NAVIGATION-BY-PREFERENCE: A NEW CONVERSATIONAL RECOMMENDER WITH PREFERENCE-BASED FEEDBACK

Group 5

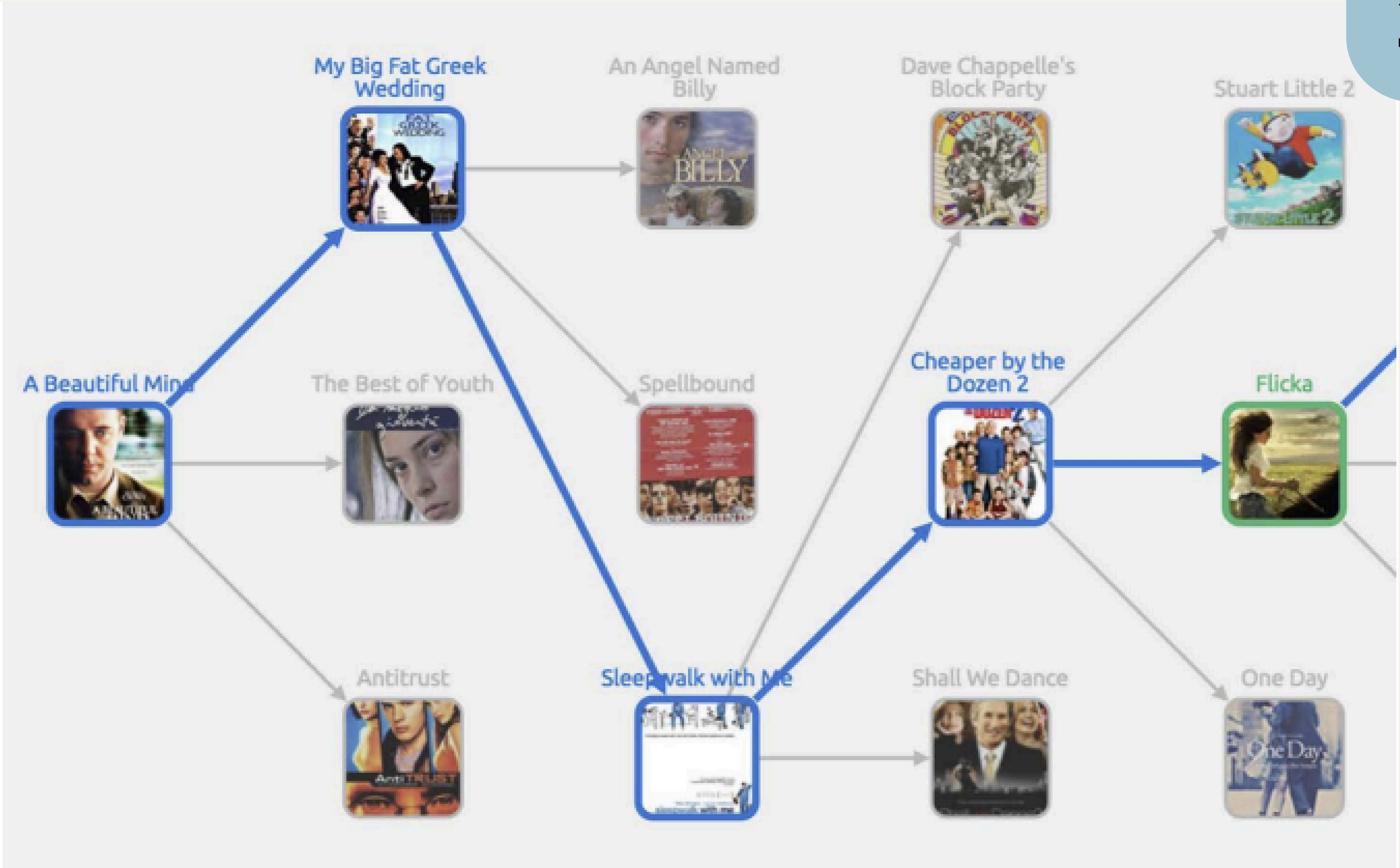
DAIICT | 2024

MOTIVATION

- Traditional systems offer a single shot of recommendations.
- This can be limiting if users aren't satisfied with their top n recommendation
- (Eg = users with different short term and long term preferences)
- Hence, to overcome the drawback of single shot recommendations we use conversational recommenders.

