User Simulation

[DAT640] Information Retrieval and Text Mining

Krisztian Balog
University of Stavanger

November 5, 2024

In this module

1. Simulation-Based Evaluation
2. Simulating Interactions with Search Systems
3. Simulating Interactions with Conversational Information Access Systems

Simulation-Based Evaluation

Evaluation Methodologies

- Reusable test collections
  - Standard evaluation methodology for making relative comparisons between two systems in a repeatable and reproducible manner
  - Limited ability to capture many aspects of users and interactions adequately; the user is abstracted away
- User studies
  - Provides the highest fidelity in terms of capturing real users' interactions with an actual system in a controlled setting
  - Costly to run, not reproducible
- Online evaluation
  - Observing real users of a fully operational system and assessing the system's performance by analyzing the recorded user behaviour
  - Enables measuring the actual utility of a system; scalable
  - Not reproducible, no control over users

Challenges and Simulation-based Evaluation

- None of the previous methodologies enable comparison of multiple interactive information access systems using reproducible experiments
  - Test collection-based evaluation is static in nature
  - Lack of reproducibility when real users are involved
- It is important to evaluate the *overall effectiveness* of a system
  - Commonly, complex tasks are decomposed into a series of smaller and simpler components
  - These can be abstracted, studied and addressed in isolation (using reusable test collections)
  - However, the evaluation of individual components alone is insufficient
  - The ultimate goal is to evaluate the *whole* system from a user's perspective
- The evaluation of an interactive system's overall effectiveness must involve a user in some way
  - The involvement of real users inherently leads to non-reproducible experiments
  - Simulated users can be controlled and thus enable reproducible experiments

User Simulation

- Informal definition: having an intelligent agent to simulate how a user interacts with a system
- User simulation has many uses, including
  - Performing **large-scale automatic evaluation** of interactive systems (i.e., without the involvement of real users)
  - Gaining **insight into user behaviour** to inform the design of systems and evaluation measures
  - **Analyzing system performance** under various conditions and user behaviours (answering what-if questions, such as "What is the influence of X on Y?")
  - **Generating synthetic data** with the purpose of training machine learning models, especially reinforcement learning
- For relative comparisons of systems, simulation does not need to be perfect; it is enough to identify relative system differences

User Simulation

- It is assumed that there is some information available about
  - the **system** (S) and its **user interface** (e.g., search engine with a query box and navigable search result lists)
  - the **user's task** (T) (e.g., collecting as many relevant information items as possible or finding a suitable product to purchase)
  - the **user** (U) (e.g., background knowledge, context)
- User simulator: a formal/computational model that determines how a given user U would behave when interacting with system S to perform some task T given any particular interaction context

Simulation Approaches

- Two broad approaches:
  - **Model-based**: can be rule-based (based on knowledge about how users behave) or interpretable probablistic models (parameters set heuristically or estimated based on observed user data)
  - **Data-driven**: maximize accuracy of fitting any observed real user data, without necessarily imposing interpretability (supervised ML)
- Accurate simulation of observable behaviour may require simulation of latent behaviour (e.g., cognitive state of a user), which makes simulation more interpretable (via interpretable generative models)
- Interpretability is desirable to enable the testing of verifiable hypotheses about users and ensure that evaluation results are meaningful
  - Varying the parameters corresponds to the simulation of different kind of users

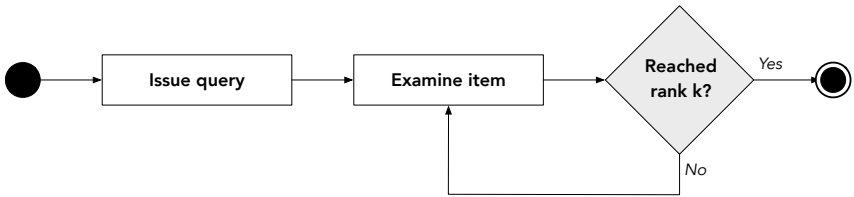
Partial vs. Complete User Simulation

- Simulation of an action of a user: Given an interaction context (system environment), predict what action a user would take (e.g., given a snippet in a list of search results, predict whether a user would click on it)
- Simulation of a sequence of actions of a user: Given an interaction context, predict the whole sequence of multiple actions that a user would take (need to consider dependency between actions)
- Simulation of a user's interactions in a whole session of finishing a task (there may be multiple sequences of interactions)
- Simulation of a user's general preferences and behaviour across tasks

