A cross-collection mixture model for comparative text mining

Conference Paper · January 2004

DOI: 10.1145/1014052.1014150 · Source: DBLP

3 authors, including:

Atulya Velivelli

15 PUBLICATIONS **370** CITATIONS

SEE PROFILE

Bei Yu

Syracuse University

40 PUBLICATIONS **1,103** CITATIONS

SEE PROFILE

A Cross-Collection Mixture Model for Comparative Text Mining

ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana Champaign

Atulya Velivelli
Department of Electrical and Computer Engineering
University of Illinois at Urbana Champaign

Bei Yu
Graduate School of Library and Information Science
University of Illinois at Urbana Champaign

ABSTRACT

In this paper, we define and study a novel text mining problem, which we refer to as Comparative Text Mining (CTM). Given a set of comparable text collections, the task of comparative text mining is to discover any latent common themes across all collections as well as summarize the similarity and differences of these collections along each common theme. This general problem subsumes many interesting applications, including business intelligence and opinion summarization. We propose a generative probabilistic mixture model for comparative text mining. The model simultaneously performs cross-collection clustering and within-collection clustering, and can be applied to an arbitrary set of comparable text collections. The model can be estimated efficiently using the Expectation-Maximization (EM) algorithm. We evaluate the model on two different text data sets (i.e., a news article data set and a laptop review data set), and compare it with a baseline clustering method also based on a mixture model. Experiment results show that the model is quite effective in discovering the latent common themes across collections and performs significantly better than our baseline mixture model.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Text Mining

General Terms: Algorithms

Keywords: Comparative text mining, mixture models, clustering

1. INTRODUCTION

Text mining is concerned with extracting knowledge and patterns from text [5, 6]. While there has been much research in text mining, most existing research is focused on one single collection of text. The goals are often to extract basic semantic units such as named entities, to extract relations between information units, or to extract topic themes.

KDD'04, August 22–25, 2004, Seattle, Washington, USA.

In this paper, we study a novel problem of text mining referred to as *Comparative Text Mining* (CTM). Given a set of comparable text collections, the task of comparative text mining is to discover any latent common themes across all collections as well as summarize the similarity and differences of these collections along each common theme. Specifically, the task involves: (1) discovering the different common themes across all the collections; (2) for each discovered theme, characterize what is in common among all the collections and what is unique to each collection. The need for comparative text mining exists in many different applications, including business intelligence, summarizing reviews of similar products, and comparing different opinions about a common topic in general.

In this paper, we study the CTM problem and propose a generative probabilistic mixture model for CTM. The model simultaneously performs cross-collection clustering and within-collection clustering, and can be applied to an arbitrary set of comparable text collections. The mixture model is based on component multinomial distribution models, each characterizing a different theme. The common themes and collection-specific themes are explicitly modeled. The proposed model can be estimated efficiently using the Expectation-Maximization (EM) algorithm.

We evaluate the model on two different text data sets (i.e., a news article data set and a laptop review data set), and compare it with a baseline clustering method also based on a mixture model. Experiment results show that the model is quite effective in discovering the latent common themes across collections and performs significantly better than our baseline mixture model.

The rest of the paper is organized as follows. In Section 2, we briefly introduce the problem of CTM. We then present a baseline simple mixture model and a new cross-collection mixture model in Section 3 and Section 4. We discuss the experiment results in Section 5.

2. COMPARATIVE TEXT MINING

2.1 A motivating example

With the popularity of e-commerce, online customer evaluations are becoming widely provided by online stores and third-party websites. Pioneers like amazon.com and epinions.com have accumulated large amounts of customer input including reviews, comments, recommendations and advice, etc. For example, the number of reviews in epinions.com

is more than one million[4]. Given a product, there could be up to hundreds of reviews, which is impossible for the readers to go through. It is thus desirable to summarize a collection of reviews for a certain type of products in order to provide the readers the most salient feedbacks from the peers. For review summarization, the most important task is to identify different semantic aspects of a product that the reviewers mentioned and to group the opinions according to these aspects to show similarities and differences in the opinions.

For example, suppose we have reviews of three different brands of laptops (Dell, IBM, and Apple), and we want to summarize the reviews. A useful summary would be a tabular representation of the opinions as shown in Table 1, in which each row represents one aspect (subtopic) and different columns correspond to different opinions.

