Rated Aspect Summarization of Short Comments

Yue Lu
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
yuelu2@uiuc.edu

ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
czhai@cs.uiuc.edu

Neel Sundaresan
eBay Research Labs
2145 Hamilton Avenue
San Jose, CA 95125
nsundaresan@ebay.com

ABSTRACT

Web 2.0 technologies have enabled more and more people to freely comment on different kinds of entities (e.g. sellers, products, services). The large scale of information poses the need and challenge of automatic summarization. In many cases, each of the user-generated short comments comes with an overall rating. In this paper, we study the problem of generating a "rated aspect summary" of short comments, which is a decomposed view of the overall ratings for the major aspects so that a user could gain different perspectives towards the target entity. We formally define the problem and decompose the solution into three steps. We demonstrate the effectiveness of our methods by using eBay sellers' feedback comments. We also quantitatively evaluate each step of our methods and study how well human agree on such a summarization task. The proposed methods are quite general and can be used to generate rated aspect summary automatically given any collection of short comments each associated with an overall rating.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Text Mining

General Terms

Algorithms

Keywords

short comments, rating prediction, rated aspect summarization

1. INTRODUCTION

As Web 2.0 technologies facilitate users to contribute rather than just retrieve information, now more and more people can freely comment on different kinds of entities (e.g. sellers, products, services). The user-contributed content in turn helps other users to make better judgments. Generally, given a target entity, we could obtain many user-generated short comments each often also with an overall rating. For example, users review and rate the products on CNET[1] from one to five stars; on eBay[2], buyers leave feedback comments to the seller and rate the transaction as positive, neutral or negative. Usually the number of comments about a target entity is of a very large scale, such as hundreds of thousands, and the number is consistently growing as more and more people keep contributing online. So the question is how to help a user better digest such a large number of comments.

In this paper, we propose to generate a "rated aspect summary" which provides a decomposed view of the overall ratings for the major aspects so that a user can gain different perspectives towards the target entity. This kind of decomposition is quite useful because different users may have quite different needs and the overall ratings are generally not informative enough. For example, a prospective eBay buyer may compromise on shipping time but not on product quality. In this case, it is not sufficient for the buyer to just know the overall ratings of a seller, and it would be highly desirable for the buyers to know the ratings of a seller on the *specific* aspect about product quality.

Rated aspect summarization can potentially help users make wiser decisions by providing more detailed information. This problem setup is illustrated in Figure 1. The input data represents what users normally can see through

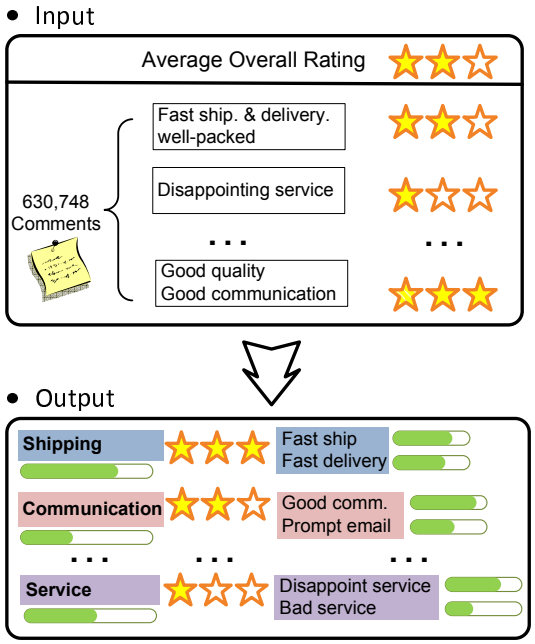

Figure 1: Problem Setup

[1]http://www.cnet.com/
[2]http://www.ebay.com/

WWW 2009, April 20–24, 2009, Madrid, Spain.
ACM 978-1-60558-487-4/09/04.

a community comment website, which generally consists of a large number of short comments with companion overall ratings. With such data, a user can only get an overall impression by looking at the average overall rating; it is infeasible to go over the large number of comments for more detailed analysis. In contrast, in the generated rated aspect summary (shown as output), the overall rating is decomposed into several aspects; each aspect has support information (the green bars) showing the confidence on the aspect rating; representative phrases with support information further enrich the rated aspects, and can serve as indices to navigate into a set of specific comments about this aspect.

This kind of rated aspect summarization is also helpful even if users do explicitly give ratings for some given aspects, because (1) we may still want to further decompose the ratings into finer sub-aspects. For example, people typically rate "food" in restaurant reviews, but users usually want to know in what sense the food is good or bad. Is there concern about healthiness or about taste? (2) the given aspects may not cover all the major aspects discussed in the text comments. In the eBay system, there are four defined aspects to rate a seller, called Detailed Seller Ratings (DSR), namely "Item as described", "Communication", "Shipping time" and "Shipping and handling charges". But it would be difficult to know the seller's performance on "packaging", "price", or "service", which might be more useful for some potential buyers.

To the best of our knowledge, this rated aspect summarization problem has not been studied in the existing work, though it is related to some existing work on opinion summarization (the connection will be further discussed in Section 5). Specifically, no previous work has attempted or proposed algorithms to decompose an overall rating into ratings on ad hoc aspects learned from the comments.

We hope to solve this novel summarization problem with no human supervision, or with minimum supervision in the case when the user wants to specify keywords to describe aspects that should be used to summarize the comments and decompose the rating. We propose to solve the rated aspect summarization problem in three steps: (1) extract major aspects; (2) predict rating for each aspect from the overall ratings; (3) extract representative phrases. In the first step, we propose a topic modeling method, called Structured PLSA, modeling the dependency structure of phrases in short comments. It is shown to improve the quality of the extracted aspects when compared with two strong baselines. In the second step, we propose to predict the aspect ratings using two different approaches, both un-supervised: Local Prediction uses the local information of the overall rating of a comment to rate the phrases in that comment; Global Prediction rates phrases based on aspect level rating classifiers which are learned from overall ratings of all comments. After the first two steps, we have the comments segmented into different aspects and different rating values. Then we could select phrases that represent what have been mostly said in this aspect.

