Aspects and Objects in Sentiment Analysis

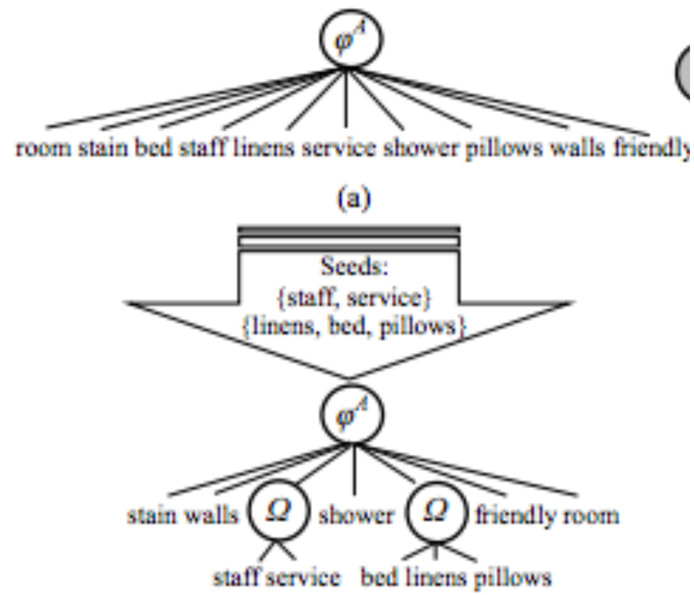
Jared Kramer and Clara Gordon

April 29, 2014

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

… I liked the food, but the service was terrible….



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

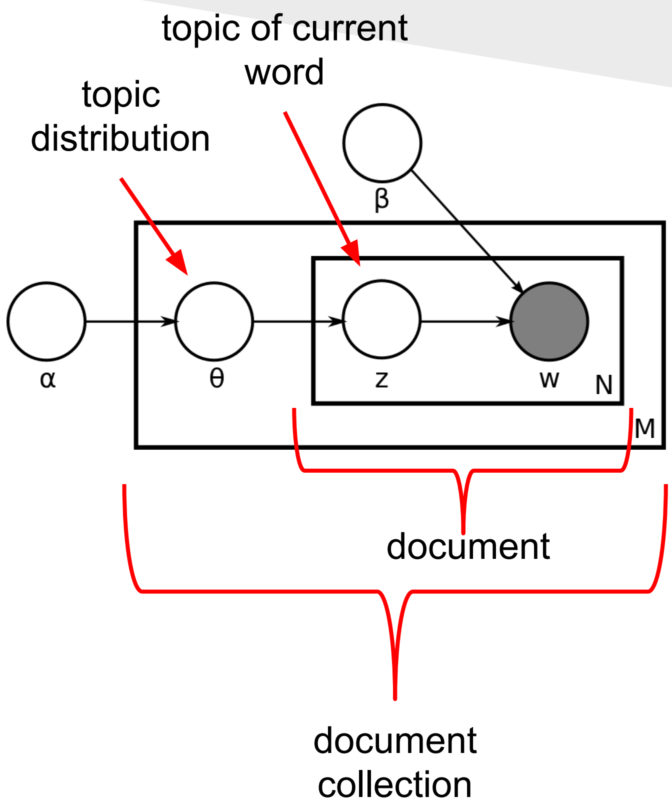


Image credit: http://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

Motivation and Intuition

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

DF-LDA
Topic
staff
friendly
helpful
beds
front
room
comfortable
large
receptionist
housekeeping

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

Seeded Aspect and Sentiment (SAS) Model: Notation

Components

$v_{1...V}$: non-seed terms in vocabulary

$Q_{l=1...C}$: seed sets

$Sent^{d}_{s}$: sentence s of doc d

$w_{d,s,j}$: jth 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 → (Ψ^{O}_{t})
  - Each non-seed term and seed set → (Ψ^{A}_{t})
    - Each term in seed set → $\Omega_{t,l}$
- 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

