Hierarchical Implicit/Explicit Feedback Recommender System

Kody J. H. Law
KLAI Ltd., London, UK, and
Mathematics Department, University of Manchester
Manchester, M13 9PL, UK

Abstract

In the modern attention economy, ranking is a ubiquitous task–across relevant news feeds in social media, websites in search, products in e-commerce, music and movies in audio and video streaming services, etc. Actions (tasks) in task-oriented dialogue systems (TODS) can be viewed through this lens also. Current recommender systems often deliver ranked items only and feedback comes mostly from clicks, dwell-time, and other implicit feedback. They are therefore prone to wasting substantial resources on ambiguous items, especially when the target item is buried in a larger set of candidate items and the user needs to navigate multiple slates–this scenario is expected to become more prevalent with the next generation of resource-constrained wearable computing platforms, where TODS will be bandwidth-constrained and users will have a low tolerance for errors. We propose the Mixed-Slate Agent (MSA) method, which replaces the item-only slate with a mixed-slate including either a fixed or dynamic set of binary facet or attribute queries, selected by maximizing an acquisition function depending on the joint item/response belief state. A partially observable Markov decision process (POMDP) on the item belief-state formalises the dialog. This explicit feedback loop is used only for immediate disambiguation and is embedded inside any existing recommender system. The resulting method is called the Hierarchical Implicit/Explicit Feedback Recommender System (HIER). For K-element slates out of N ranked items, our method can deliver up to a factor of $\mathcal{O}(N \log_2 K / K \log_2 N)$ asymptotic improvement in scroll depth in comparison to the usual top-K approach. Numerical experiments on a toy problem, a realistic simulated goal-space environment, and real e-commerce and movie recommendation datasets demonstrate the impact of the method.

1 Introduction

A typical modern recommender system involves candidate generation, ranking, and possibly several stages of re-ranking, which may include in-session or real-time re-ranking [Covington et al., 2016, Liu et al., 2022]. Candidate generation and ranking typically operate on an intrinsically passive system assumption: the world-user state gives rise to the user's preferences over items, and those are inputs which are reflected in the output ranking, akin to a supervised learning problem. Contextual bandit and reinforcement learning methods incorporate implicit feedback for exploration to improve the global ranking model [Li et al., 2010, Agrawal and Goyal, 2012, Sun and Zhang, 2018]. The output ranking from the recommender system is the starting point of the present work, and we aim to make a case for leveraging queries from the system to the user for *explicit feedback within a single session*, with the objective of disambiguating the user's goal as swiftly as possible. It is worth noting that active learning frameworks are also routinely applied in recommender systems [Boutilier et al., 2003, Elahi et al., 2016], but these explicit exploration strategies are typically designed to improve the

39th Conference on Neural Information Processing Systems MLxOR Workshop (NeurIPS 2025).

global ranking model, for example in the context of a cold start, rather than an immediate ephemeral ranking. Recommender systems typically operate in a *low-bandwidth interaction* regime, with button clicks being the primary mode of interaction, although they may be seeded with search text from the user.

Task-oriented dialogue systems (TODS) [Zhang et al., 2020] on the other hand typically operate in the high-bandwidth user response regime, for example text or voice, and are hence seeded with a user request or command. The rest of the dialogue is about clarifying and confirming the user intent. There has been some work incorporating further context into TODS [Kottur et al., 2021, Wu et al., 2023], but the value added has been limited because typically (i) the space of intents and slots is small/finite, and (ii) the user is able to explicitly specify their intent to initiate the dialogue. This is likely to change in the future, for two primary reasons. First, 2025 has been named the year of Agents [Barron's, 2025], and along with multi-agentic systems [Chen et al., 2023] extensible frameworks are emerging, like Apple Intents/Shortcuts [Apple Inc., 2025], Alexa Skills/Routines [Amazon, 2025], Model Context Protocol [Anthropic, 2024], etc. Soon the number of intents and slots may be in the millions or even billions, on par with modern recommender systems. Second, one can imagine band-width constrained scenarios, for example where the interface is a wearable and voice is not an option due to environmental circumstances, e.g. a noisy road or a quiet library. This scenario is expected to become more prevalent with future computing platforms.

In such scenarios the goal is to formulate low-bandwidth dialogue for large goal-spaces, much like in recommender systems, and prior context becomes crucial. Furthermore, users are much less tolerant to errors in this scenario—it only takes a few errors for a user to stop using the new technology and revert to classical interfaces and approaches. In the context of recommender systems more broadly, users may often settle for a sub-optimal item, although this reduces watch-time on YouTube [Covington et al., 2016] and monthly churn on Netflix [Gomez-Uribe and Hunt, 2015]. Therefore, new hybrid approaches are required which blend the strengths of dialogue systems and recommender systems. It is worth noting that Conversational Recommender Systems are picking up momentum in this direction, but they typically intersperse individual high-bandwidth natural language queries with recommendation streams [Sun and Zhang, 2018, Lei et al., 2020, Deng et al., 2021].