Since this is a new task, there is no existing data set that can be used to evaluate it. We opt to create our own test set using the seller feedback comments from eBay. We design measures to evaluate each of the three components in a rated aspect summary (i.e., aspects, ratings of aspects, and representative phrases). The extracted aspects are evaluated by comparing aspect coverage and clustering accuracy against human generated aspect clusters; we use the DSR ratings in eBay as the gold standard to evaluate the aspect rating prediction, and evaluation metrics include both aspect rating correlation and ranking loss; we calculate precision and recall of the representative phrases against human labeled phrases. Evaluation results show that our proposed methods can generate useful rated aspect summaries from large amounts of short comments and overall ratings. The PLSA approach, especially the proposed Structured PLSA which leverages the phrase structures in the short comments, outperforms the k-means clustering method. Our results also show that Global Prediction generates more accurate rating prediction, but Local Prediction is sufficient at predicting a few representative phrases in each aspect.

The rest of the paper is organized as follows. In Section 2, we formally define the novel problem of rated aspect summarization. After that, we present our methods in Section 3. Then we discuss our experiments and evaluation results in Section 4. The connections with existing work are made in Section 5. Finally, we conclude in Section 6;

2. PROBLEM DEFINITION

In this section, we formally define the problem we study in this paper.

Given a large number of short comments about a target entity, each associated with an overall rating indicating different levels of overall opinion, our goal is to generate a rated aspect summary, i.e. an aspect summary with a rating for each aspect, in order to help users better digest the comments along different dimensions of the target entity. There are two application scenarios:

1. no supervision: If there is no prior knowledge of the aspects, we just automatically decompose the overall rating into purely ad hoc aspects based on the data.

2. minimum supervision: If the user could provide a couple of keywords specifying aspects he or she would be interested in, we should accommodate targeted aspect decomposition.

Formally, we denote the collection of short comments by $T = \{t_1, t_2, ...\}$, where each $t \in T$ is associated with an overall rating of $r(t)$.

Definition (Overall Rating) An overall rating $r(t)$ of a comment t is a numerical rating indicating different levels of overall opinion of t, and $r(t) \in \{r_{min}, ..., r_{max}\}$.

Usually, it is infeasible for a user to go over all the overall ratings of a large number of comments. A common way used in many real applications is to summarize them with a single number: the average overall ratings of the whole collection.

Definition (Average Overall Rating) The average overall rating of a collection of comments $R(T)$ is a score averaged over all the overall ratings: $R(T) = \frac{\sum_{t \in T} r(t)}{|T|} \in [r_{min}, r_{max}]$.

In short comments, such as the eBay feedback text, most opinions are expressed in concise phrases, such as "well packaged", "excellent seller". So with the help of some shallow parsing techniques, we could extract those phrases and identify the head term and the modifier. This also allows us to take advantage of the phrase structure to learn aspects.

Definition (Phrase) A phrase $f = (w_m, w_h)$ is in the form of a pair of head term w_h and modifier w_m. Usually the head term is an aspect or feature, and the modifier expresses some opinion towards this aspect.

Then each comment is represented by a bag of phrases $t = \{f = (w_m, w_h) | f \in t\}$ instead of a regular bag of words. After that, rated aspect summarization could be naturally decomposed into three steps:

1. identify k major aspect clusters
2. predict aspect rating for each aspect
3. extract representative phrases to support or explain the aspect ratings

Some of the concepts are defined as follows:

Definition (Aspect Cluster) An aspect cluster A_i is a cluster of head terms that share similar meaning in the given context. Those words jointly represent an aspect that users would comment on and/or would be interested in. We denote $A_i = \{w_h | A(w_h) = i\}$, where $A(.)$ is a mapping function from some aspect clustering algorithm that maps a head term to a cluster label.

Definition (Aspect Rating) An aspect rating $R(A_i)$ is a numerical measure with respect to the aspect A_i, showing the degree of satisfaction demonstrated in the comments collection T toward this aspect, and $R(A_i) \in [r_{min}, r_{max}]$.

Definition (Representative Phrase) A representative phrase $rf = (f, s(f))$ is a phrase f with a support value $s(f)$, where $s(f) \in [1, \infty)$ indicating how many phrases in the comments that this phrase can represent.

Note that, we use $r(.)$ to denote a discrete rating (an integer between r_{min} and r_{max}), and $R(.)$ to denote an average rating over a number of discrete ratings, which is a rational number (usually non-integer) between r_{min} and r_{max}. We can now define the rated aspect summary we would like to generate as follows.

Definition (Rated Aspect Summary) A rated aspect summary is a set of tuples $(A_i, R(A_i), RF(A_i))_{i=1}^{k}$, where A_i is a ratable aspect, $R(A_i)$ is the predicted rating on A_i, and $RF(A_i)$ is a set of representative phrases in this aspect.

3. METHODS

We propose several methods to solve the problem of rated aspect summarization in three steps as defined in Section 2.

3.1 Aspect Discovery and Clustering

As stated in Section 2, in short comments, opinions on different aspects are usually expressed in concise phrases. And we suppose each phrase is parsed into a pair of head term w_h and modifier w_m in the form of $f = (w_m, w_h)$. Usually the head term is about an aspect or feature, and the modifier expresses some opinion towards this aspect. In the first step, our task is to identify k interesting aspects and cluster head terms into those aspects. We propose three different approaches.

3.1.1 *k-means Clustering*

Intuitively, the structure of phrases could help with the clustering of the head terms, because if two head terms tend to use the same set of modifiers, they should share similar meaning. For example, head terms that are usually modified by "fast" should be more similar to each other compared with head terms modified by "polite" or "honest". So in the first attempt, we try to use the relation between modifiers and head terms by representing each head term w_h as a vector $v(w_h)$ in the form of

$$v(w_h) = (c(w_h, w_m^1), c(w_h, w_m^2), ...)$$

where $c(w_h, w_m^i)$ is the number of co-occurrences of head term w_h with modifier w_m^i. Then we apply k-means [10], a standard clustering algorithm shown to be effective in many clustering tasks, to a set of such vectors. The clusters output by k-means form the aspects of interest. However, the space of modifiers is usually of very high dimensionality, ranging from several hundreds to thousands. Due to the *curse of dimensionality*, the sparsity of the data could affect the clustering performance.