Table 1: A tabular summary

Subtopics	Dell	IBM	Apple
Battery life	long enough	short	short
Memory	good	bad	good
Speed	slow	fast	fast

It is, of course, very difficult, if not impossible to produce such a table completely automatically. However, we can achieve a less ambitious goal – identifying the semantic aspects and identifying the common and specific characteristics of each product in an *unsupervised* way. This is a concrete example of comparative text mining.

2.2 The general problem

The example above is only one of the many possible applications of comparative text mining. In general, the task of comparative text mining involves: (1) discovering the common themes across all the collections; (2) for each discovered theme, characterize what is in common among all the collections and what is unique to each collection. It is very hard to precisely define what a theme is, but it corresponds roughly to a topic or subtopic. The granularity of themes is application-specific. CTM is a fundamental task in exploratory text analysis. In addition to opinion comparison and summarization, it has many other applications, such as business intelligence (comparing different companies), customer relationship management (comparing different groups of customers), and semantic integration of text (comparing component text collections).

CTM is challenging in several ways: (1) It is a completely unsupervised learning task; no training data is available. (It is for the same reason that CTM can be very useful for many different purposes – it makes minimum assumptions about the collections and in principle we can compare any arbitrary partition of text.) (2) We need to identify themes *across* different collections, which is more challenging than identifying topic themes in one single collection. (3) The task involves a discrimination component – for each discovered theme, we also want to identify the unique information specific to each collection. Such a discrimination task is difficult given that we do not have training data. In a way, CTM goes beyond the regular one-collection text mining by requiring an "alignment" of multiple collections based on common themes.

Since no training data is available, in general, we must rely on unsupervised learning methods, such as clustering, to perform CTM. In this paper, we study how to use probabilistic mixture models to perform CTM. Below we first describe a simple mixture model for clustering, which represents a straightforward application of an existing text mining method, and then present a more sophisticated mixture model specifically designed for CTM.

3. CLUSTERING WITH A SIMPLE MIXTURE MODEL

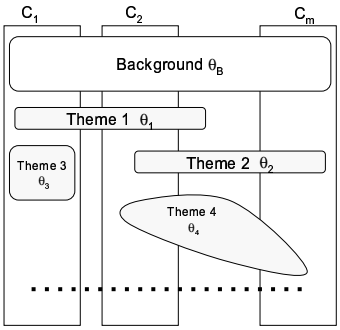

Figure 1: The Simple Mixture Model

A naive solution to CTM is to treat the multiple collections as one single collection and perform clustering. Our hope is that some clusters would represent the common themes across the collections, while some others would represent themes specific to one collection (see Figure 1). We now present a simple multinomial mixture model for clustering an arbitrary collection of documents, in which we assume there are k latent common themes in all collections, and each is characterized by a multinomial word distribution (also called a unigram language model). A document is regarded as a sample of a mixture model with these theme models as components. We fit such a mixture model to the union of all the text collections we have, and the obtained component multinomial models can be used to analyze the common themes and differences among the collections.

Formally, let $\mathcal{C} = \{C_1, C_2, ..., C_m\}$ be m comparable collections of documents. Let $\theta_1, ..., \theta_k$ be k theme unigram language models and θ_B be the background model for all the collections. A document d is regarded as a sample of the following mixture model (based on word generation).

$$p_d(w) = \lambda_B p(w|\theta_B) + (1 - \lambda_B) \sum_{j=1}^{k} [\pi_{d,j} p(w|\theta_j)]$$

where w is a word, $\pi_{d,j}$ is a document-specific mixing weight for the j-th aspect theme, and $\sum_{j=1}^{k} \pi_{d,j} = 1$. λ_B is the mixing weight of the background model θ_B. The log-likelihood of all the collections $\mathcal{C}$ is

$$\log p(\mathcal{C}|\Lambda) = \sum_{i=1}^{m} \sum_{d \in C_i} \sum_{w \in V} [c(w,d) \times \log(\lambda_B p(w|\theta_B) + (1 - \lambda_B) \sum_{j=1}^{k} (\pi_{d,j} p(w|\theta_j)))]$$

where V is the set of all the words (i.e., vocabulary), $c(w,d)$ is the count of word w in document d, and $\Lambda = (\{\theta_j, \pi_{d,j}\}_{j=1}^{k}$

is the set of all the theme model parameters. The purpose of using a background model is to "force" clustering to be done based on more discriminative words, leading to more informative and more discriminative component models. We control this effect through θ_B.

