Learning Attitudes and Attributes from Multi-Aspect Reviews

Julian McAuley, Jure Leskovec, Dan Jurafsky
Stanford

***Abstract*—Most online reviews consist of plain-text feedback together with a single numeric score. However, there are multiple dimensions to products and opinions, and understanding the 'aspects' that contribute to users' ratings may help us to better understand their individual preferences. For example, a user's impression of an audiobook presumably depends on aspects such as the *story* and the *narrator*, and knowing their opinions on these aspects may help us to recommend better products. In this paper, we build models for rating systems in which such dimensions are *explicit*, in the sense that users leave separate ratings for each aspect of a product. By introducing new corpora consisting of five million reviews, rated with between three and six aspects, we evaluate our models on three prediction tasks: First, we uncover which parts of a review discuss which of the rated aspects. Second, we *summarize* reviews, by finding the sentences that best explain a user's rating. Finally, since aspect ratings are *optional* in many of the datasets we consider, we recover ratings that are missing from a user's evaluation. Our model matches state-of-the-art approaches on existing small-scale datasets, while scaling to the real-world datasets we introduce. Moreover, our model is able to 'disentangle' content and sentiment words: we automatically learn content words that are indicative of a particular aspect as well as the aspect-specific sentiment words that are indicative of a particular rating.**

I. Introduction

Online reviews, consisting of numeric ratings and plain-text feedback, are a valuable source of data for tasks such as product recommendation, summarization, and sentiment analysis. Making effective use of such reviews means understanding *why* users evaluated products the way they did. Did a user dislike an audiobook because of the narrator or because of the story? If a user prefers toys of a certain brand, is it because that brand's toys are fun, or because they are educational? If a user describes a beer as having 'grapefruit tones', what feature of the beer does this refer to, and are the words 'grapefruit tones' praise or criticism?[1]

Naturally, users' opinions are *multifaceted*, and answering such questions means understanding the different *aspects* that contribute to their evaluation. For example, consider the beer-rating website *BeerAdvocate*, one of the datasets included in our study. When a user evaluates a beer, their opinion is presumably influenced by the beer's look, smell, taste, and feel (palate). Furthermore, their opinions about such aspects may be conflicted: if a beer has a bad taste but a good palate, it might be described as having 'stale hops, but a velvety body'; how can we learn that 'body' refers to palate, 'hops' refers to taste, and 'stale' and 'velvety' refer to negative and positive sentiments about those aspects?

[1]This is an extended version of our ICDM paper [24].

'Partridge in a Pear Tree', brewed by 'The Bruery'

Dark brown with a light tan head, minimal lace and low retention. Excellent aroma of dark fruit, plum, raisin and red grape with light vanilla, oak, caramel and toffee. Medium thick body with low carbonation. Flavor has strong brown sugar and molasses from the start over bready yeast and a dark fruit and plum finish. Minimal alcohol presence. Actually, this is a nice quad.

Feel: 4.5 Look: 4 Smell: 4.5 Taste: 4 Overall: 4

Figure 1. An example review from *BeerAdvocate* (of the beer 'Partridge in a Pear Tree', a Californian Quadrupel). Each review consists of five ratings and free-text feedback. Our goal is to assign each of the sentences to one of the five aspects being evaluated (in this example the six sentences discuss look, smell, feel, taste, taste, and overall impression, respectively).

To answer this, we consider product rating systems in which such aspects are *explicit*, in the sense that reviews include *multiple* ratings [30], corresponding to different aspects of each product. For example, users of *BeerAdvocate* provide ratings for each of the four sensory aspects mentioned above, in addition to their overall opinion. An example review from this corpus is shown in Figure 1.

We consider three tasks on this type of data: First, can multi-aspect ratings be used as a form of weak supervision to learn language models capable of uncovering which sentences discuss each of the rated aspects? For example, using only multi-aspect rating data from many reviews (and no other labels), can we learn that 'medium thick body with low carbonation' in Figure 1 refers to 'feel'? Moreover, can we learn that the word 'warm' may be negative when describing the *taste* of a beer, but positive when describing its *color*? Second, can such a model be used to summarize reviews, which for us means choosing a subset of sentences from each review that best explain a user's rating? And third, since ratings for aspects are optional on many of the websites we consider, can missing ratings be recovered from users' overall opinions in addition to the review content?

Although sophisticated models have been proposed for this type of data [5], [12], [22], [30], our primary goal is to design models that are *scalable* and *interpretable*. In terms of scalability, our models scale to corpora consisting of several million reviews. In terms of interpretability, while topic-modeling approaches learn distributions of words used

to describe each aspect [5], [12], [30], [31], we separately model words that describe an aspect and words that describe *sentiment about an aspect*; in this way we learn highly interpretable topic and sentiment lexicons simultaneously.

A. Present work

We introduce a new model, which we name *Preference and Attribute Learning from Labeled Groundtruth and Explicit Ratings*, or PALE LAGER for short. We introduce corpora of five million reviews from *BeerAdvocate*, *RateBeer*, *Amazon*, and *Audible*, each of which have been rated with between three and six aspects. PALE LAGER can readily handle datasets of this size under a variety of training scenarios: in order to predict sentence aspects, the model can be trained with no supervision (i.e., using only aspect ratings), weak supervision (using a small number of manually-labeled sentences in addition to unlabeled data), or with full supervision (using only manually-labeled data). Using expert human annotators we obtain groundtruth labels for over ten thousand of the sentences in our corpora.

We find that while our model naturally benefits from increasing levels of supervision, our unsupervised model already obtains good performance on the tasks we consider, and produces highly interpretable aspect and sentiment lexicons. We discover that separating reviews into individual aspects is crucial when recovering missing ratings, as conflicting sentiments may appear in reviews where multiple aspects are discussed. However, we find that this *alone* is not enough to recover missing ratings, and that to do so requires us to explicitly model relationships between aspects.

Our segmentation task is similar to those of [5], [12], [30], where the authors use aspect ratings to label and rank sentences in terms of the aspects they discuss. The same papers also discuss summarization, though in a different context from our own work; they define summarization in terms of ranking sentences across multiple reviews, whereas our goal is to choose sentences that best explain a user's multiple-aspect rating. More recently, the problem of recovering missing aspect ratings has been discussed in [13], [22].

Unlike topic modeling approaches, which learn word distributions over topics for each aspect, PALE LAGER simultaneously models words that discuss an aspect and words that discuss the associated sentiment. For example, from reviews of the type shown in Figure 1, we learn that nouns such as 'body' describe the aspect 'feel', and adjectives such as 'thick' describe positive sentiment about that aspect, and we do so without manual intervention or domain knowledge. For such data we find that it is critical to separately model sentiment lexicons *per-aspect*, due to the complex interplay between nouns and adjectives [10].

