Recommender Systems & Collective Intelligence

COMP47580

Dr. Michael O'Mahony

michael.omahony@ucd.ie

Overview

- Recommender Systems
  - Non-personalised vs. personalised approaches

- Content-based recommendation:
  - Traditional content-based (unstructured) vs. case-based (structured) recommendation
  - Recommendations are based on product/content similarity.

- Collaborative filtering:
  - *Content-free* recommendation
  - Recommendations are based on user preference data – no information about product content is required.

“Single-Shot” Recommendation

- Single-shot recommendation model:
  - Limited to single interaction between user and recommender system
  - What happens if the user is not satisfied? Rate additional items to obtain new recommendations…
  - Consider e-commerce domains:
    - User may not know exactly what s/he wants (e.g. purchasing a digital camera – different cameras have different features/specifications, prices…).
    - Single shot models do not readily facilitate an exploration of the product space
    - Move towards a different model – as the user learns more about available options and features, facilitate her to navigate through the product space (e.g. *“this camera is expensive! – show me some cheaper options”*, *“show me similar cameras but with better resolution”*)

This lecture…

- Conversational Recommender Systems:
  - Engage users in extended dialogue
  - Engagement begins with a query (specified by the user or inferred from the user’s profile)
  - Have multiple interactions (recommendation cycles) with users
    - Small number of recommendations displayed during each cycle (in some instances, just one)
    - Elicit user feedback on displayed recommendations – leverage feedback to eliminate some items in the product space and to idenfify suitable rcommendations for the next cycle…
  - Help users to more effectively navigate complex product spaces
  - Think online “sales assistant”

Case-based Recommendation

- **Item representation**: items are represented in a structured manner using a well-defined set of features and feature values
- **Similarity assessment**: case-based representation allows for sophisticated and fine-grained judgements about the similarity between items and other items (or user queries):
  - Case-level similarity – based on explicit mappings between case features, weights encode the relative importance of features:

$$\mathrm{Sim}(\mathrm{T},\mathrm{C}) = \frac{\sum_{i=1}^{n} w_i \times \mathrm{Sim}(v_{C,i}, v_{T,i})}{\sum_{i=1}^{n} w_i}$$

  - Feature-level similarity – numeric (symmetric, asymmetric similarity functions) and non-numeric (set-based, ontology-based approaches)
- This approach to item representation and similarity assessment facilitates:
  - Search and navigation of complex information spaces
  - Flexible user-feedback options

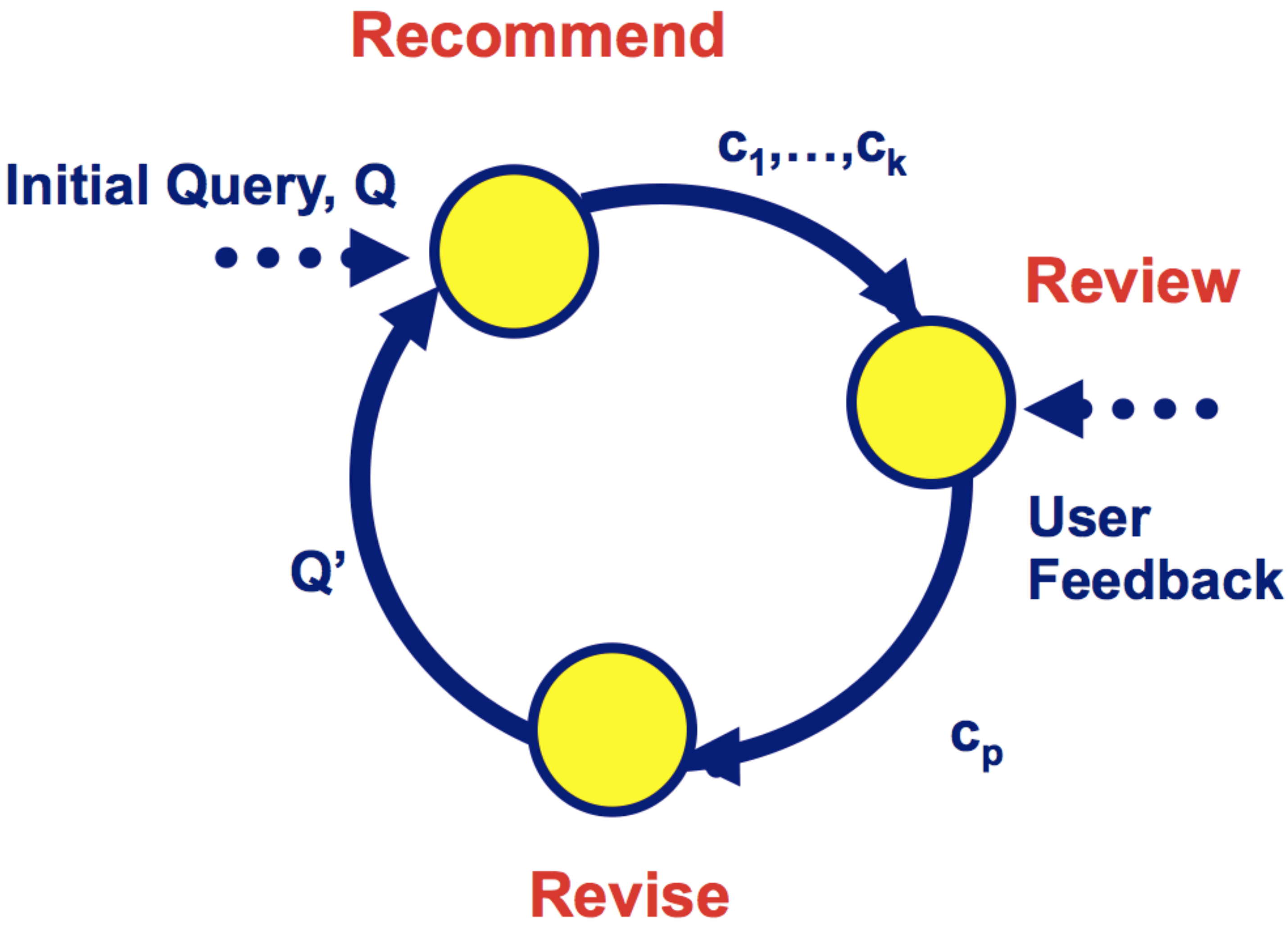
Example Cases…

Epson Expression Home XP-235 All-in-One Inkjet Printer

£44.00 & **FREE Delivery** in the UK. Details | In stock. Dispatched from and sold by Amazon. Gift-wrap available.

	 Epson Expression Home XP-235 All-in-One Inkjet Printer	 Epson Home XP-335 Expression All-in-One Inkjet Printer
Customer Rating	(74)	(89)
Price	**£44.00**	**£44.99**
Delivery	FREE Delivery	FREE Delivery
Sold by	Amazon.co.uk	Amazon.co.uk
Connectivity Technology	WiFi	USB 2.0, Wireless LAN
Resolution	1200 Dots Per Inch	1200 Dots Per Inch
Ink Colour	Multicoloured	Multicoloured
Dimensions	14.5 cm x 30 cm x 39 cm	14.5 cm x 39 cm x 30 cm
Item Weight	Information not provided	4.2 kg
Maximum Printspeed Black White	26 ppm	33 ppm
Model Year	Information not provided	2015
Scanner Type	Flatbed	Flatbed
	Add to Basket	Add to Basket

Conversational Recommender System Cycle



Considerations

- Initial query formation (user specifies initial query with some feature values or feature values can be inferred from the user's profile)
- Recommendation generation
- Presentation of candidates
- Feedback mechanism
- Recommendation explanation
- Cognitive load

Navigation & Feedback

- Conversational recommenders are distinguished by the kind of *navigation* considered and how *feedback* is elicited
- **Navigation by Asking**:
  - Users are asked a series of questions about their requirements…
  - E.g., "*What style of camera do you want - Compact or SLR?*" or "*How much optical zoom do you need?*"
  - Such feedback is referred to as *value elicitation*
- **Navigation by Proposing**:
  - Avoid posing direct questions to users
  - System engages users in recommendation cycles – during each cycle:
    - Present users with interim recommendation(s)
    - Users are asked for feedback in the form of e.g. a rating or simple preference (e.g. none of the current recommendations are ideal, but this one best meets by needs…)
    - System uses feedback to select the recommendations for the next cycle

Types of Feedback

- Value Elicitation
- Ratings-based Feedback
- Preference-based Feedback
- Critiquing-based Feedback

Value Elicitation

- Navigation by asking
- User are requested to provide specific values for features:
  - E.g., in the laptop domain, the price they are prepared to pay, memory and hard disk requirements, preferred manufacturer…
  - **Before** presentation of recommendations
- Feature-level interaction
- Provides very rich feedback:
  - Most direct approach to elicit user requirements
- High cost:
  - Requires very precise feature-level interaction
  - Users must possess a high level of domain knowledge
  - Users must be willing to answer direct/specialised questions

Value Elicitation – Example

1.	**Inquirer**:	*Where do you think I should eat tonight?*
2.	**Advisor**:	What type of food would you like?
3.	**Inquirer**:	*What types are there?*
4.	**Advisor**:	You can say things like Chinese, Indian, and Mediterranean.
5.	**Inquirer**:	*Oh, maybe a cheap Indian place.*
6.	**Advisor**:	What kind of parking would you like?
7.	**Inquirer**:	*I don't care, as long as it's in Palo Alto.*
8.	**Advisor**:	I'm sorry, I don't know of any restaurants like that, would you like to search for any price?
9.	**Inquirer**:	*No, I think I'd like Chinese instead.*
10.	**Advisor**:	How does Mandarin Gourmet on 420 Ramona sound?[1]
11.	**Inquirer**:	*No, what else do you have?*
12.	**Advisor**:	How about Jing-Jing Szechwan Hunan Gourmet on 443 Emerson Street?
13.	**Inquirer**:	*Sure, that sounds fine.*

- Adaptive Place Advisor Recommender System:
  - Dialogue between user (inquirer) and system (advisor) to recommend restaurants
  - System asks free-form questions; user may also ask questions of system
  - System selects questions to narrow down the range of possible recommendations
  - Recommendations are made when only a few items remain