The model can be estimated using any estimator. For example, the Expectation-Maximization (EM) algorithm [3] can be used to compute a maximum likelihood estimate with the following updating formulas:

$$p(z_{d,w}=j) = \frac{\pi_{d,j}^{(n)} p^{(n)}(w|\theta_j)}{\sum_{j'=1}^{k} \pi_{d,j'}^{(n)} p^{(n)}(w|\theta_{j'})}$$

$$p(z_{d,w}=B) = \frac{\lambda_B p(w|\theta_B)}{\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k} \pi_{d,j}^{(n)} p^{(n)}(w|\theta_j)}$$

$$\pi_{d,j}^{(n+1)} = \frac{\sum_{w\in V} c(w,d) p(z_{d,w}=j)}{\sum_{j'}\sum_{w\in V} c(w,d) p(z_{d,w}=j')}$$

$$p^{(n+1)}(w|\theta_j) = \frac{\sum_{i=1}^{m}\sum_{d\in C_i} c(w,d)(1-p(z_{d,w}=B))p(z_{d,w}=j)}{\sum_{w'\in V}\sum_{i=1}^{m}\sum_{d\in C_i} c(w',d)(1-p(z_{d,w'}=B))p(z_{d,w'}=j)}$$

This mixture model is closely related to the probabilistic latent semantic indexing model (PLSI) proposed in [7] and treats CTM as a single-collection text mining problem. However, such a simple model is inadequate for CTM for two reasons: (1) We have completely ignored the structure of collections. As a result, we may have clusters that represent only *some*, not all of the collections. (2) There is no easy way to identify which theme cluster represents the common information across collections and which represents specific information to a particular collection. Below we present a more sophisticated coordinated mixture model, which is specifically designed for CTM and addresses these two deficiencies.

4. CLUSTERING WITH A CROSS-COLLECTION MIXTURE MODEL

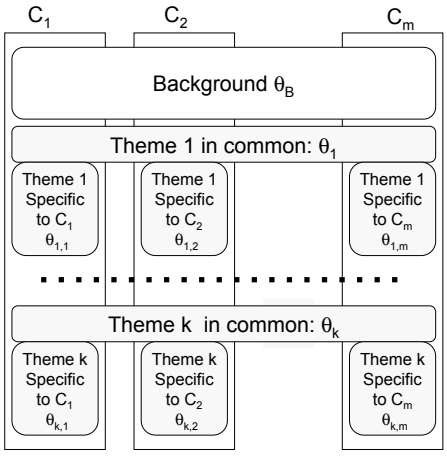

Figure 2: The Cross-Collection Mixture Model

4.1 The model

Our main idea for improving the simple mixture model for comparative text mining is to explicitly distinguish common theme clusters that characterize common information across all collections from special theme clusters that characterize collection-specific information. Thus we now consider k latent common themes as well as a potentially different set of k collection-specific themes for each collection (illustrated in Figure 2). These component models directly correspond to all the information we are interested in discovering. The sampling distribution of a word in document d (from collection C_i) is now collection-specific. Specifically, it involves the background model (θ_B), k common theme models ($\theta_1, ..., \theta_k$), and k collection-specific theme models ($\theta_{1,i}, ..., \theta_{k,i}$), which are to capture the unique information about the k themes in collection C_i. That is,

$$p_d(w|C_i) = (1-\lambda_B)\sum_{j=1}^{k}[\pi_{d,j}(\lambda_C p(w|\theta_j) + (1-\lambda_C)p(w|\theta_{j,i}))] + \lambda_B p(w|\theta_B)$$