B. Contributions

We introduce a new dataset of approximately five million reviews with multi-aspect ratings, with over ten thousand manually annotated sentences. The data we introduce requires a model that can be trained on millions of reviews in a short period of time, for which we adapt high-throughput methods from computer vision. The models we propose produce highly interpretable lexicons of each aspect, and their associated sentiments.

The tasks we consider have been studied in [5], [12], [30] (segmentation and ranking) and [13], [22] (rating prediction). Although these previous approaches are highly sophisticated, they are limited to corpora of at most a few thousand reviews. We use *complete* datasets (i.e., all existing reviews) from each of the sources we consider, and show that good performance can be obtained using much simpler models.

The main novelty of our approach lies in *how* we model such data. For each aspect we separately model the words that discuss an aspect and words that discuss sentiment about an aspect. In contrast to topic models, this means that we learn sentiment-neutral lexicons of words that describe an aspect (which contain nouns like 'head', 'carbonation', and 'flavor' in our *BeerAdvocate* data), while simultaneously learning sentiment lexicons for each aspect (which contain adjectives like 'watery', 'skunky', and 'metallic').

C. Further related work

Reviews consisting of plain-text feedback and a single numeric score have proved a valuable source of data in many applications, including product recommendation [2], feature discovery [27], review summarization [16], and sentiment analysis [33], among others. Understanding the multifaceted nature of such ratings has been demonstrated to lead to better performance at such tasks [12], [16], [26]. Matrix factorization techniques [18] use nothing but numeric scores to learn latent 'aspects' that best explain users' preferences, while clustering and topic-modeling approaches use nothing but review text to model 'aspects' in terms of words that appear in document corpora [4], [9], [11]. While such approaches may accurately model the data, the topics they learn are frequently not interpretable by users, nor are they representative of ratable aspects. This is precisely the issue addressed in [31], which attempts to uncover latent topics that are in some way similar to aspects on which users vote.

[29] and [30] used multi-aspect ratings as a means to summarize review corpora. As in our work, the authors of [30] use topic-models to identify 'topics' whose words are highly correlated with the aspects on which users vote, and like us they apply their model to assign aspect labels to sentences in a review. Some more recent works that deal with multi-aspect rating systems are [1], [5], [12], [23], [22]. Finally, [17] examine whether similar models can be applied *without* explicit aspect ratings, so that ratable aspects might be uncovered using only overall ratings.

A number of works model the relationship between *aspects* and *opinions on aspects* [21], [36]. In [27] sentiment

is associated with objective product features; similarly, [19] assigns sentiment labels to different product 'facets' in a corpus of camera reviews, where users manually specify facets that are important to their evaluation. [5] and [12] demonstrate that sentence aspects can be inferred from sentence sentiment, and they also introduce a publicly-available dataset which we include in our study.

Noting that aspect ratings are optional in many multi-aspect review systems, the problem of recovering missing ratings is discussed in [13], and more recently in [22]. First, we confirm their finding that multiple-aspect rating prediction depends on having separate sentiment models for each aspect. We then extend their work by explicitly modeling relationships between aspects.

Our work is also related to the discovery of *sentiment lexicons* [28], [34]. Unlike topic-modeling approaches, whose 'topics' are per-aspect word distributions, our goal is to separately model words that discuss an aspect and words that discuss sentiment about an aspect. Lexicon discovery is discussed in [25], and has been used for review summarization by [3], though such approaches require considerable manual intervention, and are not learned automatically. The use of language in review systems such as those we consider is discussed in [10] and [14], whose findings are consistent with the lexicons we learn.

II. Datasets

The beer-rating websites *BeerAdvocate* and *RateBeer* allow users to rate beers using a five-aspect rating system. Ratings are given on four sensory aspects (feel, look, smell, and taste), in addition to an overall rating. From *BeerAdvocate* we also obtain reviews of *Pubs*, which are rated in terms of food, price, quality, selection, service, and vibe.

All *Amazon* product reviews allow users to rate items in terms of their overall quality. The *Toys & Games* category allows users to provide further feedback, by rating products in terms of fun, durability, and educational value.

Finally, the audiobook rating website *Audible* allows users to rate audiobooks in terms of the author and the narrator, in addition to their overall rating.

These datasets are summarized in Table I. The 'CC' column shows the average correlation coefficient across all pairs of aspects. Our *Pubs* data has the lowest correlation between aspects, and in fact some aspects are negatively correlated (price is negatively correlated with both food and service, as we would expect for pub data).

We briefly mention a variety of other websites that provide similar rating systems, including *TheBeerSpot*, *BeerPal*, *Yahoo! Hotels*, *Yahoo! Things to Do*, *TigerDirect*, *BizRate*, *TripAdvisor*, and *DP-Review*, among others.

We also obtained the *CitySearch* dataset used in [5], [12]. This dataset consists of 652 reviews, which are labeled using four aspects (food, ambiance, price, and staff). This data differs from our own in the sense that aspects are not determined from rating data (*CitySearch* includes only overall ratings), but rather aspects and user sentiment are determined by human annotators. Sentiment labels are *per sentence* (rather than per review), so to use their data with our method, we treat their sentiment labels (positive, negative, neutral, and conflicted) as four different ratings. We adapt our method so that ratings are indexed *per sentence* rather than *per aspect*, though we omit details for brevity.

A. Groundtruth labels

Since we wish to learn which aspects are discussed in each sentence of a review, to evaluate our method we require groundtruth labels for a subset of our data. Our first author manually labeled 100 reviews from each of our datasets, corresponding to 4,324 sentences in total. Labels for each sentence consist of a single aspect, in addition to an 'ambiguous/irrelevant' label.

For our *BeerAdvocate* data, we obtained additional annotations using crowdsourcing. Using Amazon's Mechanical Turk, we obtained labels for 1,000 reviews, corresponding to 9,245 sentences. To assess the quality of these labels, we computed Cohen's kappa coefficient (to be described in Section IV-D), a standard measure of agreement between two annotators. Unfortunately, the Mechanical Turk annotations agreed with our own labels in only about 30% of cases, corresponding to $\kappa = 0.11$, which is not significantly better than a random annotator (the 4% of sentences labeled as 'ambiguous' were not used for evaluation).

To address this, we used the crowdsourcing service *oDesk*, which allows requesters to recruit individual workers with specific skills. We recruited two 'expert' beer-labelers based on their ability to answer some simple questions about beer. Both annotators labeled the same 1,000 reviews independently, requiring approximately 40 hours of work. These experts agreed with a kappa score of 0.93, and obtained similar scores against the 100 reviews labeled by our first author (who is also a beer expert).

Since *RateBeer* reviews are similar to those of *BeerAdvocate*, rather than annotating reviews from both corpora, from *RateBeer* we obtained annotations from non-English reviews (which are far more common in the *RateBeer* corpus). We identified 1,295 Spanish and 19,998 French reviews, and annotated 742 sentences from the two corpora, again with the help of expert labelers.