Ratings-based Feedback

- Navigation by proposing
- Users are requested to provide explicit ratings for recommendation candidates:
  - Positive/negative, 1-5 stars…
  - Post presentation of recommendations
- Case-level interaction
- Cost:
  - Users express qualitative indication of interest at the level of an individual case(s)
  - But may need to rate many cases…
  - Detailed domain knowledge not necessary but user must consider the case as a whole


http://serendipity.ucd.ie:8080/store/index.jsp
Serendipity
View Basket | Checkout | About Us | Contact Us
Home
Jewellery
Phones
Weddings
Electronics
Computers
Browse: Weddings
Wedding Dresses
Bridesmaid Dresses
Mother of the Bride Dresses
Wedding Accessories
Bridal Veils and Wedding Veils
Prom Dresses
Your Basket
There are no items in your basket.
Impression - 2663
Impression - 2624
Impression - 2648
Recommended Price $690
Recommended Price $730
Recommended Price $550
Our Price $495
Our Price $515
Our Price $425
Add To Basket
Pic 1/4
Pic 1/3
Pic 1/4
Preference
Designer
Impression
Colors
Ivory/Champagne
Ivory
Sizes
2
Neckline
Strapless
Waist/Silhouette
Mermaid
Pieces/Separates
One-piece
Train Length
Sweep
Fabric
Lace
Chiffon
Formality
Formal
Sleeve length
Strapless
Hem Line
Floor Length
Lace
Filters
Size
PleaseSelect
Neckline
Bateau
Bateau, Scalloped
Drape
Designer
Alfred Angelo
Bonny
DaVinci
Waist / Silhouette
A-Line
A-Line, Empire
Ballgown
Colour
Antique
Black
Blue
Price
Min
Max
No Limit
more...
Update

Preference-based Feedback

- Navigation by proposing
- Users are requested to indicate a preference (usually positive, but can be negative) for one candidate recommendation from those presented:
  - Post presentation of recommendations
- Case-level interaction
- Cost:
  - Users consider and choose one preferred case out of *k* presented cases – simpler than ratings-based
  - Must consider and compare cases as a whole
  - Relatively low domain expertise required
- Limited in its ability to guide the navigation process
  - Not always clear why user has selected one case over alternatives

Critique-based Feedback

- Navigating by proposing
- Typically users are presented with a single recommendation during each cycle
- Users are requested to provide a directional preference on a feature-value (e.g. price less than €1000)
  - Post presentation of recommendations
- Feature-level interaction
- Powerful form of feedback
  - Individual critiques act as a filter over the remaining cases (items)
- Low cost:
  - Does not require detailed domain knowledge
  - Provide feedback at feature-level rather than considering entire case

Case Study 1

Critique-based Feedback

Critique-based Feedback

- Feedback is in the form of a directional feature constraint:
  - E.g., a user may ask for a camera that is *cheaper* than the current recommendation – example of *unit* critique
- Each critique acts as a filter over the item space:
  - In the next recommendation cycle, items (e.g. restaurants, cameras etc.) that do not satisfy the critique are eliminated from consideration
- Compound critiques:
  - Designed to cover *multiple* features
  - E.g. show me a *sportier* car than the current recommendation, i.e. a a car with a larger *engine size* and with better *acceleration*
  - Allows the recommender to take larger steps through the item-space and eliminate more items during each recommendation cycle

“Find Me” Systems

Figure 1: Entry point for the Entree system

Entrée – A Restaurant Recommender System

- Users start off by indicating what kind of a restaurant they are looking for

Two modes of interaction:

- Term-based selection
- Similarity-based selection

See reading (**Burke 2000**)

Retrieval and User Feedback in Entrée

Retrieval and Recommendation

- The system retrieves restaurants that are considered to be similar to the user's current requirements

User Feedback

- The user indicates a feature critique or 'tweak' to constrain the value range of that feature

{cheaper, quieter, livelier…}

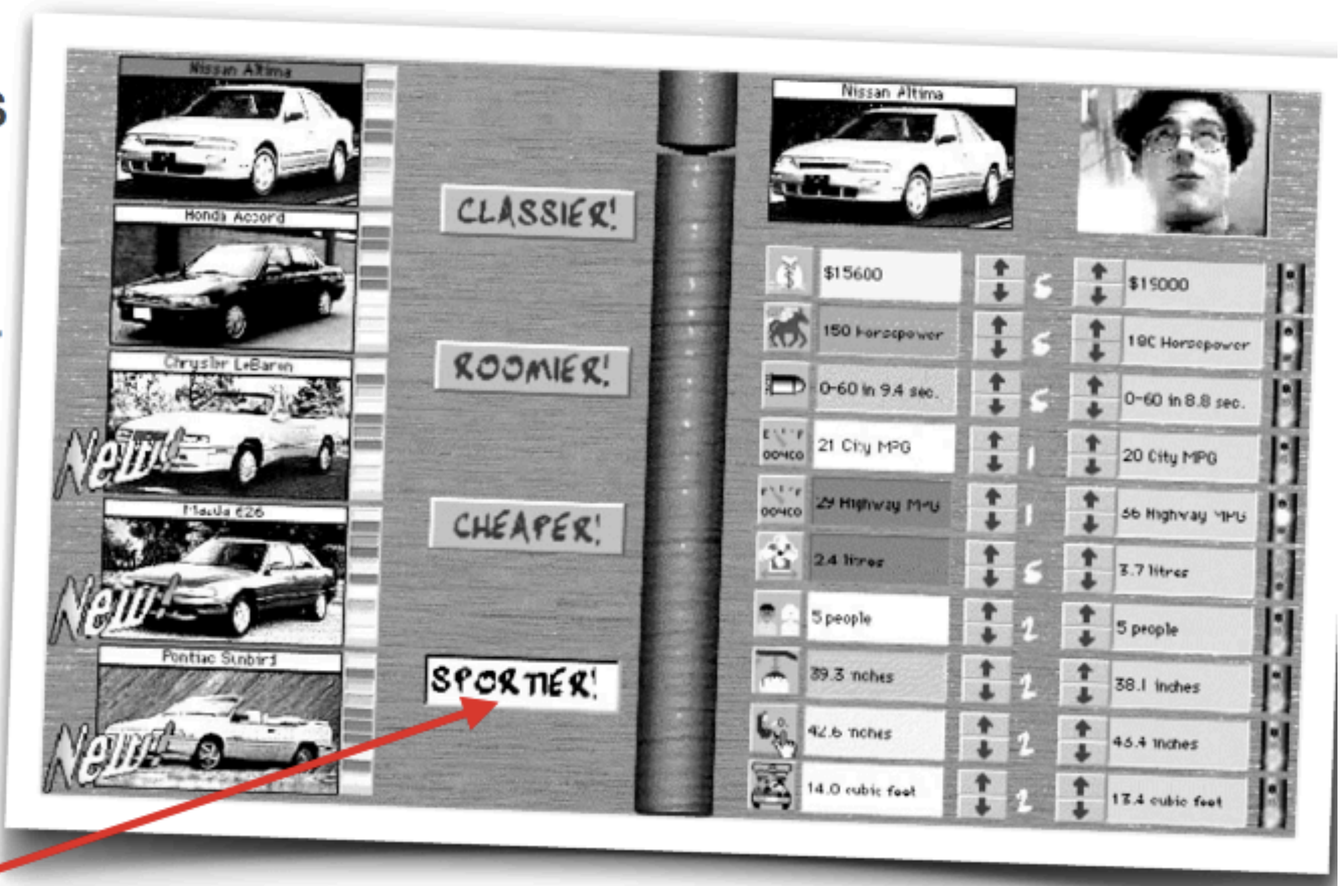
Unit Critique

Figure 2: Similarity-based retrieval in Entree

Compound Critiques

"Find-Me" systems

- [Burke et al.]
- Car Navigator
- Critique allows user to manipulate multiple features
- Hard-coded, static and fixed



Compound Critique

Dynamic Critiquing

Dynamic Critiquing

- Flexible Approach
- Unit critique
- Compound critique
- Generate compound critiques on the fly

See reading:
- **McCarthy 2005a**

Mining Dynamic Critiques

Generate Critique Patterns

- Re-describe every other case in relation to current case

Mine Compound Critiques

- Identify recurring patterns within set of critique patterns – Apriori [Agrawal et al.]

Select & Present

- Choose *Compound Critiques* to present to user

Generate Critique Patterns

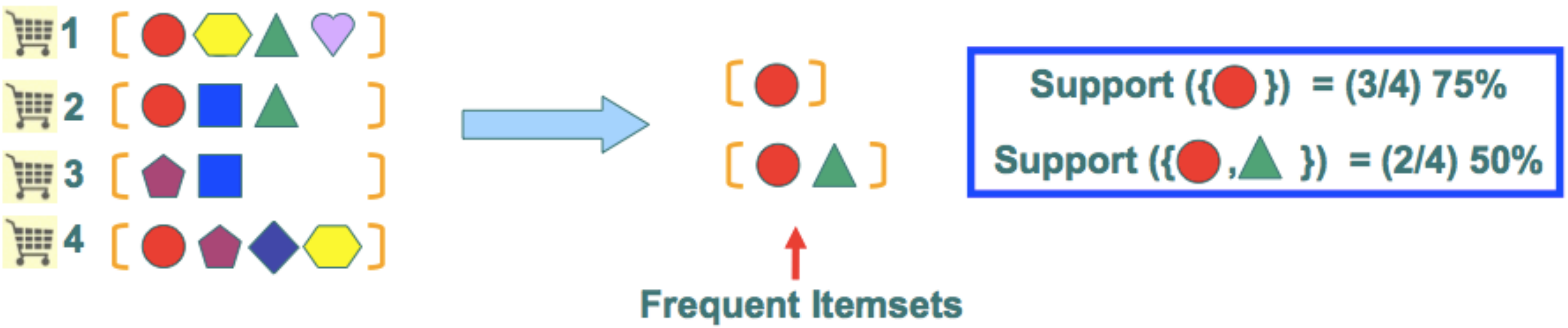
	Current Case	Case c from CB	Critique Pattern
HolidayType	Education	Language	!=
Price (Euro)	3738	2039	<
NumberOfPersons	2	1	<
Region	Egypt	Malta	!=
Transportation	Plane	Plane	=
Duration (Days)	14	21	>
Season (Month)	October	September	!=
Accommodation	FourStars	TwoStars	!=
Hotel	Anlage Arabia	Sprachkurs Malta	!=

Feature “Price” is assigned a “<” critique since case *c* is cheaper than the current case

Feature “Duration” is assigned a “>” critique since case *c* is longer than the current case

Mining Compound Critiques

- Key Idea
  - Identify recurring subsets of patterns within the pattern base
  - E.g. *Market-Basket Analysis*



- Apriori
  - [*Agrawal et al.*]
  - Use to find frequent itemsets
  - Rank itemsets by 'support' value

Mining Compound Critiques

	Price	Holiday Type	Duration	No. of Persons
Case 1	>	!=	<	>
Case 2	>	!=	>	<
Case 3	>	=	>	>
Case 4	>	!=	<	>
Case 5	<	=	>	<
Case 6	>	!=	<	<

- Candidate Compound Critiques
  - { [Price '>'], [Holiday Type '!='] }
  - { [Holiday Type '='], [Duration '>'] }

