Lecture 7: Conversational Recommender Systems for E-Commerce

Derek Bridge

Problems

- The following problems remain, irrespective of whether we use filter-based, similarity-based, utility-based or diversity-enhanced retrieval:
  - seldom are we able to specify all our requirements up-front
  - seldom are we satisfied with the initial set of results
- We've been assuming a single-shot system
  - submit query, view results, end of story
- If not satisfied, our only option is to revise the query and submit again
  - typically with no guidance!
  - can lead to 'stonewalling'

Solution

- A conversational recommender system
  - an iterative approach
  - users can elaborate their requirements as part of an extended recommendation dialog

Conversational systems

- Single-shot systems: independent queries
  - we assumed the user could supply his/her requirements up-front
  - e.g. value-elicitation by form-filling
- Conversational systems: a dialog
  - Navigation-by-asking
    - recommender selects and asks questions
    - value-elicitation: user may answer the questions
  - Navigation-by-proposing
    - recommender makes interim recommendations
    - user provides feedback on these recommendations (e.g. critiques)

Navigation-by-asking: desiderata

- Questions should be few in number
- Questions should have a comprehensible ordering/grouping
- Each question should be comprehensible
- Each question should have low answering cost
- …

Navigation-by-asking

- Let's focus on minimizing the number of questions
- *Statically-defined dialog*
  - will not minimize the number of questions since next question is fixed ➔ insensitive to user's answers to previous questions
- *Dynamically-defined dialog*
  - next question is chosen based on an analysis of the distribution of remaining candidate items
- For simplicity, let's assume filter-based retrieval
  - i.e. exact-matching

Check your intuitions

- Suppose these are the candidate items:

Id	Colour	Size	Weight
1	red	small	light
2	red	small	light
3	red	large	heavy
4	blue	small	heavy
5	blue	small	heavy
6	red	small	light
7	red	small	light
8	blue	small	heavy
9	blue	large	heavy
10	blue	large	medium

- You can ask the user to supply a preferred *colour* or a preferred *size* or a preferred *weight*.
  Which would you ask first?

Check your intuitions, continued

- We'll suppose the user gives us an answer to our first question. In the lecture, delete parts of the table that are no longer relevant:

Id	Colour	Size	Weight
1	red	small	light
2	red	small	light
3	red	large	heavy
4	blue	small	heavy
5	blue	small	heavy
6	red	small	light
7	red	small	light
8	blue	small	heavy
9	blue	large	heavy
10	blue	large	medium

- What would you ask next?

Information gain

- Let *C* be the remaining candidate items
- Suppose attribute *A* has a set of possible values, *V*
  - e.g. for *A* = Colour, *V* = {red, blue}
- Let $C_{A=v}$ be those members of *C* for which *A* = *v*
- The *information gain* for an attribute *A*, *Gain(A)*:

$$Gain(A) = -\sum_{v \in V} \frac{size(C_{A=v})}{size(C)} \times \log\left(\frac{size(C_{A=v})}{size(C)}\right)$$

- Log? Should be $\log_2$ but you can use the button on your calculator labeled log, which is $\log_{10}$. This will not change the outcome here

Worked example

- Let's compute, *Gain(Colour)*:

$$Gain(A) = -\sum_{v \in V} \frac{size(C_{A=v})}{size(C)} \times \log\left(\frac{size(C_{A=v})}{size(C)}\right)$$

Information gain

- Compute *Gain(Size)* and *Gain(Weight)* in your own time
- But here are the answers, so that you can check yours against mine:
  - *Gain(Colour)* = 0.3
  - *Gain(Size)* = 0.14
  - *Gain(Weight)* = 0.41

Question

- Consider our rental property example
- The attributes are
  - Type, Rent, Bdrms, Bthrms, Furnished, Location
- Our definition of *Gain* is probably not suitable for all these attributes
- Where does the problem lie?
- How might we fix it?

Dynamic dialog

- Let *Candidates* be the entire product catalog
- Repeat the following until *Candidates* is small enough to display on the screen or all candidates have the same values for all attributes
  - Compute the information gain of each unasked attribute
  - Choose the attribute with highest information gain
  - Ask the user for his/her preferred value for this attribute
  - Remove from *Candidates* all products which do not have this value for this attribute

Discussion

- Our treatment assumes filter-based retrieval
  - however, a variation has been defined that works for similarity-based/utility-based retrieval
  - S.Schmitt (2002): *simVar: A similarity-influenced question-selection criterion for e-sales dialog*, Artificial Intelligence Review, vol.18(304), pp.195-221
- We have only considered minimizing dialog length
  - it easy to incorporate question costs, if they are known (which they rarely are)
  - comprehensible ordering/grouping might be achievable by incorporating a similarity measure *between questions*
  - if users have the option of declining to answer a question, we have the opportunity to learn answering preferences in order to personalize dialogs

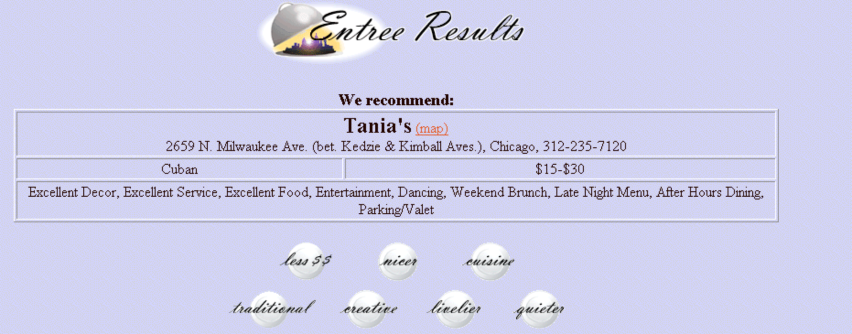
Navigation-by-proposing: intuition

- Asking the user questions, whether up-front (e.g. form-filling) or incrementally (navigation-by-asking) still requires that s/he
  - knows his/her own mind
  - is able to articulate his/her preferences
- On the other hand, if we show the user one or more items (interim recommendations), s/he may more easily be able to say
  - what s/he likes about them
  - what s/he dislikes about them