3.1.2 *Unstructured PLSA*

Probabilistic latent semantic analysis (PLSA) [5] and its extensions [19, 12, 11] have recently been applied to many text mining problems with promising results. If we ignore the structure of the phrases, we could apply PLSA on the head terms to extract topics, i.e. aspects.

As in most topic models, the general idea is to use a unigram language model (i.e., a multinomial word distribution) to model a topic. For example, a distribution that assigns high probabilities to words such as "shipping", "delivery", "days", would suggest a topic such as "shipping time". In order to identify multiple topics in text, we would fit a mixture model involving multiple multinomial distributions to the text data and try to figure out how to set the parameters of the multiple word distributions so that we can maximize the likelihood of the text data.

We define k unigram language models: $\Theta = \{\theta_1, \theta_2, ..., \theta_k\}$ as k theme models, each is a multinomial distribution of head terms, capturing one aspect. A comment $t \in T$ can then be regarded as a sample of the following mixture model.

$$p_t(w_h) = \sum_{j=1}^{k} [\pi_{t,j} p(w_h | \theta_j)]$$

where w_h is a head term, $\pi_{t,j}$ is a comment-specific mixing weight for the j-th aspect ($\sum_{j=1}^{k} \pi_{t,j} = 1$). The log-likelihood of the collection T is given by

$$\log p(T|\Lambda) = \sum_{t \in T} \sum_{w_h \in V_h} \{c(w_h, t) \times \log \sum_{j=1}^{k} [\pi_{t,j} p(w_h | \theta_j)]\}$$

where V_h is the set of all the head terms, $c(w_h, t)$ is the count of head term w_h in comment t, and Λ is the set of all model parameters.

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:

$$p(z_{t,w_h,j}) = \frac{\pi^{(n)}_{t,j} p^{(n)}(w_h|\theta_j)}{\sum_{j'=1}^{k} \pi^{(n)}_{t,j'} p^{(n)}(w_h|\theta_{j'})}$$

$$\pi^{(n+1)}_{t,j} = \frac{\sum_{w_h \in V_h} c(w_h,t) p(z_{t,w_h,j})}{\sum_{j'} \sum_{w_h \in V_h} c(w_h,t) p(z_{t,w_h,j'})}$$

$$p^{(n+1)}(w_h|\theta_j) = \frac{\sum_{t \in T} c(w_h,t) p(z_{t,w_h,j})}{\sum_{w'_h \in V_h} \sum_{t \in T} c(w'_h,t) p(z_{t,w'_h,j})}$$

where $p(z_{t,w_h,j})$ represents the probability of head term w_h in comment t assigned to the jth aspect.

After that, we have a set of theme models extracted from the text collection $\{\theta_i | i = 1, ..., k\}$, and now we could group each head term $w_h \in V_h$ into one of the k aspects by choosing the theme model with the largest probability of generating w_h, which is our clustering mapping function:

$$A(w_h) = \arg\max_j p(w_h|\theta_j)$$

Intuitively, if two head terms tend to co-occur with each other (such as, "ship" and "delivery" co-occurring in "fast ship and delivery") and one term is assigned a high probability, then the other generally should also be assigned a high probability in order to maximize the data likelihood. Thus this model generally captures the co-occurrences of head terms and can help cluster the head terms into aspects based on co-occurrences in comments.

3.1.3 Structured PLSA

Using a similar intuition as in the k-means clustering method, we try to incorporate the structure of phrases into the PLSA model, using the co-occurrence information of head terms and their modifiers.

Similar to Unstructured PLSA, we define k unigram language models of head terms: $\Theta = \{\theta_1, \theta_2, ..., \theta_k\}$ as k theme models. Each modifier could be represented by a set of head terms that it modifies:

$$d(w_m) = \{w_h | (w_m, w_h) \in T\}$$

which can then be regarded as a sample of the following mixture model.

$$p_{d(w_m)}(w_h) = \sum_{j=1}^{k} [\pi_{d(w_m),j} p(w_h|\theta_j)]$$

where $\pi_{d(w_m),j}$ is a modifier-specific mixing weight for the j-th aspect, which sums to one, i.e. $\sum_{j=1}^{k} \pi_{d(w_m),j} = 1$. The log-likelihood of the collection of modifiers V_m is

$$\log p(V_m|\Lambda) = \sum_{w_m \in V_m} \sum_{w_h \in V_h} \{c(w_h, d(w_m)) \times \log \sum_{j=1}^{k} [\pi_{d(w_m),j} p(w_h|\theta_j)]\}$$

where $c(w_h, d(w_m))$ is the number of co-occurrences of head term w_h with modifiers w_m, and Λ is the set of all model parameters. Using a similar EM algorithm as in Section 3.1.2, we could estimate the k theme models and obtain the clustering mapping function. For completeness, we are showing the updating formulas as follows:

$$p(z_{d(w_m),w_h,j}) = \frac{\pi^{(n)}_{d(w_m),j} p^{(n)}(w_h|\theta_j)}{\sum_{j'=1}^{k} \pi^{(n)}_{d(w_m),j'} p^{(n)}(w_h|\theta_{j'})}$$

$$\pi^{(n+1)}_{d(w_m),j} = \frac{\sum_{w_h \in V_h} c(w_h, d(w_m)) p(z_{d(w_m),w_h,j})}{\sum_{j'} \sum_{w_h \in V_h} c(w_h, d(w_m)) p(z_{d(w_m),w_h,j'})}$$

$$p^{(n+1)}(w_h|\theta_j) = \frac{\sum_{w_m \in V_m} c(w_h, d(w_m)) p(z_{d(w_m),w_h,j})}{\sum_{w'_h \in V_h} \sum_{w_m \in V_m} c(w'_h, d(w_m)) p(z_{d(w_m),w'_h,j})}$$

where $p(z_{d(w_m),w_h,j})$ represents the probability of head term w_h associated with modifier w_m assigned to the jth aspect.

Compared with Unstructured PLSA, this method models the co-occurrence of head terms at the level of the modifiers they use instead of at the level of comments they occur. Since we are working on short comments, there are usually only a few phrases in each comment, so the co-occurrence of head terms in comments is not very informative. In contrast, Structured PLSA model goes beyond the comments and organizes the head terms by their modifiers, which could use more meaningful syntactic relations.