INTRODUCTION

- In a conversational recommendation system, a set of n recommendation is suggested
- Then the user is asked to choose one of the recommendations
- This item becomes query for next cycle
- May take several cycles


My Big Fat Greek Wedding
An Angel Named Billy
Dave Chappelle's Block Party
Stuart Little 2
A Beautiful Mind
The Best of Youth
Spellbound
Cheaper by the Dozen 2
Flicka
Antitrust
Sleepwalk with Me
Shall We Dance
One Day

NOVELTIES

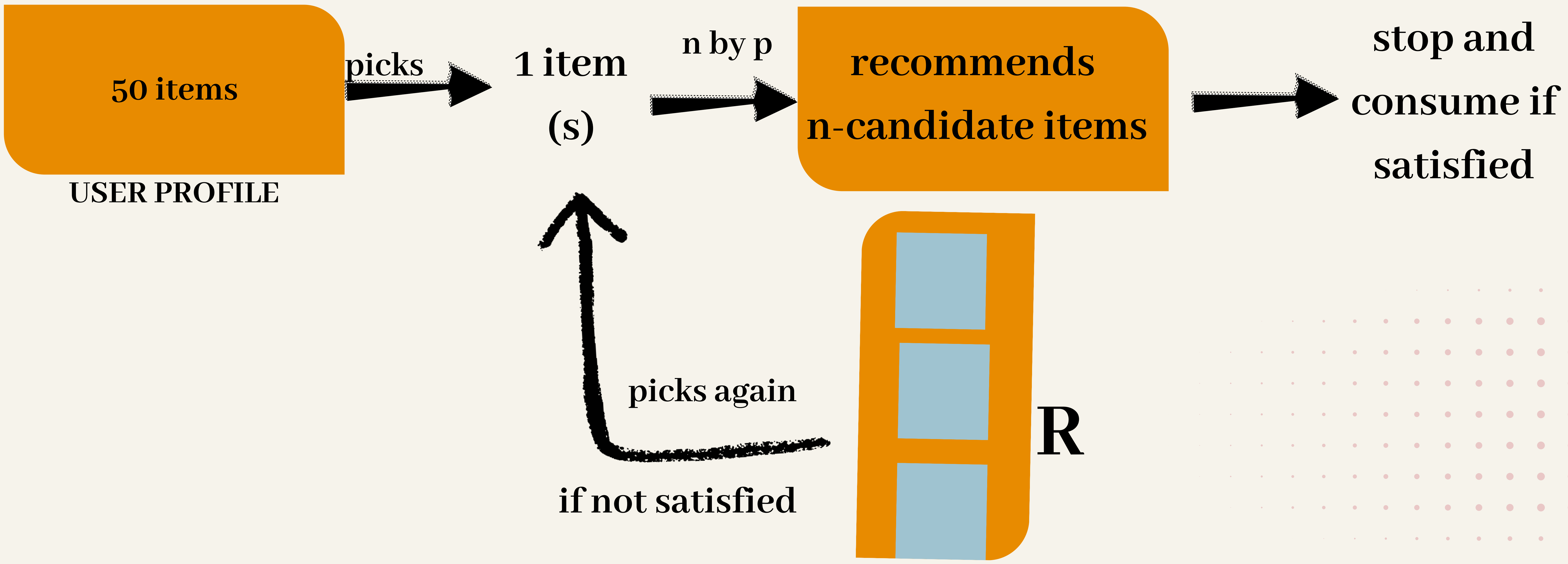
- Works for unstructured feature values
- Takes into account the combination of short term as well as the long term goals instead of just short term goals
- Evaluation done combining both offline experiments and user trial

NAVIGATION BY PREFERENCE

- It is a conversational recommendation system that used feedback based on user preferences (i.e short term /long term).
- The most important advantage of this system is its ability to work in domains that have unstructured item representations.
- It does not involve learning of models from the training sets of session data.

NAVIGATION BY PREFERENCE

High level working of n-by-p



NOTATIONS

P - user profile of active user (set of items liked by user) $P \subseteq \mathbb{I}$

$\mathbb{I}$ - set of all items (movies)

I - all items that might me recommended to user (candidate items).

$$I = \{i : i \in \mathbb{I} \setminus P\}$$

fi - set of features (tags) for item i

$$sim(i, j) = \frac{|fi \cap fj|}{|fi \cup fj|}$$

Ni - set of related items (i.e candidate items that are neighbours of i)

$$Ni = \{j \in I, \quad j, i : sim(i, j) > \theta\}$$

WHICH ITEMS TO RECOMMEND IN NEXT CYCLE ?

- Candidates that are similar to s i.e Ns.
- But not every member of Ns should be recommended

WHY ?