Traditional (Test Collection-based) Evaluation

- Components of an IR test collection
  - Collection of documents
  - A set of queries
  - Corresponding relevance judgments
- System is run to generate retrieval results for each query
- Retrieval performance is measured for each query using various evaluation metrics (e.g., Precision, Recall, NDCG) $\Rightarrow$ perceived utility of a result list from the user's perspective

Traditional Evaluation Measures as Naive User Simulators



- User model: Sequentially browse the ranked list of results up to rank position k and examine each item
- E.g., Precision@k, Recall@k, MAP

Measures based on Explicit Models of User Behaviour

Virtually all IR measures attempt to quantify the performance of a search result based on a combination of four factors:

- The assumed **user task** (e.g., high precision vs. high recall)
- The assumed **user behaviour** when interacting with the results
- Measurement of the **reward** a user would receive from examining the result
  - Early IR measures defined reward based on relevance-based gains
  - Later, novelty and diversity of the search results were also considered
- Measurement of the **effort** a user would need to make in order to receive the reward
  - Uniform vs. longer documents would take more effort/time

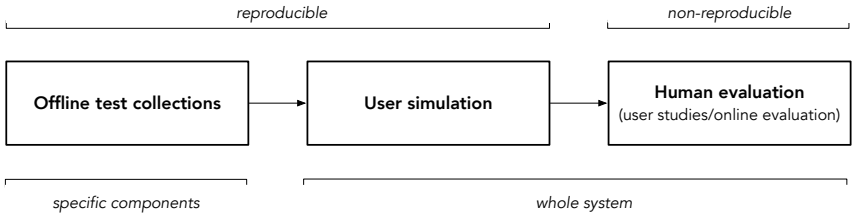
Measures based on Explicit Models of User Behaviour

Virtually all IR measures attempt to quantify the performance of a search result based on a combination of four factors:

- The assumed **user task** (e.g., high precision vs. high recall)
- The assumed **user behaviour** when interacting with the results
- Measurement of the **reward** a user would receive from examining the result
  - Early IR measures defined reward based on relevance-based gains
  - Later, novelty and diversity of the search results were also considered
- Measurement of the **effort** a user would need to make in order to receive the reward
  - Uniform vs. longer documents would take more effort/time

Limited to evaluating a ranked list of results; insufficient in highly interactive settings

User Simulation in the Evaluation Workflow



Simulation is not meant to replace but to complement other evaluation methodologies!

- Components individually evaluated offline using reusable test collections
- Online testing with real users is constrained by the traffic volume of the service
- User simulation offers a cost-effective and efficient way to explore a large number of system variations before committing to online experiments
- Ultimately, systems should be tested with real users (crucial validation step for the simulation results)

Simulating Interactions with Search Systems

Mathematical Framework

- Markov decision process (MDP)
  - Formally be described by a finite state space $\mathcal{S}$, a finite action set $\mathcal{A}$, a set of transition probabilities P, and a reward function R
  - At a given point in time, the simulated user is in state $s \in \mathcal{S}$, and by executing action $a \in \mathcal{A}$, they transition into a new state s' according to the transition probability $P(s'|s, a)$ and receive reward $R(a, s)$
  - The Markov property ensures that this transition depends only on the current state and action (which simplifies modeling and reduces computational complexity)

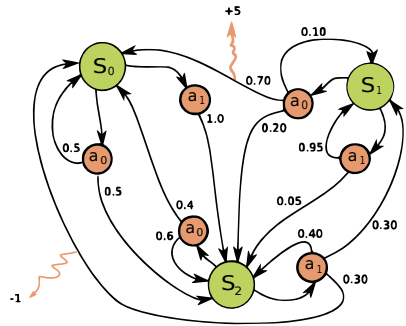


Figure: Markov decision process[a]

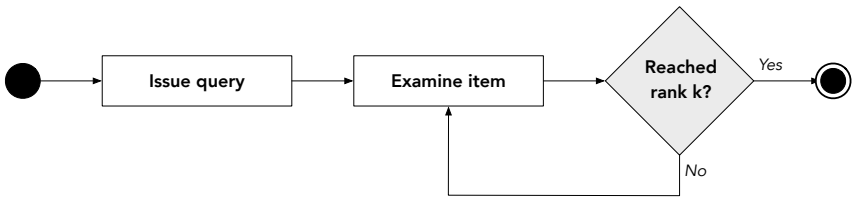
[a] https://en.wikipedia.org/wiki/Markov_decision_process