Aspect (seeds)	ME-SAS		SAS		ME-LDA		DF-LDA
	Aspect	Sentiment	Aspect	Sentiment	Aspect	Sentiment	Topic
Staff (staff service waiter hospitality upkeep)	attendant	friendly	attendant	friendly	staff	friendly	staff
	manager	attentive	waiter	nice	maintenance	nice	***friendly***
	waitress	polite	waitress	dirty	**room**	courteous	***helpful***
	maintenance	nice	manager	**comfortable**	upkeep	**extremely**	**beds**
	bartender	**clean**	maintenance	nice	**linens**	nice	**front**
	waiters	pleasant	**helpful**	**clean**	room-service	**clean**	room
	housekeeping	**slow**	waiters	polite	receptionist	polite	**comfortable**
	receptionist	courteous	housekeeping	**extremely**	**wait**	**little**	**large**
	waitstaff	rude	receptionist	courteous	**pillow**	helpful	receptionist
	janitor	professional	**polite**	efficient	waiters	**better**	housekeeping

Quantitative

Aspect	ME-LDA			DF-LDA			DF-LDA-Relaxed			SAS			ME-SAS		
	P@10	P@20	P@30	P@10	P@20	P@30	P@10	P@20	P@30	P@10	P@20	P@30	P@10	P@20	P@30
Dining	0.70	0.65	0.67	0.50	0.60	0.63	0.70	0.70	0.70	0.80	0.75	0.73	0.90	0.85	0.80
Staff	0.60	0.70	0.67	0.40	0.65	0.60	0.60	0.75	0.67	0.80	0.80	0.70	1.00	0.90	0.77
Amenities	0.80	0.80	0.67	0.70	0.65	0.53	0.90	0.75	0.73	0.90	0.80	0.70	1.00	0.85	0.73

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

Brainstorming Session

- 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.

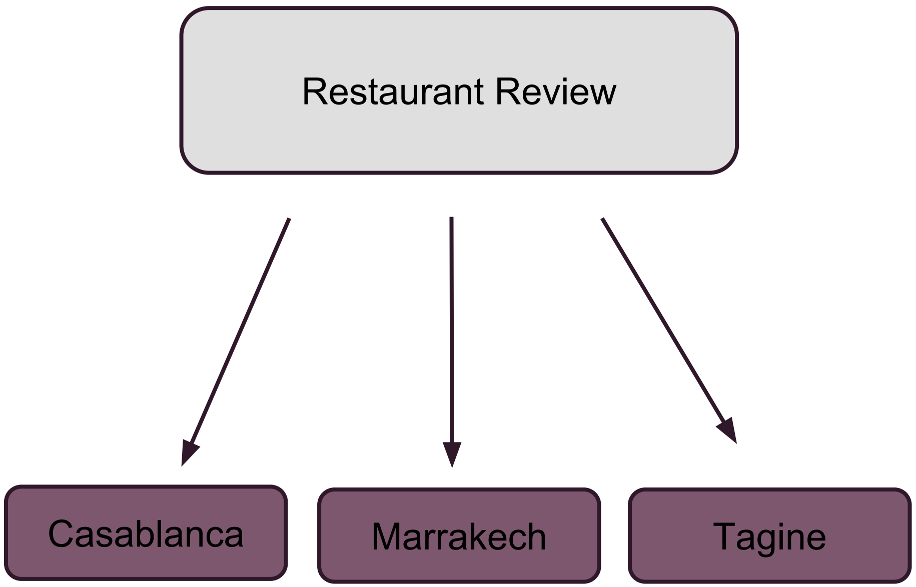
- The model described in this paper uses seed words to discover aspects:
  - Does this defeat the purpose?
  - Potential for bootstrapping?

Do we believe the results?

