Use of Social Computing on Ecommerce Platforms

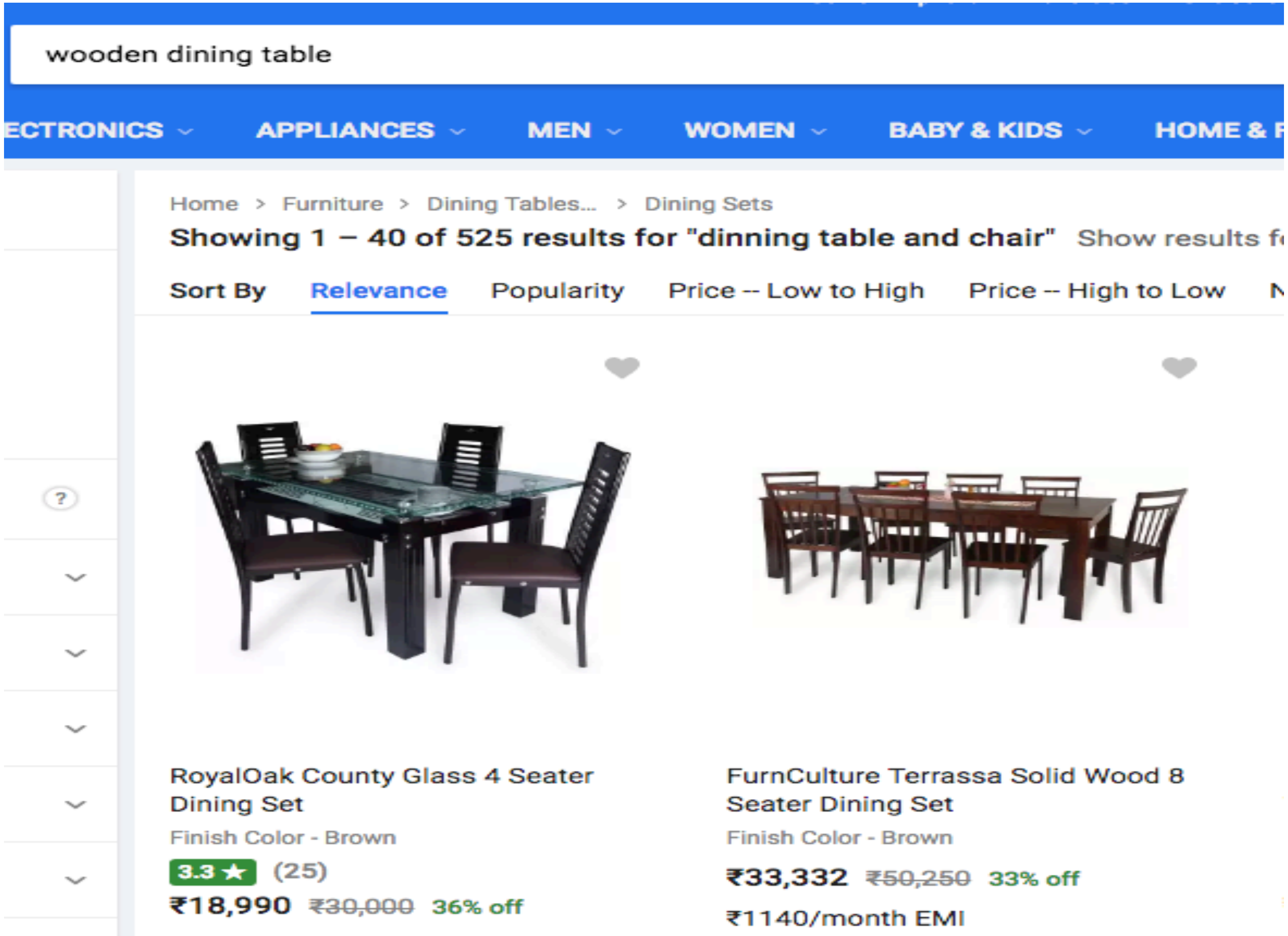
Saptarshi Ghosh
Department of CSE, IIT Kharagpur
Social Computing course, CS60017

Application: Query augmentation

- For a user-specified query
  - Suggest modified / augmented queries
  - Auto-complete queries

- Goal: increased user satisfaction
  - Help user to formulate good queries
  - Suggested queries will show better results to the user