Partially observable Markov decision processes (POMDP) are a natural framework in which to formulate both TODS [Young et al., 2013, Williams and Young, 2007, Budzianowski et al., 2018] and Recommender Systems [Renoux et al., 2020, Araya-López et al., 2010]. The present explicit-feedback recommender loop is built upon a POMDP model for actively inferring the item/action/task belief distribution using a general acquisition function for the next query depending on the joint belief distribution over items and response. Active Collaborative Filtering [Boutilier et al., 2003] pioneered explicit query selection via expected value of information (EVOI), which is a special case of our approach, but its item-rating questions target offline global/background improvement of the recommender model. Our Mixed-Slate Agent (MSA) dialog instead generates queries in-session with the objective of immediate disambiguation, updating the slate in real time and requiring only low-bandwidth binary feedback. In the special case of expected information gain (EIG) [Lindley, 1956, Houlsby et al., 2011], the setting is similar to 20 Questions [Jedynak et al., 2012, Suresh, 2017, Chen et al., 2018] and information pursuit for decision trees [Geman and Jedynak, 2002, Jahangiri et al., 2017]. We will constrain our attention to binary system queries for the time being since they equate to buttons, which we envision as a core component of the initial (G)UI design. However, this can be arbitrarily generalized, for example to queries which are literally custom-generated multimodal UIs. As such, the system queries will be attributes, categories, keys, or facets, associated to the items [Hearst, 2006, Tunkelang, 2009, He et al., 2015]. They will be hard-coded and known, although a small error may still be assigned to the binary likelihood probabilities to relieve brittleness. The expectation is that items have been filtered by earlier stages of ranking, and the number of items N is in the 100s or 1000s (or more). A prior probability distribution associated to these items is assumed. If the system does not deliver probabilities by default, this can be derived either from a Gibbs measure associated to item scores or as a 'uniform plus epsilon' modification in case only a ranking is available.

The method is completed by embedding the inner explicit MSA loop inside an outer recommendation engine such as HSTU [Zhai et al., 2024], which processes the implicit feedback logs nightly as usual. Optionally we can include a supervised learning module to learn the attribute/item adjacency matrix. This can be valuable for catching secondary attribute-item ambiguity which can arise when items are

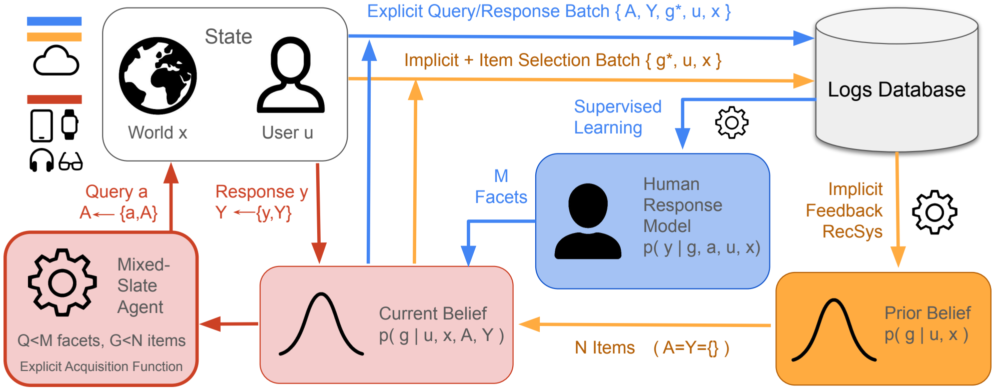
Explicit Query/Response Batch { A, Y, g*, u, x }
Implicit + Item Selection Batch { g*, u, x }
State
World x
User u
Logs Database
Supervised Learning
M Facets
Human Response Model p(y | g, a, u, x)
Implicit Feedback RecSys
Query a
A⟵ {a,A}
Response y
Y ⟵{y,Y}
Mixed-Slate Agent
Q<M facets, G<N items
Explicit Acquisition Function
Current Belief p(g | u, x, A, Y)
N Items (A=Y={})
Prior Belief p(g | u, x)

Figure 1: HIER method flowchart. The MSA explicit feedback loop is shown in red. After this ephemeral on-device session, the observed logs are sent to the cloud for the standard implicit-feedback recommendation system and learning the human response model.

concatenated. The full system is called the Hierarchical Implicit/Explicit Feedback Recommender System (HIER).