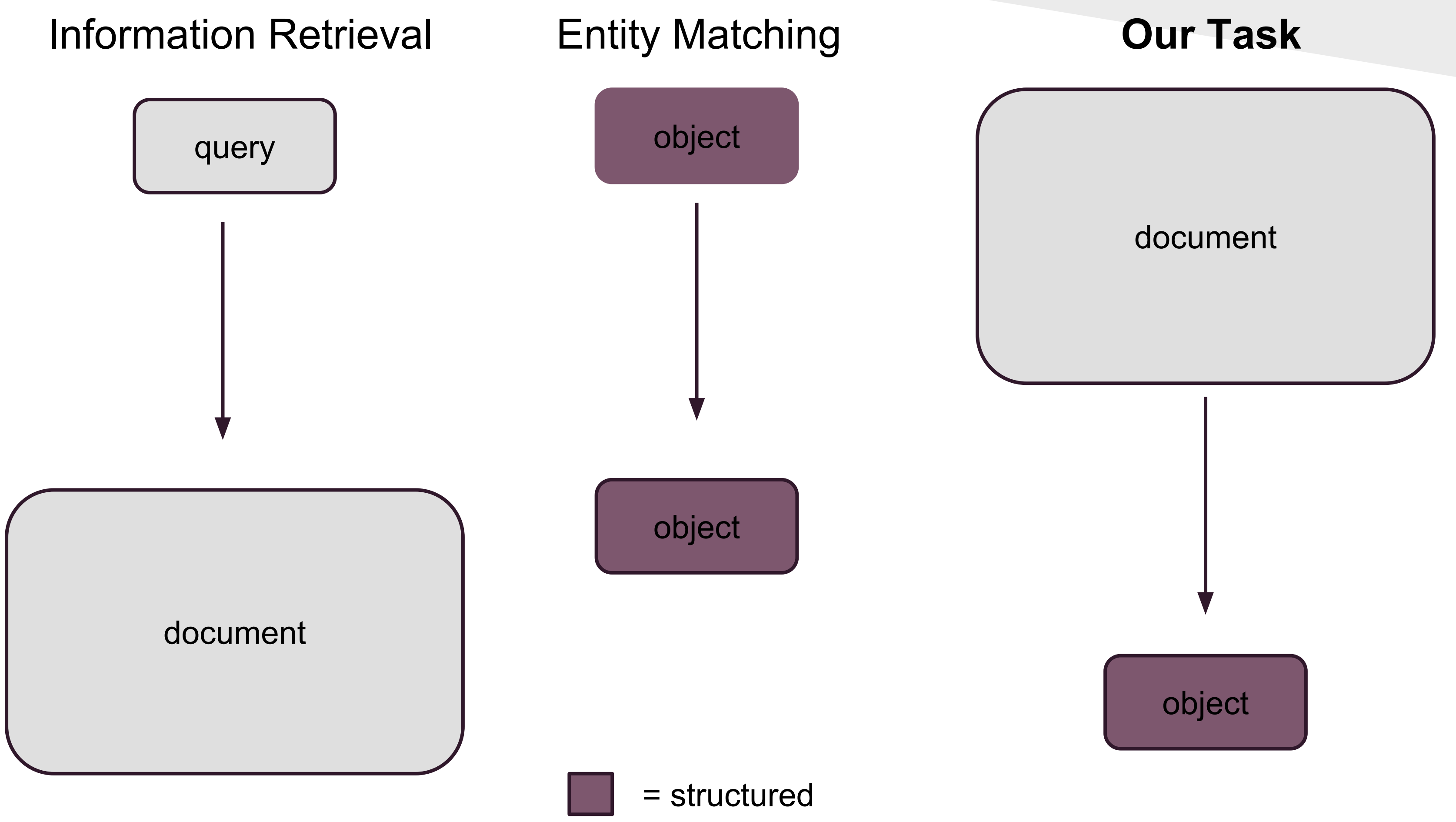
Despite these criticisms, for the most part we do believe these results.

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.