where λ_B is the weight on the background model θ_B and λ_C is the weight on the common theme model θ_j (as opposed to the collection-specific theme model $\theta_{j,i}$). Intuitively, when we "generate" a word, we first decide whether to use the background model θ_B according to λ_B; the larger λ_B is, the more likely we will use θ_B. If we decide not to use θ_B, then we need to decide which theme to use; this is controlled by $\pi_{d,j}$, the probability of using theme j when generating words in d. Finally, once we decide which theme to use, we still need to decide whether we should use the common theme model or the collection-specific theme model, and this is controlled by λ_C, the probability of using the common model. The weighting parameters λ_B and λ_C are intentionally to be set by the user, and their interpretation is as follows. λ_B reflects our knowledge about how noisy the collections are. If we believe the text is verbose, then λ_B should be set to a larger value. In our experiments, a value of $0.9 - 0.95$ often works well. λ_C indicates our emphasis on the commonality, as opposed to the speciality in comparative text mining. A larger λ_C would allow us to learn a richer common theme model, whereas a smaller one would learn a weaker common theme model, but stronger special models. The optimal value depends on the specific applications.

According to this generative model, the log-likelihood of the whole set of collections is

$$\log p(\mathcal{C}) = \sum_{i=1}^{m}\sum_{d\in C_i}\sum_{w\in V}[c(w,d)\log[\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}(\lambda_C p(w|\theta_j) + (1-\lambda_C)p(w|\theta_{j,i}))]]$$

4.2 Parameter estimation

We estimate the background model θ_B using all the available text in the m text collections. That is,

$$\hat{p}(w|\theta_B) = \frac{\sum_{i=1}^{m}\sum_{d\in C_i} c(w,d)}{\sum_{i=1}^{m}\sum_{d\in C_i}\sum_{w'\in V} c(w',d)}$$

Since λ_B and λ_C are set manually, this leaves us with the following parameters to estimate: (1) the common theme models, $\theta = \{\theta_1, ..., \theta_k\}$; (2) the special theme models for each collection C_i, $\theta_{C_i} = \{\theta_{1,i}, ..., \theta_{k,i}\}$; and (3) the theme mixing weights for each document d: $\pi_d = \{\pi_{d,1}, ..., \pi_{d,k}\}$.

$$
\begin{aligned}
p(z_{d,C_i,w}=j) &= \frac{\pi_{d,j}^{(n)}(\lambda_C p^{(n)}(w|\theta_j)+(1-\lambda_C)p^{(n)}(w|\theta_{j,i}))}{\sum_{j'=1}^{k}\pi_{d,j'}^{(n)}(\lambda_C p^{(n)}(w|\theta_{j'})+(1-\lambda_C)p^{(n)}(w|\theta_{j',i}))}\\
p(z_{d,C_i,w}=B) &= \frac{\lambda_B p(w|\theta_B)}{\lambda_B p(w|\theta_B)+(1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}^{(n)}(\lambda_C p^{(n)}(w|\theta_j)+(1-\lambda_C)p^{(n)}(w|\theta_{j,i}))}\\
p(z_{d,C_i,j,w}=C) &= \frac{\lambda_C p^{(n)}(w|\theta_j)}{\lambda_C p^{(n)}(w|\theta_j)+(1-\lambda_C)p^{(n)}(w|\theta_{j,i})}\\
\pi_{d,j}^{(n+1)} &= \frac{\sum_{w\in V}c(w,d)p(z_{d,C_i,w}=j)}{\sum_{j'}\sum_{w\in V}c(w,d)p(z_{d,C_i,w}=j')}\\
p^{(n+1)}(w|\theta_j) &= \frac{\sum_{i=1}^{m}\sum_{d\in C_i}c(w,d)(1-p(z_{d,C_i,w}=B))p(z_{d,C_i,w}=j)p(z_{d,C_i,j,w}=C)}{\sum_{w'\in V}\sum_{i=1}^{m}\sum_{d\in C_i}c(w',d)(1-p(z_{d,C_i,w'}=B))p(z_{d,C_i,w'}=j)p(z_{d,C_i,j,w'}=C)}\\
p^{(n+1)}(w|\theta_{j,i}) &= \frac{\sum_{i=1}^{m}\sum_{d\in C_i}c(w,d)(1-p(z_{d,C_i,w}=B))p(z_{d,C_i,w}=j)(1-p(z_{d,C_i,j,w}=C))}{\sum_{w'\in V}\sum_{i=1}^{m}\sum_{d\in C_i}c(w',d)(1-p(z_{d,C_i,w'}=B))p(z_{d,C_i,w'}=j)(1-p(z_{d,C_i,j,w'}=C))}
\end{aligned}
$$

Figure 3: EM updating formulas for the cross-collection mixture model

As in the simple mixture model, we can also use the EM algorithm to compute a maximum likelihood estimate. The updating formulas are shown in Figure 3. Each EM iteration involves scanning all the text once, so the algorithm is quite scalable.

