UNIVERSITY OF ALBERTA

Personalized and Explainable Aspect-based Recommendation using Latent Opinion Groups

Introduction

- Recommendation of products and services:
    - A key feature of e-commerce websites
- Traditional Recommender systems:
    - Recommending new items to the users which they potentially like
    - Item can be a product or service
    - No indication why an item is recommended to them.

Explainable recommender system

- Describing why the recommender offers a specific item
- Using different types of reasons:
  - Similar-user based explanation
  - Similar-item based explanation
  - Feature based explanation
  - Aspect-based explanation

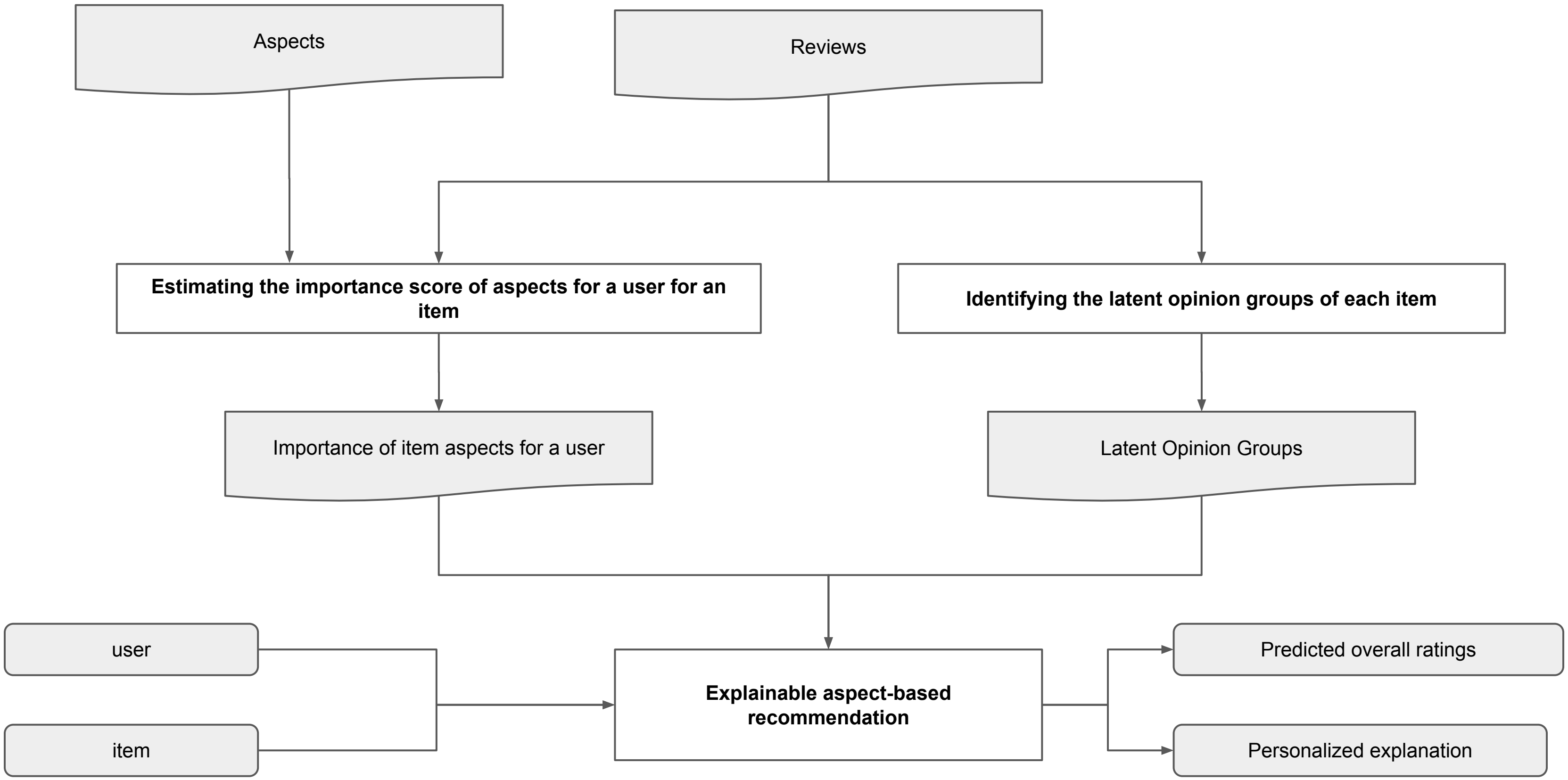
Explainable aspect-based recommendation

- Subjective attributes of an item, e.g, “affordability”
  - Consider a high-quality camera
    - One user may find it “affordable”
    - The other one may find it “too expensive”
- Using the user’s preferences
  - To explain why an item is recommended based on the item’s aspects