Support value of (4/6) 67%
Support value of (2/6) 33%

Select and Present

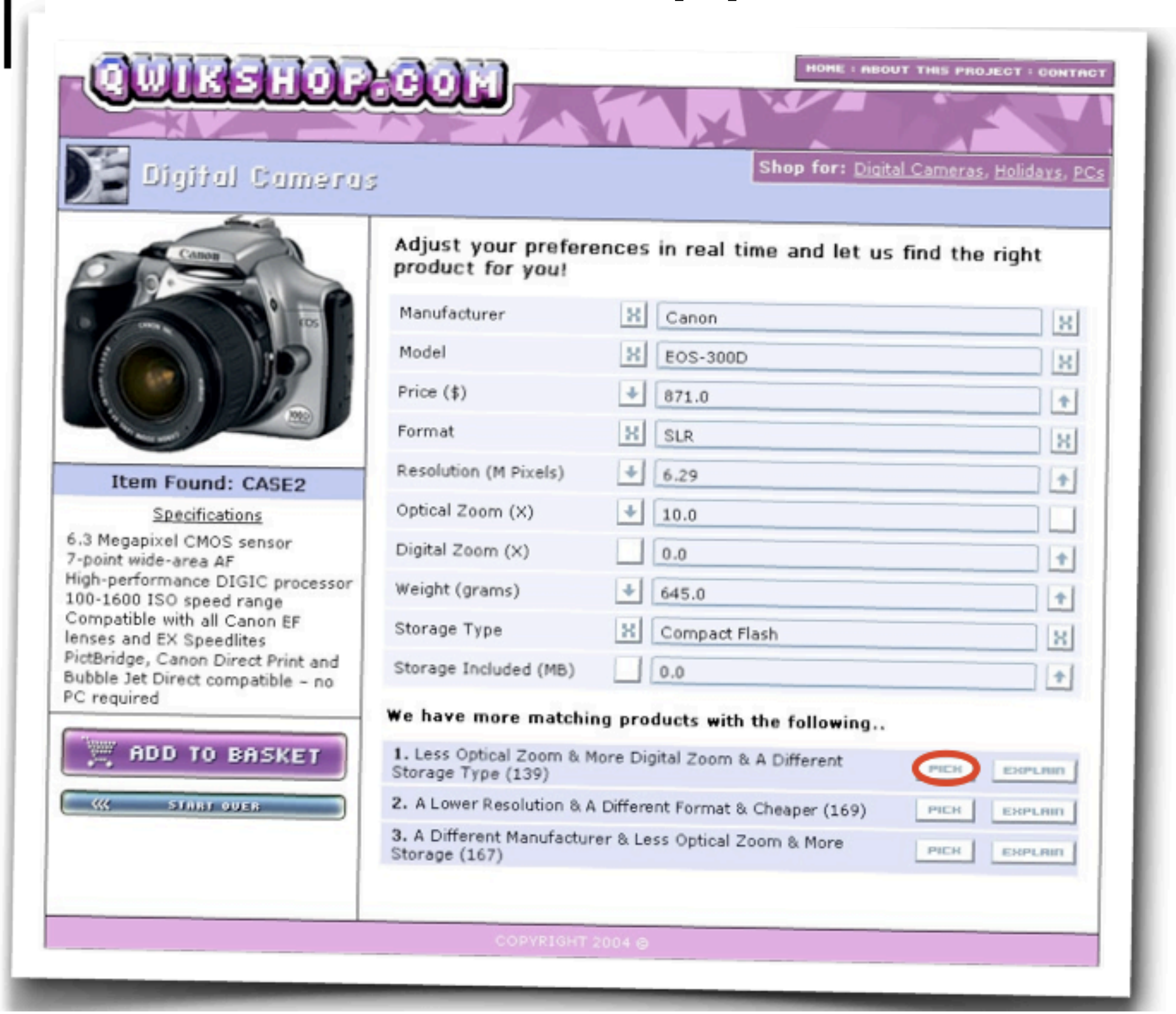
Three Selection Strategies

- Low-Support (LS) – Pick compound critiques with lowest support
- High-Support (HS) – Pick compound critiques with highest support
- Random (RAND) – Pick a set of random compound critiques

Present to User

- { [Price ‘>’], [Holiday Type ‘!=’] }
- “A Different Type of Holiday which is More Expensive”

QWIKSHOP Application



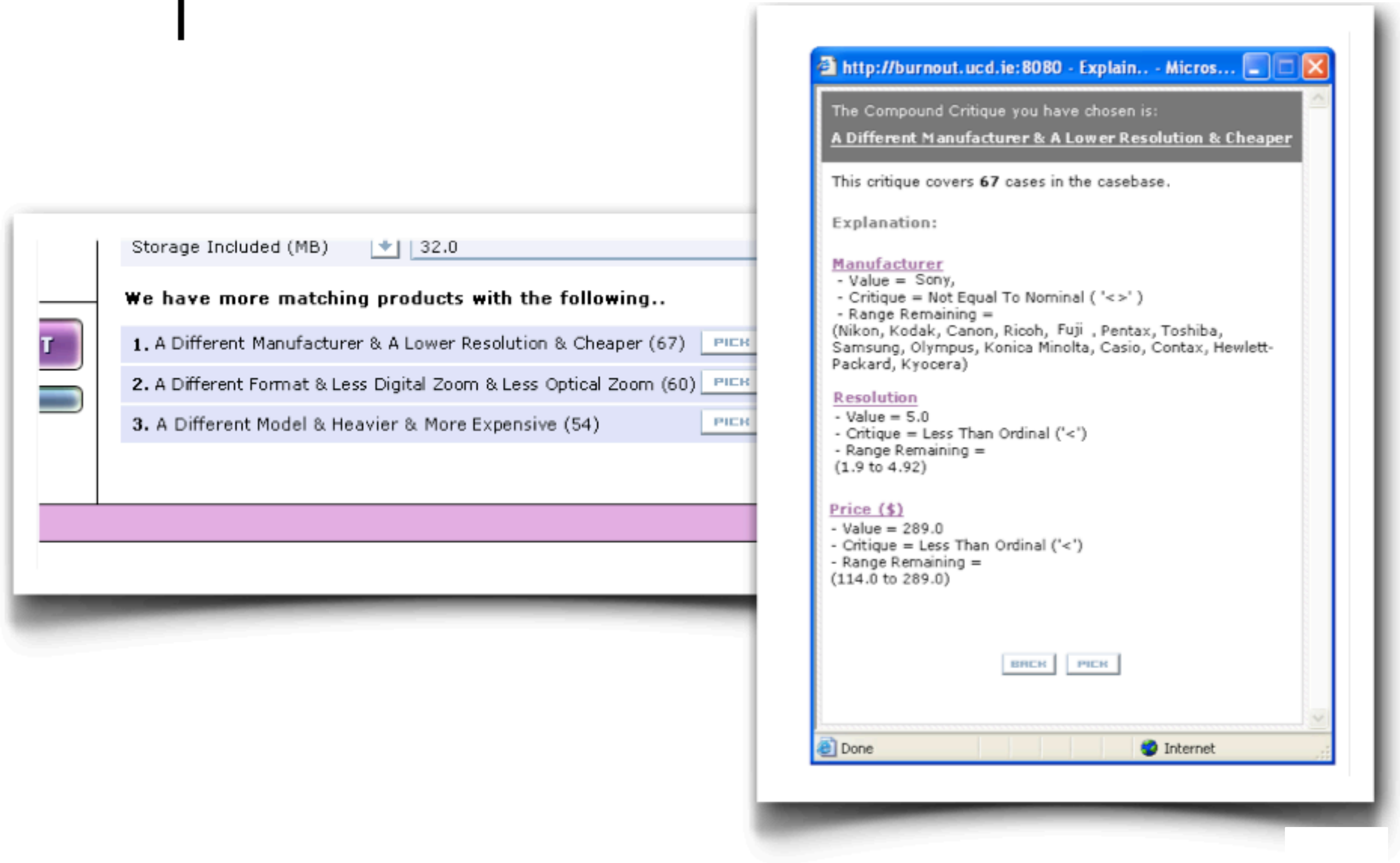
Explanatory Benefits

Explanatory Benefits

The explanatory power of compound critiques is evident in the information they convey to the user;

- They highlight the common interactions that occur between features of the remaining cases, e.g. more expensive and greater resolution tells the user that if they want a camera with greater resolution they should expect to pay more.
- They map out the product space by highlighting the recommendation opportunities that exist, e.g. cheaper and less resolution tells the user there exists products in the product space that satisfy that critique.

Explanatory Benefits



Diversity

Problem

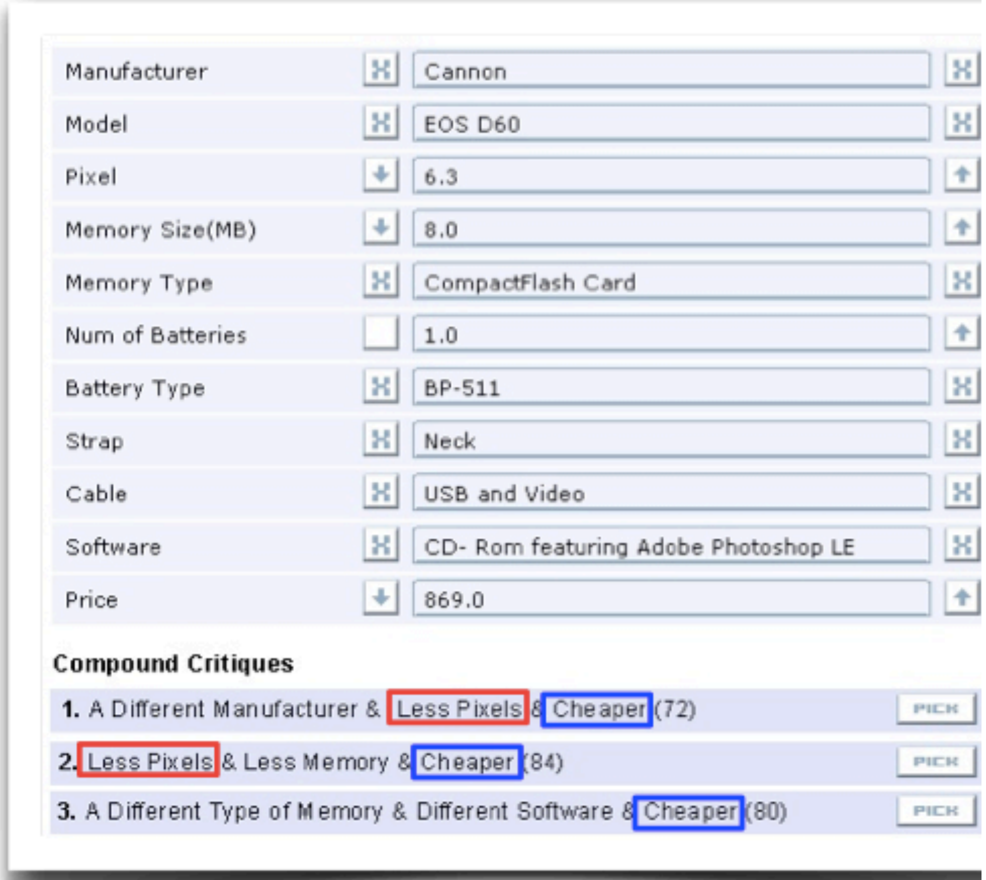
- Compound critiques presented, critique some of the features in the same way