Code, data, and groundtruth labels shall be released at publication time.

III. The Pale Lager Model

Pale Lager models aspects, and ratings on aspects, as a function of the words that appear in each sentence of a review. Our goal is to simultaneously learn which words discuss a particular aspect, and which words are associated with a particular rating. For example, in our *BeerAdvocate* data, the word 'flavor' might be used to discuss the 'taste'

Table I
DATASET STATISTICS.

DATASET	ASPECTS	#USERS	#ITEMS	#REVIEWS	CC
Beer (beeradvocate)	feel, look, smell, taste, overall	33,387	66,051	1,586,259	0.64
Beer (ratebeer)	feel, look, smell, taste, overall	40,213	110,419	2,924,127	0.66
Pubs (beeradvocate)	food, price, quality, selection, service, vibe	10,492	8,763	18,350	0.29
Toys & Games (amazon)	durability, educational, fun, overall	79,994	267,004	373,974	0.65
Audio Books (audible)	author, narrator, overall	7,009	7,004	10,989	0.74

aspect, whereas the word 'amazing' might indicate a 5-star rating. Thus if the words 'amazing flavor' appear in a sentence, we would expect that the sentence discusses 'taste', and that the 'taste' aspect has a high rating. As a first approximation, nouns can be thought of as 'aspect' words, and adjectives as 'sentiment' words; we find that this intuition closely matches the parameters we learn.

We first introduce the notation used throughout the paper. Suppose our review corpus $(\mathcal{R}, \mathcal{V})$ consists of *reviews* $\mathcal{R} = \{r_1 \dots r_R\}$ and *ratings* $\mathcal{V} = \{v_1 \dots v_R\}$. Next assume that each review r_i is divided into *sentences* $s \in r_i$, and that each rating v_i is divided into K *aspects* $\{v_{i1} \dots v_{iK}\}$ (e.g. ratings on smell, taste, overall impression, etc.). Finally, assume that each sentence is further divided into *words*, $w \in r_{is}$.

We assume that each sentence in a review discusses a single aspect; alternately we could model one aspect per *word* or per *paragraph*, though one aspect per sentence matches what appears in existing work [5], [12], [30].

Our goal is to differentiate words that discuss an aspect from words that discuss the associated sentiment. To do so, we separate our model into two parameter vectors, θ and ϕ, which respectively encode these two properties. In our model, the probability that a sentence s discusses a particular aspect k, given the ratings v associated with the review, is

$$P^{(\theta,\phi)}(\text{aspect}(s) = k \mid \text{sentence } s, \text{rating } v) = \frac{1}{Z_s^{(\theta,\phi)}} \exp \sum_{w \in s} \Big\{ \underbrace{\theta_{kw}}_{\text{aspect weights}} + \underbrace{\phi_{kv_kw}}_{\text{sentiment weights}} \Big\}. \quad (1)$$

The normalization constant Z_s is

$$Z_s^{(\theta,\phi)} = \sum_{k=1}^{K} \exp \sum_{w \in s} \Big\{ \theta_{kw} + \phi_{kv_kw} \Big\}. \quad (2)$$

Note that θ_k is indexed by the aspect k, so that we learn which words are associated with each of the K aspects. Alternately, ϕ_{kv_k} is indexed by the aspect k, *and the rating for that aspect* v_k; this way, for each aspect we learn which words are associated with each star rating. While using θ and ϕ together has no more expressive power than using ϕ alone, we find that separating the model in this way is critical for interpretability. Another option would be to have a single sentiment parameter ϕ_{v_k} for all aspects; however, we find that each aspect uses different sentiment words (e.g. 'delicious' for taste, 'skunky' for smell), so it is beneficial to learn sentiment models per-aspect [10].

Assuming that aspects for each sentence are chosen independently, we can write down the probability for an entire review (and an entire *corpus*) as

$$p^{(\theta,\phi)}(\text{aspects}|\mathcal{R}, \mathcal{V}) = \prod_{i=1}^{R} \prod_{s \in r_i} p^{(\theta,\phi)}(\text{aspect}(s)|s, v_i). \quad (3)$$

We will now show how to learn aspect labels and parameters so as to maximize this expression.

IV. LEARNING

We describe three learning schemes, which use increasing levels of supervision in the form of sentence labels. As we show in Section VI, increased supervision leads to higher accuracy, though even without supervision we can obtain good performance given enough data.

A. Unsupervised Learning

Unsupervised learning proceeds by choosing the parameters $(\hat{\theta}, \hat{\phi})$ and the latent aspect assignments $\hat{t}$ so as to maximize the log-likelihood of the corpus:

$$(\hat{\theta}, \hat{\phi}), \hat{t} = \underset{(\theta,\phi),t}{\text{argmax}} \underbrace{\log p^{(\theta,\phi)}(t|\mathcal{R}, \mathcal{V})}_{\text{corpus probability}} - \underbrace{\Omega(\theta, \phi)}_{\text{regularizer}}. \quad (4)$$

Optimization proceeds by coordinate ascent on (θ, ϕ) and t, i.e., by alternately optimizing

$$t^i = \underset{t}{\text{argmax}} \log p^{(\theta,\phi)^i}(t|\mathcal{R}, \mathcal{V}) \quad (5)$$

$$(\theta, \phi)^{t+1} = \underset{(\theta,\phi)}{\text{argmax}} \log p^{(\theta,\phi)}(t^i|\mathcal{R}, \mathcal{V}) - \Omega(\theta, \phi) \quad (6)$$

until convergence, i.e., until $t^i = t^{i-1}$. Optimizing (eq. 5) merely consists of maximizing (eq. 1) independently for each sentence. Noting that the model is concave in (θ, ϕ), optimization of (eq. 6) proceeds by gradient ascent, where partial derivatives can be easily calculated. We regularize using the squared ℓ_2 norm, $\Omega(\theta, \phi) = ||\theta||_2^2 + ||\phi||_2^2$.

Being a local optimization procedure, coordinate ascent is sensitive to initialization. We initialize θ by setting $\theta_{k,k} = 1$ for each aspect k (e.g. $\theta_{taste,\text{`}taste\text{'}} = 1$). In practice this means that initially a sentence is assigned to an aspect if it explicitly mentions the name of that aspect. Other parameters were initialized randomly; we selected the model with the highest log-likelihood among 64 random restarts.

Finally, we note that (eq. 1) is underconstrained, in the sense that adding a constant to θ_{kw} and subtracting the same

s_1: Clear copper colored brew, medium cream colored head.
s_2: Floral hop nose, caramel malt.
s_3: Caramel malt front dominated by a nice floral hop backround.
s_4: Grapefruit tones.
s_5: Very tasty hops run the show with this brew.
s_6: Thin to medium mouth.
s_7: Not a bad choice if you're looking for a nice hop treat.