2 The Mixed-Slate Agent (MSA)

The objective of the HIER system is to *discover the user's true target goal* g^* when engaged. The inputs to the system are described as follows. The user u and the context x give rise to the space of goals $\mathcal{G}$, and a prior over those goals, p_0. This occurs in the Recommender System module in the cloud, and we can think of it as retrieval and ranking. Then the space of goals $\mathcal{G}$ gives rise to a space of system queries $\mathcal{Q}$, which in turn gives rise to the space of responses $\mathcal{R}$ (assumed to be the same for all queries as a matter of convenience). The likelihood describes the relationship between elements $(y, a, g) \in \mathcal{R} \times \mathcal{Q} \times \mathcal{G}$. This may either be rule-based category facet assignment ($y = 1$ if the facet applies or 0 otherwise), or learned more generally by the Human Response Model.

Given these inputs, we can now define the MSA, which is a greedy/myopic policy for the POMDP. At each step, the input to the MSA is a belief distribution over goals and responses for any query $a'_n \in \mathcal{Q}$

$$g_{n-1} \sim p_{n-1} = p_0(\cdot \mid a_{1:n-1}, y_{1:n-1}), \quad y_n \sim p(\cdot \mid g_{n-1}, a'_n)\,.$$

The output is the action which maximizes the *acquisition function* Φ_n

$$a_n = \arg\max_{a'_n \in \mathcal{Q}} \Phi_n(a'_n), \quad \Phi_n(a'_n) = \mathbb{E}_{y_n}\left[U_{n-1}[p_{n-1}(g_{n-1}, y_n \mid a'_n)]\right],$$

where $U_{n-1}[p_{n-1}(g_{n-1}, y_n \mid a)]$ is a utility function of the joint goal-response belief state, whose output depends on the response y_n but not g_{n-1}, and $\mathbb{E}_{y_n}$ denotes expectation with respect to y_n.

The acquisition function itself is given by expected information gain (EIG), maximum marginal entropy (MaxEnt), or expected maximum probability (MaxProb). These are given explicitly in Appendix A. The action a is a slate of Q binary queries and G goals, where $Q + G = K$, and we let $Q = G = K/2$ by default. See Appendix B for a description of the likelihood and further discussion of batch optimization strategies.

Complexity. The complexity of the method can be understood as follows. The cognitive load required from the user to respond to a slate of K items is $\log_2 K$, according to Hick's law [Hick, 1952], which is identical to a classical top-K query. If we begin with N items, then the total number of items remaining after a single slate is $N - K$ or 1, following a top-K ($G = K$) slate, whereas it is bounded below by $N/(K + 1)$ following an all-query slate ($Q = K$).

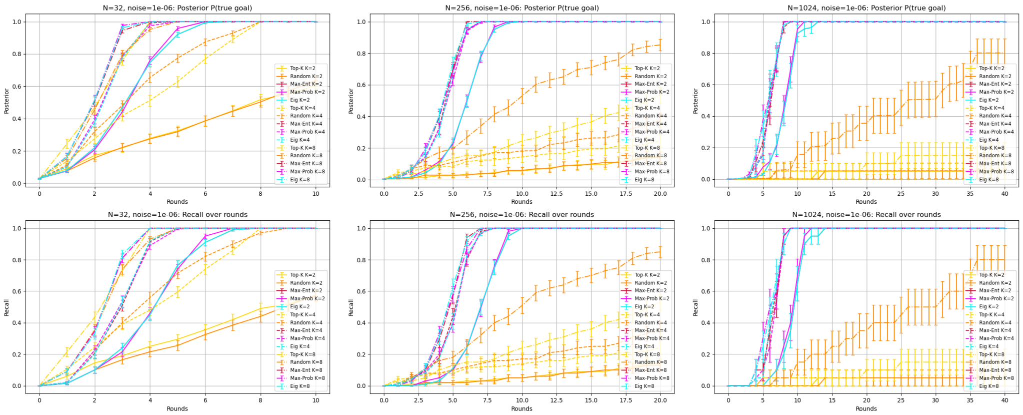

Figure 2: Toy problem, generated data, $\varepsilon = 0$.

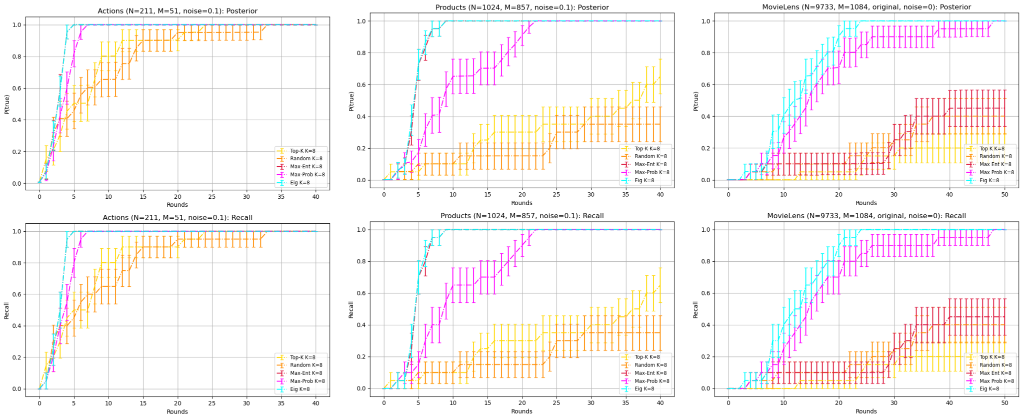

Figure 3: Results associated to digital actions (left), Amazon products (middle), and movielens (right).

