The Problem

- Most online reviews don’t just offer a single opinion on a product
- Users are interested in finer-grained information about product features
- Other sentiment tasks, like automatic summarization, rely on this fine-grained information
- Aspect grouping is a subjective task
  - Grouping task benefits from seed user input
Aspect Extraction
(Mukherjee & Liu, 2012)

- Semi-unsupervised method for extracting aspects (features of the product being reviewed)
- User provides seed aspect categories
- Two subtasks:
  - Extracting aspect terms from reviews
  - Clustering synonymous aspect terms

- Parallels with:
  - Topic modeling
  - Joint sentiment and aspect models
  - DF-LDA model (Andrezejewski, 2009)
    - Must-link and cannot-link constraints

- Novel contribution: two semi-supervised ASMs that both extract aspects and performs grouping, while jointly modeling aspect and sentiment
Previous Approaches

- Latent Dirichlet Allocation (LDA)
  - Topic model that assigns Dirichlet prior to:
    - Distribution of topics in document
    - Distribution of words in topic
  - Determine topics using “higher-order co-occurrence”
    - Co-occurrence of same terms in different contexts

Motivation and Intuition

- Unsupervised methods for extracting and grouping aspects are, well, unsupervised.

By adding seeds, you can tap into human intuition and guide the creation of the statistical model.
The Two Flavors

**Flavor 1**
- Extracting aspects without grouping them
- Grouping can be done in a later step

**Flavor 2**
- Extract and group in a single step, using a sentiment switch
- Usually unsupervised
- Their approach falls into this category more-or-less
Components

\( v_{1...V} \): non-seed terms in vocabulary
\( Q_{l=1...C} \): seed sets
\( Sent_{d_s} \): sentence \( s \) of doc \( d \)
\( w_{d,s,j} \): \( j \)th term of \( Sent_{d_s} \)
\( r_{d,s,j} \): switch variable for \( w_{d,s,j} \)

Distributions

\( \Psi^A_{t=1...T} \): aspect distribution
\( \Psi^O_{t=1...T} \): sentiment distribution
\( \Omega_{t,l} \): distribution of seeds in set \( Q_l \)
\( \psi_{d,s} \): aspect and sentiment terms in \( Sent_{d_s} \)

Counts:

- \( V \) non-seed terms
- \( C \) seed sets
- \( T \) aspect models
Algorithm Overview

- For each aspect $t$, draw Dirichlet distribution over:
  - sentiment terms $\rightarrow (\Psi^O_t)$
  - Each non-seed term and seed set $\rightarrow (\Psi^A_t)$
    - Each term in seed set $\rightarrow \Omega_{t,l}$
- For each document $d$:
  - Draw various distributions over the sentiment and aspect terms
- For each word $w_{d,s,j}$:
  - Draw Bernoulli distribution for switch variable $r_{d,s,j}$

- Authors assume that a review sentence usually talks about one aspect.
  - True?
  - Is a sentence with two aspects only able to yield one?

ME-SAS variant

- Intuition: “aspect and sentiment terms play different syntactic roles in a sentence”
- Uses Max-Ent priors to model the aspect-sentiment switching (instead of switch variable $r_{d,s,j}$)
## Results

### Qualitative

<table>
<thead>
<tr>
<th>Aspect (seeds)</th>
<th>ME-SAS</th>
<th>SAS</th>
<th>ME-LDA</th>
<th>DF-LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Staff</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>attendant</td>
<td>friendly</td>
<td>staff</td>
<td>friendly</td>
</tr>
<tr>
<td></td>
<td>manager</td>
<td>polite</td>
<td>maintenance</td>
<td>nice</td>
</tr>
<tr>
<td></td>
<td>waitess</td>
<td></td>
<td>upkeep</td>
<td></td>
</tr>
<tr>
<td></td>
<td>maintenance</td>
<td></td>
<td>room</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bartender</td>
<td></td>
<td>linens</td>
<td></td>
</tr>
<tr>
<td></td>
<td>waiters</td>
<td></td>
<td>room-service</td>
<td></td>
</tr>
<tr>
<td></td>
<td>housekeeping</td>
<td></td>
<td>receptionist</td>
<td></td>
</tr>
<tr>
<td></td>
<td>receptionist</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>waitstaff</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>janitor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Service</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Hospitality</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Upkeep</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Quantitative