- It is because we will not recommend an item more than once in whole dialog.
- Also the items not selected in previous round might be treated as negative feedback

- Since the user's most recent choice is $s \in R$
- To reflect negative feedback (i.e $R \setminus \{s\}$), we have to discard some members of Ns.
- We refer to this subset of Ns as $S \subseteq Ns$. This is called the selection consistent candidate set.
- $L = \bigcup_{i \in P} Ni$ (i.e User's long term preferences)

(candidate items that are neighbours of each item i in user profile)

GOALS OF SET "R"

- To reflect long term preferences of user
  (measure how much Ni overlaps with L)

- To reflect short term preferences of user
  (measure how much Ni overlaps with S as a whole)

- To ensure diversity
  (make sure that Ni covers parts of S that were not covered in previous recommendation cycles)

TYPES

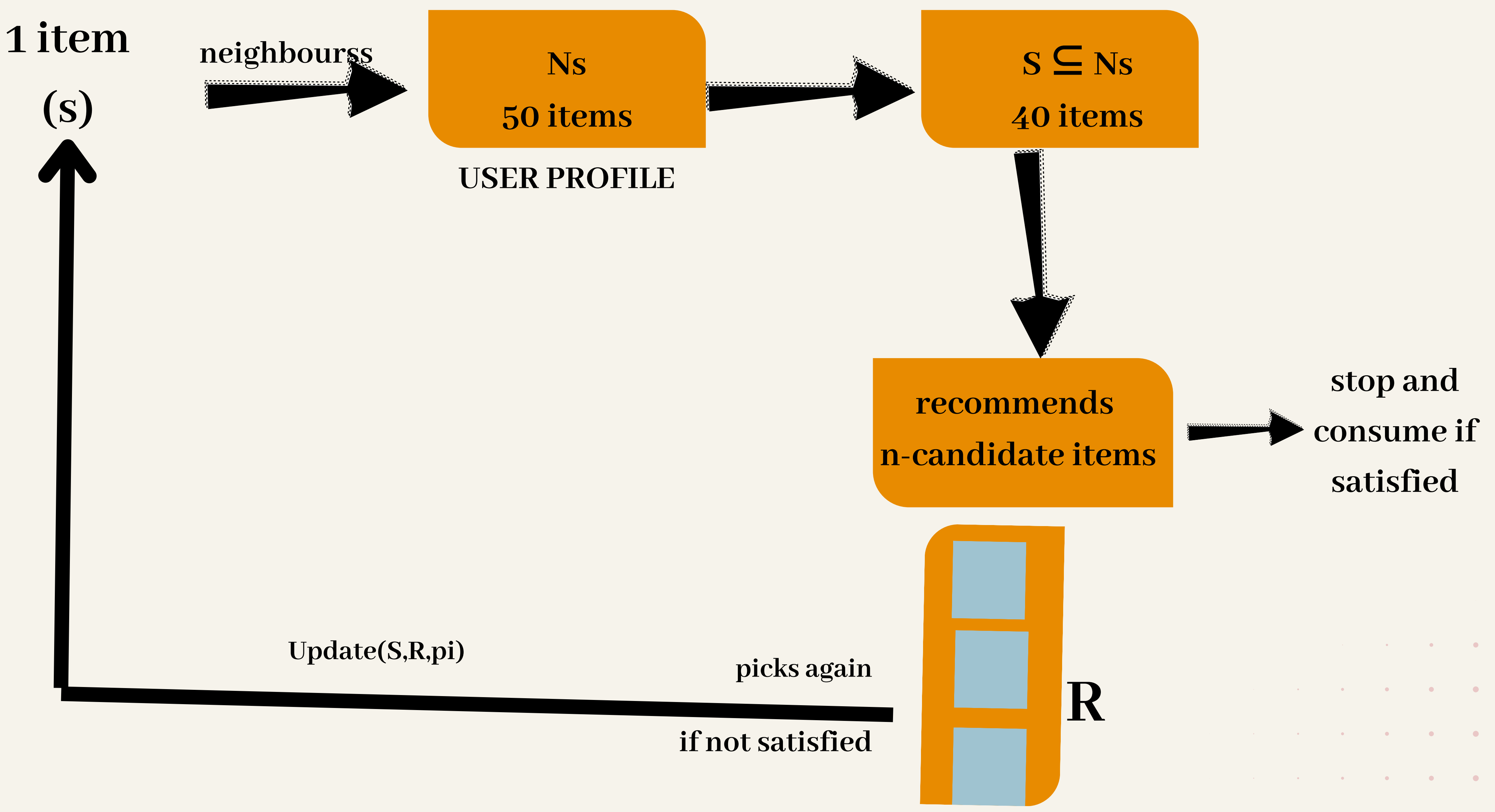
n-by-i-p

In this, only feedback from the most recent cycle of recommendation affects the next cycle.

n-by-c-p

In this, feedback from the earlier cycles of recommendation also affects the next cycle.