The maximum number of all-query slates can be derived as follows. Suppose we have N uniformly distributed items (up-to ε so that they have a ranking), without loss of generality. Then the starting entropy is $H_0 = \log_2 N$. If the items are not uniformly distributed, then we consider the effective number of items as $N_{\text{eff}} = 2^{H_0}$. We can obtain a maximum expected information gain of $\log_2 K$ with each slate, and if this is achieved then the maximum number of slates is bounded by $\log_2 N / \log_2 K$. Defining $\phi(N) = N/\log_2 N$, the theoretical speedup is $\mathcal{O}(\phi(N)/\phi(K))$ under this scenario. In general, we can expect improvement from linear to logarithmic in the number of slates, similarly to a binary search tree. See Appendix B.2 for more discussion on the outer loop(s).

3 Numerical Experiments

Figure 2 illustrates the various methods on toy generated data for $N = 32, 256, 1024$ items and even mixed slates of $K = 2, 4, 8$. The second dataset we consider is a generated set of digital actions, built by gpt-o3 [OpenAI, 2025] by generating domains, intents, slots, and values, as well as categories for the resulting actions. Results are presented in the left panels of Figure 3. The middle and right panels are for a subset of Amazon products [Leskovec et al., 2007], and Movielens movie and tag-relevance [Vig et al., 2012] datasets.

The toy dataset is built with randomly generated Dirichlet(a) prior and Bernoulli(0.5) base likelihoods. The actual likelihood used in practice is corrupted by a prescribed level of noise ε, so that its values are in $\{\varepsilon/2, 1 - \varepsilon/2\}$. In Figures 2 and 5 (in the Appendix) illustrate the various methods for $N = 32, 256, 1024$ items and even mixed slates of $K = 2, 4, 8$, and for noise levels of $\varepsilon \approx 0$ and $\varepsilon = 0.25$, respectively. Figure 6 compares batch oblivious (K largest top-1) with batch optimal (K

optimal) acquisition functions for $N = 32$, showing similar performance. The second dataset we consider is a generated set of digital actions, built by gpt-o3 [OpenAI, 2025] by generating domains, intents, slots, and values, as well as categories for the resulting actions. Complete results are presented in Figure 7 in the Appendix.

3.1 Real-world Datasets

Next, we look at 2 real-world datasets. First, we consider the Amazon Products dataset [Leskovec et al., 2007], comprised of a set of $\approx 5e5$ products, and extract categories from the meta-data. This is far too large to consider all-at-once at the edge and should be viewed as the database from which we retrieve a random sub-sample of $N = 1024$ items. Next, we look at the Movielens tag genome dataset [Vig et al., 2012, Kotkov et al., 2021]. This provides a learned tag relevance matrix for $M \approx 1e3$ tags over $N \approx 1e4$ movies, which we use as the HRM/likelihood – note this is the one example with non-trivial probabilities, i.e. not in $\{\varepsilon/2, 1 - \varepsilon/2\}$. We assume no model mis-specification, i.e. the user simulator is identical to the HRM, noting that this will be minimal once the system is running online with large numbers of users.

3.2 Observations from the Experiments

The following are notable observations:

- MSA always wins the race to disambiguation in comparison to the baselines for $\varepsilon \approx 0$ (Fig 2), while in the noisy case $\varepsilon = 0.25$ (Fig 5) top-K just barely wins when $N = 32$ and $K = 8$, a nd it is competitive with $K = 4$.
- Between the 3 acquisition functions there is no difference beyond noise for the toy model, but we start to see a notable distinction in Figures 7, 8, and 9.
- All methods do better with larger K, but the gain is greater for the baseline methods.
- The gain of MSA over the baselines improves with N, becoming quite huge for $N = 1024$; conversely, for small $N = 32$ and large K, top-K may actually be better for the first few rounds (low recall). See left panels of Figures 2 and 5.
- The noisy channel increases the required number of slates.

4 Conclusion

This work presents the Hierarchical Implicit/Explicit Recommender (HIER), in which an inner explicit feedback loop accelerates recall of the user's target item, without any opportunity cost to the outer Implicit Recommender System. The inner explicit loop is governed by the Mixed-Slate Agent (MSA), which is composed of attribute or facet queries as well as items, endowing the user with the control to sort through items rapidly. Numerical experiments illustrate the value and impact of the method on two synthetic data examples and two real data examples.