Implications

- Limits the scope of feedback options
- Limits the applicability of the compound critiques offered
- Off-putting and contrary to user expectations

Solution

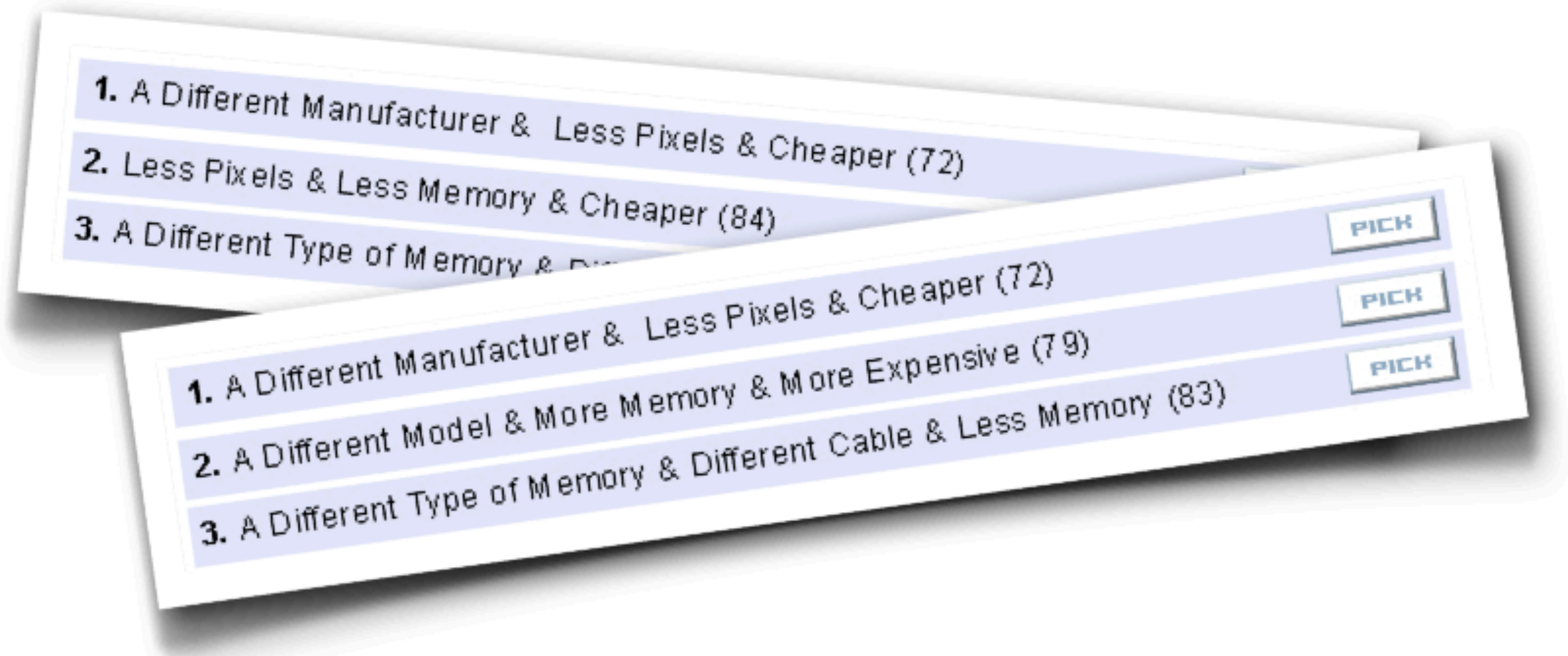
- Present compound critiques which cover a wider range of feature critique combinations



"Less Pixels" occurs in 2 compound critiques

"Cheaper" is common to all 3 compound critiques

Diversity Enhancement



- Use the Bounded Greedy Approach
- Produces compound critiques with fewer overlapping unit critiques
- Diverse critiques more likely to satisfy user as they cover more options
- Reduce session lengths

Cognitive Load

Level of effort associated with thinking and reasoning.

Important for us to understand the level of effort required by the user during system interactions.

Measured by interaction/response time.

Cognitive Load in Conversational Recommenders

Recommender systems make cognitive demands on users in some way.

Limiting factors:

- Users rarely have a clear understanding of what product they are looking for.
- Users may not be proficient in the required product description knowledge.
- Users may be unable to compare presented suggestions.
- Users are often incapable of retaining learned information in short-term memory.

Cognitive Load in Conversational Recommenders

Case Evaluation Level

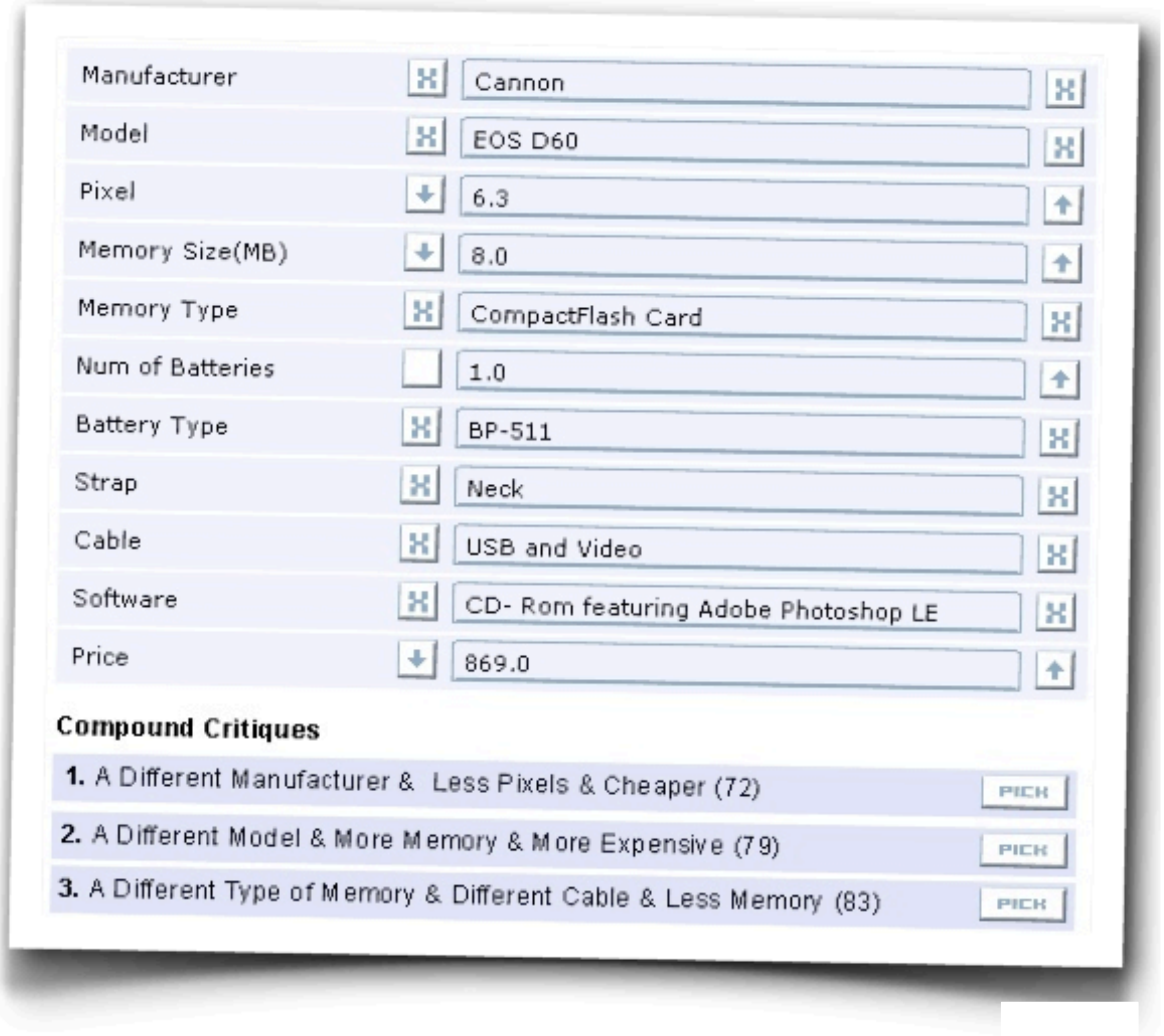
- Cognitive load experienced at case level.
- Comparative cognitive cost.
- i.e. Preference based feedback.

	Air Durham (men) $100 Buy It!	Air Structure Triax (men) $90 Buy It!	Air Span Triax (men) $85 Buy It!
Surface	Road Shoe	Road Shoe	Road Shoe
Midsole	PU midsole	Phylon midsole	Phylon midsole
Width	Standard	Standard	Standard
Motion Control	★★★★	★★★☆	★★☆☆
Feel: Impact Protection	★★★☆	★★☆☆	★★☆☆
Feel: Responsiveness	★☆☆☆	★★☆☆	★★☆☆
Breathability	★★☆☆	★★☆☆	★★☆☆
Water Resistance	No	No	No
Weight	14.5 ounces	12.6 ounces	12 ounces