MDP in User Simulation

- The reward function can be used to encapsulate the costs and rewards based on observed data (from logs or user studies)
- Transition probabilities are modeled explicitly based on some model of user behavior
- Policy is also based on an explicit model of user behavior; does not need to be optimal, but needs to be controllable by the system designer

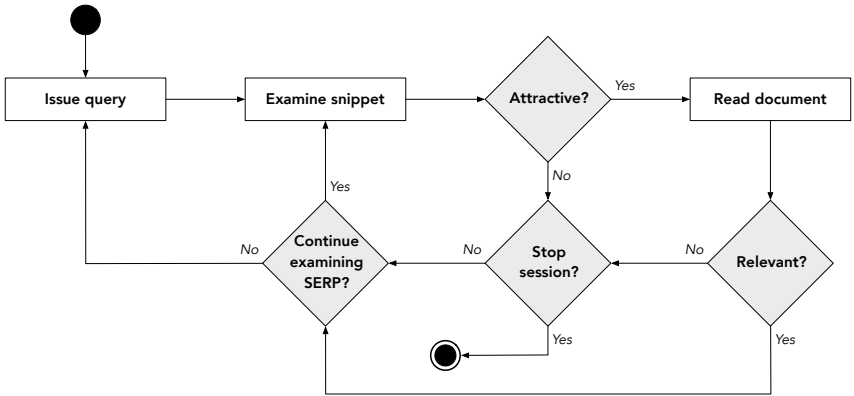
Workflow Models

- Simulation relies on simplified models (of workflows and user behaviour), which allows for "unnecessary complications" to be abstracted away
- The main research challenge is determining what elements of human behaviour to capture in these abstractions, while keeping the models as simple as possible



Naive searcher model, corresponding to highly abstracted user

Search Workflows



Searcher model

Search Workflows

Complex Searcher Model

(A) Topic examination

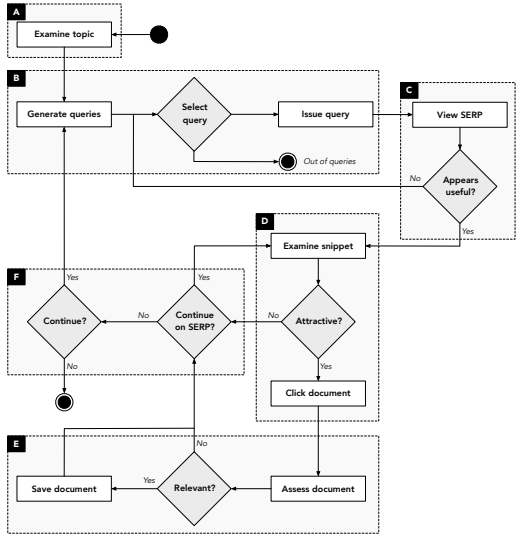
(B) Querying

(C) SERP examination

(D) Result summary examination

(E) Document examination

(F) Deciding to stop



Simulating Queries

- Generate individual queries for a *known item search*
  - A document d is sampled from the collection as the known item
  - The query is generated iteratively by adding a term t_i from the language model of d
- Generate query reformulations
  - Assuming a fixed set of terms $t_1, \ldots, t_m$ for a given topic
  - Based on strategies such as *expansion*, *replacement*, and a combination of the two
  - For example,
    - $q_1 = \{t_1\} \rightarrow q_2 = \{t_1, t_2\} \rightarrow q_3 = \{t_1, t_2, t_3\} \rightarrow \ldots$
    - $q_1 = \{t_1\} \rightarrow q_2 = \{t_2\} \rightarrow q_3 = \{t_3\} \rightarrow \ldots$

Simulating Scanning Behaviour

- Concerned with how the user processes the list of results presented to them in response to their search query
- Not directly observable! Commonly, **sequential browsing** is assumed
- *Cascade model*
  - The user examines each result and decides whether the snippet is deemed relevant enough to warrant a click
  - Snippets below a clicked result are not examined (i.e., the user would stop after having found a relevant result)
- *User browsing model*
  - At each rank position, the user first decides whether to look at the snippet or not ("attractive" or not)
  - Then, resume the scan of the result list from the next rank position (whether the result gets clicked or not)
  - Models the event that user *examines* the snippet ($P(E = 1|R_i, C_1, ..., C_{i-1})$) and, independently from it, whether they find the snippet *attractive* ($P(A = 1|R_i)$)
  - Model parameters are estimated from clicks

Complex Presentation Layouts

- Current approaches rarely consider modern SERPs and alternative presentation layouts, where the top-down traversal assumption is challenged
- Given the complexity of such interfaces, model-based approaches might have inherent limitations compared to data-driven approaches

(a) A traditional "ten blue links" layout.