3.1.4 Incorporating Aspect Priors

In many cases, we have some domain knowledge about the aspects. For instance, "food" and "service" are the major aspects in comments on restaurants. And sometimes a user may have specific preference on some aspects. For example, a buyer may be especially into the "packaging" aspect. In the probabilistic model framework, we could use conjugate prior to incorporate such human knowledge to guide the clustering of aspects.

Specifically, we build a unigram language model $\{p(w_h|a_j)\}_{w_h \in V_h}$ for each aspect that we have prior knowledge about. For example, a language model for a "packaging" aspect may look like

$$p(\texttt{packaging}|a_1) = 0.5$$
$$p(\texttt{wrapping}|a_1) = 0.5$$

The we could define a conjugate prior (i.e., a Dirichlet prior) on each unigram language model, parameterized as $Dir(\{\sigma_j p(w_h|a_j) + 1\}_{w_h \in V_h})$, where σ_j is a confidence parameter for the prior. Since we use a conjugate prior, σ_j can be interpreted as the "equivalent sample size" which means that the effect of adding the prior would be equivalent to adding $\sigma_j p(w_h|a_j) + 1$ pseudo counts for head term w_h when we estimate the topic model $p(w_h|\theta_j)$. Basically, the prior serves as some "training data" to bias the clustering results.

The prior for all the parameters is given by

$$p(\Lambda) \propto \prod_{j=1}^{k} \prod_{w_h \in V_h} p(w_h|\theta_j)^{\sigma_j p(w_h|a_j)}$$

where $\sigma_j = 0$ if we do not have prior knowledge on some aspect θ_j.

We can then use the Maximum A Posteriori (MAP) estimator to estimate all the parameters as follows (for Unstructured PLSA and Structured PLSA respectively)

$$\hat{\Lambda} = \arg\max_{\Lambda} p(T|\Lambda) p(\Lambda)$$

$$\hat{\Lambda} = \arg\max_{\Lambda} p(V_m|\Lambda) p(\Lambda)$$

The MAP estimate can be computed using essentially the same EM algorithm as presented above with only slightly different updating formula for the component language models. The new updating formulas are: (for Unstructured PLSA and Structured PLSA respectively)

$$p(w_h|\theta_j)^{(n+1)} = \frac{\sum_{t\in T} c(w_h,t)p(z_{t,w_h,j}) + \sigma_j p(w_h|a_j)}{\sum_{w'_h\in V_h}\sum_{t\in T} c(w'_h,t)p(z_{t,w'_h,j}) + \sigma_j}$$

$$p(w_h|\theta_j)^{(n+1)} = \frac{\sum_{w_m\in V_m} c(w_h,d(w_m))p(z_{d(w_m),w_h,j}) + \sigma_j p(w_h|a_j)}{\sum_{w'_h\in V_h}\sum_{w_m\in V_m,} c(w'_h,d(w_m))p(z_{d(w_m),w'_h,j}) + \sigma_j}$$

3.2 Aspect Rating Prediction

In the second step, we already have k aspect clusters of head terms in the form of a clustering mapping function $A(.)$. We want to predict the rating for each aspect from the overall rating without any supervision nor any external knowledge. We first propose two methods for classifying each phrase f into a rating $r(f)$ as the same scale as the overall ratings and then aspect ratings could be calculated by aggregating ratings of the phrases within each aspect.

3.2.1 Local Prediction

In the first method, we assume that the overall rating a user gives is consistent with what he or she writes in the comment. In other words, each phrase mentioned in a comment shares the same rating as the overall rating of the comment. This kind of prediction only uses the *local* information which is the overall rating of the exact comment that the phrase appears in. So the rating classifier for a phrase is

$$r(f \in t) = r(t) \in \{r_{min}, ..., r_{max}\}$$

which basically classifies the phrase into the same overall rating as the comment.

3.2.2 Global Prediction

In the second method, we do not blindly rate each phrase as the same as the overall rating of the comment it appears in. Instead, we first learn aspect level rating classifiers using the *global* information of the overall ratings of all comments. Then each phrase is classified by the globally learned rating classifier. The main idea is that by learning rating classifiers globally, we hope to correct some errors made when we only have local information available.

Specifically, for each aspect A_i, we estimate $r_{max}-r_{min}+1$ rating models empirically, each corresponding to a rating value $r \in \{r_{min}, ..., r_{max}\}$. Each rating model is a unigram language model of modifiers capturing the distribution of modifiers with the given rating value. We estimate the rating model by the empirical distribution:

$$p(w_m|A_i, r) = \frac{c(w_m, S(A_i, r))}{\sum_{w'_m\in V_m} c(w'_m, S(A_i, r))}$$

where

$$S(A_i, r) = \{f | f \in t, A(f) = i, \texttt{ and } r(t) = r\}$$

is the subset of phrases that belong to this aspect and comments containing these phrases receive the overall rating of r. After that we can classify each phrase by choosing the rating class that has the highest probability of generating the modifier in the phrase, which is basically a Naive Bayes classifier with uniform prior on each rating class.

$$r(f) = \arg\max_r \{p(w_m|A_i, r)|A(f) = i\}$$

Intuitively, the phrase rating classifier of Global Prediction should work better than that of Local Prediction. In some cases, not all the phrases in a comment is consistent with the overall rating. It is quite possible that people give a high overall rating while mentioning some short comings in the comments, and vice-versa. Suppose a comment says "slow shipping" while rated as maximum score: Local Prediction would blindly rate the phrase a maximum score; but Global Prediction could potentially tell "slow" is a low-rating on shipping, because "slow" should appear in more lowly rated comments than highly rated comments about shipping. With the globally learned classifiers, Global Prediction should be able to accommodate more noisy data, where some comments do not totally agree with their overall ratings.

3.2.3 Rating Aggregation

After we classify each phrase into different rating values using either Local Prediction or Global Prediction, the rating for each aspect A_i can be calculated by aggregating the rating of the phrases that are clustered into this aspect. A common way is to calculate the average rating of phrases within this aspect.

$$R(A_i) = \frac{\sum_{A(f)=i} r(f)}{|\{f|A(f) = i\}|}$$

$R(A_i)$ is some value between r_{min} and r_{max}, representing the average rating towards this aspect.