Critiquing

- *Critiquing* is one form of navigation-by-proposing
- How it works (roughly)
  - the system shows the user an item
  - the user supplies a critique of the item (e.g. *"cheaper"*, *"more bedrooms"*,…)
  - the system retrieves all items that satisfy the critique
  - of these items, it shows the user the one that is most similar to the one being critiqued
- This captures the idea of *"like this but…"*

Entrée: restaurant recommender



Entree Results

We recommend:

Tania's (map)

2659 N. Milwaukee Ave. (bet. Kedzie & Kimball Aves.), Chicago, 312-235-7120

Cuban	$15-$30

Excellent Decor, Excellent Service, Excellent Food, Entertainment, Dancing, Weekend Brunch, Late Night Menu, After Hours Dining, Parking/Valet

Worked example

Id	Address	Type	Bdrms	Bthrms	Rent	Furnished	Location
1	16 Oxford Road	Flat	1	1	265	Yes	Acton
2	2 Heathfield Road	House	3	2	370	Yes	Acton
3	101 Nassau Road	Flat	2	1	271	No	Barnes
4	78 Moscow Road	Flat	3	1	850	Yes	Bayswater

Worked example

- Suppose the system shows the user the following item:

Id	Address	Type	Bdrms	Bthrms	Rent	Furnished	Location
2	2 Heathfield Road	House	3	2	370	Yes	Acton

- The user selects the “cheaper” critique
- So s/he wants to see items that are
  - “like the second item but cheaper”

Worked example

- Since the item has Rent = 370, the user’s critique can be expressed as Rent < 370
- The system finds all items that satisfy the critique
  - `SELECT * FROM Properties WHERE Rent < 370;`

Id	Address	Type	Bdrms	Bthrms	Rent	Furnished	Location
1	16 Oxford Road	Flat	1	1	265	Yes	Acton
3	101 Nassau Road	Flat	2	1	271	No	Barnes

- Call these the *Candidates*

Worked example

- For each candidate item *i*, compute *sim(s, i)* where *s* is the selected item

2	2 Heathfield Road	House	3	2	370	Yes	Acton
1	16 Oxford Road	Flat	1	1	265	Yes	Acton
sim(id2, id1) = Σ:		0	0.25	0.875	0.838	1	1

2	2 Heathfield Road	House	3	2	370	Yes	Acton
3	101 Nassau Road	Flat	2	1	271	No	Barnes
sim(id2, id3) = Σ:		0	0.875	0.875	0.848	0	0.6

sim(id2, id1) = 3.963 *sim(id2, id3) = 3.198*

Worked example

- Show the user the highest scoring item:

Id	Address	Type	Bdrms	Bthrms	Rent	Furnished	Location
1	16 Oxford Road	Flat	1	1	265	Yes	Acton

- "like this but cheaper"!

Critiquing: variation

- This variant might give a more efficient dialog:
  - The system shows the user *k* items (*k* > *1*, e.g. *k* = *3*)
  - The user selects one of the items, the one that comes closest to what s/he wants
  - The user supplies a critique of the selected item
  - The system retrieves all items that satisfy the critique
  - Of these items, the system shows the user the *k* that are most similar to the one being critiqued
    - (Another variant: use Bounded Greedy Selection. Why?)

Entry points

- But what item(s) do you start with?
- Named entry:
  - user picks an item s/he knows about
- Search entry:
  - user fills in a form with some initial requirements
- Prototype entry:
  - system selects a diverse set of *k* items from the product catalog
- Navigation-by-asking entry:
  - the ExpertClerk system, H. Shimazu (2002): *ExpertClerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops*, Artificial Intelligence Review, vol.18(3-4), pp.223-244

Technical issues

- If a critique is unsatisfiable, it ought to be disabled so the user cannot select it
- Critiquing involves filter-based retrieval, then similarity-based retrieval (possibly diversity-enhanced). Is it right to use filter-based retrieval?
- If the product space is dense, critiques may result in only slow change and differences that are not perceived as significant
  - on the other hand, attempts to remedy this may make it impossible to reach some items
- No one knows whether critiques should cumulate

Slightly broader issues

- Designers have to anticipate the critiques to offer
- Some critiques may not be expressible in terms of individual attributes
  - especially lifestyle characteristics
- Should we offer 'compound critiques', which change more than one attribute at a time?
  - might help the user to see trade-offs
  - does it solve the problem above?
  - but which 'compound critiques' should we offer
    - too many possible compound critiques to show them all
    - maybe the system can select the most useful ones dynamically?

Much broader issues

- In both navigation-by-answering and navigation-by-proposing
  - the user has to have quite a lot of knowledge/understanding
- When input modalilities are more limited (e.g. handheld devices), critiquing and navigation-by-asking may impose an unreasonable burden
  - there are other forms of navigation-by-proposing requiring less user input
- In both navigation-by-answering and navigation-by-proposing
  - there has been a fixation with minimizing dialog length
  - Why might this be wrong? In other words, why might a user prefer a longer dialog than is strictly necessary?