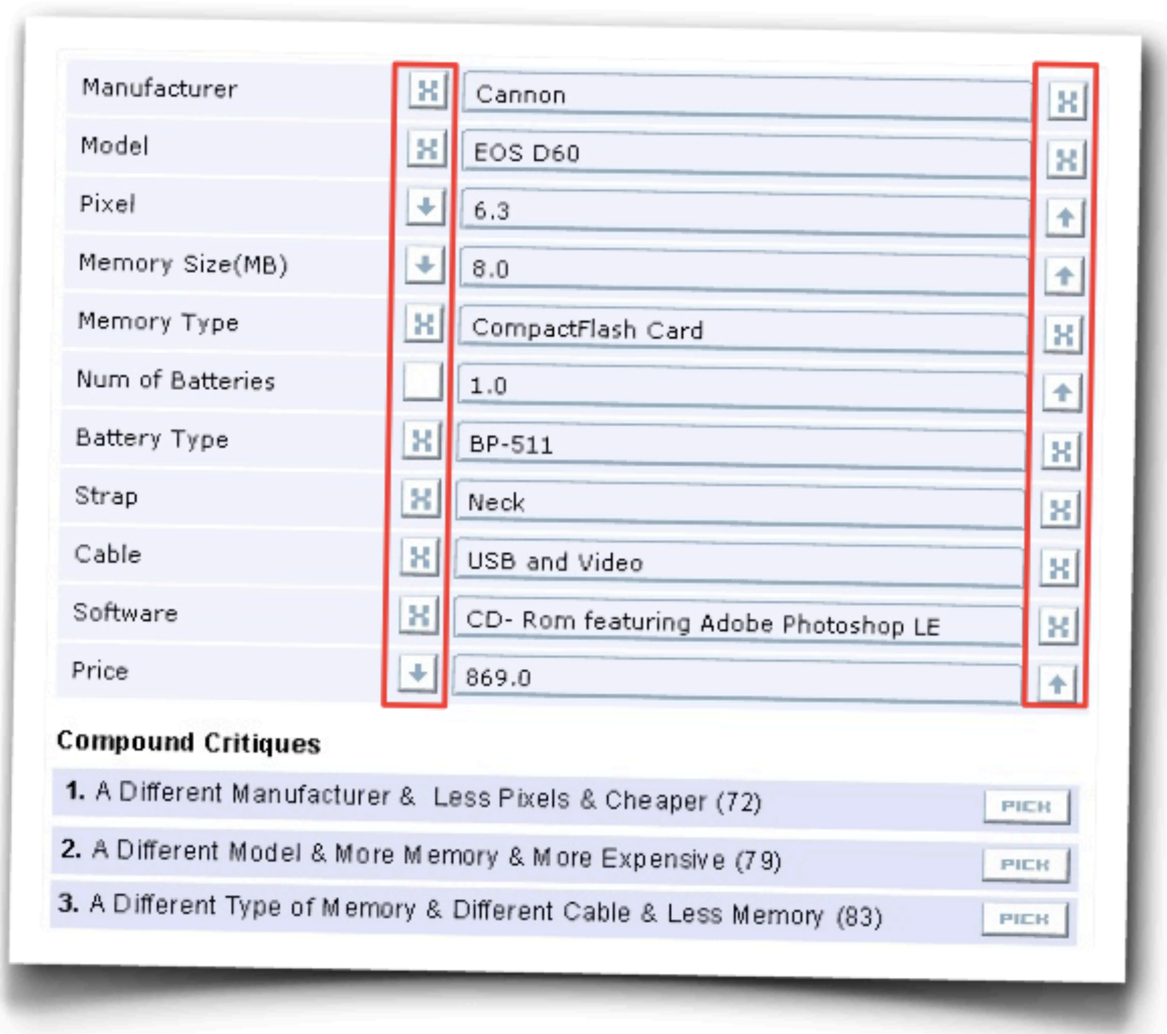
Cognitive Load in Conversational Recommenders

Feature Interaction Level

- Cognitive load experienced at feature level
- Users must breakdown and understand critique options offered.
- Unit critiques .
- Compound critiques.



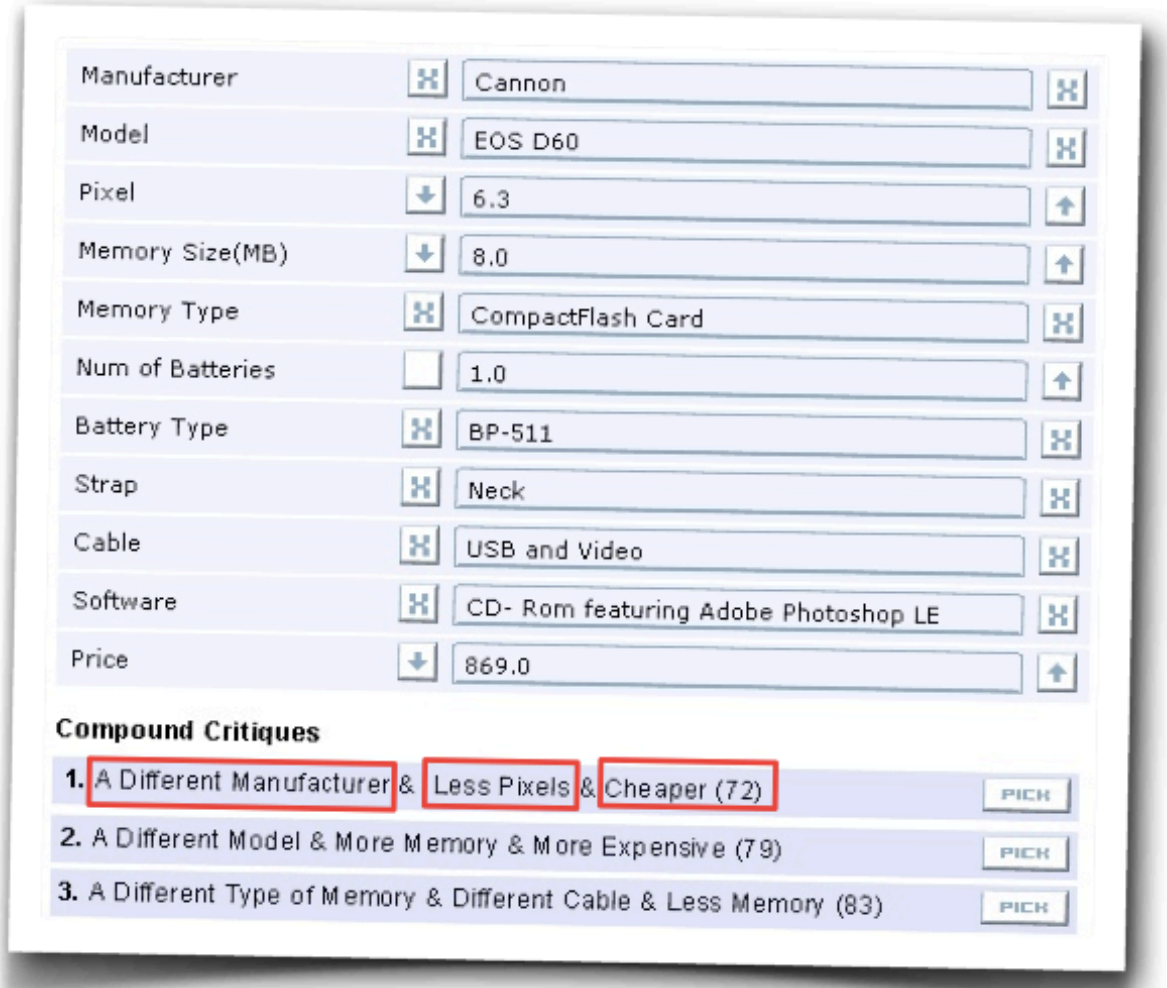
Cognitive Load – Unit Critiques



Unit Critiques

- Icons fixed.
- Do not change position from cycle to cycle.
- Feature values displayed.
- Easy to understand.

Cognitive Load – Dynamic Compound Critiquing



Compound Critiques

- 3 presented to user.
- Change on each cycle.

- Each compound critique made up of 3 unit critiques.
- No value information.

Cognitive Load – Results

Elapsed response time as a measure of cognitive load.

Increased cognitive load associated with compound critiques.

Additional cognitive effort is worthwhile

- Less elapsed time (28% reduction).
- Less recommendation cycles (71% reduction).

User behaviour differs

Cognitive load increase outweighed by recommendation efficiency improvements

Evaluating Dynamic Critiquing

Preformed both simulations and real user experiments to evaluate and validate our approach

Off-line simulations

- Reductions in session lengths of 36% - 66%

Live user studies

- Reductions in session lengths of 70% - 80%
- Number of cycles falls from 34 to 10
- Users apply compound critiques 25% of time

Case Study 2

Preference-based Feedback


http://serendipity.ucd.ie:8080/store/index.jsp
Serendipity
View Basket | Checkout | About Us | Contact Us
Home Jewellery Phones Weddings Electronics Computers
Browse: Weddings
Wedding Dresses
Bridesmaid Dresses
Mother of the Bride Dresses
Wedding Accessories
Bridal Veils and Wedding Veils
Prom Dresses
Your Basket
There are no items in your basket.
Impression - 2663
Impression - 2624
Impression - 2648
Recommended Price $690
Recommended Price $730
Recommended Price $550
Our Price $495
Our Price $515
Our Price $425
Add To Basket
Pic 1/4
Pic 1/3
Pic 1/4
Preference
Designer Impression Impression Impression
Colors Ivory/Champagne Ivory Ivory
Sizes 2 2 2
Neckline Strapless Strapless Strapless
Waist/Silhouette Mermaid Mermaid Mermaid
Pieces/Separates One-piece One-piece One-piece
Train Length Sweep Sweep Sweep
Fabric Lace Lace Chiffon
Formality Formal Formal Formal
Sleeve length Strapless Strapless Strapless
Hem Line Floor Length Floor Length Floor Length
Lace
Filters
Size PleaseSelect
Neckline Bateau Bateau, Scalloped Drape
Designer Alfred Angelo Bonny DaVinci
Waist / Silhouette A-Line A-Line, Empire Ballgown
Colour Antique Black Blue
Price Min: No Limit Max: No Limit
more...
Update

Preference-based Feedback

- Assumptions:
  - The user has some product requirement
  - The requirement is sufficiently vague (the user cannot simply specify all feature values)
  - The user provides an initial query (e.g. some initial feature values are specified or query is based on user profile) – small set of recommendations returned
  - The user is facilitated to find a good product by providing preference-based feedback on items displayed during each recommendation cycle (i.e. I prefer this item)
  - User feedback is leveraged to select the set of products to be displayed during subsequent recommendation cycles
- Practical Applications
  - Recommending restaurants, cars, cameras, property etc.