N-BY-I-P



Algorithm 1 Navigation-by-Immediate-Preference (*n-by-i-p*)

Input: s: seed item, chosen by the user
L: candidate items that are neighbours of items in P
π: update policy
n: number of recommendations per cycle
Output: $i \in I$, a candidate item to consume

```
S ← N_s
Tabu ← ∅
while |S| > n do
    R ← RECOMMEND(S, L, n)
    s, a ← user chooses s ∈ R and a ∈ {STOP, CONTINUE}
    if a = STOP then
        return s
    S ← UPDATE(s, R \ {s}, π)
    Tabu ← Tabu ∪ R
    S ← S \ Tabu
```

https://github.com/shgupta1461/RPIP_Group5/blob/main/RPIP_nbyip.ipynb

RECOMMENDING

- Greedily selects n items from S with highest score

$$\text{score}(i, S, L, R) = (1 - \eta) \cdot \text{ovrlp}(i, S, R) + \eta \cdot \text{ovrlp}(i, L \setminus S, R)$$

prevents double counting of item in both S and L

$$\text{ovrlp}(i, X, R) = \frac{2 \cdot |(N_i \setminus \text{cov}(X, R)) \cap X|}{|N_i \setminus \text{cov}(X, R)| + |X \setminus \text{cov}(X, R)|}$$

to exclude items that are already covered by neighbours of R

No explicit reference to features

HOW TO UPDATE SET "S"

- **Open**

  It simply means that we don't remove any members of set Ns.

  i.e we don't take negative feedback into account.

- **Strict**

  If any member of Ns is neighbour of any member of R′ then we remove that member from Ns and the updated set Ns is assigned to S

  i.e we don't recommend items from R′ in next recommendation

- **Relaxed**

  We remove member of Ns if it is neighbour of every (all) member of R′

HOW TO UPDATE SET "S"

- **Mean**

  If similarity of member of Ns with s is less than MEAN of similarity of Ns with Rejected members then we remove those members from Ns

  $$S \leftarrow Ns \setminus \{j \in Ns : sim(j,s) < mean(Sims(j))\}$$

- **Max**

  If similarity of member of Ns with s is less than MAX of similarity of Ns with Rejected members then we remove those members from Ns

  $$S \leftarrow Ns \setminus \{j \in Ns : sim(j,s) < max(Sims(j))\}$$

PROBLEM WITH N-BY-I-P

Current set R may contain items which are not related to previously selected items by users.

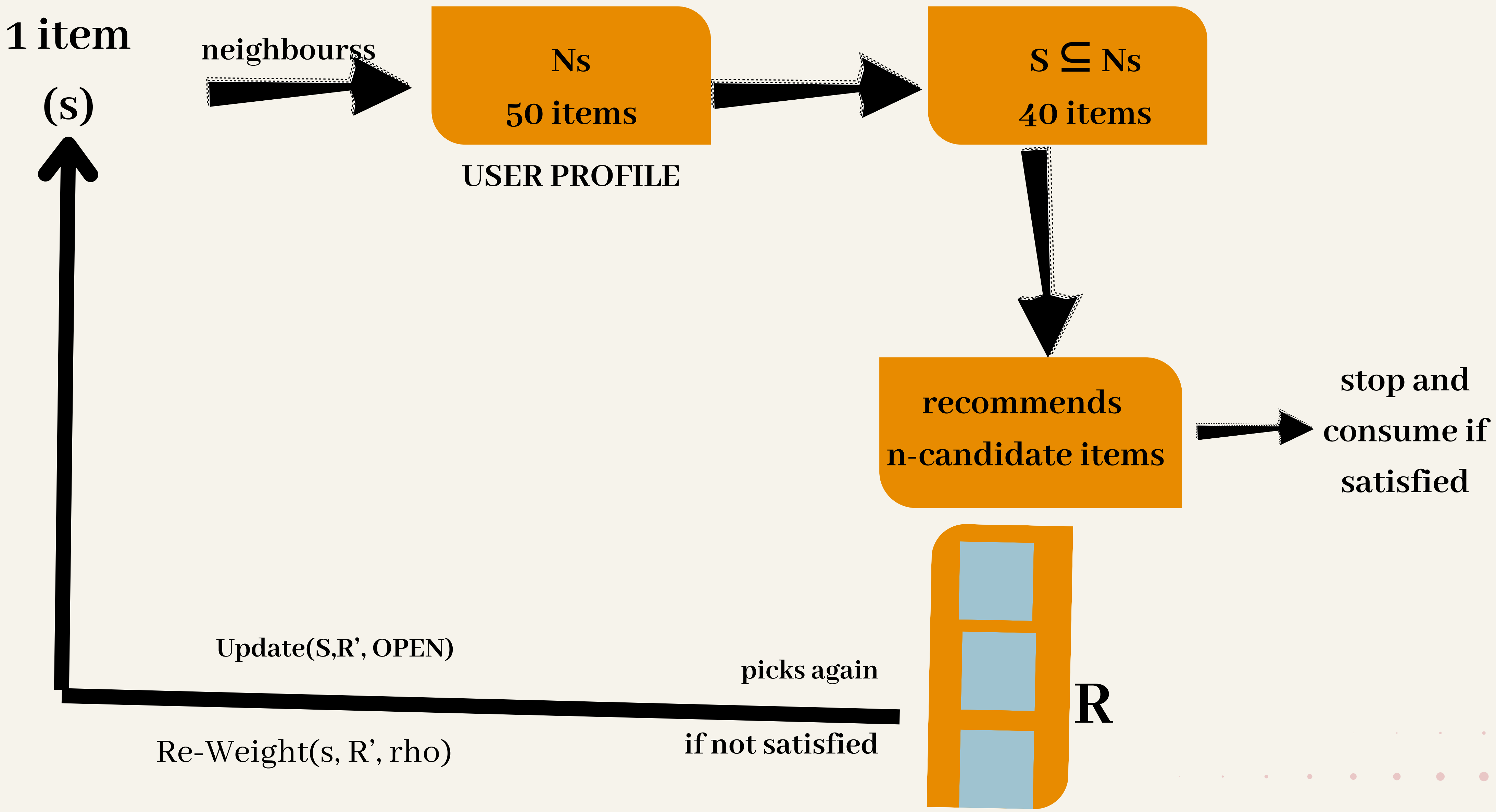
Current set R may contain items which are related to previously rejected items by users.