Query augmentation in action



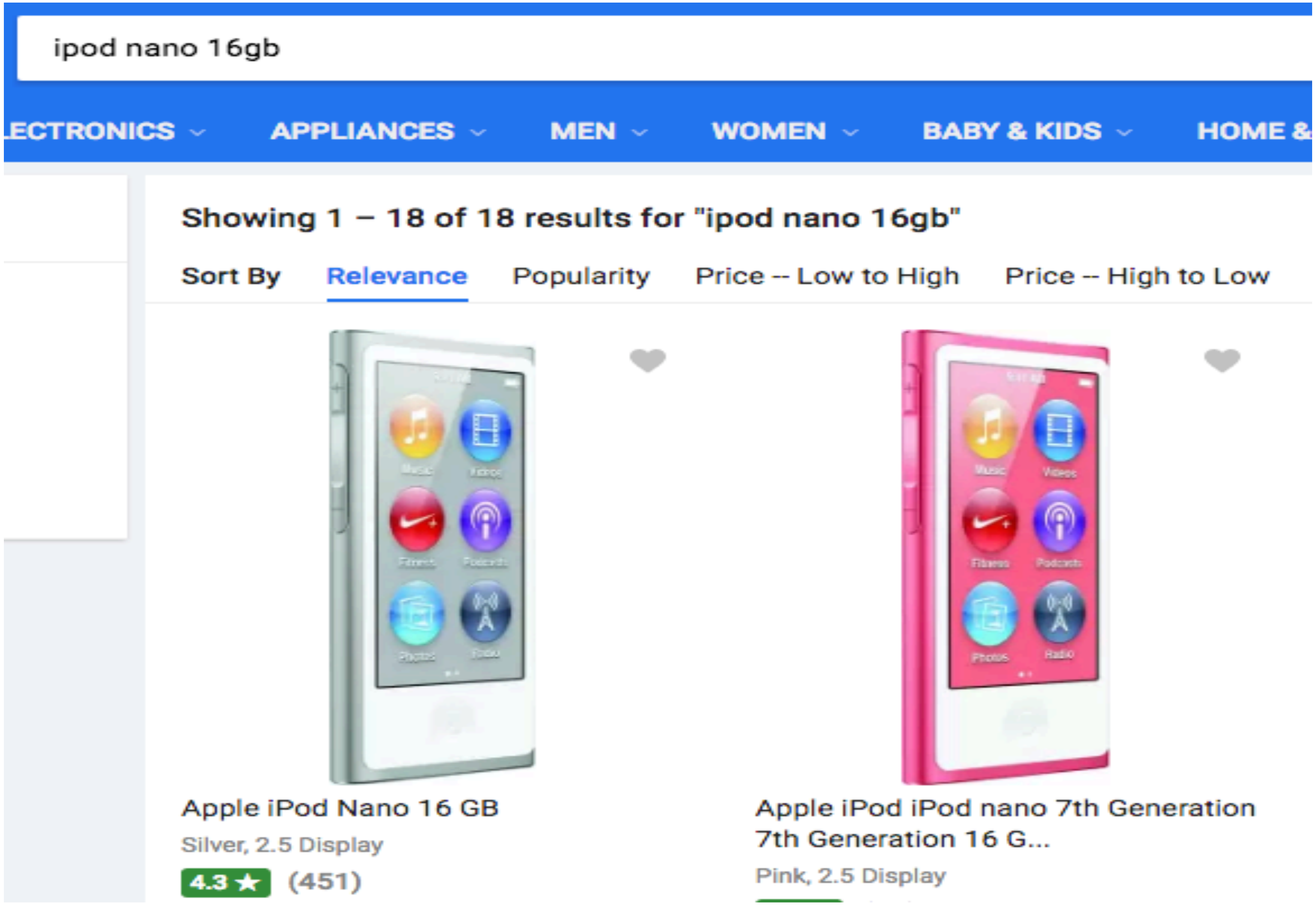
Query augmentation

- Examples taken from: Query Suggestion for E-Commerce Sites, Hasan et al., WSDM 2011
  - Published by Ebay
  - Observations are based on Ebay data, but should be generalizable to other Ecommerce sites as well