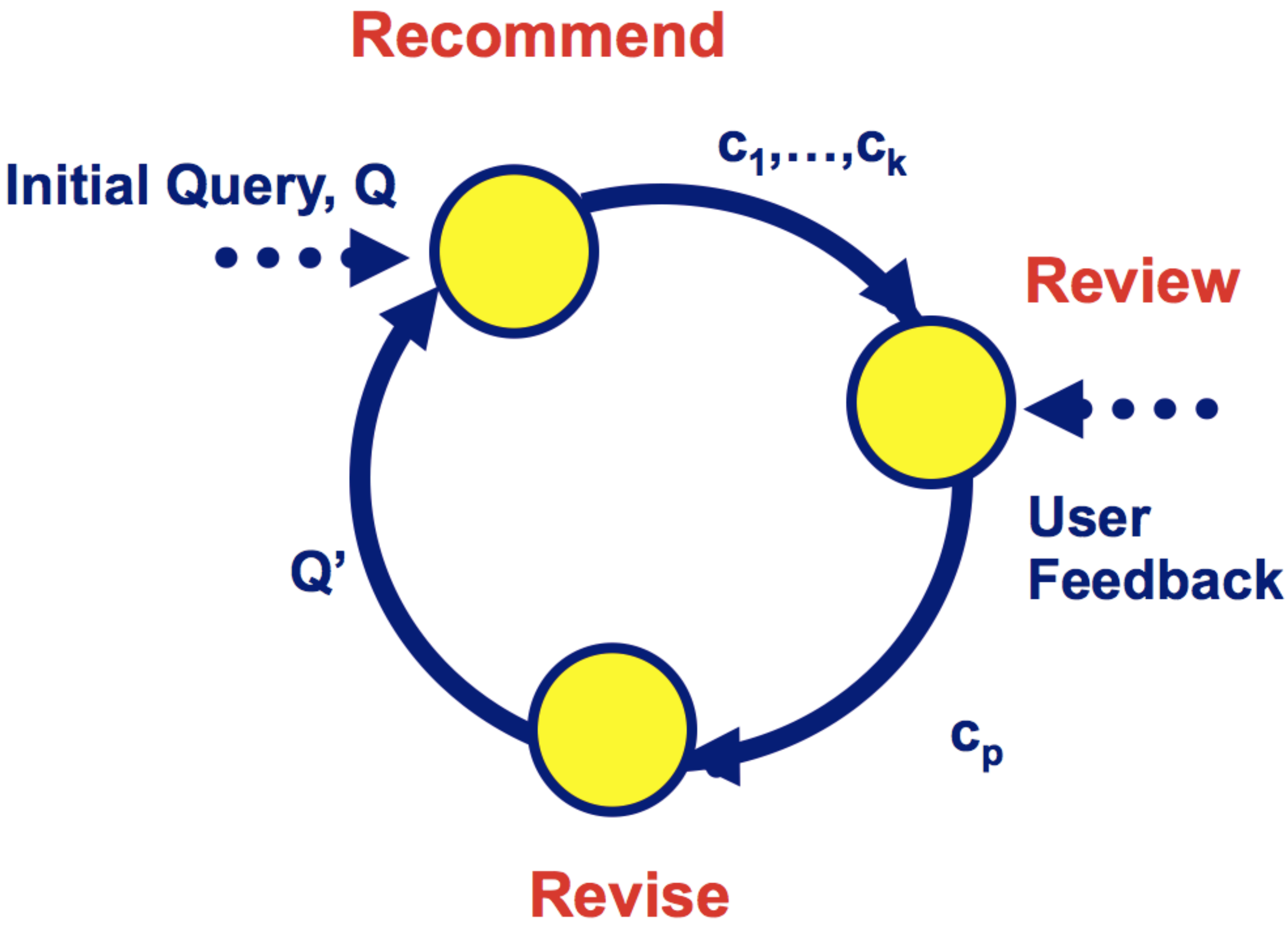
Limitations

- During each recommendation cycle, users simply indicate a preference for one item over the others:
  - Well suited to domains where users have limited domain knowledge
- Simple and relatively low cost form of feedback, but has limitations:
  - May not be clear why the user picked one item over the alternatives presented
  - Items presented may have some featues in common and some which distinguish them
- One approach: *comparison-based recommendation* (**McGinty 2002**):
  - Proposes a variety of query revision strategies that are designed to update the current query as a result of preference-based feedback

User Feedback

- The review process is conceptually simple…
- The user is simply asked to select the item that best matches their needs – form of *positive feedback*.
- Alternatively the user can provide *negative feedback* by selecting the item which best corresponds to items that the user is not interested in.
- Here, focus on positive feedback.

Conversational Recommender System Cycle



Comparison-based Recommendation Algorithm

```
define Comparison-Based-Recommend(Q, CB, k)
begin
  Do
   R ← ItemRecommend(Q, CB, k)
   c_p ← UserReview(R, CB)
   Q ← QueryRevise(Q, c_p, R)
  until UserAccepts(c_p)
end
```

```
define ItemRecommend(Q, CB, k)
begin
  CB' ← sort cases in CB in decreasing order of their sim to Q
  R ← top k cases in CB'
  return R
end
```

```
define UserReview(R, CB)
begin
  c_p ← user selects best case from R
  CB ← CB - R
return c_p
```

```
define QueryRevise(Q, c_p, R)
begin
  R' ← R - {c_p}
  For each f_i ∈ c_p
    Q ← update(Q, f_i, R')
  end For
  return Q
end
```

Query Revision

- The main idea is how the user's preference feedback can be leveraged to inform the subsequent recommendation cycle.
- The goal is to revise the current query based on what can be learned from the user's feedback.
- In the case of positive feedback – where the user selects a preferred case (item), c_p – the goal is to update query features with features from the preferred case that best reflect the user's implicit preferences.
- Query revision strategies:
  - MLT – More Like This
  - pMLT – Partial More Like This
  - wMLT – Weighted More Like This
  - LLT – Less Like This
  - MLT+LLT – More Like This + Less Like This

MLT – More Like This

- Simplest form of query revision
- Query features are assigned the feature values of the preferred case
- Advantage:
  - A partial user query (one in which only a few feature values are specified) can be instantly extended to include all features from the preferred case
- Disadvantages:
  - Not every feature of the preferred case may be preferred by the user
  - MLT may suffer from *overfitting* to user feedback, user may be guided to irrelevant parts of the item space
  - May result in an increased number of recommendation cycles where the user has to examine more cases than necessary

$$\textbf{MLT RULE: update}(Q, f_i, R') : Q.f_i := f_i$$

pMLT – Partial More Like This

- A revised form of MLT.
- Motivation is to reduce overfitting.
- A feature value from the preferred case is only assigned to the query if none of the rejected cases have the same feature value.
- Allows the recommender to focus on the those aspects of the preferred case that are unique in the current cycle.

$$\textbf{pMLT RULE: update}(Q, f_i, R') : Q.f_i := f_i \text{ if } (\neg\exists c_j \in R' : c_j.f_i = f_i)$$

wMLT – Weighted More Like This

- wMLT attempts to weight updated query features based on the degree of confidence that we can attribute to them as genuine user preferences.

- The main idea is that, for each feature, the more feature alternatives that have been presented to the user, the more confident we can be about learning their preference for that feature.

- Based on MLT – the wMLT strategy transfers all features from the preferred case to the query but weights them according to how diverse the feature is among recommendations in the current cycle.

$$\textbf{wMLT RULE: update}(Q, f_i, R') : Q.f_i := < f_i, weight(f_i, R') >$$

$$weight(f_i, R') = \frac{\#\text{ of alternatives to } f_i \text{ in } R'}{|R'|}$$

LLT – Less Like This

- In additional to learning about features that the user is likely to prefer, it may also possible to infer features that the user tends to dislike
- The LLT strategy treats the query as a set of negative preferences and ordering cases during future recommendation cycles by prioritising cases that are most dissimilar to the negative query.
- LLT strategy – if rejected cases all have the same value for a given feature (which is different to the preferred case), then this feature value is added to the negative query.

$$\textbf{LLT RULE: update}(Q, f_i, R') : Q_{neg}.f_i := f_i' \text{ if } (\forall c_j \in R', c_j.f_i = f_i') \wedge (f_i' \neq f_i)$$

pMLT + LLT – Partial More Like This + Less Like This

- Combines pMLT and LLT to learn both positive and negative preferences in a single recommendation cycle.
- The query has two components – a positive and negative component.
- To compute the relevance between this query and a candidate case, compute the similarity between the positive query component and the case and subtract the similarity between the negative component and the case.

$$\textbf{pMLT+LLT RULE:} = \textbf{update}(Q, f_i, R') : \quad \begin{array}{l} Q_{pos}.f_i := f_i \text{ if } (\neg\exists c_j \in R' : c_j.f_i = f_i) \\ \bigwedge \\ Q_{neg}.f_i := f_i' \text{ if } (\forall c_j \in R', c_j.f_i = f_i') \wedge (f_i' \neq f_i) \end{array}$$

Evaluation Methodology

- Compare performance of the five query revision strategies against a benchmark (single shot) similarity-based approach
- PC dataset, containing 120 cases, each describing a unique PC
- Feature set: (1) *PC type*, (2) *processor type*, (3) *processor speed*, (4) *memory*, (5) *disk size*, (6) *display size* and (7) *price*.
- Adopt a leave-one-out evaluation methodology – each case (current) in turn in the dataset is temporarily removed and used in two ways:
  - First the case in the dataset which is the most similar to the current case is used as the *target case*
  - Second the current case serves as the basis for a set of random queries of varying sizes
    - Vary values of q (the number of initial features per query) from 1 to 5
    - Vary values of *k* (the number of cases returned during each recommendation cycle) from 2 to 5
  - 1200 separate queries generated

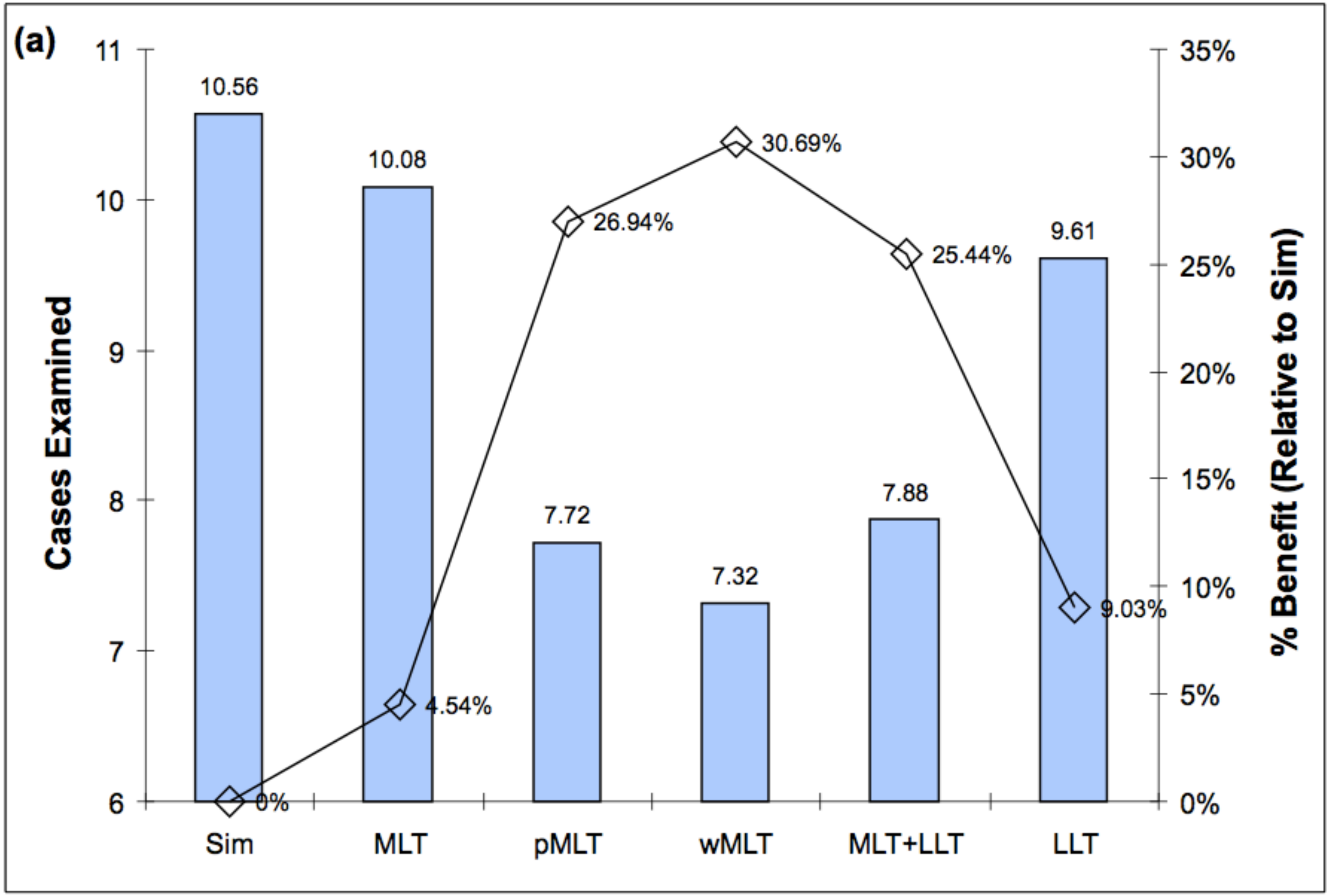
Evaluation Metrics

- Benchmark – standard single-shot similarity-based retrieval approach, rank all cases by decreasing similarity. Compute the average rank (over all queries) of the target case.