N-BY-C-P

Algorithm 3 Navigation-by-Cumulative-Preference (*n-by-c-p*)

Input: s: seed item, chosen by the user
L: candidate items that are neighbours of items in P
ρ: re-weighting policy
n: number of recommendations per cycle
Output: $i \in I$, a candidate item to consume

$S \leftarrow N_s$
$Tabu \leftarrow \varnothing$
$\textsc{Reweight}(s, \varnothing, \rho)$
while $|S| > n$ **do**
  $R \leftarrow \textsc{Recommend}(S, L, n)$
  $s, a \leftarrow$ user chooses $s \in R$ and $a \in \{STOP, CONTINUE\}$
  if $a = STOP$ **then**
    return s
  $S \leftarrow \textsc{Update}(s, R \setminus \{s\}, \pi = Open)$
  $\textsc{Reweight}(s, R \setminus \{s\}, \rho)$
  $Tabu \leftarrow Tabu \cup R$
  $S \leftarrow S \setminus Tabu$



https://github.com/shgupta1461/RPIP_Group5/blob/main/RPIP_nbycp.ipynb

RECOMMENDING

- Greedily selects n items from S with highest score

$$wscore(i, S, L, R) = (1 - \eta) \cdot wovrlp(i, S, R) + \eta \cdot wovrlp(i, L \setminus S, R)$$

prevents double counting of item in both S and L

$$wovrlp(i, X, R) = \frac{2 \cdot \sum_{j \in (Ni \setminus cov(X,R)) \cap X} wj}{|Ni \setminus cov(X, R)| + |X \setminus cov(X, R)|}$$

insted of counting items directly, it counts the weights assigned to each item.

No explicit reference to features

RE-WEIGHTING

$w_i \leftarrow w_i + \Delta w_i \quad \forall i \in I$ (all candidate items)

$$C_{ij} = \begin{cases} 1 & ; i \in N_s \\ 0 & ; \text{otherwise} \end{cases}$$

check whether i and neighbours of j are related

Δw_i is Increased - when i is related to item just selected by user (C_{is})

Δw_i is Decreased - when i is related to item just rejected by user (C_{ij})

RE-WEIGHTING

ρ = *Directional (Direc)*:

$$\Delta w_i = C_{is} - \sum_{j \in R'} C_{ij}$$

Only considers whether i is a neighbour of s or members of R'.

ρ = *Similarity (Sim)*:

$$\Delta w_i = C_{is} \cdot \mathrm{sim}(i, s) - \sum_{j \in R'} C_{ij} \cdot \mathrm{sim}(i, j)$$

Considers similarities when i is a neighbour of s or members of R'.

ρ = *Similarity-Mean (Smean)*:

$$\Delta w_i = C_{is} \cdot \mathrm{sim}(i, s) - \mathrm{mean}(\{C_{ij} \cdot \mathrm{sim}(i, j) : j \in R'\})$$

Considers similarity when i is a neighbour of s, and the mean similarity when i is a neighbour of members of R'.

ρ = *Similarity-Max (Smax)*:

$$\Delta w_i = C_{is} \cdot \mathrm{sim}(i, s) - \max(\{C_{ij} \cdot \mathrm{sim}(i, j) : j \in R'\})$$

Considers similarity when i is a neighbour of s, and the maximum similarity when i is a neighbour of members of R'.

ρ = *Recency (Rcy)*:

$$\Delta w_i = C_{is} \cdot \mathrm{sim}(i, s)^{1/d} - \sum_{j \in R'} C_{ij} \cdot \mathrm{sim}(i, j)^{1/d}$$

As per *Sim* above, but with updates counting more for later recommendations.