3.3 Representative Phrases Extraction

In the third step, we are trying to pull out some representative phrases in order to provide the users with some textual clues for better understanding of the predicted aspect rating. If our aspect clusters and aspect rating predictions are accurate, we would expect the phrases that are classified into the same aspect and same rating to be very similar to each other. So we could segment the collection of comments T into subsets of phrases for each aspect A_i and each rating value r,

$$F(A_i, r) = \{f|A(f) = i, r(f) = r\}$$

Then we could extract the top three phrases with the highest frequency in each subset. The support value for a phrases f is the frequency of the phrase in the subset

$$s(f) = c(f, F(A_i, r))$$

4. EXPERIMENTS

Rated aspect summarization is a new task which has not been studied before, so there is no existing data set available to evaluate it. In this section, we describe how we create a data set using the sellers' feedback comments on eBay. Then we present our experimental results and show both qualitative and quantitative evaluation of our methods using this data set.

4.1 Data Set and Preprocessing

We create a data set by collecting feedback comments for 28 eBay sellers with high feedback scores for the past year. The feedback score of a seller is defined as the accumulated number of positive feedback. In eBay, the feedback mechanism works as follows: after each transaction, the buyer is supposed to leave some feedback for the seller, including (1) an overall rating as positive, neutral or negative (2) Detailed Seller Ratings (DSRs) on four given aspects "Item as described", "Communication", "Shipping time" and "Shipping and handling charges" at the scale of 5 stars (3) some short comments in free text.

Then for preprocessing, we utilize the POS tagging and chunking function of the OpenNLP toolkit[3] to identify phrases in the form of a pair of head term and modifier. Some statistics of the data set is shown in Table 1.

Statistics	Mean	STD
# of comments per seller	57,055	62,395
# of phrases per comment	1.5533	0.0442
overall rating (positive %)	0.9799	0.0095

Table 1: Statistics of the Data Set

There are a few observations from the statistics: (1) Those sellers with high feedback scores receive large number of comments, 57,055 on average. But the number also varies across different sellers, as the standard deviation is very high. (2) The buyers usually use only a few phrases in each comment. After parsing, there are about 1.5 phrases per comment. Note that, the original data is more noisy. For example, user invented superlative "AAA+++" does not provide much detailed information on aspects. Our preprocessing reduces the data by about 40% in terms of the number of tokens. (3) The average overall ratings are usually very high, nearly 0.98 are positive, so they are not discriminative.

4.2 Sample Result of Rated Aspect Summarization

A sample rated aspect summarization of one of the sellers is shown in Table 2. The first column shows the automatically discovered and clustered aspects using Structured PLSA. We empirically set the number of aspects to be 8. The top two head terms in each aspect are displayed as the aspect label. The second column is the predicted ratings for different aspects using Global Prediction. Due to the mostly-positive nature of the eBay feedback, we treat both neutral and negative as rating of 0, and positive as rating of 1. So our predicted rating for each aspect would be a value between 0 and 1. Then we uniformly map our predicted rating to the 5 star ratings to produce a score between 0 and 5 as in the second column of the table. The last two columns show three representative phrases together with their frequency for each aspect and for rating 1 and 0 respectively.

It can be observed that

1) We can discover the major aspects and cluster the head terms in a meaningful way. Aspect 1 is about whether the seller truly delivers what is promised; Aspect 3 shows whether the buyers would recommend this seller; Aspect 7 talks about price. Almost all aspects are coherent and separable except that aspect 2 and aspect 4 are both talking about "shipping time".

2) The aspect ratings help us gain some insight towards this seller's performance on different aspects.

3) Although some phrases are noisy, such as "not did" and "i ordered" and some phrases are miss-classified into ratings, such "new condition" and "new item" misclassified into the rating 0 class, majority of the phrases are informative and indicate the ratings they belong to. In addition, the frequency counts could help users tell whether these opinions are representative of the major opinions.

4) We could see some correlation between the predicted aspect ratings and the phrase frequency counts: usually a high aspect rating maps to a large number of phrases in rating 1 and a small number of phrases in rating 0 and vice-versa.

We also show a sample comparison of two sellers in Table 3. Due to the limit of space, only part of the summary is displayed. We can see that although two sellers have very similar overall rating (98.66% positive V.S. 98.16% positive), Seller1 is better at providing good shipping while Seller2 is stronger at good communication. This clearly provides more detailed information than the overall rating, showing the benefit of decomposing an overall rating into aspect ratings.

Aspects	Seller1	Seller2
OVERALL	98.66%	98.16%
described	4.7967	4.8331
communication	4.5956	**4.9462**
shipping	**4.9131**	4.2244

Table 3: Sample Comparison of Two Sellers

4.3 Evaluation of Aspect Discovery and Clustering

In order to quantitatively evaluate the effectiveness of aspect discovery and clustering, we ask users to manually generate some aspect clusters as our gold standard. For each seller, we display no more than 100 head terms that have support no less than 0.1%. (for a typical seller, there are about 80 terms) We also display the term frequency and five most frequent phrases with this head term. An example is

```
price   608  0.012
        great price, good price, fair price,
        nice price, reasonable price
```

where the head term is "price", which appears 608 times in this seller's feedback comments, accounting for 1.2% of all the head terms; and the most frequent phrases with this head term are "great price, good price, fair price, nice price, reasonable price". These phrases are displayed mainly to provide the user with some context for clustering the head terms in case there is any ambiguity. Then we ask the users to cluster them into no more than 8 clusters based on their meanings. If more than 8 clusters are formed, the user is supposed to keep the top 8 clusters with highest support. Each cluster is supposed to be an aspect that a buyer would comment on. Some head terms that do not look like aspects (maybe because of parsing errors) or do not fit into top 8 clusters could be ignored.