The shortcomings of previous methods

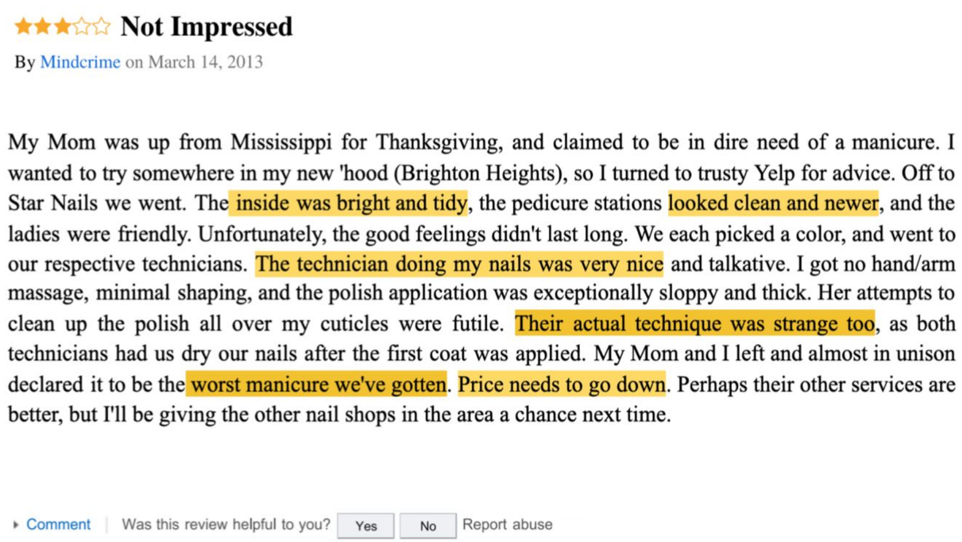
- Describing item aspects using fixed sets of words
- Not providing any explanation for how the aspects led to recommendation
- Assuming all users paying similar attention to all aspects
- Ignoring the effect of combination of aspects on user’s experience

Our method: Personalized Recommendations based on Latent Opinion Groups (PRLOG)



How to model aspects and sentiments expressed in the reviews

(a) A sample Tripadvisor review



(b) A sample Yelp review

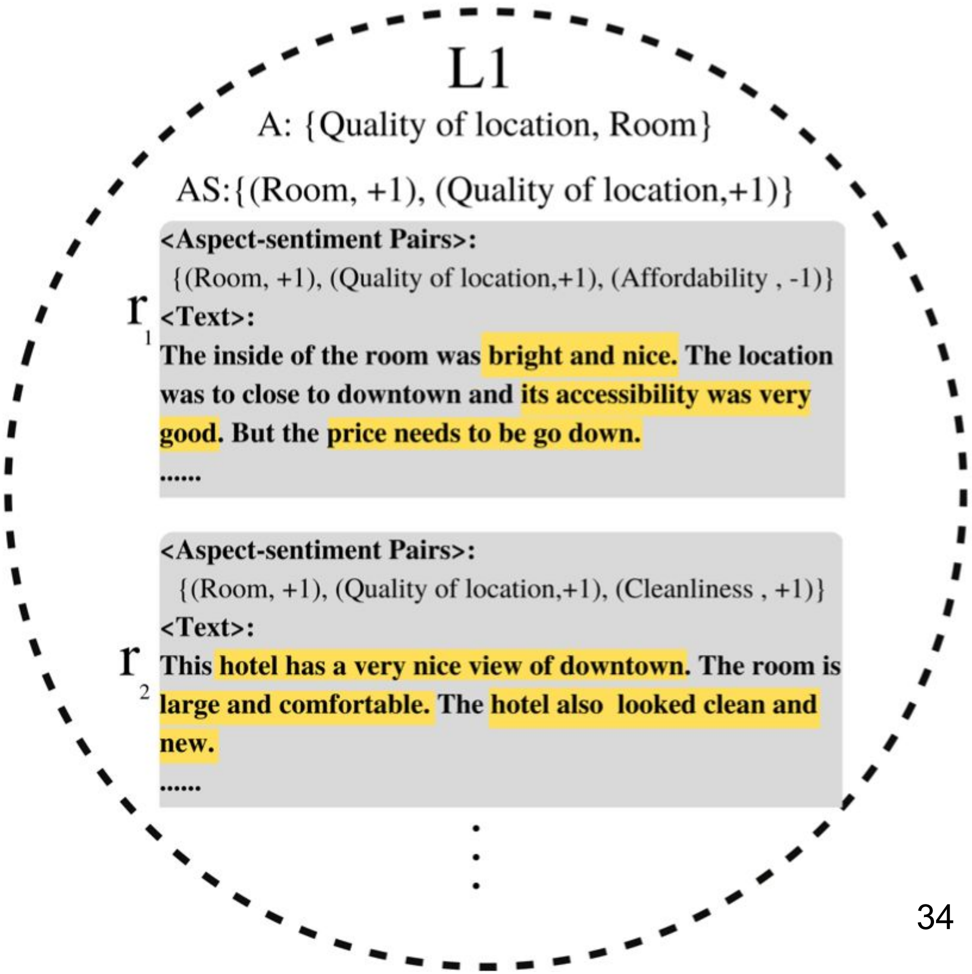
Capturing the dependency between sentiments and aspects