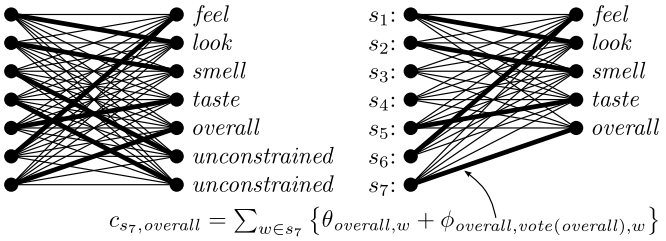

Figure 2. Both our segmentation and summarization tasks can be expressed as weighted bipartite graph cover. Each of the sentences at left (from a *BeerAdvocate* review) must be matched to an aspect. The optimal cover is highlighted using bold edges. For the segmentation task (left graph), five nodes are constrained to match to each of the five aspects, ensuring that each aspect appears at least once in the segmentation (the remaining two unconstrained aspects are both 'smell' in this case). The summarization task (right graph) includes precisely one node for each aspect, so that each aspect is summarized using the sentence that most closely aligns with that aspect's rating.

constant from $\phi_{k \cdot w}$ has no effect on the model. To address this, we add an additional constraint that

$$\sum_v \phi_{kvw} = 1, \text{ for all } k, w. \tag{7}$$

This has no effect on performance, but in our experience leads to significantly more interpretable parameters.

B. Enforcing Diversity in the Predicted Output

An issue we encountered with the above approach is that 'similar' aspects tended to coalesce, so that sentences from different aspects were erroneously assigned a single label. For example, on *BeerAdvocate* data, we noticed that 'smell' and 'taste' words would often combine to form a single aspect. From the perspective of the regularizer, this makes perfect sense: ratings for 'smell' and 'taste' are highly correlated, and similar words are used to describe both; thus setting one aspect's parameter vector to zero significantly reduces the regularization cost, while reducing the log-likelihood only slightly.

To address this, we need to somehow enforce *diversity* in our predictions, that is, we need to encode our knowledge that all aspects should be discussed. In practice, we enforce such a constraint *per-review*, so that we choose the most likely assignments of aspects, subject to the constraint that each aspect is discussed at least once. We find this type of constraint in computer vision applications: for example, to localize a pedestrian, we might encode the fact that each of their limbs must appear exactly once in an image [6]. Instead of matching image coordinates to limbs, we match sentences to aspects, but otherwise the technology is the same. In [6], such a constraint is expressed as *bipartite graph cover*, and is optimized using *linear assignment*.

We construct a bipartite graph for each review r, which matches $|r|$ sentences to $|r|$ aspects. From (eq. 1) we define the compatibility between a sentence s and aspect k:

$$c_{sk} = \sum_{w \in s} \{\theta_{kw} + \phi_{kv_k w}\}. \tag{8}$$

Next we define edge weights in terms of this compatibility function. Noting that each of the K aspects *must* be included in the cover, our weight matrix $A^{(r)}$ is defined as

$$A^{(r)}_{s,l} = \overbrace{\begin{cases} c_{sl} & \text{if } 1 \le l \le K \\ \max_k c_{r_s k} & \text{otherwise} \end{cases}}^{\substack{\text{each of the } K \text{ aspects must have a matching sentence} \\ \text{other sentences can match any aspect}}}, \tag{9}$$

and the optimal cover is given by

$$\hat{f} = \operatorname*{argmax}_f \sum_{s \in r} A^{(r)}_{s,f(s)}, \tag{10}$$

which is found using the Kuhn-Munkres algorithm.

This entire procedure is demonstrated in Figure 2, where the assignment matrix $A^{(r)}$ is visualized using a weighted bipartite graph, so that $\hat{f}$ becomes a cover of that graph. The nodes on the left of the graph correspond to the sentences in the review, while the nodes on the right correspond to their assignments. K of the nodes on the right are constrained to match each of the K aspects, while the remaining nodes may match to any aspect.

The same bipartite matching objective can also be used for our summarization task. Here, our goal is to predict for each aspect the sentence that best explains that aspect's rating. Using the compatibility function of (eq. 8), our goal is now to choose the K sentences that are *most compatible* with the K aspects. This idea is depicted on the right of Figure 2. These constraints are discarded for reviews with fewer than K sentences.

If the hard constraint that every aspect must be discussed proves too strong, it can be relaxed by adding additional 'unconstrained' nodes: for example, adding two additional nodes would mean that each review must discuss at least $K-2$ unique aspects. Alternately, in datasets where aspects are easily separable (such as *CitySearch*), this constraint can be discarded altogether, or discarded at test time. However, for datasets whose aspects are difficult to distinguish (such as *BeerAdvocate*), this constraint proved absolutely critical.

C. Semi-Supervised Learning

The semi-supervised variant of our algorithm is no different from the unsupervised version, except that the probability

is conditioned on some fraction of our groundtruth labels t', i.e., our optimization problem becomes

$$(\hat{\theta},\hat{\phi}),\hat{t} = \underset{(\theta,\phi),t}{\operatorname{argmax}} \underbrace{\log p^{(\theta,\phi)}(t|\mathcal{R},\mathcal{V},t')}_{\text{corpus probability}} - \underbrace{\Omega(\theta,\phi)}_{\text{regularizer}}. \quad (11)$$

In addition, we initialize the parameters θ and ϕ so as to maximize the likelihood of the observed data t'.

D. Fully-Supervised Learning

Given fully-labeled data, it would be trivial to choose $\hat{\theta}$ and $\hat{\phi}$ so as to maximize the log-likelihood of (eq. 4). However, a more desirable option is to learn parameters so as to directly optimize the criterion used for evaluation.

Cohen's kappa statistic is a standard accuracy measure for document labeling tasks [7]. It compares an annotator or algorithm's performance to that of a random annotator:

$$\kappa(a,b) = \frac{P(a \text{ agrees with } b) - 1/K}{1 - 1/K}. \quad (12)$$

$\kappa = 0$ corresponds to random labeling, $0 < \kappa \leq 1$ corresponds to some level of agreement, while $\kappa < 0$ corresponds to disagreement. If two annotators a and b label a corpus with aspects $t^{(a)}$ and $t^{(b)}$, then

$$P(a \text{ agrees with } b) = 1 - \Delta_{0/1}(t^{(a)}, t^{(b)}), \quad (13)$$

where $\Delta_{0/1}$ is the $0/1$ loss. Critically, since kappa is a monotonic function of the $0/1$ loss, a predictor trained to minimize the $0/1$ loss will maximize Cohen's kappa statistic. We train a predictor based on the principle of regularized risk minimization, i.e., we optimize

$$\hat{\theta},\hat{\phi} = \underset{\theta,\phi}{\operatorname{argmin}} \underbrace{\Delta_{0/1}(t^{(\theta,\phi)},t')}_{\text{empirical risk}} + \underbrace{\Omega(\theta,\phi)}_{\text{regularizer}}, \quad (14)$$

so that $\hat{\theta}$ and $\hat{\phi}$ are chosen so as to minimize the $0/1$ loss on some training data t' provided by an annotator.