ρ = *Recency-Mean (Rmean)*:

$$\Delta w_i = C_{is} \cdot \mathrm{sim}(i, s)^{1/d} - \mathrm{mean}(\{C_{ij} \cdot \mathrm{sim}(i, j)^{1/d} : j \in R'\})$$

As per *Smean* above, but with updates counting more for later recommendations.

ρ = *Recency-Max (Rmax)*:

$$\Delta w_i = C_{is} \cdot \mathrm{sim}(i, s)^{1/d} - \max(\{C_{ij} \cdot \mathrm{sim}(i, j)^{1/d} : j \in R'\})$$

As per *Smax* above, but with updates counting more for later recommendations.

NOTION OF 'JUMP'

- It is possible that no item in current set R suits to user but the one from previous cycle is more suitable.
- User can select either to consume that previous item or create a new set of recommendations based on that item.

OFFLINE EXPERIMENTS

The goal for the Offline Experiments conducted are:

- "n-by-i-p" versus "n-by-c-p": These represent different ways of considering user feedback, either focusing on the most recent feedback or all feedback across dialog cycles.
- Different update and re-weighting policies for handling negative feedback.
- The influence of a parameter called η, which controls the balance between short-term and long-term user preferences.
- These experiments were used to decide which configurations to use in user trial.

EXPERIMENT SETTINGS

- The hetrec dataset included over 2000 users, 6000 movies, 80000 keywords and half a million ratings.
- 500 randomly chosen users are taken and profiles were created for users based on movies they liked (rated 4 or 5).
- Simulated dialogs were created between users and different configurations of the n-by-p system (maximum cycles for recommendations =15).

HOW TO CAPTURE SHORT TERM PRFERENCES

- Randomly choose a target item t from Ns

↓

- Recommend n-most similar items to target t

↓

- Stop the simulated dialog when $t \in R$
  (i.e target item is one of the recommendation.

↓

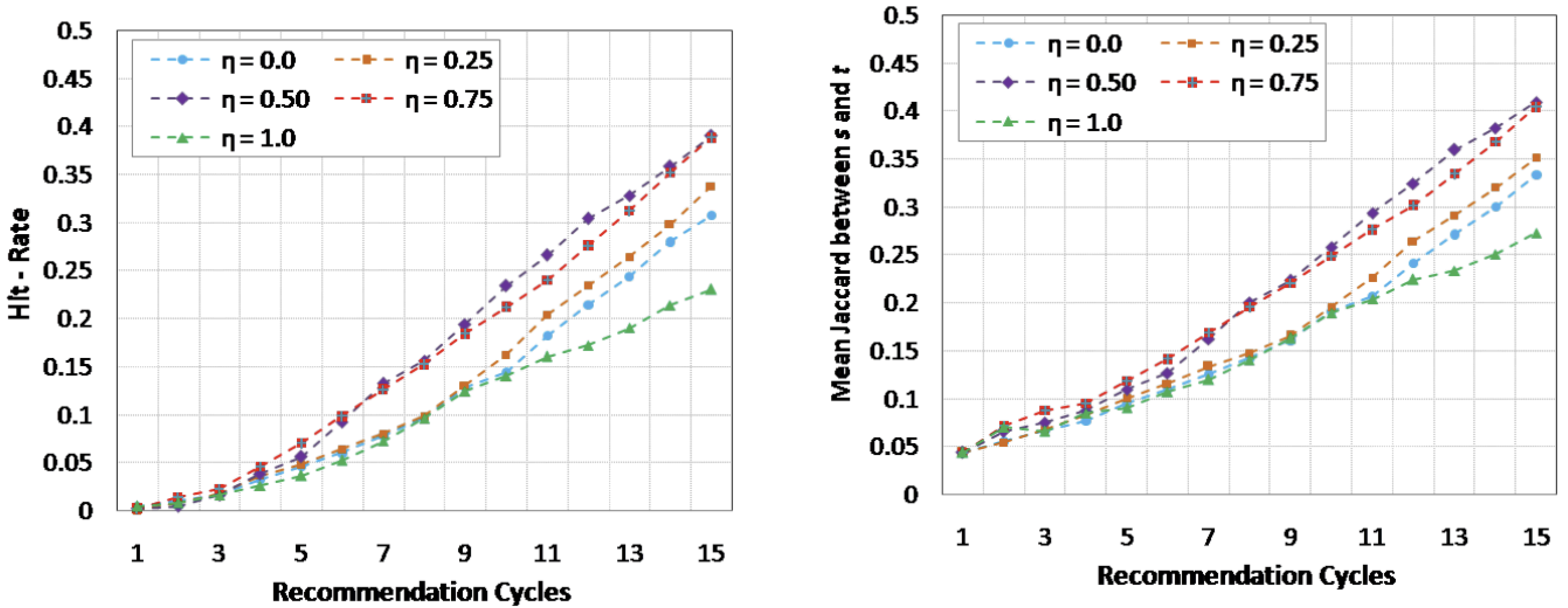
Measure Hit rate for each cycle

EXPERIMENT METRICS

- **Hit-rate: The proportion of simulated users who found their target item within 15 cycles.**
- **Jaccard similarity: A measure of similarity between the item selected by the simulated user and their target.**
- **Diversity and surprise: Metrics indicating the variety and unexpectedness of recommendations (Div_cont and S_cont).**