Acknowledgments and Disclosure of Funding

K.J.H.L. is grateful to the team in Reality Labs at Meta for hosting him and inspiring this exciting research direction: "I had too many interesting discussions to list, but you know who you are and please know that I appreciate you."

References

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 39.1–39.26, 2012.

Amazon. Alexa skills kit & routines. Developer documentation, 2025. Accessed May 2025.

Anthropic. Model context protocol (mcp) specification. White paper, v1.1, 2024.

Apple Inc. App intents framework. Developer documentation, 2025. Accessed May 2025.

María Araya-López, Olivier Buffet, Vincent Thomas, and Francois Charpillet. A pomdp extension with belief-dependent rewards. In *Advances in Neural Information Processing Systems 23 (NeurIPS 2010)*, pages 64–72, Vancouver, Canada, 2010.

Barron's. Nvidia ceo says 2025 is the year of ai agents, 2025. URL `https://www.barrons.com/articles/nvidia-stock-ceo-ai-agents-8c20ddfb`.

Craig Boutilier, Richard Zemel, and Benjamin Marlin. Active collaborative filtering. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 98–106, 2003.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, 2018.

Giacomo Capannini, Claudio Lucchese, Franco Maria Nardini, Raffaele Perego, and Fabrizio Silvestri. Efficient diversification of web search results. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '11)*, pages 1211–1212. ACM, 2011. doi: 10.1145/2009916.2010131. URL `https://arxiv.org/abs/1105.4255`.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents, 2023.

Yihong Chen, Bei Chen, Xuguang Duan, Jian-Guang Lou, Yue Wang, Wenwu Zhu, and Yong Cao. Learning-to-ask: Knowledge acquisition via 20 questions. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1216–1225, 2018.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)*, pages 191–198, Boston, MA, USA, 2016. ACM. doi: 10.1145/2959100.2959190.

Y. Deng et al. Unified conversational recommendation policy learning via graph-based rl. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1431–1441, 2021.

Mehdi Elahi, Francesco Ricci, and Neil Rubens. A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, 20:29–50, 2016.

Donald Geman and Bruno Jedynak. An active testing model for tracking roads in satellite images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):1–14, 2002.

Carlos A Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):1–19, 2015.

Xiaoting He, Tianqi Chen, Min-Yen Kan, and Xiao Chen. Trirank: Review aware explainable recommendation by modelling aspects. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1661–1670, Melbourne, Australia, October 2015.

Marti Hearst. Design recommendations for hierarchical faceted search interfaces. pages 1–5, 2006.

W. E. Hick. On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, 4(1): 11–26, 1952.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. volume 1050, page 24, 2011.

Ehsan Jahangiri, Erdem Yoruk, Rene Vidal, Laurent Younes, and Donald Geman. Information pursuit: A bayesian framework for sequential scene parsing. *arXiv preprint arXiv:1701.02343*, 2017.

Bruno Jedynak, Peter I Frazier, and Raphael Sznitman. Twenty questions with noise: Bayes optimal policies for entropy loss. *Journal of Applied Probability*, 49(1):114–136, 2012.

Denis Kotkov, Alexandr Maslov, and Mats Neovius. Revisiting the tag relevance prediction problem. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1768–1772, 2021.

Satwik Kottur, Seungwhan Moon, Alborz Geramifard, and Babak Damavandi. Simmc 2.0: A task-oriented dialog dataset for immersive multimodal conversations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4903–4912, 2021.

W. Lei et al. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 304–312, 2020.

Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5–es, 2007.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual bandit approach to personalized news article recommendation. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 661–670, 2010.

D. V. Lindley. On a measure of the information provided by an experiment. *Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

Weiwen Liu, Jiarui Qin, Ruiming Tang, and Bo Chen. Neural re-ranking for multi-stage recommender systems. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 698–699, 2022.

OpenAI. Openai o3 and o4-mini system card, April 2025. URL `https://openai.com/index/o3-o4-mini-system-card/`. Accessed 2025-08-26.

Jérémy Renoux, Teresa S. Veiga, Pedro U. Lima, and Matthijs T. J. Spaan. A unified decision-theoretic model for information gathering and communication planning. In *Proceedings of the 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 67–74, 2020.

Rodrygo LT Santos, Jie Peng, Craig Macdonald, and Iadh Ounis. Explicit search result diversification through sub-queries. In *European conference on information retrieval*, pages 87–99. Springer, 2010.

Yueming Sun and Yi Zhang. Conversational recommender system. In *The 41st international acm sigir conference on research & development in information retrieval*, pages 235–244, 2018.

Ananda Theertha Suresh. A bayesian strategy to the twenty-questions game. M.sc. thesis, Duke University, 2017.

Daniel Tunkelang. *Faceted Search*. Morgan & Claypool, San Rafael, CA, 2009.