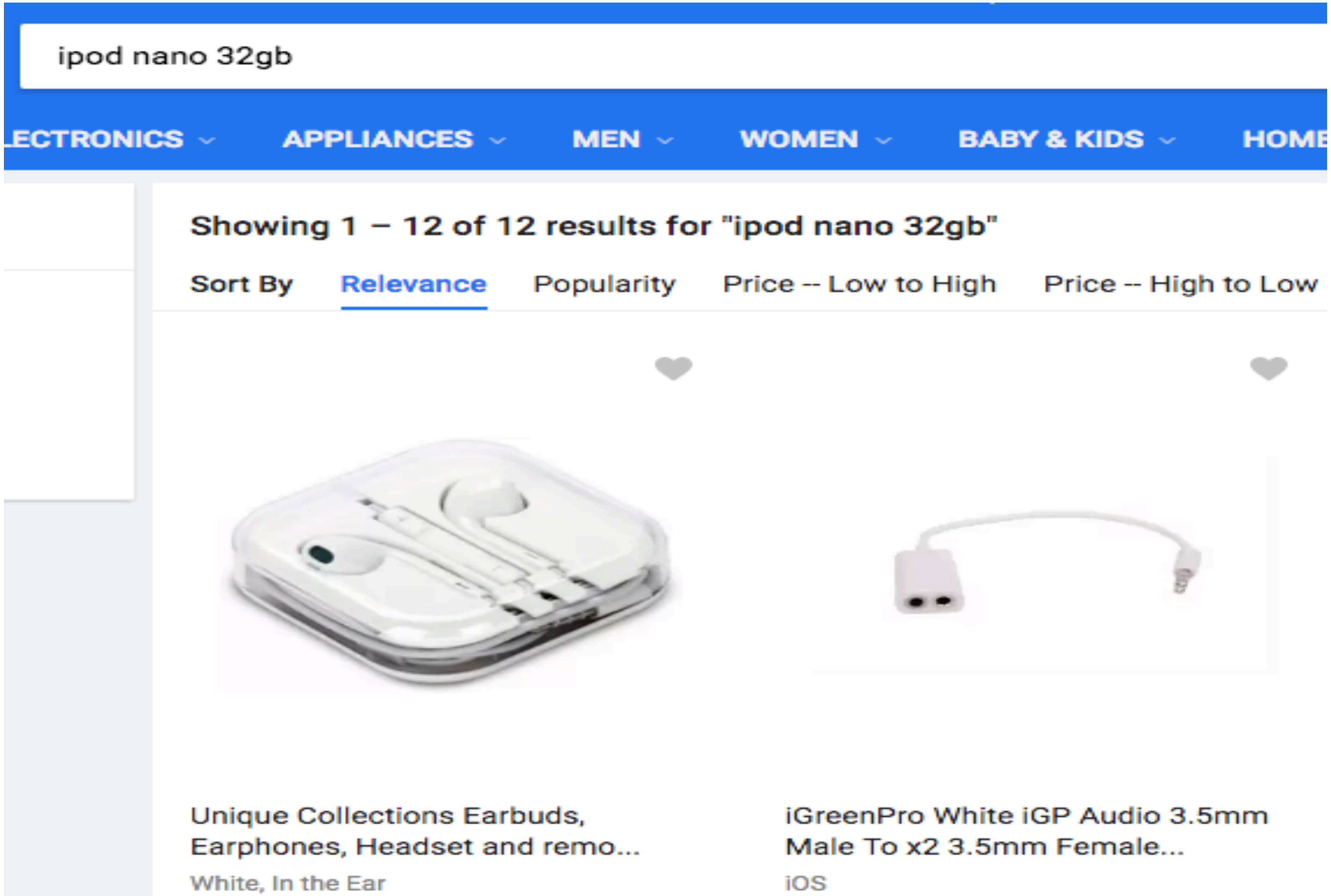
Need for query augmentation

- Mismatch between seller-buyer vocabulary
  - Item descriptions written by sellers usually more technical
  - "persian rug" vs. "carpet"
  - "gucci purse" vs. "designer handbag"
- Lack of domain knowledge of buyers
  - "ipod nano 32gb" → "ipod nano 16gb"
- Transient inventory – items may get sold and no longer be available, seasonal buzz items, …

Flipkart results for “ipod nano 16gb”



Flipkart results for “ipod nano 32gb”



Types of query augmentation

- Query refinement
  - Specialization: “ipod nano” → “ipod nano 16 gb”
  - Generalization: “blue ipod nano” → “ipod nano”