- Similarly minded reviewers sharing similar views about the particular item
  - Latent Opinion Groups (LOG)
  - Clustering reviews of an item into clusters that express the same combination of sentiments towards a set of aspects

What does a LOG look like?

A LOG for an item i, L is defined as a tuple <R, A, AS>:

- $R = \{r_1, r_2\}$



How do we discover LOGs for an item?

- Using frequent itemset mining method Apriori:
  - Identifying all itemsets occurring in a database of transactions with frequency above a minimum support threshold
  - Frequent itemset is a set of aspect sentiment pairs

Importance of item aspects for a user

- Aggregate aspect importance score for a user and an item:
  - Text-based aspect importance score
  - Ratings-based aspect importance score

Text-based aspect importance score

- How likely a user *u* is going to talk about aspect *a* when reviewing item *i*, $P(a|u,i)$
    - Users and items are assumed independent from each other:
    - Aspects are represented by sets of words
    - $P(u|w_j)$ and $P(i|w_j)$ should be estimated

How to handle word-mismatching

- Different users may choose different words to describe the same item aspects
- Using the Semantic Entity Retrieval Toolkit (SERT) to estimate $P(u|w_j)$ and $P(i|w_j)$:
    - Learning word-embedded vector representation for entities and words

Ratings-based aspect importance score

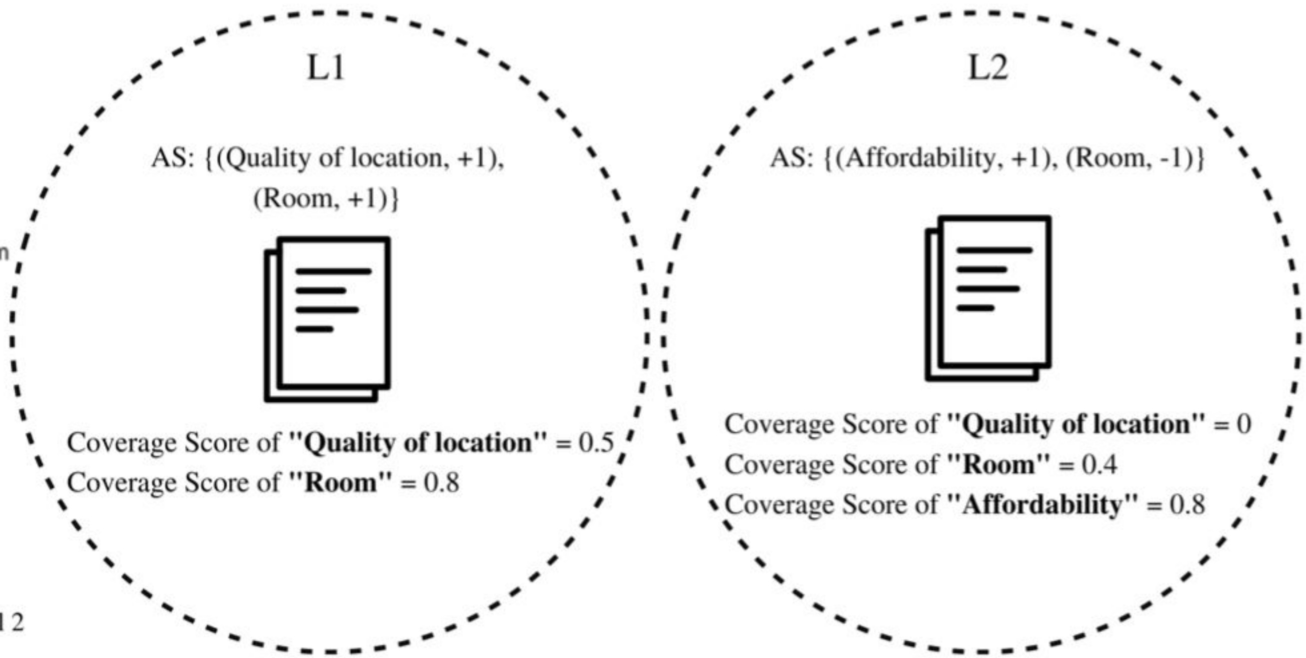
- Using larger rating scale for more important aspects
- Measuring the importance of an aspect for a user:
    - Standard deviation
- Assuming aspect are equally important for each item