EXPERIMENT RESULTS

	n-by-i-p					n-by-c-p						
η	*Strict*	*Relaxed*	*Open*	*Mean*	*Max*	*Direc*	*Sim*	*Smean*	*Smax*	*Rcy*	*Rmean*	*Rmax*
0.00	0.102	0.042	0.204	0.146	0.098	0.112	0.070	0.308	0.282	0.028	0.204	0.174
0.25	0.058	*0.034*	0.152	0.072	0.044	*0.112*	0.040	*0.338*	*0.270*	*0.016*	*0.196*	*0.158*
0.50	0.032	*0.046*	0.036	0.038	0.050	*0.084*	0.040	**0.390**	*0.250*	*0.018*	*0.222*	*0.154*
0.75	0.016	*0.038*	0.014	0.014	0.042	0.072	*0.042*	0.388	*0.266*	*0.018*	*0.162*	*0.134*
1.00	0.012	*0.030*	0.010	0.024	0.042	0.070	0.040	0.230	0.220	*0.020*	0.112	0.122



(a) Hit-rate in each cycle

(b) Mean Jaccard similarity in each cycle between the user's selected item and their target item

Figure 2: Results per cycle for *n-by-c-p* with $\rho = Smean$ and different values of η for *random* targets.

USER TRIAL

Purpose

To evaluate the effect of incorporating long-term preferences alongside short-term preferences in recommendation accuracy and user satisfaction.

Systems Compared

- Smean@0.5: Utilizes a mix of short-term and long-term preferences ($\eta = 0.5$) as the best-performing setup from previous offline experiments.
- Smean@0.0 Baseline: Similar to Smean@0.5 but exclusively focuses on short-term preferences ($\eta = 0.0$), serving as the control group.

TRIAL DATA

- A total of 102 people participated in trial.
- Trial used a subset of the dataset from previous offline experiments.
- Featured 1851 movies released between 2000 and 2011, to maximize participant familiarity.
- Random user assignment to either Smean@0.5 or Smean@0.0.
- Goal: Assess how the integration of long-term preferences affects user interactions and system performance.

USER TRIAL PROTOCOL

- Participants begin by selecting 10 movies they like, establishing their long-term preferences.

↓

- Users select a movie from their profile they think is best to watch with a selected family member with differing tastes.

↓

- Users select from a list of eight family members for whom they think movie preferences differ.
- They then choose a movie they believe both would enjoy.
- Process may be repeated up to three times to refine the ephemeral goal.