[3]http://opennlp.sourceforge.net/

No.	Aspects	Ratings	Phrases of Rating 1	Phrases of Rating 0
1	described,promised	4.8457	as described (3993) as promised (323) as advertised (149)	than expected (68) than described(43) i ordered (10)
2	shipped,arrived	4.3301	quickly shipped (162) great thanks (149) quickly arrived (138)	open box (39) wrong sent (29) back sent (15)
3	recommended, was	3.9322	highly recommended (236) highly recommend (115) exactly was (84)	back be (42) defective was (40) not have (37)
4	shipping,delivery	4.7875	fast shipping (5354) quick shipping (879) fast delivery (647)	good shipping (170) slow shipping (81) reasonable shipping (32)
5	transaction, item	4.6943	great item (1017) great transaction (704) smooth transaction (550)	wrong item (70) new condition (48) new item (34)
6	seller,product	4.9392	great seller (2010) great product (1525) good seller (866)	poor communication (12) defective product (12) personal comm (9)
7	works,price	4.3830	great works (1158) great price (642) good price (283)	perfectly works (132) fine works (90) not working (29)
8	buy, do	4.0917	will buy (356) would buy (347) again buy (271)	not did (105) not work (91) didnt work (49)

Table 2: A Sample Result of Rated Aspect Summarization

After obtaining the human annotated gold standard for 12 sellers, we evaluate the aspect clustering algorithms by both **Aspect Coverage** and **Clustering Accuracy**.

Aspect Coverage aims at measuring how much an aspect clustering algorithm could recover the major aspects that human have identified. If the most frequent term in an algorithm output cluster matches one of the terms in a human identified cluster, we count that as an aspect match. Top K clusters are the K clusters with the largest size. Then we define Aspect Coverage at top K as the number of aspect matches within top K clusters divided by K.

However, Aspect Coverage only evaluates the most frequent term in each cluster (it could be treated as the label of a cluster); it does not measure the coherence of terms within a cluster. So we propose to use Clustering Accuracy to measure the clustering coherence performance. Given a head term w_h, let $h(w_h)$ and $A(w_h)$ be the human annotated cluster label and the label generated by some algorithm, respectively. The clustering accuracy is defined as follows:

$$\texttt{Clustering Accuracy} = \frac{\sum_{w_h \in V_h} \delta(h(w_h), map(A(w_h)))}{|V_h|}$$

where $|V_h|$ is the total number of head terms, $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and $map(A(w_h))$ is the permutation mapping function that maps each cluster label $A(w_h)$ to the equivalent label from the human annotation. The best mapping can be found by using the Kuhn-Munkres algorithm [9].

We compare three aspect clustering methods on Aspect Coverage in Figure 2 and on Clustering Accuracy in Table 4. As seen in Figure 2, both probabilistic methods, i.e. Unstructured PLSA and Structured PLSA, are good at picking up a small number of the most significant aspects (when K is small). As the number of clusters increases, the performance of three methods converge to a similar level, around 0.8. This indicates that all of the three methods could discover the 8 major aspects reasonably well. However, based on Table 4, structured PLSA achieves the best performance of Clustering Accuracy, 0.52 in bold font, meaning that the clusters are most coherent with respect to human generated clusters. This is consistent with our analysis in Section 3.1.

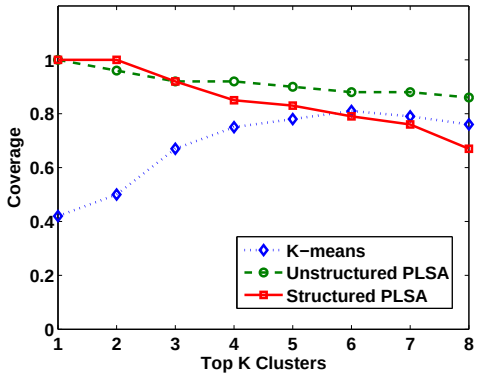

Figure 2: Evaluation of Aspect Coverage

Method	Clustering Accuracy
k-means	0.36
Unstructured PLSA	0.32
Structured PLSA	**0.52**

Table 4: Evaluation of Cluster Accuracy

We would also like to test how human agree on the coherence in such clustering task, so that we could have some sense of the "upper bound" performance. Three users are asked to label the same set of three sellers. Then the human agreement is evaluated as the clustering accuracy between each pair of users, as shown in Table 5. It can be seen that human agreement could vary a lot, from 0.5484 to 0.7846, across different annotator pairs and different data they work on. The average agreement is 0.6738. We plot the human agreement curve with different cutoffs of head term support values in Figure 3. The higher the support value is, the smaller number of head terms there will be.

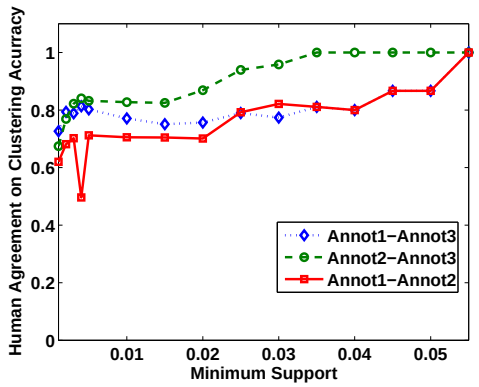

Figure 3: Human Agreement Curve on Clustering Accuracy

We would expect human to agree more on smaller number of terms. Indeed, the three curves of Clustering Accuracy, denoting three pairs of annotators, converge to 1 at some point of support value 5.5%, where there are only three or four terms left. Before that point of minimum support, most agreement still stays no more than 0.8. All these evidences show that aspect discovery and cluster could be a subjective and difficult task.

	Seller1	Seller2	Seller3	AVG
Annot1-Annot2	0.6610	0.5484	0.6515	0.6203
Annot1-Annot3	0.7846	0.6806	0.7143	0.7265
Annot2-Annot3	0.7414	0.6667	0.6154	0.6745
AVG	0.7290	0.6319	0.6604	**0.6738**

Table 5: Human Agreement on Clustering Accuracy

4.4 Evaluation of Aspect Rating Prediction

It is more difficult to evaluate the aspect rating prediction with human generated gold standard, because it would be too costly to ask human to read all the comments and rate each ad hoc aspect. Instead, we use the DSR ratings by buyers as the gold standard. As discussed in Section 3.1.4, we could use the descriptions for the four DSR criteria as priors when estimating the four aspect models, so that the discovered aspects would align with the DSR criteria defined in the eBay system. After that, we map our predicted ratings into $[0, 5]$ in order to allow comparison with the actual DSR ratings provided by buyers. Note that our algorithms do not use any information from the true DSR ratings. Instead we predict the DSR ratings based on only the comments and the overall ratings. If our algorithms are accurate, the predictions are expected to be similar to the true DSR ratings by the buyers who wrote the comments.