<table>
<thead>
<tr>
<th>Aspect</th>
<th>ME-LDA</th>
<th>DF-LDA</th>
<th>DF-LDA-Relaxed</th>
<th>SAS</th>
<th>ME-SAS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P@10</td>
<td>P@20</td>
<td>P@30</td>
<td>P@10</td>
<td>P@20</td>
</tr>
<tr>
<td><strong>Dining</strong></td>
<td>0.70</td>
<td>0.65</td>
<td>0.67</td>
<td>0.50</td>
<td>0.60</td>
</tr>
<tr>
<td><strong>Staff</strong></td>
<td>0.60</td>
<td>0.70</td>
<td>0.67</td>
<td>0.40</td>
<td>0.65</td>
</tr>
<tr>
<td><strong>Amenities</strong></td>
<td>0.80</td>
<td>0.80</td>
<td>0.67</td>
<td>0.70</td>
<td>0.65</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>0.85</td>
<td>0.77</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Critiques

Pros:
- Sentiment analysis is highly domain specific
  - Just a small amount of user-provided, domain-specific goes a long way to improve performance

Cons:
- More explanation of the intuitions behind the distributions used in the model would be helpful
If we had this model available to us to build an application, what would it look like?
Who are the users?

- From the paper:
  - “asking users to provide some seeds is easy as they are normally experts in their trades and have a good knowledge what are important in their domains”

- Is this true?

- Who are the users the authors have in mind?
This is about joint sentiment and aspect discovery, right?

- We don’t know how the sentiment side does because they don’t report evaluation.
- They actually report sentiment words in aspect categories as errors for this paper.
- The model described in this paper uses seed words to discover aspects:
  - Does this defeat the purpose?
  - Potential for bootstrapping?
Do we believe the results?

Despite these criticisms, for the most part we do believe these results.
Matching Reviews to Objects using a LM (Dalvi et al, 2009)

- Problem: determine entity (object) described by an online review using text only
- “IR in reverse:” review is query, and objects are “documents” in collection
- Advantage: expands range of search when aggregating user opinions: blogs, message boards, etc.
Problems with Traditional IR

- IR methods incompatible with problem
  - tf-idf: restaurant named “Food” will have a high idf score, causing it to be the match for
- **Long** queries, **short** documents
  - Predictable language in query, structured document
- Innovation: “mixture” language model: assumes two different types of language in review
  - Generic review language
  - Object-specific language

...the **food was** great... when we finished with our **food**....

- Soup
- The Sandwich Shop
- Food
Model Notation

General intuition behind generative model: state a model for documents, and select the document most likely to have been generated by the query.

- $r_e = r \cap \text{text}(e)$
- $P_e(w)$: probability word in review describes object
- $P(w)$: probability word is generic review language
- Parameter $\alpha$: $\alpha = P_e(w)$, $1 - \alpha = P(w)$
- $Z(r)$: normalizing function based on review length and word counts
Estimating review probability:

\[ P(r|e) = Z(r) \prod_{w \in r} ((1 - \alpha) P(w) + \alpha P_e(w)) \]

Matching object to review:

\[ e^* = \arg\max_e \sum_{w \in r_e} \log \left( 1 + \frac{\alpha P_e(w)}{1 - \alpha P(w)} \right) \]

** uniform assumption for review language allows us to ignore words outside \( r_e \)
Parameter Estimation

- Similar to a traditional LM, but requires estimation because total term frequency counts aren’t available
- $P(w)$ calculated using reviews with all object-related language removed
- $\alpha$ estimated using development set: 0.002
  - Experiments showed performance is not sensitive to this parameter

\[
P_e(w) = \frac{g(w)}{\sum_{w' \in \text{text}(e)} g(w')}
\]

\[
g(w) = \log\left(\frac{1}{\text{freq}(w)}\right)
\]
Dataset

- ~300K Yelp reviews, describing 12K restaurants
- Processing: removed reviews with no mention of the restaurant
- Expanded set of 681K restaurants from Yahoo! Local
- Final dataset: 25K reviews, describing 6K restaurants
- Evenly divided test and training sets, with 1K reserved as development data
Results