4.3 Using the model

Once the model is estimated, we will have k collection-specific models for each of the m collections and k common theme models across all collections. Each of these models is a word distribution or unigram language model. The high probability words can characterize the theme/cluster extracted. Such words can often be used directly as a summary or indirectly (e.g., through a hidden Markov model) to extract relevant sentences to form a summary of the corresponding theme. The extracted word distributions can also be used in many other ways, e.g., to classify other text documents or to link the related passages in the text collections so that a user can navigate the information space for comparative analysis.

We can input our bias for CTM through setting λ_B and λ_C manually. Specifically, λ_B allows us to input our knowledge about the noise (stop words) in the data – if we know the text data is verbose, then we should set λ_B to a high value, whereas if the data is concise and mostly content-bearing keywords, then we need to set λ_B to a smaller value. Similarly, λ_C allows us to input a trade-off between extracting common theme models (setting λ_C to a higher value) vs. extracting collection-specific models (setting λ_C to a smaller value). Such biases cannot be learned by the maximum likelihood estimator. Indeed, maximizing the data likelihood is only a means to achieve our ultimate goal, which is why we want to regularize our model in a meaningful way so that we can impose certain preferences while maximizing the data likelihood. The flexibility and control provided by λ_B and λ_C make it possible for a user to control the focus of the results of comparative text mining.

5. EXPERIMENTS AND RESULT ANALYSIS

We evaluated the Simple Mixture model (SimpMix) and the Cross-Collection Mixture model (CCMix) on two domains – war news and laptop reviews.

5.1 War news

The War news data consists of news excerpts on two comparable events: (1) Iraq war and (2) Afghanistan war, both of which occurred in the last two years. The Iraq war news excerpts were a combination of 30 articles from the CNN and BBC web sites over the last one year span. The Afghanistan war data consists of 26 news articles downloaded from the CNN and BBC web sites for one year starting from Nov. 2001. Our goal is to compare these two wars and find out their common and specific characteristics.

The results of using either the simple mixture model or the cross-collection mixture model are shown in Table 2, where the top words of each theme model are listed along with their probabilities. We set $\lambda_B = 0.95$ for SimpMix and set $\lambda_b = 0.9$, $\lambda_C = 0.25$ for CCMix; in both cases, the number of clusters is fixed to 5. Variations of these parameters are discussed later.

We see that although there are some interesting themes in the results of SimpMix (e.g., cluster3 and cluster4 appear to be about American and British inquiry into the presence of weapons in Iraq, respectively, while cluster2 suggests the presence of British soldier in Basra, a town in southern Iraq), they are all about Iraq war. We do not see any obvious theme *common* to both Iraq war and Afghanistan war. This is expected given that SimpMix pools all documents together without exploiting the collection structure.

In contrast, the results of CCMix explicitly suggest the common themes and the corresponding collection-specific themes. For example, cluster3 clearly suggests that in both wars, there has been loss of lives. Furthermore, the top words in the corresponding Iraq theme include names of some key defense people that are involved in the Iraq war (e.g., "Hoon" is the last name of the british defense secretary and "Sanchez" is the last name of the U.S General in Iraq). In comparison, the top words in the corresponding Afghanistan theme includes the name of the U.S Defense secretary who had an important role in the Afghan war.