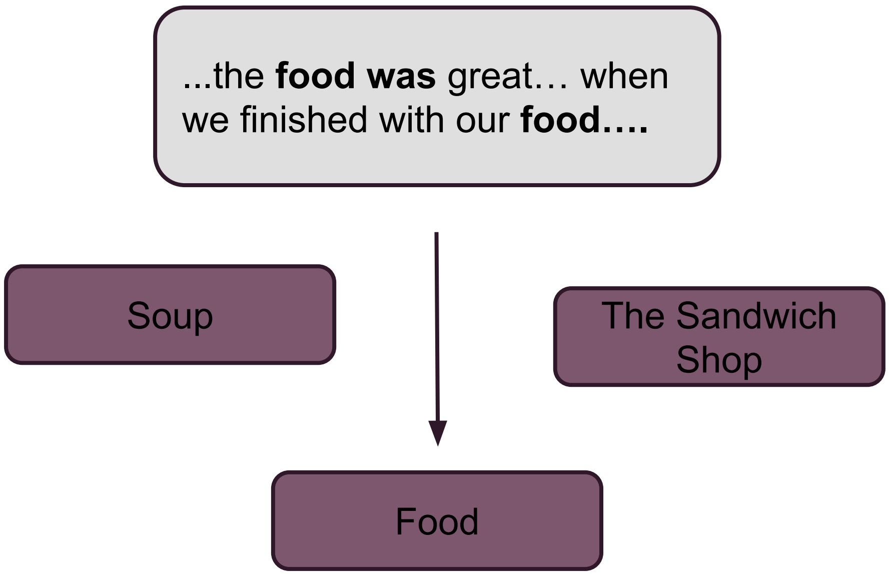


Context


Information Retrieval
query
document
Entity Matching
object
object
Our Task
document
object
= structured

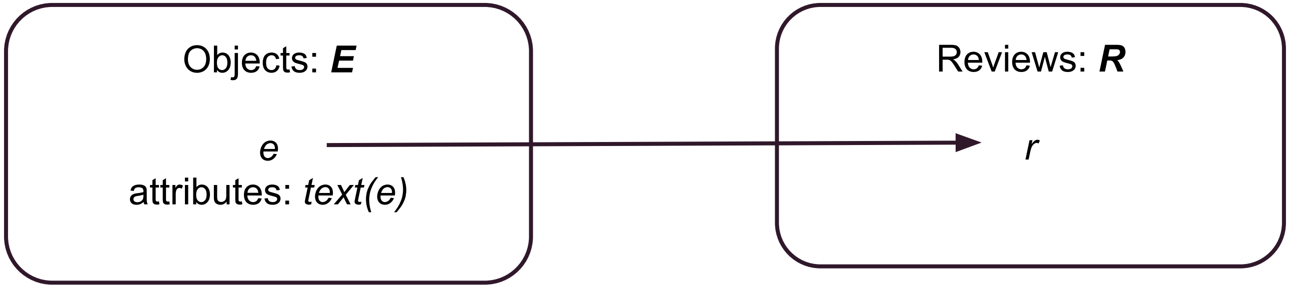
Problems with Traditional IR

- 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



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 α: $\alpha = P_e(w)$, $1 - \alpha = P(w)$
- $Z(r)$: normalizing function based on review length and word counts

Model Definition

Estimating review probability:

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

$$\prod_{w \in r_e} ((1-\alpha)P(w) + \alpha P_e(w)) \longrightarrow \prod_{w \in r_e} \left(1 + \frac{\alpha}{1-\alpha} \frac{P_e(w)}{P(w)}\right)$$

Matching object to review:

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

** 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
- α estimated using development set: 0.002
  - Experiments showed performance is not sensitive to this parameter

g(w) = log(1/ freq(w))

$$P_e(w) = \frac{\overbrace{g(w)}}{\sum_{w' \in \text{text}(e)} g(w')}$$

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

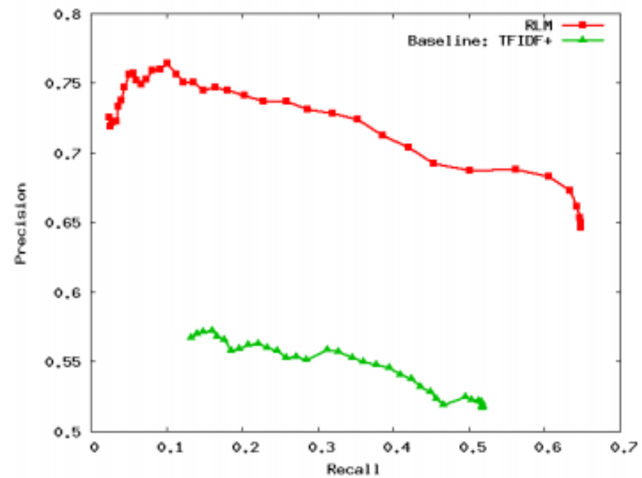
- Baseline algorithm: TFIDF+
    - Treats objects as queries, review as documents