- **Baseline algorithm: TFIDF+**
  - Treats objects as queries, review as documents

  \[ e^* = \arg \max_e \sum_{w \in r_e} \log f(w) \]

  RLM: \( f(w) = 1 + \frac{\alpha P_e(w)}{1 - \alpha P(w)} \)

  TFIDF+: \( f(w) = \frac{N}{df(w)} \)

- RLM outperforms TFIDF+ particularly for longer reviews
- Longer reviews more difficult to categorize in general: more confounding proper noun mentions
Critiques

Pros:

- Good example of using relatively simple LM techniques to gain a significant advantage over tf-idf
- Methods could be expanded to other IR tasks with long queries and short “documents”
  - Ex: topic of customer emails

Cons:

- Data processing removed ~11/12 of original Yelp review set
  - Suggests only a small fraction of reviews are suitable for object classification
- Proliferation of structured review sites calls into question usefulness of method
- Questionable assumptions: uniform distribution of review language
Main RQ:
- Beyond identifying aspects, can we rank them according to importance?

Building on Previous Work:
- Frequency alone has been used as an indicator of importance
- Is frequency enough?
- Is frequency a good idea at all?

Define importance:
The aspects that most influence a consumer’s opinion about a product.
Central Idea:
“we assume that consumer’s overall opinion rating on a product is generated based on a weighted sum of his/her specific opinions on multiple aspects of the product, where the weights essentially measure the degree of importance of the aspects” (p. 1497)

Do we agree with this assumption?
Aspect Ranking: Data

- 11 products in 4 domains:
  - All electronics products

- 2 types of reviews crawled from 4 web sites:
  - Pros + Cons
  - Free text

- Manually annotated by several people for aspect importance and sentiment (importance = average of gold standard)
Overview

1. Extract aspects via dependency parsing
   - Take frequent NPs from Pros/Cons, use them to train an SVM for the free text.
   - Expand via synonymy *(thesaurus.com)*
   - Problems?

2. Classify the sentiment of these aspects
   - Train SVM (again) on Pros/Cons, classify sentiment expressions in free text closest to aspects.
   - Problems?
   - This seemed almost unrelated to the core goals of the paper
3. Determine aspects importance

- Assume the opinion of a review can be represented as a vector of aspects with a corresponding vector of weights (importance).

- Their model’s job is to create that weight vector.

- Opinion is seen as being drawn from a Normal Distribution (why?) and use MLE given corpus data to optimize the weights.
### Aspect Identification

<table>
<thead>
<tr>
<th>Data set</th>
<th>Hu’s Method</th>
<th>Wu’s Method</th>
<th>Our Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canon EOS</td>
<td>0.681</td>
<td>0.686</td>
<td>0.728</td>
</tr>
<tr>
<td>Fujifilm</td>
<td>0.685</td>
<td>0.666</td>
<td>0.710</td>
</tr>
<tr>
<td>Panasonic</td>
<td>0.636</td>
<td>0.661</td>
<td>0.706</td>
</tr>
<tr>
<td>MacBook</td>
<td>0.680</td>
<td>0.733</td>
<td>0.747</td>
</tr>
<tr>
<td>Samsung</td>
<td>0.594</td>
<td>0.631</td>
<td>0.712</td>
</tr>
<tr>
<td>iPod Touch</td>
<td>0.650</td>
<td>0.660</td>
<td>0.718</td>
</tr>
<tr>
<td>Sony NWZ</td>
<td>0.631</td>
<td>0.692</td>
<td>0.760</td>
</tr>
<tr>
<td>BlackBerry</td>
<td>0.721</td>
<td>0.730</td>
<td>0.734</td>
</tr>
<tr>
<td>iPhone 3GS</td>
<td>0.697</td>
<td>0.736</td>
<td>0.740</td>
</tr>
<tr>
<td>Nokia 5800</td>
<td>0.715</td>
<td>0.745</td>
<td>0.747</td>
</tr>
<tr>
<td>Nokia N95</td>
<td>0.700</td>
<td>0.737</td>
<td>0.741</td>
</tr>
</tbody>
</table>
Aspect Ranking: Results and Evaluation