Predicting sentiments towards aspects

- A weighted average of sentiments expressed towards each aspect in all LOGs of that item:
  - The contribution score of each LOG is based on two factors:
    - Which aspects are mentioned in a LOG for this item and how often
    - How important those aspects are to the user

Different contribution scores of LOGs

Predicted aspect importance score for user 1 for Hotel 2



The overall rating prediction

- Linear combination of predicted sentiments of user u towards aspects of item i, weighted by the importance of aspects

User 1

Hotel 1

Predicted aspect importance score for user 1 for hotel1

We recommend Hotel 1 to you because you can expect the "Room" to be good and the "Quality of location" to be excellent in this hotel based on the experience of other users who visited this hotel and they talked the aspects that are important to you in their reviews .

Evaluation

- Partition the reviews collection into training and test sets
  - Split is done on a per-user basis
  - In ratios of 80% and 20%
- Datasets
  - Yelp including reviews from Hotel, Beauty Spa and Restaurants
    - Opinion parser for aspect extraction
    - 19,10 and 16 aspects, respectively
  - Tripadvisor dataset
    - 7 predefined aspects
- Baseline
  - SULM by Bamen et al.

Parameter settings

- Minimum support used by Apriori algorithm
  - 0.03 for all datasets
- Parameter of convex combination of text-based and ratings-based aspect importance score:
  - 0.5 for Tripadvisor dataset
    - Assigning the equal weight to both source of information
  - 1 for Yelp dataset
    - No predefined aspects

Aspect Ranking Performance

Dataset	Method	Precision@3	Precision@5
BeautySpa	PRLOG	**0.36**	**0.30**
	SULM	0.22	0.19
Hotel	PRLOG	**0.42**	**0.38**
	SULM	0.40	0.32
Restaurant	PRLOG	**0.26**	**0.42**
	SULM	0.19	0.16
Tripadvisor	PRLOG	**0.30**	**0.42**
	SULM	0.23	0.35