If not for the diversity constraint of Section IV-B, optimization of (eq. 14) would be independent for each sentence, and could be addressed using a multiclass SVM or similar technique. However, the diversity constraint introduces *structure* into the problem so that predictions cannot be made independently. Thus we require an optimization technique designed for structured output spaces, such as that of [32]. The use of bipartite graph cover objectives in structured learning is addressed in [6], where an objective similar to that of (eq. 10) is used to match keypoints in images. We adapt their framework to our problem, which can be shown to minimize a convex upper bound on (eq. 14).

V. Learning to Predict Ratings from Text

In many websites with multiple aspect ratings, ratings for aspects are *optional*, while only 'overall' ratings are mandatory. For example, our 10,989 *Audible* reviews represent only those where all three aspects (author, narrator, overall) were rated. In total there were 199,810 reviews in our crawl that included an overall vote but were missing an aspect rating. Predicting such missing ratings may help us to understand *why* users voted the way they did. We will learn models for this task from users who entered complete ratings.

A naïve solution would be to learn parameters γ_{kv_kw} for each aspect k and rating v_k, using fully-rated reviews as training data. That is, each rating v_{ik} for review r_i and aspect k would be predicted according to

$$v_{ik}^{(\gamma)} = \underset{v}{\operatorname{argmax}} \sum_{w \in r_i} \gamma_{kvw}. \quad (15)$$

We shall see in Section VI that this proves ineffective when users have mixed feelings about different aspects: both positive and negative words appear together in reviews, making it difficult to 'tease-apart' users' opinions.

An appealing solution to this problem consists of using *segmented* text to predict ratings for each aspect, i.e.,

$$v_{ik}^{(\gamma)} = \underset{v}{\operatorname{argmax}} \underbrace{\sum_{s \in r_i} \delta(\hat{t}_{is} = k)}_{\text{sentences labeled with aspect } k} \sum_{w \in s} \gamma_{kvw}. \quad (16)$$

However, we found that this approach *also* performs poorly, even when highly accurate sentence labels are available [22]. A simple explanation is that different aspects are highly correlated: for example, when learning naïve predictors from unsegmented text on *BeerAdvocate* data as described in (eq. 15), we found that the word 'skunky' was among the strongest 1-star predictors *for all aspects*, even though the word clearly refers only to smell. Not surprisingly, a product that smells 'skunky' is unlikely to be rated favorably in terms of its taste; by predicting ratings from segmented text as in (eq. 16), we fail to exploit this correlation.

Instead, the model we propose uses segmented text, but explicitly encodes relationships between aspects. Our model is depicted in Figure 3. In addition to conditioning on segmented text, the 'smoothness' term α encodes how likely two ratings are to co-occur for different aspects:

$$v_i^{(\gamma,\alpha)} = \underset{v}{\operatorname{argmax}} \sum_k \sum_{s \in r_i} \delta(t_{is} = k) \sum_{w \in s} \gamma_{kv_kw} + \sum_{i \neq j} \alpha_{ijv_iv_j}. \quad (17)$$

For example, $\alpha_{smell,taste,1,5}$ encodes the penalty for a 1-star 'smell' vote to co-occur with a 5-star 'taste' vote; in practice α prevents such an unlikely possibility from occurring.

We train each of the above predictors (eqs. 15, 16, and 17) so as to minimize the ℓ_2 error of the prediction compared to the groundtruth ratings used for training, i.e.,

$$(\hat{\gamma},\hat{\alpha}) = \underset{\gamma,\alpha}{\operatorname{argmin}} \sum_{i=1}^{R'} \sum_{k \neq overall} ||v_{ik}^{(\gamma,\alpha)} - v_{ik}||_2 + \Omega(\gamma,\alpha).$$

We optimize this objective using a multiclass SVM in the case of (eqs. 15 and 16), though for (eq. 17) the term α introduces dependencies between ratings, so we again use structured learning techniques as in Section IV-D [32].

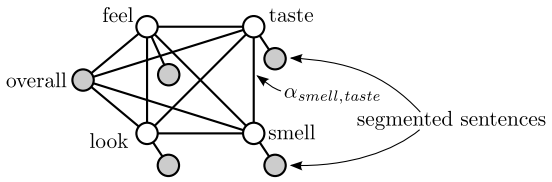

Figure 3. Graphical model for predicting ratings from segmented text. Grey nodes represent observations, whereas white nodes represent variables. The model incorporates both segmented text and relationships between aspects.

Table II
CITYSEARCH RESULTS, USING ACCURACY SCORES FROM [22].

Always label as 'food'	0.595
LDA [4]	0.477
MultiGrain LDA [31]	0.760
Segmented Topic Models [8]	0.794
Local LDA [5]	0.803
Support Vector Machine	0.830
PALE LAGER, unsupervised	0.751
PALE LAGER, semi-supervised	0.805
PALE LAGER, fully-supervised	0.892

VI. EXPERIMENTS

We evaluate PALE LAGER on our segmentation, summarization, and rating prediction tasks. Segmentation requires us to predict aspect labels for each sentence in our review corpora, while summarization requires us to choose one sentence per aspect for each review. For the first two tasks we report the accuracy (i.e., the fraction of correct predictions), which is related to Cohen's kappa by (eq. 12). For rating prediction we report the ℓ_2 error of the predicted ratings, after scaling ratings to be in the range $[0, 1]$. Even in the largest experiments we report, PALE LAGER could be trained in a few hours using commodity hardware.

We randomly split groundtruth data from each of our corpora into training and test sets. Our *unsupervised* algorithm uses entire corpora, but ignores groundtruth sentence labels; our *semi-supervised* algorithm also uses entire corpora, and conditions on the labeled training data; our *fully-supervised* algorithm uses *only* the labeled training data. All algorithms are evaluated on groundtruth labels from the test set. For our rating prediction task, we further split our unlabeled data into training and test sets, so that our segmentation algorithms are not trained using the ratings we are trying to predict.

A. Review Segmentation

Figure 4 (top) shows the performance of PALE LAGER on the seven datasets we consider. As expected, semi-supervised learning improves upon unsupervised learning (by 45% on average), and fully-supervised learning outperforms semi-supervised learning by a further 17%; the sole exception occurs on *Audible* data, which is possibly due to overfitting. Despite the good performance of our unsupervised method on *BeerAdvocate* data, it performs poorly on non-English *RateBeer* data. The simplest explanation is merely the paucity of non-English data, revealing that while this task can be approached without supervision, it requires many reviews to do so (though this could be addressed using seed-words). Once we add supervision, we observe similar performance across all three beer datasets.