- Count the average (over all queries) number of cases the user needs to examine before the target case is presented for each query revision strategy and compare to the benchmark.

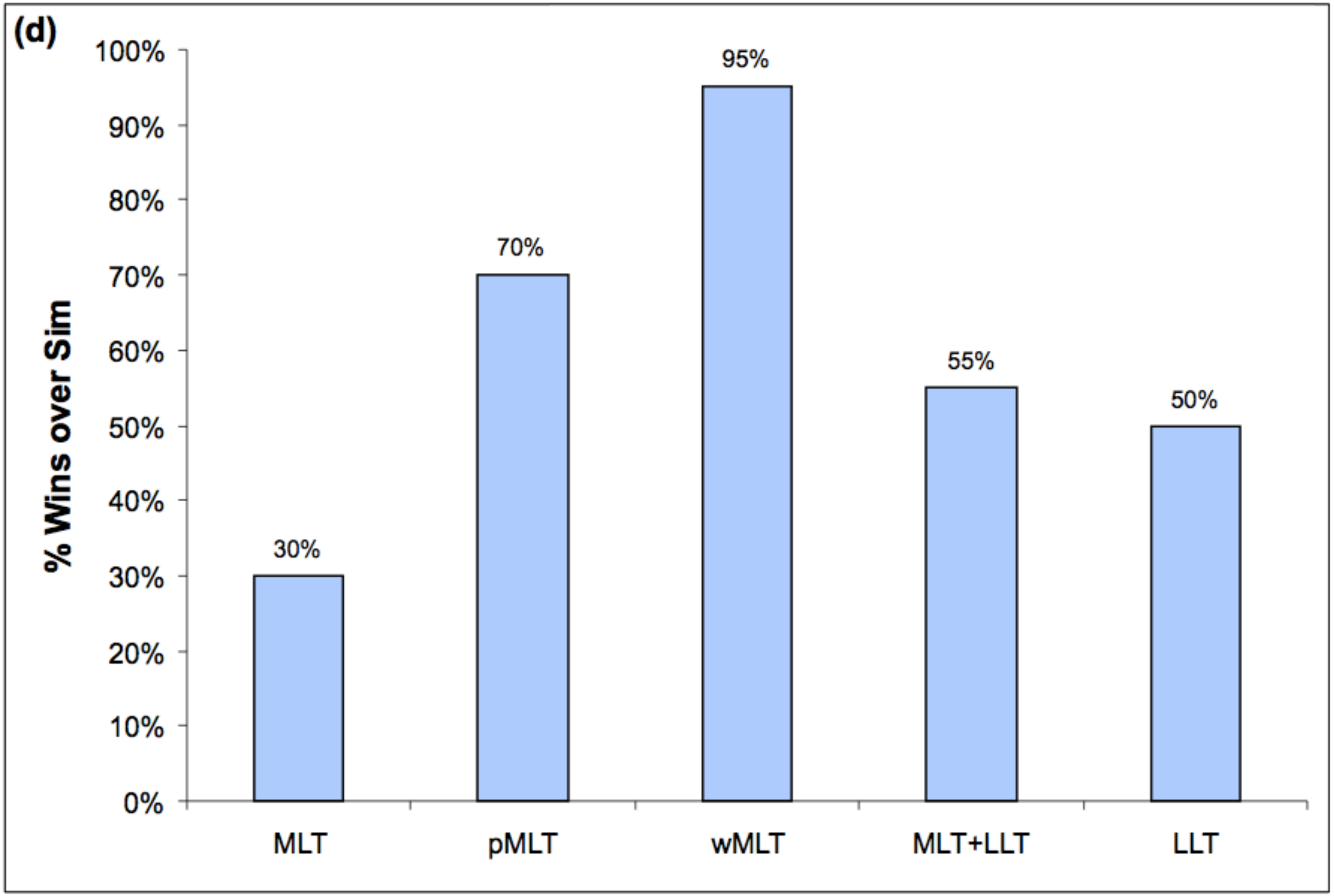
- The number of “wins” for each query revision strategy compared to the benchmark:
  - Count the number of trials where a strategy results in fewer cases being examined before the target case is presented

Results

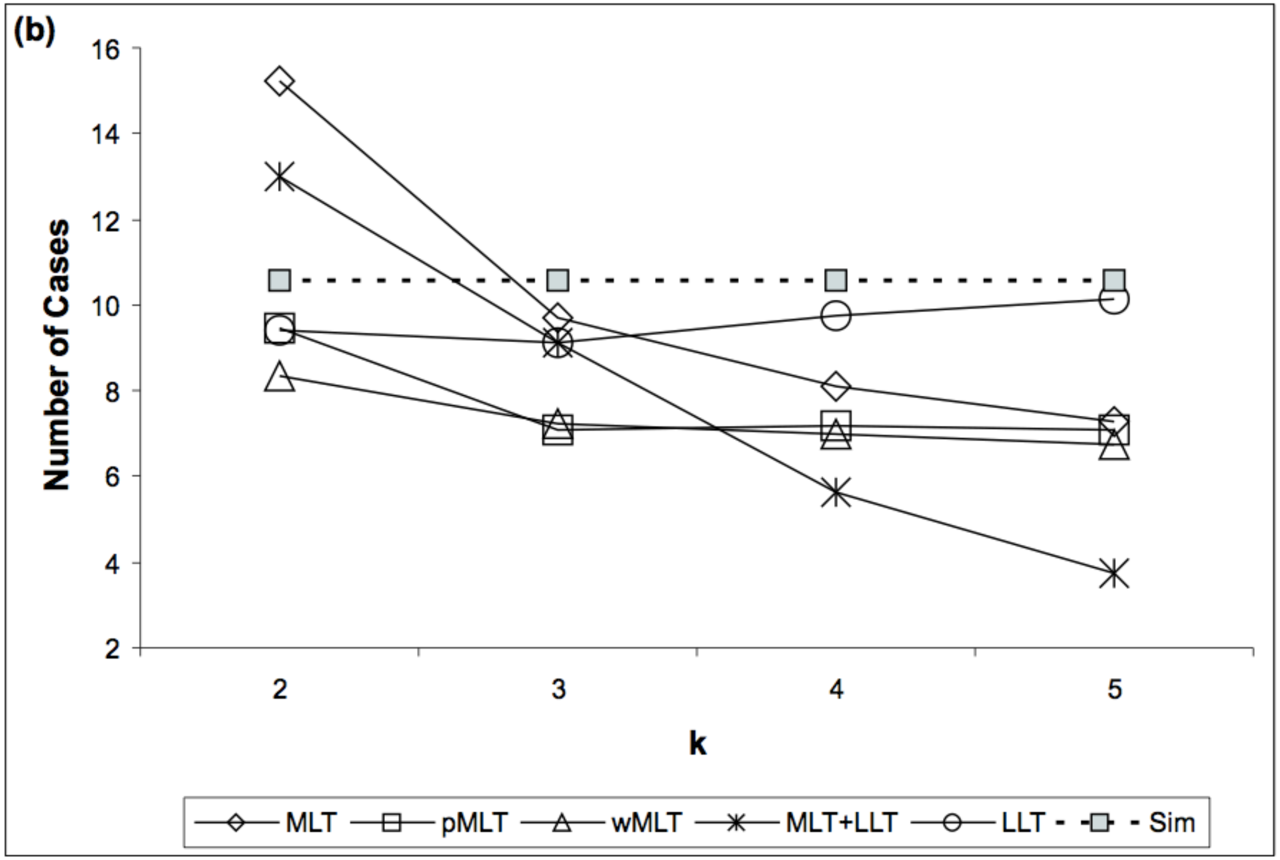

(a)
11
10
9
8
7
6
Cases Examined
10.56
10.08
7.72
7.32
7.88
9.61
0%
4.54%
26.94%
30.69%
25.44%
9.03%
35%
30%
25%
20%
15%
10%
5%
0%
% Benefit (Relative to Sim)
Sim
MLT
pMLT
wMLT
MLT+LLT
LLT

Results

trials where a strategy results in fewer cases being examined before the target case is presented