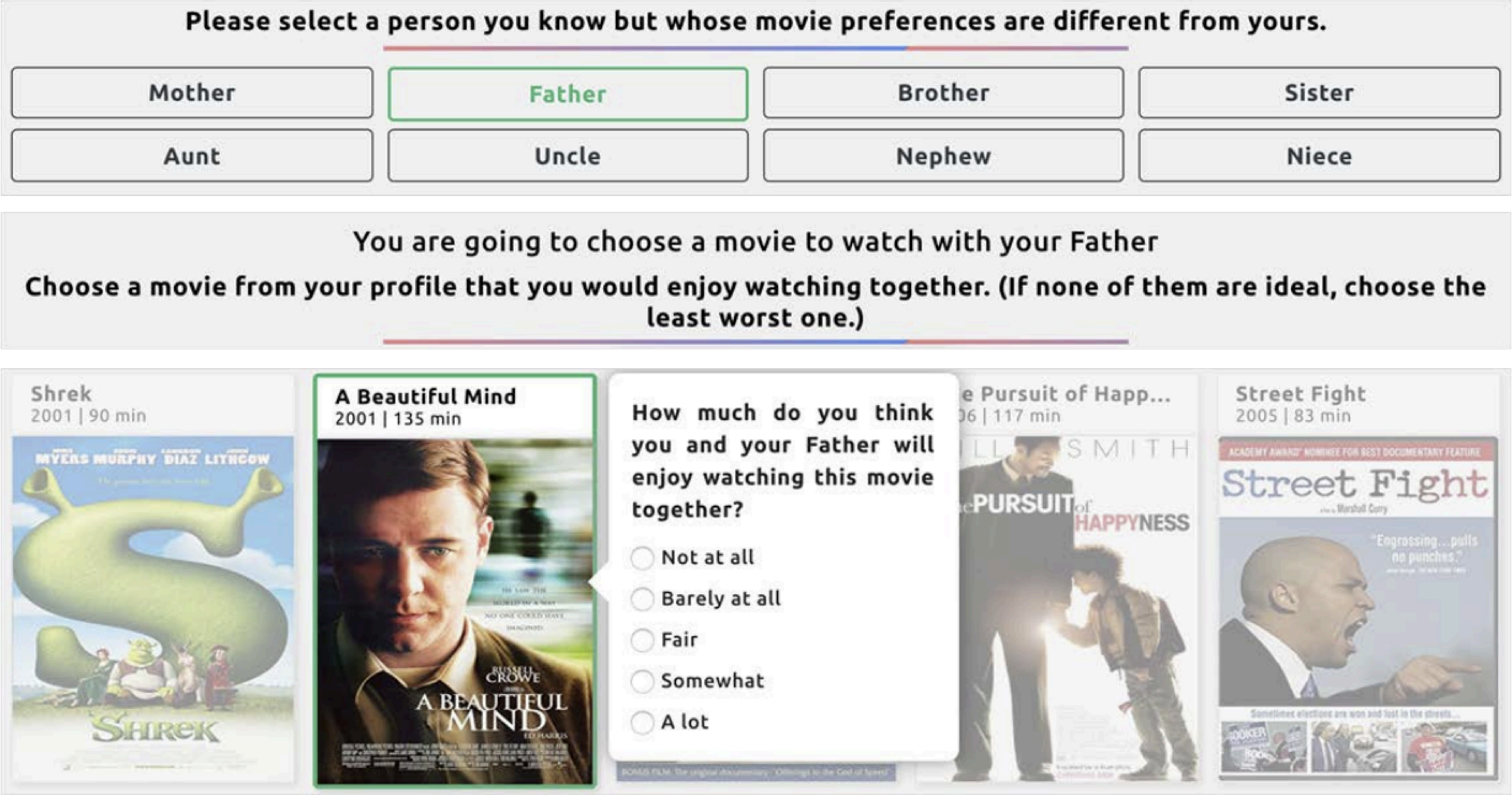


Figure 3: Parts of screenshots showing selection of a companion and a seed movie.

TRIAL

- Eight-cycle dialog where the system presents three movie recommendations in each cycle.
- After eight cycles, users are presented with a tree of 24 movies.
- They select the movie believed to best suit the viewing with their companion.
- Maintains consistency across trials, allowing for fair comparison despite possible discovery of an 'ideal' movie early in the session.

USER EXPERIMENT

- **Evaluation Metrics:**

1. **Familiarity: Whether users had seen the movie before.**
2. **Relevance: Users' judgment on potential enjoyment of the movie with a companion.**
3. **Serendipity: Surprise element of the recommendation.**
4. **Effectiveness: Helpfulness of the recommendations.**
5. **Satisfaction: Overall enjoyment of using the system.**

- **Statistical Analysis:**

One-sided Z-test and t-test used to determine significance of differences.

USER TRIAL RESULTS

Responses to survey questions in the user trial.

User's Response	Smean@0.5				Smean@0.0			
	Relevance	*Serendipity*	*Effectiveness*	*Satisfaction*	*Relevance*	*Serendipity*	*Effectiveness*	*Satisfaction*
Not at all	0	2	0	1	3	5	5	2
Barely at all	1	4	5	5	7	4	4	5
Fair	11	12	14	12	16	21	17	14
Somewhat	18	19	21	20	12	12	16	18
A lot	21	14	11	13	13	9	9	12

IMPROVEMENT FORMULA

Formula:

$$\text{improvement} = \frac{\sum_{v=1}^{5} v \cdot \omega_v - \sum_{v=1}^{5} v \cdot \alpha_v}{\sum_{v=1}^{5} 5 \cdot \omega_v - \sum_{v=1}^{5} v \cdot \alpha_v}$$

- **Measures how much the system improved the satisfaction from the initial movie to the recommended final movie.**
- **A higher value indicates more significant improvement relative to the best possible outcome.**

USER EFFORT

Measure of effort	*Smean@0.5*	*Smean@0.0*
Nodes displayed	26.85	29.56
Node mouse-overs	19.28	22.84
Edge mouse-overs	10.03	11.92
Cycles	5.36	4.12
Time taken (secs.)	251.93	300.58

All values are averaged over participants who liked their final selected movie Somewhat or A lot.

CONCLUSIONS

- Preference based feedback does not use features explicitly.
- n-by-p is highly configurable (i.e 60 configurations)
- The system is interpretable and easy to understand the relationship between pairs of consecutive items in preference chain.
- User efforts are minimal.

THANK YOU

Presented By :

Shubham Gupta (202318052)
Kavisha Madani (202318007)
Shashwat Parikh (202318062)
Mayan Bhut (202318043)