As a baseline we compare PALE LAGER to *Latent Dirichlet Allocation* [4]. We train LDA with different numbers topics, and use our training labels to identify the optimal correspondence between topics and aspects (so in this sense the process is *semi-supervised*). Our semi-supervised model outperforms this baseline in 5 out of 7 cases and by 48% on average; two exceptions occur in datasets where users tend to focus on their overall evaluation and do not discuss aspects (e.g. toy reviews rarely discuss durability or educational value). We acknowledge the existence of more sophisticated variants of LDA, though we are not aware of suitable alternatives that scale to millions of reviews; we used *Online LDA* as implement in *Vowpal Wabbit* [15], which required a few hours to train on our largest dataset.

In all experiments fully-supervised learning outperforms semi-supervised learning, even though the semi-supervised algorithm has access to both labeled *and* unlabeled data. An explanation is that our semi-supervised algorithm optimizes the log-likelihood, while the fully-supervised algorithm directly optimizes the accuracy score used for evaluation. It is certainly possible that by using latent-variable structured learning techniques our fully-supervised algorithm could be extended to make use of unlabeled data [35].

1) Performance on CitySearch data: To our knowledge, the 652 review *CitySearch* dataset from [5], [12] is the only publicly-available dataset for the aspect labeling task we consider. In Table II we report published results from [22]. For comparison we used the same seed-words as in their study, which aid performance significantly. Note that in this dataset supervision takes the form of *per-sentence* ratings, rather than *per-aspect* ratings, though our method can be adapted to handle both. PALE LAGER is competitive with highly sophisticated alternatives, while requiring only a few seconds for training. The supervised version of our model, which jointly models text and ratings, outperforms (by 7%) an SVM that uses text data alone. Overall, this is a promising result: our method is competitive with sophisticated alternatives on a small dataset, and scales to the real-world datasets we consider.

B. Review Summarization

In the context of our model, summarization means identifying a subset of sentences that best explain a user's multiple-aspect rating. Specifically, for each review r_i and aspect k, we choose the sentence that maximizes the score for that aspect given the aspect's rating v_{ik}, as described in Section IV. This setup is motivated by the findings of

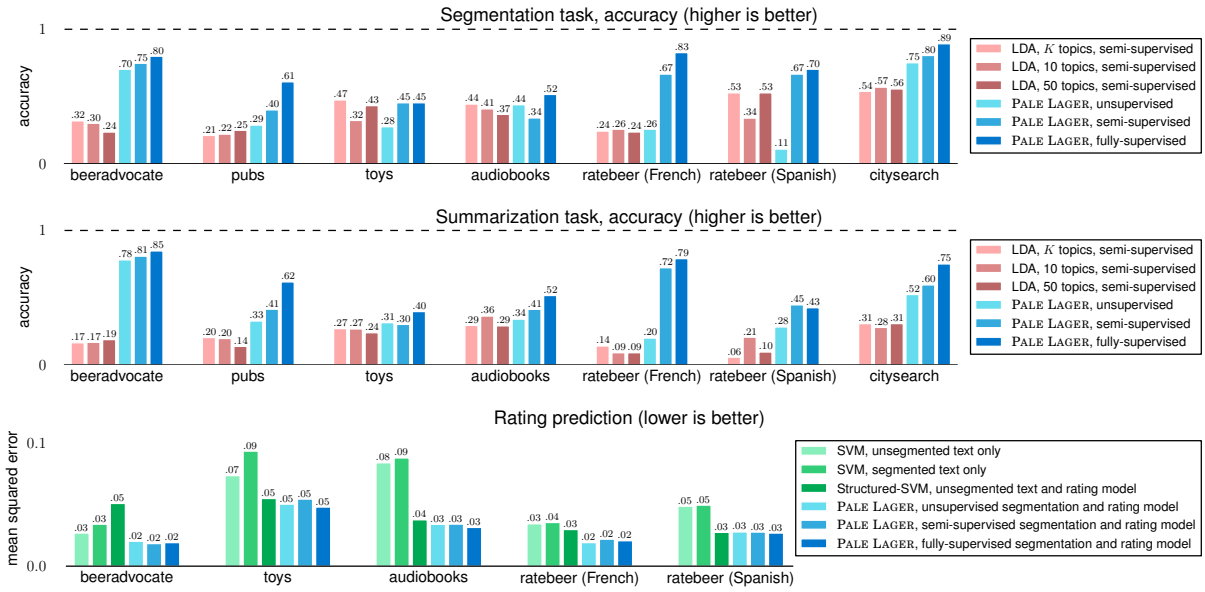

Figure 4. Performance of PALE LAGER and baselines on our segmentation task (top), our summarization task (middle), and our rating prediction task (bottom). Results are shown in terms of accuracy (higher is better) and mean squared error (lower is better).

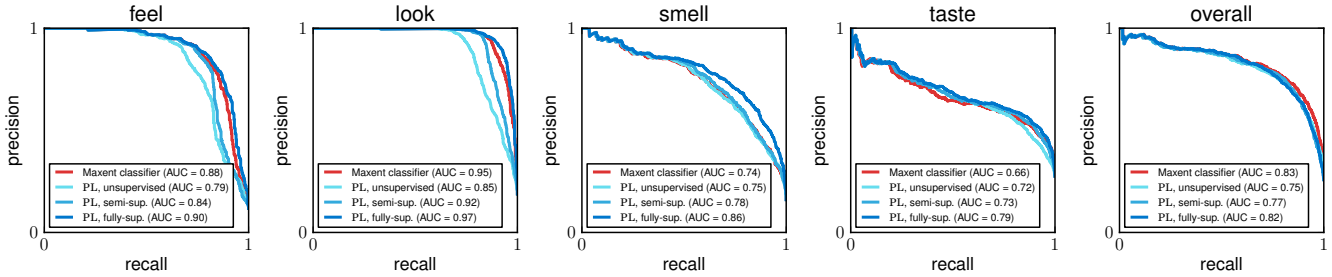

Figure 5. Precision recall curves for sentence ranking (blue curves are the PALE LAGER model). Like in [30], we find that our unsupervised method achieves performance close to that of a fully-supervised Maximum Entropy classifier. However, we also report that our semi-supervised method matches the performance of the maxent classifier, and our fully supervised method outperforms maximum entropy classification significantly. In terms of Mean Average Precision, Maxent = 0.82 (fully-supervised), PALE LAGER = 0.76 (unsupervised), 0.81 (semi-supervised), 0.87 (fully-supervised).