Cluster4 and cluster5 are also meaningful themes. The common theme captured in Cluster4 is the Monday briefings by an official spokesman of a political administration during both wars; the corresponding special themes indicate the difference in the topics discussed in the briefings (e.g., weapon inquiry for Iraq war and Bin Laden for Afghanistan war). The common theme of Cluster5 is about the diplomatic role

Table 2: War news results using SimpMix model (top) vs. CCMix model (bottom)

	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5
Common theme words	will 0.019	british 0.017	weapons 0.022	inquiry 0.052	countries 0.026
	let 0.012	soldiers 0.015	kay 0.021	intelligence 0.036	contracts 0.023
	united 0.012	baghdad 0.015	rumsfeld 0.017	dossier 0.024	allawi 0.012
	god 0.011	air 0.011	commission 0.014	hutton 0.021	hoon 0.012
	inspectors 0.011	basra 0.011	group 0.014	claim 0.019	russian 0.010
	your 0.010	mosque 0.010	senate 0.011	wmd 0.019	international 0.010
	nation 0.010	southern 0.01	survey 0.010	mps 0.018	russia 0.009
	n 0.010	fired 0.010	paper 0.010	committee 0.017	reconstruction 0.009

	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5
Common theme words	us 0.042	mr 0.029	killed 0.036	monday 0.036	united 0.042
	nation 0.030	marines 0.025	month 0.032	official 0.032	nations 0.04
	will 0.024	dead 0.023	deaths 0.023	i 0.029	with 0.03
	action 0.022	general 0.022	one 0.023	would 0.028	is 0.025
	re 0.022	defense 0.019	died 0.022	where 0.025	it 0.024
	border 0.019	key 0.018	been 0.022	do 0.025	they 0.023
	its 0.017	since 0.018	drive 0.018	spokesman 0.022	diplomatic 0.023
	ve 0.016	first 0.016	according 0.015	political 0.021	blair 0.022
Iraq theme words	god 0.022	iraq 0.022	troops 0.016	intelligence 0.049	n 0.03
	saddam 0.016	us 0.021	hoon 0.015	weapons 0.034	weapons 0.024
	baghdad 0.013	baghdad 0.017	sanchez 0.012	inquiry 0.028	inspectors 0.023
	your 0.012	nato 0.015	billion 0.01	commission 0.017	council 0.016
	live 0.01	iraqi 0.013	spokeswoman 0.008	independent 0.016	declaration 0.015
Afghan theme words	paper 0.021	story 0.028	taleban 0.026	bin 0.031	northern 0.040
	afghan 0.019	full 0.026	rumsfeld 0.020	laden 0.031	alliance 0.040
	meeting 0.014	saturday 0.016	hotel 0.012	steinberg 0.027	kabul 0.030
	euro 0.012	e 0.015	front 0.011	taliban 0.023	taleban 0.025
	highway 0.012	rabbani 0.012	dropped 0.010	chat 0.019	aid 0.020

played by the United Nations (UN). The corresponding special themes again suggest the difference between the two wars. The Iraq theme indicates the role of UN in sending weapon inspectors to Iraq; the Afghanistan theme refers to Northern Alliance that received aid from the UN and came to power in Afghanistan after the defeat of Taliban.

5.2 Laptop customer reviews

This data set was constructed to test our models for comparing opinions of customers on different laptops. We manually downloaded the following 3 review sets from epinions.com [4], filtering out the misplaced ones: Apple iBook (M8598LL/A) Mac Notebook (34 reviews), Dell Inspiron 8200 (8TWORH) PC Notebook (22 reviews), IBM ThinkPad T20 2647 (264744U) PC Notebook (42 reviews).

The results on this data set are generally similar to those on war news. Due to the limit of space, we only show the CCMix results in Table 3, which are obtained by setting λ_C=.7 and λ_B=.96 and fixing the number of clusters to 8. Here we again see many very interesting common themes; indeed, the top two words in the common themes can provide a very good summary of the themes (e.g., "sound and speakers" for cluster1, "battery hours" for cluster5, and "Microsoft Office" for cluster8). However, the special themes, although suggesting some differences among the three laptops, are much harder to interpret. This may be because there is a great deal of variation in product-specific opinions in the data, which makes the data extremely sparse for learning a coherent collection-specific theme for each of the eight themes.