- Related query: suggestions that are neither specialization nor generalization

- Which type of suggestions to give?
  - Depends on factors like type of buyer, category of item

Dependence on types of buyer

- Focused buyer
  - Intends to make a specific purchase
  - Better to give focused, specialized suggestions
  - Generalization or related queries might be distracting

- Exploratory buyer
  - Exploring the inventory
  - Generalization or related queries helps to explore

- Challenge to distinguish between the two types

Dependence on category of item

- Electronics category
  - Usually buyers know what item they want to buy, might not know technical specifications
  - Specializations or generalizations work better

- Antiques category
  - Most users exploring without knowing what exactly to buy
  - Related suggestions might work better

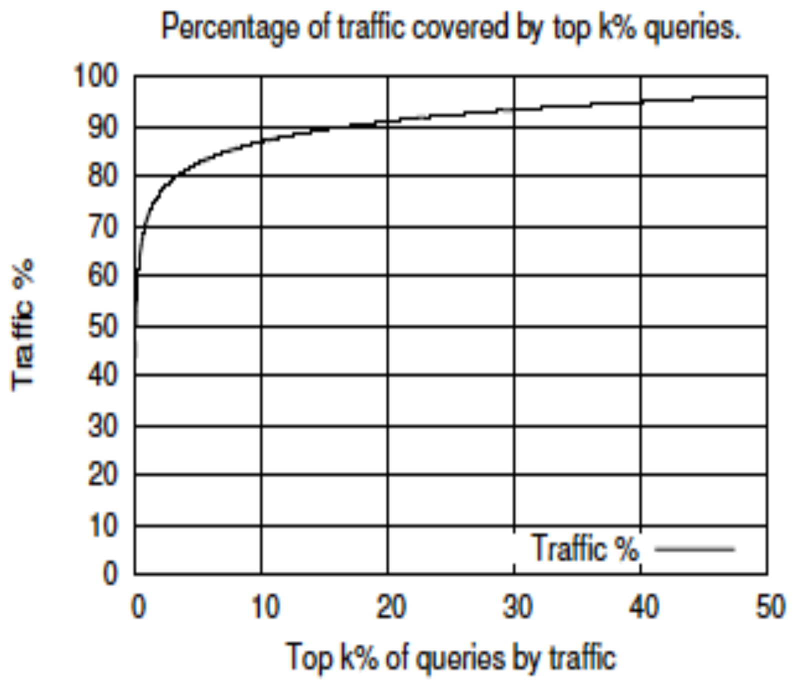
Challenges

- Queries and items are heavily transient
  - Typically low overlap between distinct queries on a day and distinct queries on the next day (~30% on Ebay)
  - Buzz queries or seasonal queries (Halloween, Christmas) can come up during wrong time period

Challenges

- Long tail query distribution
  - Head queries: asked frequently
  - Tail queries: asked rarely

- Statistics from eBay:
  - 20% head queries cover 91% of search traffic
  - Query frequency distribution is usually power-law



Why care about the rest 80% queries in the long tail?

Importance of tail queries

- Tail queries have low recall
  - Low query frequency <--> low recall in inventory
  - Correlation between demand and supply

- Low recall → shoppers need query suggestions more
  - Click Through Rate (CTR) on suggested queries much higher for queries which have low recall

- For tail queries, not enough information in query logs

How to evaluate query suggestions?

- Most common measure: Click Through Rate (CTR)
  - A suggested query is helpful if users click on the results that it retrieves

- Another intuitive measure: higher purchase
  - But, suggestions with higher CTR may not lead to higher purchase
  - Depends on the value of the suggested item, personal choice of the buyer, …

Methods for Query Augmentation

- Use query logs → learn from past user behavior

- A graph-based method
  - Inferring semantic query relations from collective user behavior, Parikh et al., CIKM 2008

- Learning from how users recover from bad queries
  - User behavior in Zero-Recall eCommerce Queries, Singh, SIGIR 2011
  - A Study of Query Term Deletion using Large-scale Ecommerce Search Logs, Yang et al., ECIR 2014

Graph based augmentation

- Inferring semantic query relations from collective user behavior, Parikh et al., CIKM 2008

- Each query: a bag of distinct words

- Build a graph
  - Each node is a query
  - Edges between nodes (queries) added based on various estimates of similarity between queries