Performance of prediction of sentiments towards aspects

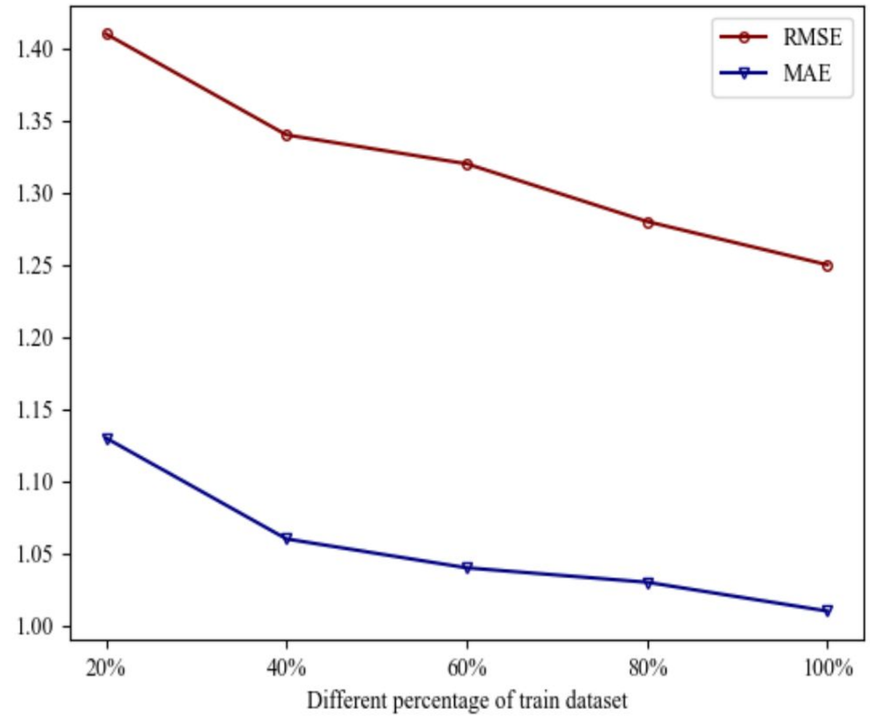
Dataset	Method	Precision
BeautySpa	PRLOG	**0.93**
	SULM	0.62
Hotel	PRLOG	**0.96**
	SULM	0.52
Restaurant	PRLOG	**0.96**
	SULM	0.58
Tripadvisor	PRLOG	**0.52**
	SULM	0.15

Overall rating prediction performance

Dataset	Method	RMSE	MAE	Accuracy
BeautySpa	PRLOG	**1.53**	**1.26**	**0.66**
	SULM	1.66	1.44	0.54
Hotel	PRLOG	**1.17**	**0.91**	**0.59**
	SULM	1.31	1.07	0.52
Restaurant	PRLOG	**1.16**	**0.94**	**0.61**
	SULM	1.27	1.04	0.54
Tripadvisor	PRLOG	**1.35**	**1.16**	**0.75**
	SULM	1.47	1.31	0.53

The effect of different percentage of previous reviews for a target user

- Restaurant Dataset
  - A group of 50 reviewers
    - Author of at least 30 reviews
  - Only including a certain percentage of previous reviews



Contributions

1. Handling natural variation of wording in the user's reviews in the aspect importance score estimation
2. Providing a better explanation
   a. Predicting how the user will like or dislike different item aspects
3. Improving the recommendation performance by
   a. Considering the dependency between sentiments towards aspects
   b. Considering the fact that different users place different value on different item aspects

Future Work

- Multi-aspect paper review assignment Problem:
  - Investigation of additional similarity functions
  - Systematic configuration of the parameters of our method
  - Validating the usefulness of our method in a real world application
- Personalized and explainable aspect-based recommendation
  - Modeling latent opinion groups more sophisticatedly
  - Optimizing the parameters of our method systematically
  - Using more-complex machine learning models to predict sentiments towards aspects

Appendices

Dynamic Setting of Parameter λ

- Setting lambda based on the amount of text available in the reviews of the user.
  - If text-based aspect importance score is close to 1, the amount of text is good and lambda should be 0.5.
  - If text-based aspect importance score is close to 0, it seems the amount of text is not reliable then lambda should be 1.

$$\lambda = \frac{1}{1 + e^{\beta_{\mathrm{Text}}(u,a,i)} - 1/\beta_{\mathrm{Text}}(a,u,i)}$$

Measures	Dynamic lambda	Lambda = 0.5
RMSE	1.31	1.35
MAE	1.11	1.16

Dynamic Setting of Parameter λ

Measures	Dynamic lambda	Lambda = 0.5
RMSE	**1.31**	1.35
MAE	**1.11**	1.16

Dynamic Setting of Minimum Support

- Setting the minimum support dynamically:
  - At least two reviews in each LOG of an item

Dataset	Method	RMSE	MAE	Accuracy
BeautySpa	PRLOG	**1.56**	**1.22**	**0.62**
	SULM	1.66	1.44	0.54
Hotel	PRLOG	1.57	1.24	**0.58**
	SULM	**1.31**	**1.07**	0.52
Restaurant	PRLOG	**1.15**	**0.88**	**0.63**
	SULM	1.27	1.04	0.54
Tripadvisor	PRLOG	**1.31**	**1.11**	**0.75**
	SULM	1.47	1.31	0.53

Adjusting the language models using LRAs

- Assigning different weights to each term using LRAs.
  - Adopting cluster-based information retrieval model

$$P(t|\theta_r) = \lambda \boxed{P(t|r)} + (1-\lambda)[\beta \boxed{P(t|\boxed{C_r})} + (1-\beta)\boxed{P(t|C)}]$$