5.3 Parameter tuning

When we vary λ_B and λ_C in CCMix, the results are generally different. Specifically, when λ_B is set to a small value, non-informative stop words tend to show up in common themes. A reasonable value for λ_B is generally higher than 0.9 – in that case, the model *automatically* eliminates the non-informative words from the theme clusters, allowing for more discriminative clustering. Indeed, in all our experiments, we have intentionally retained all the stop words, and the model is clearly able to filter out non-informative words, though in some cases, they still show up as top words in the common themes of the news data. They can be "eliminated" by using an even higher λ_B, but then we may end up having insufficient information to learn a common theme reliably. λ_C affects the vocabulary allocation between the common and collection-specific themes. In the news data experiments, when we change λ_C to a value above 0.4, the collection-specific terms would dominate the common theme models. In the laptop data experiments, when λ_C is less than 0.7, we lose many content keywords of the common themes to the corresponding collection-specific themes. Both λ_B and λ_C are intentionally left for a user to tune so that we can incorporate application-specific bias into the model.

6. RELATED WORK

The most related work to our work is the coupled clustering method presented in [8], which appears to be one of the very few studies considering the clustering problem in multiple collections. They extend the information bottleneck approach to discover common clusters across different collections. Comparative text mining goes beyond this by analyzing both the similarities and collection-specific differences. We also use a completely different approach based on probabilistic mixture models. Another related work is [10], where cross-training is used for learning classifiers from multiple document sets. Our work differs from it in that we perform unsupervised learning. The aspect models studied in [7, 2] are also related to our work but they are closer to our baseline model and are not designed for comparing multiple collections. There are many studies in document clustering [1]. Again, the difference lies in that they consider only one collection and thus are similar to the baseline model.

Our work is also related to document summarization, especially multiple document summarization (e.g.,[9, 12]). Indeed, we can the results of CTM as a special form of summary of multiple text collections. However, an important difference is that while a summary intends to retain the *explicit* information in text (to maintain fidelity), CTM aims at extracting non-obvious *implicit* patterns.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we define and study a novel text mining problem referred to as comparative text mining. It is con-

Table 3: Laptop review results using CCMix model

	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6	Cluster7	Cluster8
COMMON	sound 0.035	port 0.023	ram 0.105	m 0.027	battery 0.129	t 0.039	cd 0.095	office 0.037
	speakers 0.035	jack 0.021	mb 0.037	trackpad 0.018	hours 0.080	modem 0.017	drive 0.076	microsoft 0.021
	playback 0.034	ports 0.018	memory 0.034	chip 0.013	life 0.060	internet 0.017	rw 0.055	little 0.018
	feel 0.019	will 0.018	256mb 0.027	improved 0.012	5 0.038	later 0.014	dvd 0.049	basic 0.015
	pros 0.017	your 0.017	128mb 0.021	volume 0.012	end 0.016	configuration 0.014	combo 0.025	6 0.014
	cons 0.017	warm 0.013	tech 0.020	did 0.011	3 0.016	free 0.013	drives 0.023	under 0.013
	market 0.017	keep 0.012	128 0.020	latch 0.011	high 0.015	vga 0.012	rom 0.020	mhz 0.012
	size 0.014	down 0.012	support 0.018	make 0.010	processor 0.014	were 0.012	floppy 0.017	word 0.011
DELL	rests 0.026	banias 0.019	options 0.039	inspiron 0.061	dells 0.032	fans 0.019	apoint 0.017	0 0.046
	palm 0.022	svga 0.014	sodimm 0.025	pentium 0.052	ran 0.017	shipping 0.017	blah 0.015	angle 0.018
	9000 0.020	record 0.014	eraser 0.021	8200 0.03	prong 0.015	2nd 0.016	hook 0.011	portion 0.0154
	smart 0.018	supposedly 0.013	crucial 0.018	toshiba 0.027	requiring 0.014	tracking 0.015	tug 0.011	usb 0.0153
	reader 0.018	rebate 0.013	sdram 0.018	440 0.026	second 0.011	spoke 0.015	2499 0.011	specials 0.014
APPLE	magazine 0.011	osx 0.040	macos 0.019	macos0.016	g4 0.016	iphoto 0.031	airport 0.075	appleworks 0.060
	ipod 0.010	quartz 0.015	personal 0.018	netscape 0.013	interlaced 0.016	itunes 0.027	burn 0.035	word 0.021
	strong 0.01	instance 0.014	shield 0.016	apache 0.009	mac 0.016	import 0.021	4x 0.018	result 0.016
	icon 0.009	underneath 0.012	airport 0.016	ie5 0.008	imac 0.014	book 0.018	reads 0.014	spreadsheet 0.013
	choppy 0.008	cooling 0.012	installation 0.015	ll 0.008	powermac 0.012	quicktime 0.016	schools 0.013	excel 0.012
IBM	technology 0.023	rj 0.033	exchange 0.023	company 0.021	thinkpad 0.077	thinkpads 0.020	t20 0.04	list 0.015
	outdated 0.020	chik 0.018	hassle 0.016	570 0.017	ibm 0.047	connector 0.018	ultrabay 0.030	factor 0.013
	surprisingly 0.018	dsl 0.017	disc 0.015	turn 0.017	covers 0.029	connectors 0.018	tells 0.021	months 0.013
	trackpoint 0.014	45 0.015	t23 0.012	buttons 0.015	lightest 0.028	bluetoot 0.018	device 0.021	cap 0.013
	recommend 0.013	pacbell 0.012	cdrw 0.015	numlock 0.012	3000 0.027	sturdy 0.011	number 0.020	helpdesk 0.0128