[20]; they show that users prefer summaries that discuss *sentiments* about various aspects of a product, rather than merely the aspects themselves.

Results for this task are shown in Figure 4 (middle). As before, increased supervision improves performance in almost all cases (semi-supervised learning beats unsupervised learning by 34%, and fully-supervised learning further improves performance by 17%). Note that summarization is not necessarily 'easier' than segmentation, and both have higher scores on different datasets. Summarization is easiest when users discuss a variety of aspects, while segmentation is easiest when users primarily discuss 'easy to classify' aspects. In practice, performance on both tasks is highly correlated. For this task, PALE LAGER outperforms LDA significantly, since LDA incorrectly labels infrequently-discussed aspects, and doesn't make use of rating data.

1) Aspect Ranking: Some works define summarization in terms of *ranking* [5], [30], [29]. For each aspect, probabilities are computed for each sentence, which are sorted to produce a ranking. Summarization can then be cast as retrieving the most relevant sentences for each aspect. Although the data from [30] are not available, for the sake of comparison we reproduce their experimental setup and baselines.

Figure 5 shows aspect ranking results on *BeerAdvocate* data. On their own data, [30] reported that their unsupervised method performed only 5% worse than a fully-supervised maxent classifier. We report a similar result for our own unsupervised method (MAP=0.76 vs. 0.82), though we find that unsupervised learning outperforms maxent classification for two out of five aspects. Furthermore, our semi-supervised algorithm matches the performance of maxent classification, and our fully-supervised method outperforms it by 7%.

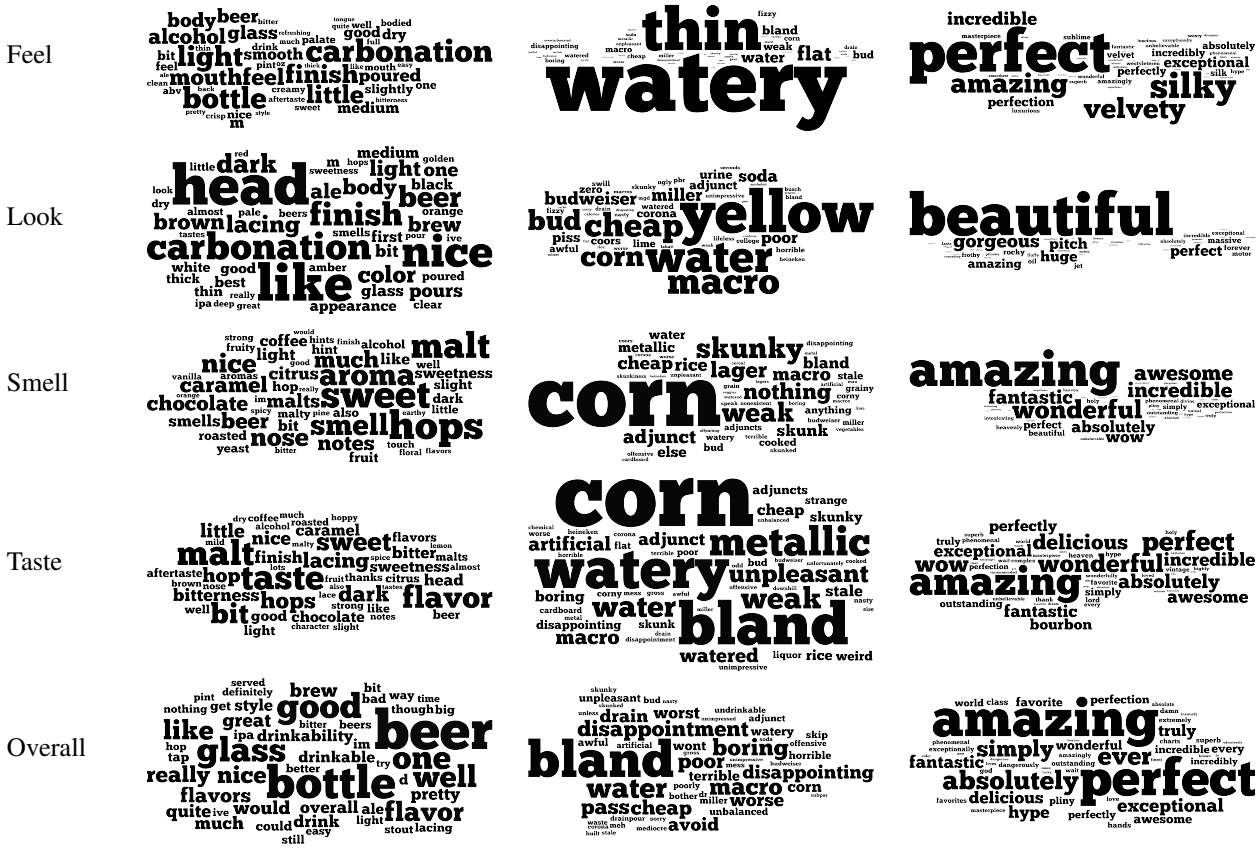

Figure 6. Word-cloud visualization of aspect parameters θ_k and sentiment parameters ϕ_{kv_k} learned from *BeerAdvocate* data. Word sizes reflect the weights θ_{kw} and ϕ_{kv_kw} for each word w. Rows show different aspects k; the left column shows 'aspect' weights θ_k, the center column shows 2-star 'sentiment' weights $\phi_{k,2}$, and the right column shows 5-star sentiment weights $\phi_{k,5}$ (1-star sentiment weights proved too unwholesome for publication). Parameters in this figure were learned using the *unsupervised* version of our model.

C. Rating Prediction

In many of the datasets we consider, only 'overall' ratings are compulsory while aspect ratings are optional. In this section we try to recover such missing aspect ratings. To measure performance on this task we train on half of our reviews to predict ratings for the other half. Naturally we ensure that none of the data used for evaluation were used during any stage of training, i.e., the segmentation models used in this experiment were *not* trained using the reviews on which we predict ratings.

Rating prediction performance is shown in Figure 4 (bottom). We exclude *Pubs* data as it includes no overall rating, and *CitySearch* data as ratings are per-sentence rather than per-review. As expected, ratings predicted from unsegmented text are inaccurate, as conflicting sentiments may appear for different aspects. More surprisingly, using *segmented* text does *not* solve this problem (in fact it is 32% *worse*), even when we have accurate aspect labels. A similar result was reported by [22], who found that models capable of segmenting text from ratings are not necessarily good at predicting ratings from text, and in fact such models do not outperform simple Support Vector Regression baselines. This occurs because aspect ratings are *correlated*, and predicting ratings from segmented text fails to account for this correlation.

Our pairwise rating model, which explicitly models relationships between aspects, largely addresses this issue. Combining our rating model with unsegmented text already decreases the error by 23% compared to the SVM baseline, and combining our rating model with *segmented* text decreases the error by a further 22%.