Relative frequency of *t* in *r*'s profile

Smoothing with latent research areas language model

The most similar latent research area of *r* to the query paper

Smoothing with Collection language model

$$P(t|\theta_p) = \alpha \boxed{P(t|p)} + (1-\alpha)p(t|\boxed{C_p})$$

Relative frequency of *t* in the paper *p*

The most similar latent research area to the query paper

The MARTA-LRA objective function

- We use a greedy forward selection algorithm in order to maximize the following objective function:

$$S(\boxed{g}, p) = v \sum_{r \in g} \boxed{R(r,p)} - (1-v)\left[\frac{1}{|g|} \sum_{r \in g} \sum_{\substack{r' \in g \\ r \neq r'}} \boxed{RD(r,r')}\right]$$

A group of reviewers assigned to the paper *p*

The relevance score of reviewer *r* to the paper *p*

Similarity between two reviewers r and r'

- Maximizing coverage, confidence and load balancing simultaneously
  - Examining the paper's quote and reviewers quote in each step

Parameter Settings

- The summarization of different parameters of our method

Parameters	Definition	Equation	Value range
λ	Reviewer's Profile Smoothing Parameter	(2.8)	0, 0.1,. . . ,. . . ,**0.9**, 1
β	Reviewer's Latent Research Area Balance Factor	(2.9)	0, 0.1,. . . ,**0.5**,. . . , 1
α	Query's Latent Research Area Balance Factor	(2.10)	0, 0.1,. . . ,**0.6**,. . . , 1
v	Diversity factor	(2.11)	0, 0.1,. . . ,**0.6**,. . . , 1
p	Number of LSA components		15, 20, **25**, 30
k	Number of clusters		10, 15, **25**, 30, 35, 40

Text-based aspect importance score

- How likely a user *u* is going to talk about aspect *a* when reviewing item *i*, $P(a|u,i)$

$$\beta_{\text{Text}}(a,u,i) = P(a|u,i) = \frac{P(u,i|a)P(a)}{P(u,i)}$$

- Users and items are assumed independent from each other:

$$P(a|u,i) = \frac{P(u|a)P(i|a)P(a)}{P(u)P(i)}$$

- Aspects are represented by sets of words
  - $P(u|a) = P(u|w_1,\ldots,w_k)$
  - $P(i|a) = P(i|w_1,\ldots,w_k)$

Ratings-based aspect importance score

- Users are more discriminating in their set of reviews when rating item aspects that they care about
- To measure the importance of aspect to a user:
  - Collecting all the ratings of aspect *a* from all the reviews by user *u*
  - Compute its standard deviation
- All explicitly defined aspects of an item are equally relevant for each item

$$\beta_{\text{Rating}}(a, u, i) = \underbrace{\boxed{\eta_{rating}(a, u)}}_{\text{stddev(ratings(a,u))}} \cdot \boxed{\eta_{rating}(a, i)}.$$

$1/|A|$, $|A|$ is the number of all aspects in the entire set of reviews

The Estimation of Contribution Score of LOGs

- How often aspect *a* is mentioned in the reviews of LOG *L*:
  - Coverage Score of an aspect *a* by LOG *L*, *C(a, L)*
  - Aggregating the relative frequency with which *a* is mentioned in each review *r* of *L*

$$C(a,L) = \begin{cases} \sum_{r \in R_L} \frac{f(a,r)}{\sum_{a \in A_r} f(a,r)} & \text{if } a \in A_L\text{ ;} \\ 0 & \text{otherwise,} \end{cases}$$

- The aggregated importance score of aspect *a* for user *u* and item *i*

$$W(L,u,i) = \sum_{a \in \mathcal{A}} C(a,L)\beta(a,u,i)$$

Predicting Sentiments towards Aspects

- A weighted average of sentiments expressed towards each aspect in all LOGs of that item:
  - The weight of each LOG is called the contribution score.
  - The contribution score of each LOG is computed based on two factors:
    - Which aspects are mentioned in a LOG for this item and how often
    - How important those aspects are to the user

The contribution score of LOG, *L* of an item *i* given a user *u*

$$S(a,u,i) = \frac{\sum_{L \in \mathcal{L}_i} \boxed{W(L,u,i)} \cdot S(a,L)}{\sum_{L \in \mathcal{L}_i} W(L,u,i)}$$