(b) A product search layout.

(c) A video recommendation layout.

(d) An advertisement layout.

Simulating Clicks

- Mimic a user's decision on whether to click on a search result (to view it in detail) after being exposed to a result (snippet)
- Often integrated with the modeling of scanning behaviour
- Many tradeoffs to be made, especially interpretability vs. prediction accuracy
  - *Position-based simulation*: clicking probability only depends on the rank positions:
    - $P(Click = 1|Rank = i, R_1, R_2, ..., R_k) \approx P(Click = 1|Rank = i)$
    - Naive but generally applicable to any simulation scenario
  - *Content-based simulation*: snippet content is used to model the probability of clicking
    - Intuitively more accurate, but learned models are prone to overfitting and may lose interpretability
- *Perfect snippet assumption* (implicit): user is assumed to be able to tell whether a result is relevant based on the snippet and would always click on a result if it is relevant

Simulating Document Processing

- Processing (i.e., reading and understanding) a document requires an **effort** from the user and yields some **utility** to them (enabling the user to acquire new information, thus changing cognitive state)
- *Dwell time* is often used as a proxy for effort
  - Time (in seconds) needed to process a document of length l, measured in words

$$T_D(l) = al + b$$

    User is reading at a rate of a seconds per word, and then uses a constant amount of b seconds to make an assessment about the document's relevance
- *Relevance* is used as a proxy for utility
  - Commonly, leveraging ground truth relevance assessments in existing test collections
  - Alternatively, predict whether the user would find the document relevant
    - Represent the user's knowledge state as a language model that evolves based on the documents encountered
  - Note that utility is meant to be a broader concept than topical relevance!
    - Includes quality, novelty, importance, credibility, etc.
    - Encompasses everything that the user values, e.g., a witty or engaging writing style

Simulating Stopping Behaviour

Users can decide to stop the search process at various points

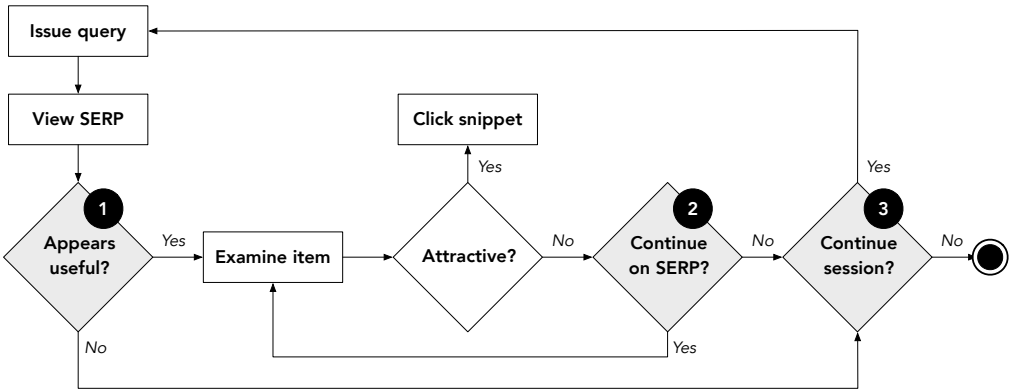


Figure: Excerpt from the updated Complex Searcher Model, highlighting various stopping decision points: (1) SERP-level stopping, (2) query-level stopping, and (3) session-level stopping

Simulating Stopping Behaviour

- Several user studies (interviews) to understand *why* people decide to stop
- Users do not apply predetermined criteria, but rather base stopping decisions on the feeling of "good enough"
  - Factors include time constraints, diminishing returns of further information seeking, and increasing redundancy of information encountered
- Different heuristic rules to quantitatively characterize the sense of "good enough," for example,
  - *Satisfaction*: encountering a predefined number of relevant snippets
  - *Searcher frustration*: observing a certain number of non-relevant snippets
  - *Satisfaction or frustration*: stopping as soon as one of the two conditions is met
  - *Time-based*: total amount of time spent on the SERP or time elapsed after the last relevant document found

Question

How can we tell if a simulator is good (i.e., mimics human behaviour sufficiently well)?