Query similarity: textual

- Connect a query *q* to
  - All queries that can be formed by adding one or more terms to *q* (specializations)
  - All queries that can be formed by removing one or more terms from *q* (generalizations)
- Edges
  - Bidirectional: traversal in one direction implies specialization, traversal in reverse implies generalization
  - Can be weighted based on term overlap

Query similarity: textual

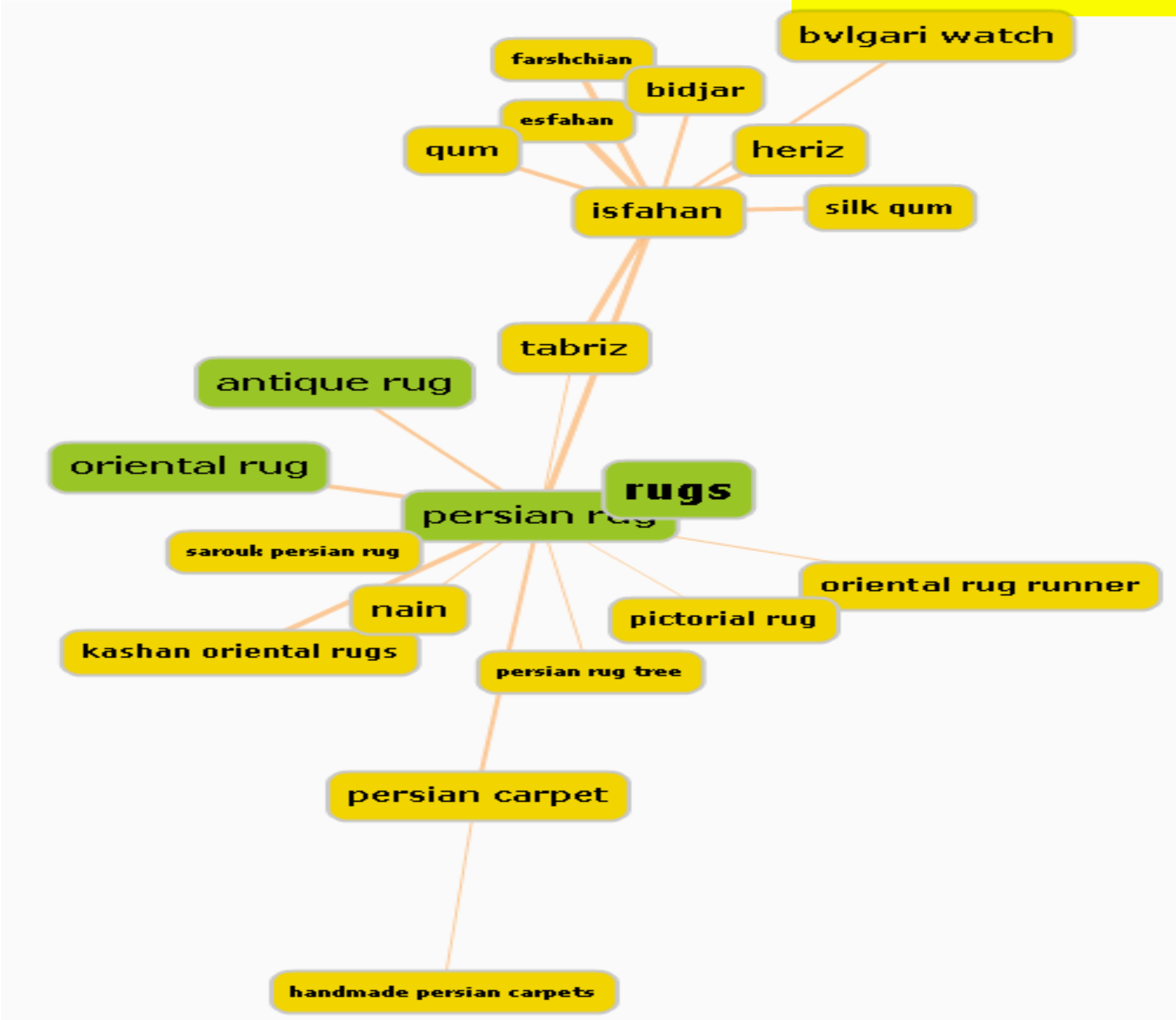
Query similarity: user session-based

- If a user issued a sequency of queries during a session Q1 → Q2 → Q3 → Q4, connect Q1 to Q2, Q2 to Q3, Q3 to Q4