Jesse Vig, Shilad Sen, and John Riedl. The tag genome: Encoding community knowledge to support novel interaction. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(3):1–44, 2012.

Jason D. Williams and Steve Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422, 2007.

Tzu-Lun Wu, Satwik Kottur, Andrea Madotto, Mohamed Azab, Peter Rodriguez, Babak Damavandi, Nanyun Peng, and Seungwhan Moon. Simmc vr: A task oriented multimodal dialog dataset with situated and immersive vr streams. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pages 6273–6291, Toronto, Canada, July 2023.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Jiayuan He, et al. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. pages 58484–58509, 2024.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, Minlie Huang, and Xiaoyan Zhu. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, 63(10): 2011–2027, 2020.

A Acquisition functions

The precise form of the acquisition functions are given below.

Expected Information Gain:

$$\Phi_n(a) = I(g_{n-1}, y_n \mid a) = \int_{G\times\mathbb{R}} p_{n-1}(dg_{n-1}, dy_n \mid a) \log \left(\frac{p_{n-1}(g_{n-1}, y_n \mid a)}{p_{n-1}(y_n \mid a) p_{n-1}(g_{n-1} \mid a)} \right) .$$

Maximum (marginal) Entropy:

$$\Phi_n(a) = H(y_n \mid a) = -\int_{\mathbb{R}} p_{n-1}(dy_n \mid a) \log(p_{n-1}(y_n \mid a)) ,$$

$$p_{n-1}(y_n \mid a) = \int_G p_{n-1}(dg_{n-1}, y_n \mid a) .$$

Maximum Probability:

$$\Phi_n(a) = \mathbb{E}_{y_n} \left[\max_g p_{n-1}(g \mid y_n, a) \right] .$$

Note that the first and the third acquisitions can be written as expected value of information (EVOI). See the Appendix for derivations.

B Likelihood

As mentioned above, the queries will be slates of basic binary queries. For N queries and M goals we can represent the goal and binary query spaces as $\mathcal{G} = \{1, \ldots, N\}$ and $\mathcal{T} = \{1, \ldots, M\}$, and the likelihood of a positive response is represented by an $N \times M$ matrix P such that for $(g, t) \in \mathcal{G} \times \mathcal{T}$, the probability of an affirmative response is given by $p(y = 1 \mid g, t) = P(g, t)$. We will select a batch of K binary queries $a = (t_1, \ldots, t_K)$ at each time. Typically, we will assume a single-select type of response so that a K-fold 'no' is obtained if none of the queries is selected, and otherwise a single 'yes' is obtained. The response space is therefore $\mathcal{R} = \{0, \ldots K\}$, with probabilities

$$p(y = 0 \mid g, a) \propto \prod_k \left(1 - P(g, t_k)\right) , \quad p(y = k \mid g, a) \propto P(g, t_k) .$$

B.1 Batch Strategies

The space of K binary query slates has dimension $\binom{N}{K}$, which quickly becomes intractable for brute force methods. Therefore, we consider graded approximations

- Batch optimal: a maximizes AF brute force (small N and small K).
- Batch greedy: t_k maximizes AF conditioned on $(t_1, \ldots, t_{k-1}$), for $k = 1, \ldots, K$. Note we still require expectation over the 2^k-dimensional response-space each time, as this is still before any observation. [1]
- Batch oblivious: top K binary queries which maximize the single binary query AF.

For non-explicit baselines, we will also consider slates with all items (top-K) and randomly chosen queries. In practice we select a slate of Q queries and G goals, where $Q + G = K$, and we let $Q = G = K/2$ by default[2].

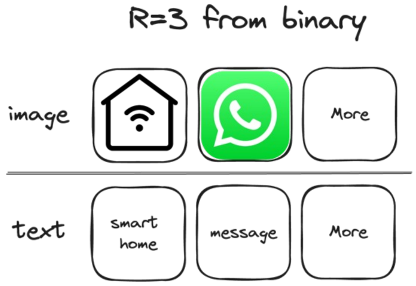

Figure 4: An example batch of 2 binary queries represented with R=3 buttons (2 modality options).

[1] As an intermediate version between oblivious and greedy, one can artificially impose a diversity penalty as in Santos et al. [2010], Capannini et al. [2011].

[2] Note that goals are trivially binary queries with $P(g, g') \approx \delta_{g,g'}$, so we can let $\mathcal{G} \subset \mathcal{T}$, and Q, G can be selected automatically and dynamically. This improves algorithm performance, but there is a hidden user-experience cost.