## Aspect Ranking

<table>
<thead>
<tr>
<th>#</th>
<th>Frequency</th>
<th>Correlated</th>
<th>Hybrid</th>
<th>Our Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Phone</td>
<td>Phone</td>
<td>Phone</td>
<td>Usability</td>
</tr>
<tr>
<td>2</td>
<td>Usability</td>
<td>Usability</td>
<td>Usability</td>
<td>Apps</td>
</tr>
<tr>
<td>3</td>
<td>3G</td>
<td>Apps</td>
<td>Apps</td>
<td>3G</td>
</tr>
<tr>
<td>4</td>
<td>Apps</td>
<td>3G</td>
<td>3G</td>
<td>Battery</td>
</tr>
<tr>
<td>5</td>
<td>Camera</td>
<td>Camera</td>
<td>Camera</td>
<td>Looking</td>
</tr>
<tr>
<td>6</td>
<td>Feature</td>
<td>Looking</td>
<td>Looking</td>
<td>Storage</td>
</tr>
<tr>
<td>7</td>
<td>Looking</td>
<td>Feature</td>
<td>Feature</td>
<td>Price</td>
</tr>
<tr>
<td>8</td>
<td>Battery</td>
<td>Screen</td>
<td>Battery</td>
<td>Software</td>
</tr>
<tr>
<td>9</td>
<td>Screen</td>
<td>Battery</td>
<td>Screen</td>
<td>Camera</td>
</tr>
</tbody>
</table>

Looks pretty good, though the order does not match the gold standard
Aspect Ranking: Results and Evaluation

Aspect Ranking
Metric: Normalized Discounted Cumulative Gain
(More points given to important aspects at the top of the list)

<table>
<thead>
<tr>
<th>Data set</th>
<th>Frequency @5</th>
<th>Frequency @10</th>
<th>Frequency @15</th>
<th>Correlation @5</th>
<th>Correlation @10</th>
<th>Correlation @15</th>
<th>Hybrid @5</th>
<th>Hybrid @10</th>
<th>Hybrid @15</th>
<th>Our Method @5</th>
<th>Our Method @10</th>
<th>Our Method @15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canon EOS</td>
<td>0.735</td>
<td>0.771</td>
<td>0.740</td>
<td>0.735</td>
<td>0.762</td>
<td>0.779</td>
<td>0.735</td>
<td>0.798</td>
<td>0.742</td>
<td>0.862</td>
<td>0.824</td>
<td>0.794</td>
</tr>
<tr>
<td>Fujifilm</td>
<td>0.816</td>
<td>0.705</td>
<td>0.693</td>
<td>0.760</td>
<td>0.756</td>
<td>0.680</td>
<td>0.816</td>
<td>0.759</td>
<td>0.682</td>
<td>0.863</td>
<td>0.801</td>
<td>0.760</td>
</tr>
<tr>
<td>Panasonic</td>
<td>0.744</td>
<td>0.807</td>
<td>0.783</td>
<td>0.763</td>
<td>0.815</td>
<td>0.792</td>
<td>0.744</td>
<td>0.804</td>
<td>0.786</td>
<td>0.796</td>
<td>0.834</td>
<td>0.815</td>
</tr>
</tbody>
</table>
Aspect Ranking: Final thoughts

- Despite criticisms, this seems to work.
- They made some assumptions that I don’t fully agree with.
- They actually state that frequency is not a good metric, then go ahead and use it in both the identification and ranking.
- But ultimately, their results look viable to me.
Thank you!