- Intuition: user will issue semantically related queries in a session

- Edges can be weighted based on number of sessions in which a transition appeared

Query similarity: user session-based



Can capture more semantics than purely text-based graph

E.g.,
- “rug” and “carpet”
- “isfahan”, “tabriz” are specific types of rugs

Query similarity: user session-based

- Concerns:
  - Change in user-intent within a session
  - Automated bot activity

- Remedies:
  - Only consider user sessions where buying occurred
  - Only consider a transition (edge) if it appears in at least three sessions

Query similarity: semantic

- Queries mapped to a higher dimensional space where semantic similarity can be measured

- Look at the item a user buys after issuing a query
  - Words found in Title / Description of item
  - Category, ISBN of item

- Map the query to the features of the item bought
  - Query gets mapped to a vector in the high dim space

Query similarity: semantic

- Mapping of some queries (top features only shown)

Query	Features for the Query
apple ipod	gb(4061), gen(4051), mp3(3766), video(3539), player(3164), black(3101), nano(3004), silver(2959)
apple dishes	franciscan(8721), butter(4198), glass(3974), small(3045), logo(2887) , mark(2887), vintage(2721), usa(2655)
j k rowling	potter(5412), harry(5395), 1st(5378), sorcerers(5069), stone(4521), signed(3254), chamber(2702)
1st sorcerer stone	sorcerers(11177), harry(6573), potter(6573), u(3402), american(3402), dj(3303), ed(3220), true(2981)

Query similarity: semantic

- A query: a vector in a high-dimension space
- Semantic similarity between two queries: dot product of the corresponding vectors

jessica alba	rosario dawson	Film celebrities	0.728
zune	black zune	Generalization / Specialization	0.918
harry potter	j k rowling	Book character / Book author	0.631
ps2	playstation 2	Abbreviation / Full Name	0.891
apple player	apple dishes	None other than one common word	0.000
jessica simpson	shoes	Brand / Product	0.796

Query similarity: semantic

Only those edges shown whose similarity value is at least 0.50

Query similarity: use which measure?

- Each similarity measure has pros and cons
  - Textual similarity does not capture semantic similarity
  - Textual similarity is the only usable method for new queries
  - Session based similarity might have noise due to user intent change
  - Session and semantic similarity useful only when a query has seen sufficient activity

- eBay used linear combination of all three similarity measures to form a Semantic Query Network

Learning from how users recover from bad queries

- User behavior in Zero-Recall eCommerce Queries Singh et al., SIGIR 2011

- A Study of Query Term Deletion using Large-scale Ecommerce Search Logs, Yang et al., ECIR 2014

Bad queries

- Zero-recall queries: queries which do not return any matching item

- Why do some queries not return any matching item?
  - Usually too verbose
  - Buyer may not know domain-specific terms
  - Temporal volatility of item space

How do users deal with zero recall?

- Two types of users
  - Novice users – who are new to the ecommerce site
  - Power users – experienced in using the site
  - Differentiated based on how much they have spent in buying items on the ecommerce site

- The two types of users deal differently with zero recall queries

How do users deal with zero recall?