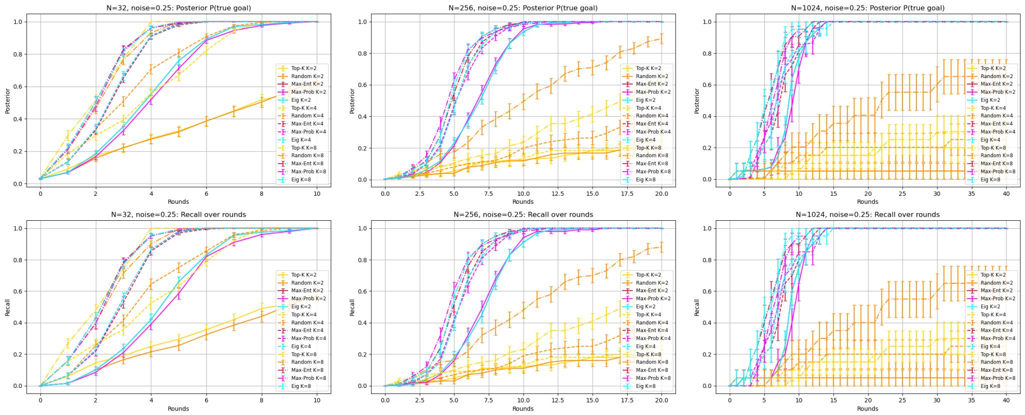

Figure 5: Toy problem, generated data, $\varepsilon = 0.25$.

B.2 The Outer Loops: Implicit-Feedback RecSys (IFR) and Human Response Model (HRM) Learning

The design of the full HIER method is that the MSA is embedded inside the existing IFR and reset for each session. By decoupling the likelihood terms associated with the dialog from other factors, online and re-ranking methods can be used in tandem within session if desired. Occasionally the system may need to feed the context back to the higher-level ranking or candidate generation stages (IFR) to replenish the item space, possibly prompted by a null-state indicating the probability that the user's goal is not in the existing set.

Let $\mathbf{w}_u, \mathbf{v}_g, \mathbf{h}_u, \mathbf{e}_a, \mathbf{f}_x$ be learned embeddings depending on parameters θ, and let $\phi = (\beta, \gamma)$ be additional parameters characterizing the effect of the world context on the respective model. In practice, the embeddings would typically be outputs of a deep/nonlinear model applied to raw features, as in large-scale two-tower recommenders Covington et al. [2016] or more recent transformer-based approaches Zhai et al. [2024]. An example architecture for the prior and the likelihood may be as follows

$$p(g \mid u, x) \quad \propto \quad \exp(\mathbf{w}_u^T \mathbf{v}_g + \beta^T \mathbf{f}_x)\,, \tag{1}$$

$$\ell(y = 1 \mid g, a, u, x) \quad = \quad \sigma(\mathbf{h}_u^T \mathbf{e}_a + \mathbf{e}_a^T \mathbf{v}_g + \gamma^T \mathbf{f}_x)\,, \tag{2}$$

where σ is a link function such as sigmoid. This is just an example and any architecture is suitable. Similarly, any recommender system method can be used to learn $p(g \mid u, x) \propto r_{u,x}(g)$, including contextual bandits and reinforcement learning. Note that the score/reward $r_{u,x}(g)$ is often modelled as a random variable in RecSys, in which case we use its expectation or mode as a point estimator.

HIER keeps the live assistant on task. The Inner MSA loop uses queries only for immediate disambiguation, which is fast and effective. The Outer RecSys and HRM learners harvest both implicit and explicit logs later, improving θ and ϕ without ever subjecting a single user to gratuitous exploration.

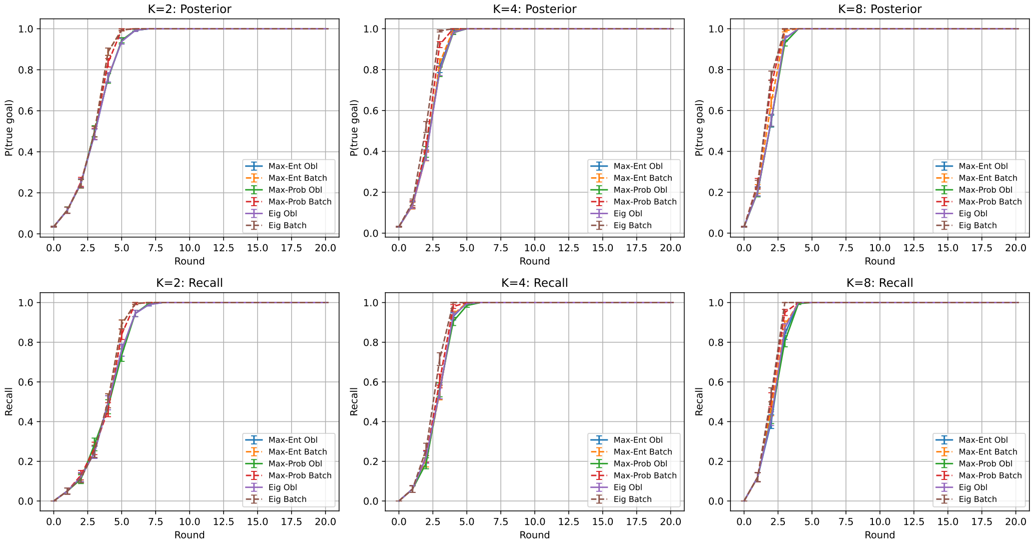

Figure 6: Toy problem ($N = 32$), generated data, batch oblivious vs batch optimal.