Since the aspect rating prediction also depends on the quality of aspect clusters, we compare our two methods of rating prediction (Local Prediction and Global Prediction) using three different aspect clustering algorithms proposed in Section 3.1. Note that, there is no easy way to incorporate such prior information into the k-means clustering algorithm. So we map the k-means clusters to four DSR criteria as a post processing step: we align the k-means cluster to a DSR if that cluster contains the description word of the DSR; if such alignment cannot be found for some DSR, we just randomly pick a cluster. We also include a baseline in our comparison which is using the positive feedback percentage to predict each aspect without extracting aspects from the comments.

We propose to evaluate the prediction from two perspectives: **Aspect Ranking Correlation** and **Ranking Loss**. Aspect Rank Correlation measures the effectiveness of ranking the four DSRs for a given seller. For example, a seller may be better at "shipping" than at "communication". We use both Kendall's Tau rank correlation and Pearson's correlation coefficient. Ranking loss [16] measures the average distance between the true and predicted ratings. The ranking loss for an aspect is equal to

$$\sum_i \frac{|actual\_rating_i - predicted\_rating_i|}{N}$$

where $N = 28$ is the number of sellers. Average ranking loss on K aspects is simply the average over each aspect. The results are shown in Table 6, and the best performance of each column is marked in bold font. A good prediction should have high correlation and low ranking loss. It can be seen that

- The aspect clustering quality indeed affects the prediction of aspect ratings. If we use k-means to cluster the aspects, no matter which prediction algorithm we use, the prediction performance is poor, even below the baseline performance especially for correlation.
- The prediction algorithm Global Prediction always performs better than Local Prediction at correlation for both Unstructured and Structured PLSA aspect clustering. This indicates that the ratings predicted by Global Prediction are more discriminative and accurate in ranking the four DSRs.
- The ranking loss performance of our methods Unstructured PLSA/Structured PLSA + Local Prediction/Global Prediction is almost always better than the baseline. The best ranking loss averaged among the four DSRs is 0.2287 given by Structured PLSA + Local Prediction compared with the baseline of 0.2865.
- The ranking loss performance also varies a lot across different DSRs. The difference is most significant on DSR 4, which is about "shipping and handling charges". However, the problem is that "charges" almost never occur in the comments, so that the aspect cluster estimated using this prior is kind of randomly related to "shipping and handling charges", resulting in the low performance on the prediction on this aspect. If we exclude this aspect and take the average of the other three ranking losses, average ranking loss performance of each algorithm improves and the best performance is achieved by Structured PLSA + Global Prediction at 0.1534 compared with 0.2365 by the baseline.

4.5 Evaluation of Representative Phrases Extraction

In order to generate gold standard for representative phrases, we utilize both the true DSR ratings and human annotation. The DSR ratings are used to generate candidate phrases at different rating level. The assumption is that if a buyer gives

Aspect Clustering	Aspect Prediction	Correlation		Ranking Loss					
		Kendal's Tau	Pearson	DSR1	DSR2	DSR3	DSR4	AVG of 4	AVG of 3
baseline		0.2892	0.3161	0.1703	0.2053	0.3332	0.4372	0.2865	0.2363
k-means	Local Prediction	0.1106	0.1735	0.1469	0.1925	0.3116	0.4177	0.2672	0.2170
k-means	Global Prediction	0.1225	-0.0250	1.3954	0.2726	0.2242	0.3750	0.5668	0.6307
Unstructured PLSA	Local Prediction	0.2815	0.4158	0.1402	0.1439	0.3092	0.3514	0.2362	0.1977
Unstructured PLSA	Global Prediction	**0.4958**	0.5781	0.2868	**0.1262**	**0.2172**	0.4228	0.2633	0.2101
Structured PLSA	Local Prediction	0.1905	0.4517	0.1229	0.1386	0.3113	**0.3420**	**0.2287**	0.1909
Structured PLSA	Global Prediction	0.4167	**0.6118**	**0.0901**	0.1353	0.2349	0.5773	0.2594	**0.1534**

Table 6: Evaluation Results on Aspect Rating Prediction

a low rating (less or equal to 3 out of 5) on an aspect, he or she will express negative opinion on this aspect in the text comments. In order to rule out the bias from our aspect clustering algorithm, we do not distinguish aspects for the phrases when displaying the phrases to the users. To summarize, we aggregate the comments with low DSR ratings and high DSR ratings respectively, and then display the most frequent 50 phrases in each set. The user is asked to select three most frequent phrases for opinions of rating 1 and rating 0 on each of the four aspects. An example output from the human annotation is as in Table 7.

Basically, the user is given a list of candidates for rating 1 phrases and a list of candidates for rating 0 phrases, and is then asked to fill in the eight cells as in Table 7. In some cases, there are no phrases that fit into some cell, such as no positive phrases for "shipping and handling charges" in this case, that cell is simply left as empty.

We apply our representative phrases extraction algorithm on top of different aspect clustering and rating prediction algorithms, and output three phrases for each of the eight cells in Table 7.

Then we could treat each cell as a "query", human generated phrases as "relevant document", and computer generated phrases as "retrieved document". Then we can calculate precision and recall as in evaluation of information retrieval:

$$\texttt{Precision} = \frac{|\{relevant\_docs\} \cap \{docs\_retrieved\}|}{|\{docs\_retrieved\}|}$$

$$\texttt{Recall} = \frac{|\{relevant\_docs\} \cap \{docs\_retrieved\}|}{|\{relevant\_docs\}|}$$

Methods	Prec.	Recall
k-means + Local Prediction	0.3055	0.3510
k-means + Global Prediction	0.2635	0.2923
Unstructured PLSA + Local Prediction	0.4127	0.4605
Unstructured PLSA + Global Prediction	0.4008	0.4435
Structured PLSA + Local Prediction	**0.5925**	**0.6379**
Structured PLSA + Global Prediction	0.5611	0.5952

Table 8: Evaluation of Representative Phrases

We report the average precision and average recall in Table 8 based on human annotation of 10 sellers. Note that when the user is filling out the cells in the table, he or she is also classifying the phrases into the four aspects and removing the phrases that are not of the right rating. So it is also an indirect way of evaluating our aspect clustering and aspect rating prediction algorithms. As we can tell from the table, (1) No matter which rating prediction algorithm we use, Structured PLSA always outperforms Unstructured PLSA which is always better than k-means; This is consistent with previous results. (2) Local Prediction always outperforms Global Prediction, independent of the underlying aspect clustering algorithm. This indicates that Local Prediction is sufficient and even better than Global Prediction at selecting only a few representative phrases for each aspect. (3) The best performance is achieved by Structured PLSA + Local Prediction at average precision of 0.5925 and average recall of 0.6379.