- Novice user

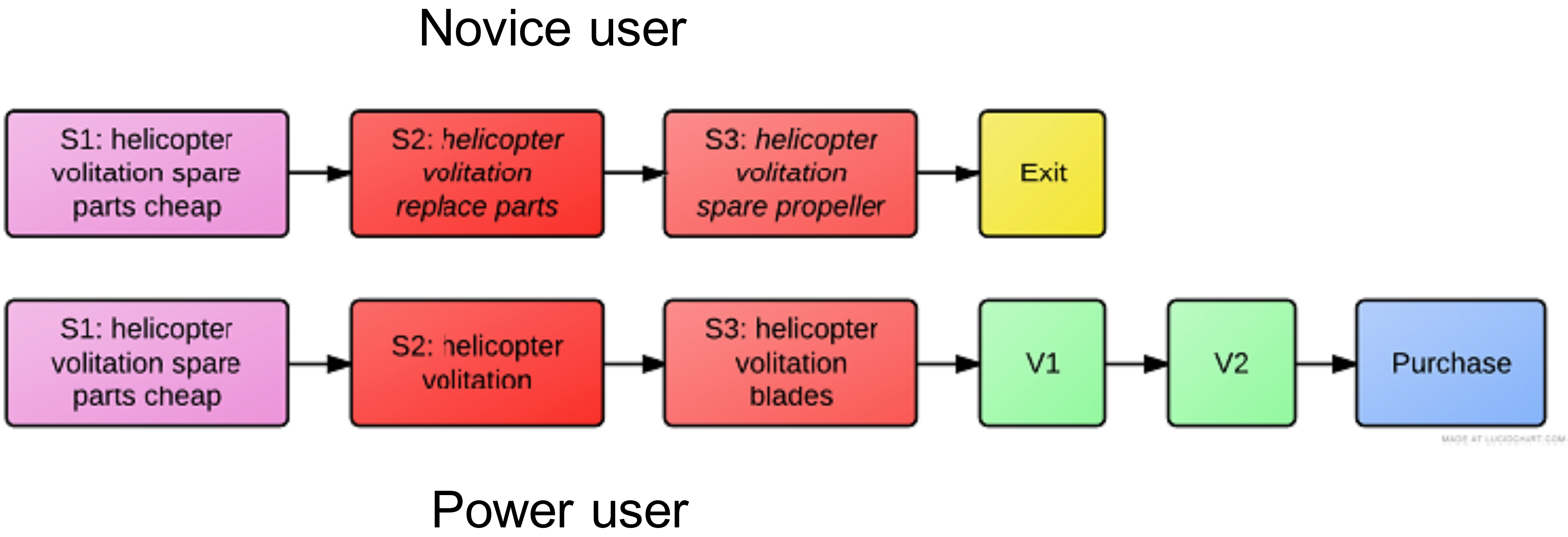
- Twice more likely to give up and exit, after seeing zero results
- Depend on assistive technologies (e.g., suggested queries) to recover

- Power user

- Usually re-formulate queries and continue trying to get relevant items
- Prefer to re-formulate queries themselves and recover

- Algorithms can learn from how they recover

Example novice and power user



How to recover from zero-recall queries?

- Primary reason for zero-recall queries:
  - Too verbose queries
  - Contain extra terms which do not match any item
  - "small carry on bag for air plane" vs. "carry on bag"
- Possible way to recovery: delete some terms
  - Which terms to delete?
  - Deleting important terms → information loss
  - Same term can have varying importance based on query context: "gap wool blazer" vs. "spark gap transmitter"

Which terms to delete in queries?

- Learn which terms to delete, from prior user behavior (query logs)
  - A Study of Query Term Deletion using Large-scale Ecommerce Search Logs, Yang et al., ECIR 2014

- Identify query transitions q1 → q2 such that
  - q1 did not lead to any click activity on results
  - q2 led to one or more clicks on results
  - q2 was formed by the user deleting one term from q1

Which terms to delete?

- Given: a query, a term in the query, category of the query (38 meta-categories from Ebay)
  - Train a logistic regression classifier to predict the probability of the term being deleted
  - Training instances (t, q, y): *t* is included in query *q*, *y*=1 if *t* was deleted by user, 0 otherwise

- Using query-dependent features for a term
  - Three types of features: lexical, history-based, context

Query-dependent features of a term

- Linguistic and lexical features
  - Whether term is conjunction/adjective/numeric/brand name
  - Term importance: probability of term appearing in the product title, conditioned on its probability of appearing in the product description
- History-based features
  - Deletion history: how often the term was deleted from queries in this category
  - Rareness (similar to IDF)
  - Is-rightmost-term (users tend to delete right-most term)

Query-dependent features of a term

- Context features: textual context of the term in the given query
  - Collocations: lexical forms of the neighboring words
  - Point-wise mutual information between all pairs of terms in the query, based on frequencies of the two terms in the query logs under the particular category

- A separate logistic regression predictor trained for each query category

Insights on term deletion

- History-based and context-based features equally important across all categories
- Importance of linguistic and lexical features vary greatly across categories
  - Adjectives are important for ‘clothing’ category, but not for ‘computer’ category
- Brand names are important

References

- Inferring semantic query relations from collective user behavior, Parikh et al., CIKM 2008
- Query Suggestion for E-Commerce Sites, Hasan et al., WSDM 2011
- User behavior in Zero-Recall eCommerce Queries Singh et al., SIGIR 2011
- A Study of Query Term Deletion using Large-scale Ecommerce Search Logs, Yang et al., ECIR 2014