep	think	using	tv	makes	our	special
price	best	item	clean	thought	special	disappointed	sound
C_2	**C_2**	**C_2**	**C_2**	**C_2**	**C_2**	**C_2**	**C_2**
months	price	ve	love	young	worst	series	saw
still	item	years	coffee	between	money	cast	watched
back	set	love	too	actors	thought	fan	loved
never	ordered	never	recommend	men	boring	stars	enjoy
worked	amazon	clean	makes	cast	nothing	original	whole
money	gift	months	over	seems	minutes	comedy	got
did	got	over	size	job	waste	actors	family
amazon	quality	pan	little	beautiful	saw	worth	series
return	received	been	maker	around	pretty	classic	season
machine	knives	pans	cup	director	reviews	action	liked

Table 6: Top ten features induced for each dimension for the KIT and DVD domains. The shaded columns correspond to the dimensions selected by the human judges. $e_2, \ldots, e_5$ are the top eigenvectors; C_1 and C_2 are the clusters.

	POL	MOV	KIT	BOO	DVD	ELE
# labels	400	150	200	350	350	200

Table 7: Transductive SVM results.

user feedback is equivalent to the effort of hand-annotating 275 documents per dataset on average.

Multiple relevant dimensions. As seen from Table 2, some human judges selected more than one dimension for some datasets (e.g., 2,3,4 for POL; 2,4 for MOV; and 3,4 for ELE). However, we never took into account these "extra" dimensions in our previous experiments. To better understand whether these extra dimensions can help improve accuracy and ARI, we conduct another experiment in which we apply 2-means to cluster the documents in a space that is defined by all of the selected dimensions. The final accuracy turns out to be 95.9%, 70.9%, and 67.5% for POL, MOV, and ELE respectively, which is considerably better than using only the optimal dimension and suggests that the extra dimensions contain useful information.

5 Conclusions

Unsupervised clustering algorithms typically group objects along the most prominent dimension, in part owing to their objective of simultaneously maximizing inter-cluster similarity and intra-cluster dissimilarity. Hence, if the user's intended clustering dimension is not the most prominent dimension, these unsupervised clustering algorithms will fail miserably. To address this problem, we proposed to integrate a novel user feedback mechanism into a spectral clustering algorithm, which allows us to mine the intended, possibly hidden, dimension of the data and produce the desired clustering. This mechanism differs from competing methods in that it requires very limited feedback: to select the intended dimension, the user only needs to inspect a small number of features. We demonstrated its viability via a set of human and automatic experiments with unsupervised sentiment classification, obtaining promising results.

In future work, we plan to explore several extensions to our proposed method. First, we plan to use our user-feedback method in combination with existing methods (e.g., Bekkerman et al. (2007)) for improving its performance. For instance, instead of having the user construct a relevant feature space from scratch, she can simply extend the set of informative features identified for the user-selected dimension. Second, since none of the steps in our method is specifically designed for sentiment classification, we plan to apply it to other non-topic-based text classification tasks.

References

Maria-Florina Balcan and Avrim Blum. 2008. Clustering with interactive feedback. In *Proceedings of ALT*, pages 316–328.

Carmen Banea, Rada Mihalcea, Janyce Wiebe, and Samer Hassan. 2008. Multilingual subjectivity analysis using machine translation. In *Proceedings of EMNLP*, pages 127–135.

Ron Bekkerman, Hema Raghavan, James Allan, and Koji Eguchi. 2007. Interactive clustering of text collections according to a user-specified criterion. In *Proceedings of IJCAI*, pages 684–689.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the ACL*, pages 440–447.

Deng Cai, Xiaofei He, and Jiawei Han. 2005. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1624–1637.

Inderjit Dhillon, Yuqiang Guan, and Brian Kulis. 2004. Kernel k-means, spectral clustering and normalized cuts. In *Proceedings of KDD*, pages 551–556.

Mark Dredze and Koby Crammer. 2008. Active learning with confidence. In *Proceedings of ACL-08:HLT Short Papers (Companion Volume)*, pages 233–236.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In Bernhard Scholkopf and Alexander Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 44–56. MIT Press.

Sepandar Kamvar, Dan Klein, and Chris Manning. 2003. Spectral learning. In *Proceedings of IJCAI*, pages 561–566.

Ravi Kannan, Santosh Vempala, and Adrian Vetta. 2004. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51(3):497–515.

Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of COLING*, pages 1367–1373.

Bing Liu, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. 2004. Text classification by labeling words. In *Proceedings of AAAI*, pages 425–430.

Tony Mullen and Nigel Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of EMNLP*, pages 412–418.

Andrew Ng, Michael Jordan, and Yair Weiss. 2002. On spectral clustering: Analysis and an algorithm. In *Advances in NIPS 14*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86.

Hema Raghavan and James Allan. 2007. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *Proceedings of SIGIR*, pages 79–86.

Dan Roth and Kevin Small. 2009. Interactive feature space construction using semantic information. In *Proceedings of CoNLL*, pages 66–74.

Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Peter Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the ACL*, pages 417–424.

Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. 2001. Constrained k-means clustering with background knowledge. In *Proceedings of ICML*, pages 577–584.

Xiaojun Wan. 2008. Using bilingual knowledge and ensemble techniques for unsupervised Chinese sentiment analysis. In *Proceedings of EMNLP*, pages 553–561.

Yair Weiss. 1999. Segmentation using eigenvectors: A unifying view. In *Proceedings of ICCV*, pages 975–982.

Yiming Yang and Jan Pedersen. 1997. A comparative study on feature selection in text categorization. In *Proceedings of ICML*, pages 412–420.