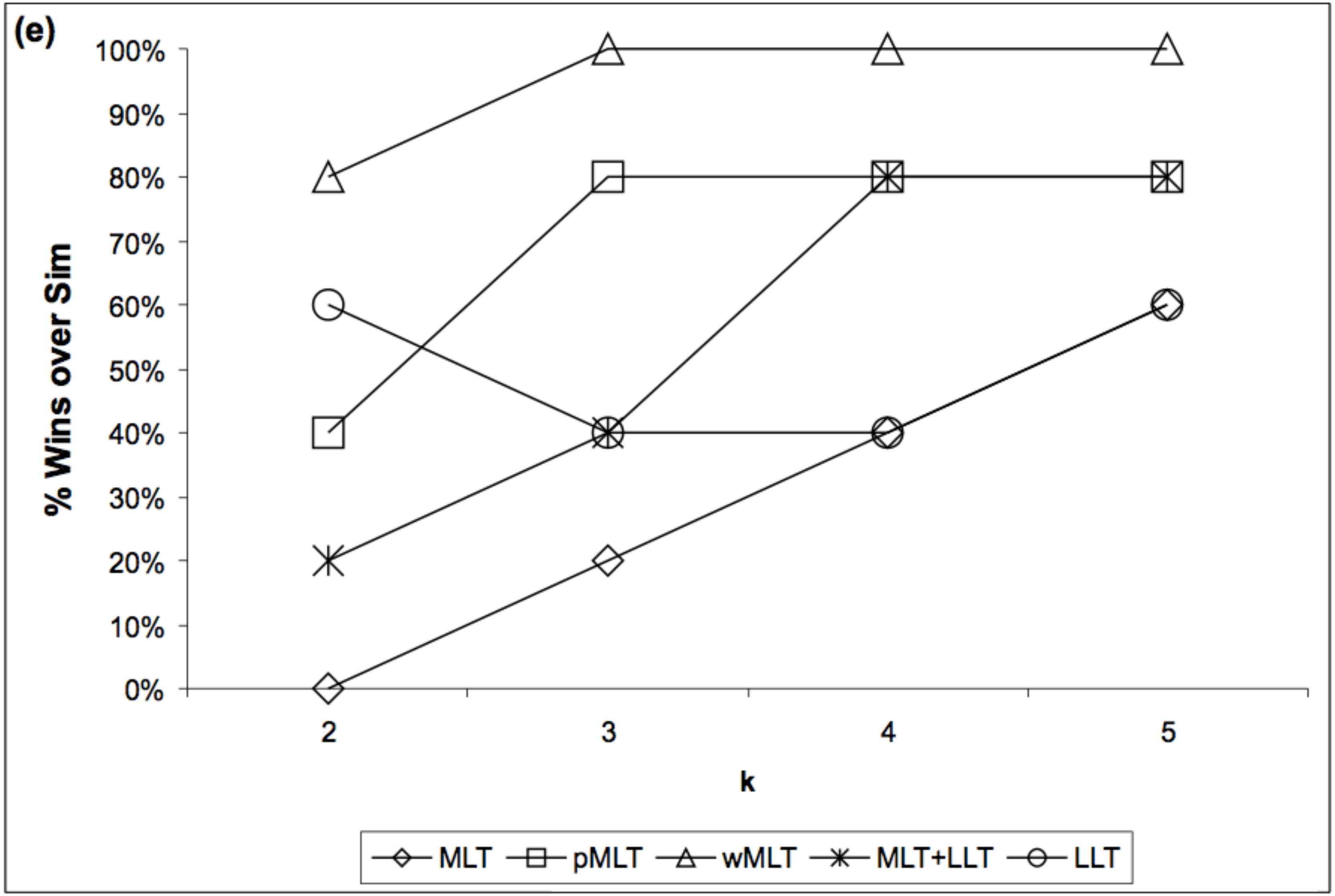
Results



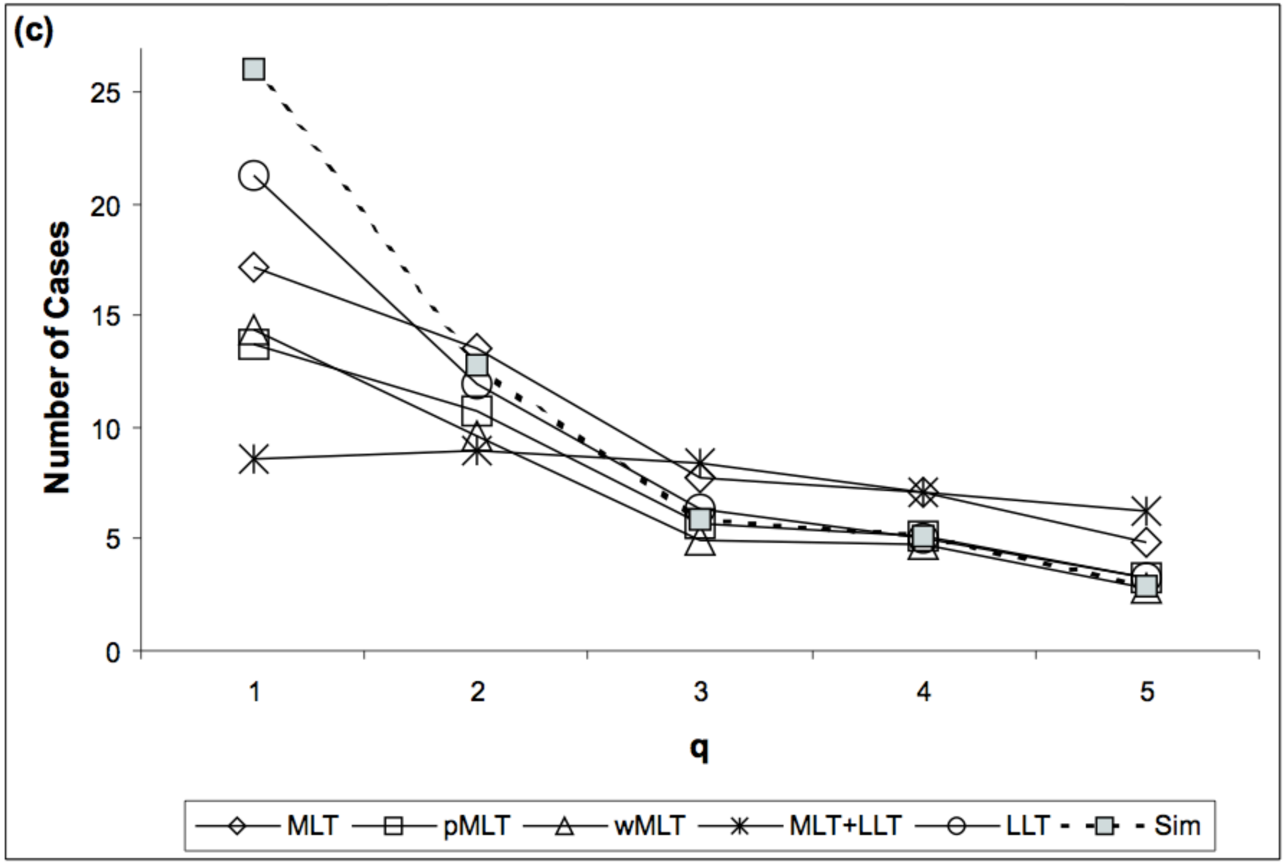
- As k increases the average number of cases examined by the user decreases
- MLT and MLT+LLT lose to Sim (benchmark) for k=2
- MLT+LLT – LLT on its own performs well at k=2, indicating poor performance is due to MLT
- Results indicate that simple MLT is not an effective strategy

Results

trials where a strategy results in fewer cases being examined before the target case is presented



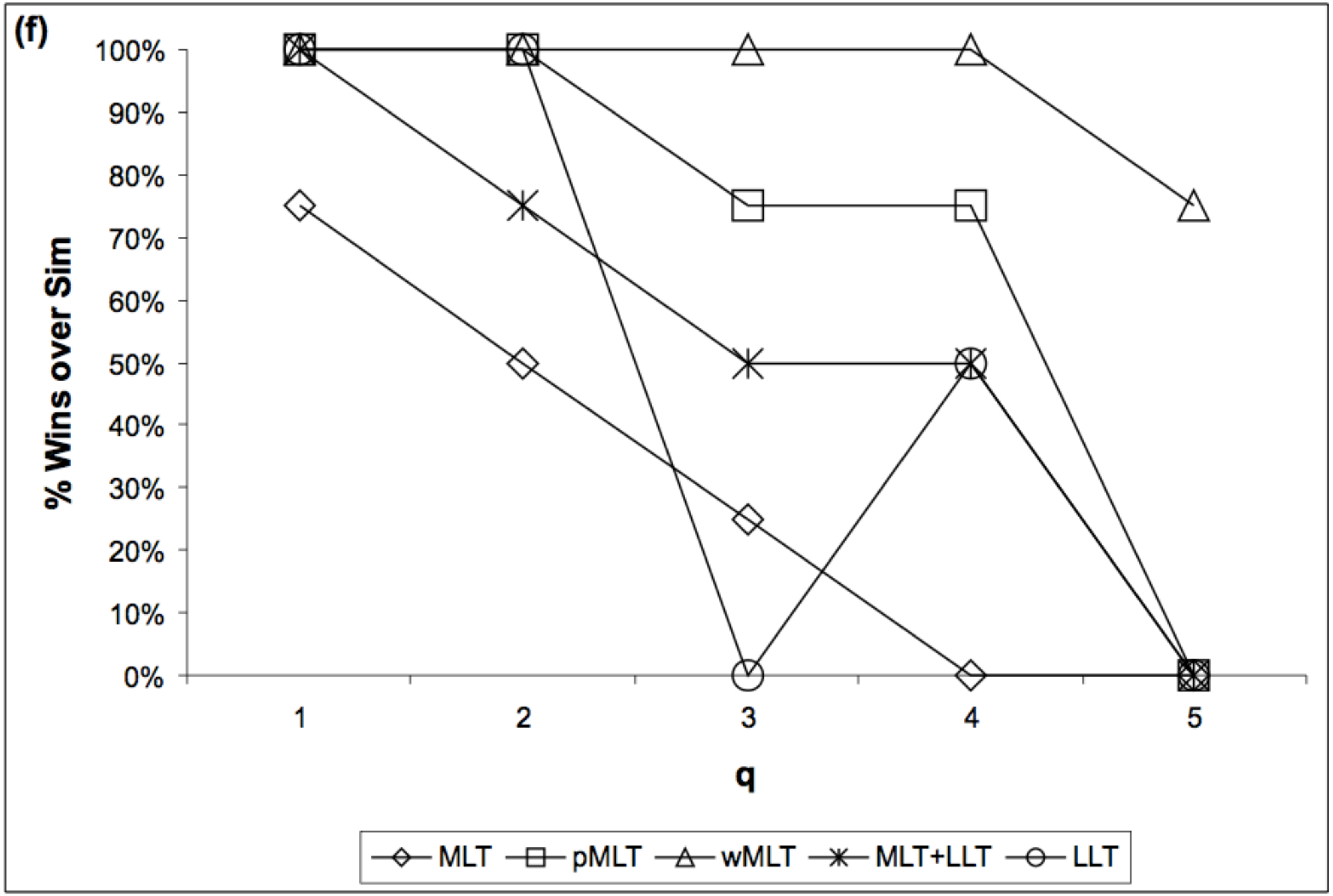
Results



- Significant reductions in number of cases examined as q increases (as expected)
- MLT again performs poorly, losing to Sim for values of q > 2
- pMLT and wMLT - best overall performance, winning outright for q > 2
- wMLT outperforms pMLT indicating value conferred by its weighting technique

Results

trials where a strategy results in fewer cases being examined before the target case is presented



Summary

- Recommender systems:
  - Non-personalised vs. personalised
  - Content/case-based vs. content-free (CF)
  - Single-shot vs. conversational:
    - Conversational – feedback, presentation, explanation, cognitive load…
- No “silver bullet” solution:
  - Dependancies: domain, available data, user base …