While segmented text improves upon unsegmented text, the level of supervision has little impact on performance for rating prediction. This is surprising, since supervision affects *segmentation* performance significantly, and in some cases we obtain good performance on rating prediction even when aspect labels are inaccurate. To understand this, note that our unsupervised algorithm learns words that are highly correlated with users' ratings, which ultimately means that the labels it predicts must in some way be predictive of ratings, even if those labels are incorrect from the perspective of human annotators. Pleasingly, this means

that we can train a model to predict aspect ratings using an *unsupervised* segmentation model; in other words good performance on our rating prediction task can be achieved without the intervention of human annotators.

D. Qualitative Analysis

We now examine the aspect and sentiment lexicons produced by our model. Figure 6 visualizes the learned parameters θ_k and ϕ_{kv_k} for the unsupervised version of our segmentation model on *BeerAdvocate* data. We make a few observations: First, the weights match our intuition, e.g. words like 'carbonation', 'head', 'aroma', and 'flavor' match the aspects to which they are assigned. Second, the highest weighted words for 'aspect' parameters are predominantly *nouns*, while the highest weighted words for 'sentiment' parameters are predominantly *adjectives*; this confirms that aspect and sentiment words fulfil their expected roles. Third, we find that very different words are used to describe different sentiments, e.g. 'watery' has high weight for feel and taste, but not for look and smell; the study 'Old Wine or Warm Beer' [10] discusses how nouns and adjectives interact in this type of data, supporting our decision to model 'aspect' and 'sentiment' words separately, and to include separate sentiment parameters for each aspect.

To explain why 'corn' is a 2-star smell and taste word, note that corn is not normally an ingredient of beer. It is used in place of barley in inexpensive, mass-produced beers (so called 'adjuncts'), which account for many of the 1- and 2-star reviews in our corpus; thus it is not surprising that the word has negative connotations among beer enthusiasts.

VII. Conclusion

By introducing corpora of five million reviews from five sources, we have studied review systems in which users provide ratings for multiple *aspects* of each product. By learning which words describe each aspect and the associated sentiment, our model is able to determine which parts of a review correspond to each rated aspect, which sentences best summarize a review, and how to recover ratings that are missing from reviews. We learn highly interpretable aspect and sentiment lexicons, and our model readily scales to the real-world corpora we consider.

Acknowledgements. We thak *oDesk*, and especially Paul Heymann, for their assistance and support in obtaining groundtruth labels. This research has been supported in part by NSF IIS-1016909, CNS-1010921, CAREER IIS-1149837, IIS-1159679, Albert Yu & Mary Bechmann Foundation, Boeing, Allyes, Samsung, Intel, Alfred P. Sloan Fellowship and the Microsoft Faculty Fellowship.

References

[1] S. Baccianella, A. Esuli, and F. Sebastiani. Multi-facet rating of product reviews. In *ECIR*, 2009.

[2] J. Bennett and S. Lanning. The Netflix prize. In *KDD Cup and Workshop*, 2007.

[3] S. Blair-Goldensohn, T. Neylon, K. Hannan, G. Reis, R. McDonald, and J. Reynar. Building a sentiment summarizer for local service reviews. In *NLP in the Information Explosion Era*, 2008.

[4] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *JMLR*, 2003.

[5] S. Brody and N. Elhadad. An unsupervised aspect-sentiment model for online reviews. In *ACL*, 2010.

[6] T. Caetano, J. McAuley, L. Cheng, Q. Le, and A. Smola. Learning graph matching. *PAMI*, 2009.

[7] J. Cohen. A coefficient of agreement for nominal scales. *Edu. and Psych. Measurement*, 1960.

[8] L. Du, W. Buntine, and H. Jin. A segmented topic model based on the two-parameter poisson-dirichlet process. *Machine Learning*, 2010.

[9] G. Erkan and D. Radev. Lexrank: Graph-based centrality as salience in text summarization. *JAIR*, 2004.

[10] A. Fahrni and M. Klenner. Old wine or warm beer: Target-specific sentiment analysis of adjectives. In *Affective Language in Human and Machine*, 2008.

[11] M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger. Pulse: Mining customer opinions from free text. In *IDA*, 2005.

[12] G. Ganu, N. Elhadad, and A. Marian. Beyond the stars: Improving rating predictions using review text content. In *WebDB*, 2009.

[13] N. Gupta, G. Di Fabbrizio, and P. Haffner. Capturing the stars: predicting ratings for service and product reviews. In *HLT Workshops*, 2010.

[14] V. Hatzivassiloglou and K. McKeown. Predicting the semantic orientation of adjectives. In *ACL*, 1997.

[15] M. Hoffman, D. Blei, and F. Bach. Online learning for latent dirichlet allocation. In *NIPS*, 2010.

[16] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD*, 2004.

[17] Y. Jo and A. Oh. Aspect and sentiment unification model for online review analysis. In *WSDM*, 2011.

[18] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.

[19] H. Lakkaraju, C. Bhattacharyya, I. Bhattacharya, and S. Merugu. Exploiting coherence for the simultaneous discovery of latent facets and associated sentiments. In *SDM*, 2011.

[20] K. Lerman, S. Blair-Goldensohn, and R. McDonald. Sentiment summarization: evaluating and learning user preferences. In *ACL*, 2009.

[21] C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *CIKM*, 2009.

[22] B. Lu, M. Ott, C. Cardie, and B. Tsou. Multi-aspect sentiment analysis with topic models. In *Workshop on SENTIRE*, 2011.

[23] Y. Lu, C. Zhai, and N. Sundaresan. Rated aspect summarization of short comments. In *WWW*, 2009.

[24] J. McAuley, J. Leskvec, and D. Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In *ICDM*, 2012.

[25] S. Mohammad, C. Dunne, and B. Dorr. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *EMNLP*, 2009.

[26] R. Ng and A. Pauls. Multi-document summarization of evaluative text. In *ACL*, 2006.

[27] A. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *HLT*, 2005.

[28] D. Rao and D. Ravichandran. Semi-supervised polarity lexicon induction. In *ACL*, 2009.

[29] B. Snyder and R. Barzilay. Multiple aspect ranking using the good grief algorithm. In *ACL*, 2007.

[30] I. Titov and R. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *ACL*, 2008.

[31] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *WWW*, 2008.

[32] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 2005.

[33] P. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *ACL*, 2002.

[34] L. Velikovich, S. Blair-Goldensohn, K. Hannan, and R. McDonald. The viability of web-derived polarity lexicons. In *HLT*, 2010.

[35] C.-N. Yu and T. Joachims. Learning structural SVMs with latent variables. In *ICML*, 2009.

[36] W. Zhao, J. Jiang, H. Yan, and X. Li. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *EMNLP*, 2010.