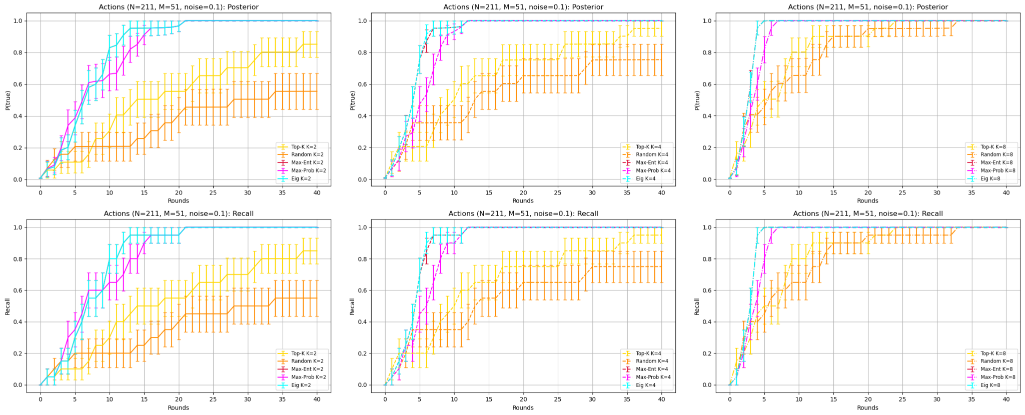

Figure 7: Generated digital actions data.

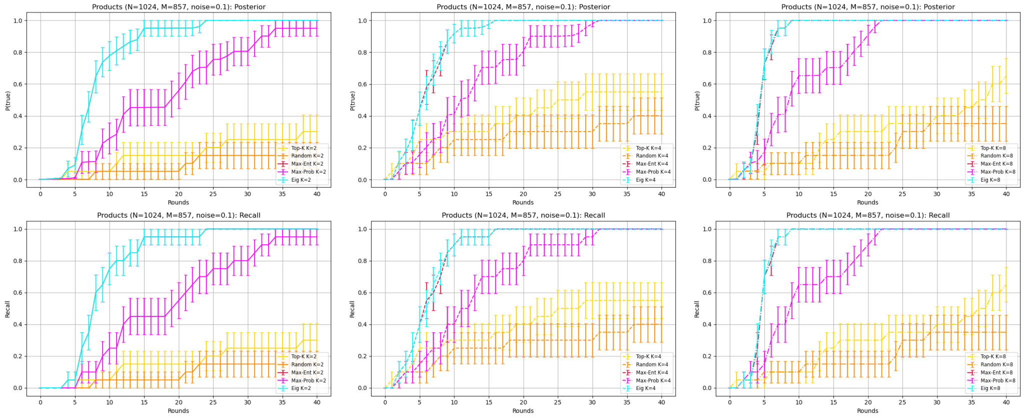

Figure 8: Amazon products data.

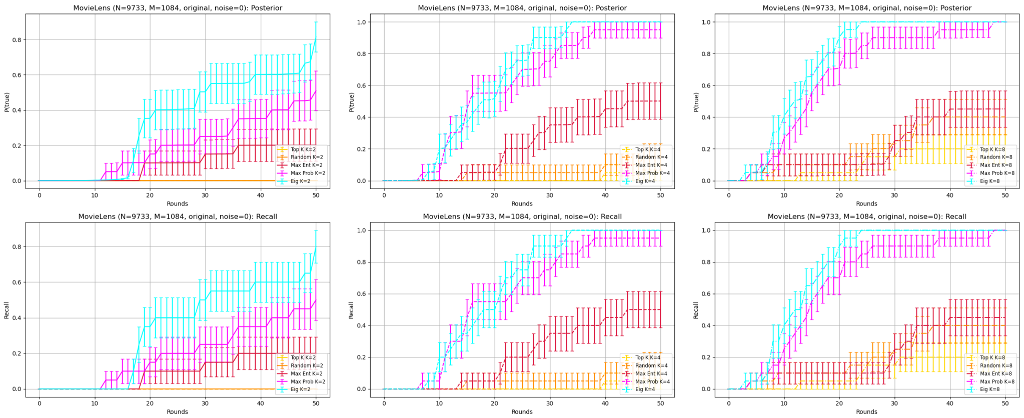

Figure 9: Movielens tag-relevance data.