cerned with discovering any latent common themes across a set of comparable collections of text as well as summarizing the similarities and differences of these collections along each theme.

We propose a generative cross-collection mixture model for performing comparative text mining. The model simultaneously performs cross-collection clustering and within-collection clustering, and can be applied to an arbitrary set of comparable text collections. We define the model and present the EM algorithm that can estimate the model efficiently. We evaluate the model on two different text data sets (i.e., a news article data set and a laptop review data set), and compare it with a baseline clustering method based on a simple mixture model. Experiment results show that the cross-collection mixture model is quite effective in discovering the latent common themes across collections and performs significantly better than the baseline simple mixture model. The proposed model has many obvious applications in opinion summarization and business intelligence. It also has many other less obvious applications in the general area of text mining and semantic integration of text. For example, our model can be used to compare the course web pages from the major computer science department web sites to discover core computer science topics. It can also be used to compare literature collections in different communities to support concept switching [11].

The work reported in this paper is just an initial step toward a promising new direction. There are many interesting future research directions. First, it may be interesting to explore how we can further improve the CCMix model and its estimation. One interesting direction is to explore the Maximum A Posterior (MAP) estimator, which would allow us to incorporate more prior knowledge in a principled way. For example, a user may already have certain thematic aspects in mind. With MAP estimation, we can easily add that bias to the component models. Second, we can generalize our model to model semi-structured data to perform more general comparative data mining. One way to achieve this goal is to introduce additional random variables in each component model so that we can model any structured data. Finally, it would be very interesting to explore how we could exploit the learned theme models to provide additional help to a user who wants to perform comparative analysis. For example, the learned common theme models can be used to construct a hidden Markov model (HMM) to identify the parts in the text collections about the common themes, and to connect them through automatically generated hyperlinks. This would allow a user to easily navigate through the common themes.

8. REFERENCES

[1] D. Baker and A. McCallum. Distributional clustering of words for text classification. In *Proceedings of ACM SIGIR 1998*, 1998.

[2] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statist. Soc. B*, 39:1–38, 1977.

[4] epinions.com, 2003. http://www.epinions.com/.

[5] R. Feldman and I. Dagan. Knowledge discovery in textual databases. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 1995.

[6] M. A. Hearst. Untangling text data mining. In *Proceedings of ACL'99*, 1999.

[7] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of ACM SIGIR'99*, pages 50–57, 1999.

[8] Z. Marx, I. Dagan, J. Buhmann, and E. Shamir. Coupled clustering: a method for detecting structural correspondence. *Journal of Machine Learning Research*, 3:747–780, 2002.

[9] K. McKeown, J. L. Klavans, V. Hatzivassiloglou, R. Barzilay, and E. E. Towards multidocument summarization by reformulation: Progress and prospects. In *Proceedings of AAAI-99*.

[10] S. Sarawagi, S. Chakrabarti, and S. Godbole. Cross-training: Learning probabilistic mappings between topics. In *Proceedings of ACM SIGKDD 2003*.

[11] B. R. Schatz. The interspace: Concept navigation across distributed communities. *Computer*, 35(1):54–62, 2002.

[12] H. Zha. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Proceedings of ACM SIGIR 2002*.