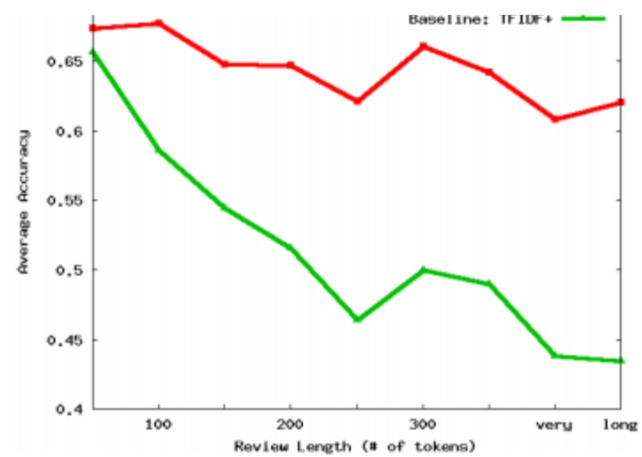
$$e^* = \arg\max_e \sum_{w \in r_e} \log f(w)$$

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

TFIDF+: f(w) = 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

Aspect Ranking: Identifying Important Product Aspects from Online Consumer Reviews
Yu, Zha, Wang, Chua, 2011

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.

Aspect Ranking: Assumptions

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)

Aspect Ranking: Methodology

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

Ranking Aspects: Methodology

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 Ranking: Results and Evaluation

Aspect Identification

Data set	*Hu's Method*	*Wu's Method*	*Our Method*
Canon EOS	0.681	0.686	**0.728**
Fujifilm	0.685	0.666	**0.710**
Panasonic	0.636	0.661	**0.706**
MacBook	0.680	0.733	**0.747**
Samsung	0.594	0.631	**0.712**
iPod Touch	0.650	0.660	**0.718**
Sony NWZ	0.631	0.692	**0.760**
BlackBerry	0.721	0.730	**0.734**
iPhone 3GS	0.697	0.736	**0.740**
Nokia 5800	0.715	0.745	**0.747**
Nokia N95	0.700	0.737	**0.741**

Aspect Ranking: Results and Evaluation

Aspect Ranking

#	*Frequency*	*Correlated*	*Hybrid*	*Our Metho*
1	Phone	Phone	Phone	Usability
2	Usability	Usability	Usability	Apps
3	3G	Apps	Apps	3G
4	Apps	3G	3G	Battery
5	Camera	Camera	Camera	Looking
6	Feature	Looking	Looking	Storage
7	Looking	Feature	Feature	Price
8	Battery	Screen	Battery	Software
9	Screen	Battery	Screen	Camera

Looks pretty good, though the order does not match the gold standard

Aspect Ranking: Results and Evaluation

Aspect Ranking

Metric: Normalized Discounted Cummulative Gain

(More points given to important aspects at the top of the list)

	Frequency			*Correlation*			*Hybrid*			*Our Method*		
Data set	*@5*	*@10*	*@15*	*@5*	*@10*	*@15*	*@5*	*@10*	*@15*	*@5*	*@10*	*@15*
Canon EOS	0.735	0.771	0.740	0.735	0.762	0.779	0.735	0.798	0.742	**0.862**	**0.824**	**0.794**
Fujifilm	0.816	0.705	0.693	0.760	0.756	0.680	0.816	0.759	0.682	**0.863**	**0.801**	**0.760**
Panasonic	0.744	0.807	0.783	0.763	0.815	0.792	0.744	0.804	0.786	**0.796**	**0.834**	**0.815**

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!