5. RELATED WORK

To the best of our knowledge, no previous study has addressed the problem of generating a rated aspect summary from an overall rating. But there are several lines of related work which we will review in this section.

Recently, there has been some work on summarizing online reviews. Hu and Liu [6] apply association mining to extract product features and decide the polarity the opinions using a seed set of adjective expanded via WordNet, but there is no attempt to cluster the aspects. A similar work of OPINE [15], which outperforms Hu and Liu's system both in feature extraction and opinion polarity identification, shares the same problem. Clustering can be very important in domains where aspects are described using different vocabulary or misspelling is common as in online short comments and it is especially important for accurate aggregation of ratings as in our new problem of rated aspect summarization. A different approach in the supervised framework is to learn the rules of aspect extraction from annotated data. For example, Zhuang and others [20] focused on movie review mining and summarization. The short coming is that the techniques are limited to the specific domain and highly dependent on the training data.

Sentiment classification is usually defined as the problem of binary classification of a document or a sentence [18, 14, 2, 7]. In some recent work, Pang and Lee generalize the definition into a rating scale [13]. Snyder and Barzilay [16] improve aspect level rating prediction by modeling the dependencies between aspects. This line of work aims at improving classification accuracy, which is different from our focus.

Our problem setup can be regarded as a generalization of the recent work on un-supervised *aspect* sentiment classification. Indeed if our rating is binary, the problem setup would reduce to un-supervised aspect sentiment classification. Existing work on sentiment classification almost all uses some external knowledge (in the form of word lists [6, 15] or training examples [11]) to distinguish positive and negative polarities. Our work is focused more on solving the rating decomposition problem in a general way and we propose general methods to leverage overall ratings associated with comments to predict ratings for specific aspects.

There is another line of the research in text mining that models the mixture of aspects in reviews, blog articles and

DSR Criteria	Phrases of Rating 1	Phrases of Rating 0
Item as described	as described (15609) as promised (1282) as expected 487	than expected (6)
Communication	great communication (1164) good communication (1018) excellent communication (266)	poor communication (22) bad communication (12)
Shipping time	fast shipping (28447) fast delivery (3919) quick shipping (3812)	slow shipping (251) slow delivery (20) not ship (18)
Shipping and handling charges		excessive postage (10)

Table 7: Sample Representative Phrases by Human Annotation

other text collections [4, 1, 19, 8]. Our aspect discovery and clustering algorithms are in line with that. The difference is that we add in a novel use of topic models to leverage the information from parsing the structure of phrases. A recent work of Titov and McDonald [17], jointly models text and aspect ratings, but their goal is to use rating information to identify more coherent aspects. Another limitation is that they assume a predefined set of aspects. In contrast, our work focuses on mining interesting aspects and automatically rate them using only the overall ratings, which to the best of our knowledge, has not been studied before.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we formally defined a novel problem of rated aspect summarization, which aims at decomposing the overall ratings for a large number of short comments into ratings on the major aspects so that a user can gain different perspectives towards the target entity. We proposed several general methods to solve the problem in three steps. With our methods, we could automatically generate a rated aspect summary that consists of (1) a number of major aspects; (2) predicted ratings for each of the major aspects; and (3) representative phrases that explain the predicted ratings. We have demonstrated the feasibility of automatically generating such a summary by using the seller feedback comments data of eBay. We also propose several ways to quantitatively evaluate such a new task. Results show that (1) aspect clustering is a subjective task with low human agreement, but our methods, especially Structured PLSA, perform reasonably well. (2) although based on simple assumption, Local Prediction is usually sufficient for predicting a few representative phrases in each aspect. But Global Prediction provides rating prediction with more discrimination in ranking different aspects. For the future work, we plan to combine the three steps into one optimization framework so that they could potentially benefit from each other. We are also planning to evaluate our methods on other kinds of data, such as product reviews. Another interesting future direction is to study how to compare entities (e.g. sellers, products) more effectively based on the rated aspects.

7. ACKNOWLEDGMENTS

This work was done as part of Yue Lu's internship work at eBay Research Labs in the summer of 2008. And we would like to thank Sunil Mohan and Jean-David Ruvini for preparing the data. This work is also supported in part by NSF under grant numbers 0425852 and 0713571.

8. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[2] H. Cui, V. Mittal, and M. Datar. Comparative experiments on sentiment classification for online product reviews. In *Twenty-First National Conference on Artificial Intelligence*, 2006.

[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] T. Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence*, Stockholm, 1999.

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

[6] M. Hu and B. Liu. Mining and summarizing customer reviews. In W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, editors, *KDD*, pages 168–177. ACM, 2004.

[7] S.-M. Kim and E. Hovy. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1367, 2004.

[8] W. Li and A. McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 577–584, 2006.

[9] L. Lovasz and M. Plummer. Matching theory. In *Annals of Discrete Mathematics*, North Holland, Amsterdam, 1986.

[10] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[11] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW*, pages 171–180. ACM, 2007.

[12] Q. Mei and C. Zhai. A mixture model for contextual text mining. In *KDD '06*, pages 649–655, 2006.

[13] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, pages 115–124, 2005.

[14] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, 2002.

[15] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 339–346, 2005.

[16] B. Snyder and R. Barzilay. Multiple aspect ranking using the good grief algorithm. In *HLT-NAACL*, 2007.

[17] I. Titov and R. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL-08: HLT*, pages 308–316, June 2008.

[18] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424, 2002.

[19] C. Zhai, A. Velivelli, and B. Yu. A cross-collection mixture model for comparative text mining. In *Proceedings of KDD '04*, pages 743–748, 2004.

[20] L. Zhuang, F. Jing, and X.-Y. Zhu. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50. ACM, 2006.