Validating Simulators

- Validating whether the simulator imitates the behaviour of real users *sufficiently well*
- Would a simulated user lead to similar retrieval performance to what is obtained from real users?
  - E.g., simulated queries against real queries
- Would a simulated user produce data that matches the characteristics of real user data?
  - How well a user simulator can predict data observed in search logs (e.g., search session statistics)?
- Does the user simulator behave as expected for it intended use (e.g., for evaluating an interactive system)
  - Tester-based framework
  - Tester: System A is expected to perform better than system B under a certain condition (e.g., for a certain kind of queries)
  - Simulator passes the test if the expected behavior is observed
  - Reliability of a user simulator and reliability of a Tester can be estimated jointly

Simulating Interactions with Conversational Information Access Systems

Recap: Conversational Information Access

- High-level categorization of systems
  - *Goal-driven* (a.k.a. *task-oriented*): aiming to assist users to complete some specific task $\Leftarrow$ our focus
  - *Non-goal-driven* (a.k.a. *chatbots*): aiming to carry on an extended conversation ("chit-chat"), usually with the purpose on entertainment

Recap: Conversational Information Access

- High-level categorization of systems
  - *Goal-driven* (a.k.a. *task-oriented*): aiming to assist users to complete some specific task $\Leftarrow$ our focus
  - *Non-goal-driven* (a.k.a. *chatbots*): aiming to carry on an extended conversation ("chit-chat"), usually with the purpose on entertainment
- **Conversational information access: tasks with an underlying information need, which can be satisfied through a conversation**
  - Includes the tasks of search, recommendation, and question answering (boundaries often blurred)

Challenges

Traditional search and recommender systems	**Conversational information access**
Limited set of user actions allowed by the system's UI	User intents need to be inferred from free text
Interactions are either driven by the user (search) or by the system (recommendation)	*Mixed initiative*: the user and system both actively participate in addressing the user's information need
Results are restricted to a ranked list of items	Results can be text of arbitrary length (incl. semi-structured elements and questions posed to the user)

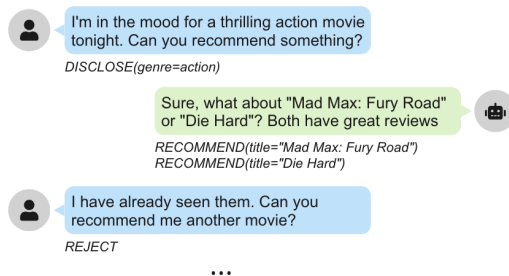
Challenges

Traditional search and recommender systems	**Conversational information access**
Limited set of user actions allowed by the system's UI	User intents need to be inferred from free text
Interactions are either driven by the user (search) or by the system (recommendation)	*Mixed initiative*: the user and system both actively participate in addressing the user's information need
Results are restricted to a ranked list of items	Results can be text of arbitrary length (incl. semi-structured elements and questions posed to the user)

$\Rightarrow$ More advanced natural language understanding capabilities are required

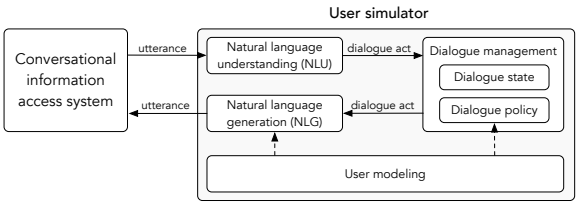
Recap: Concepts and Terminology

- A conversation is a sequence of *turns*
- A turn is a natural language *utterance* from either the user or the system
- *Dialogue act* represents the interactive function of an utterance
  - It is commonly a tuple comprising an *intent* and *slot-value* pairs
  - E.g., `GREETING` and `DISCLOSE(location=Stavanger)`



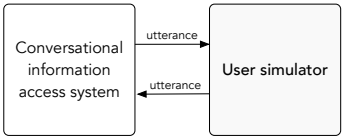
Simulator Architectures

Modular systems



- Model user responses semantically on the level of dialogue acts, then generate the corresponding natural language utterances

End-to-end systems



- Operate on the utterance level (generate textual responses directly)
- Might yield more fluent dialogues, but do not allow for interpretable user behaviour

User Dialogue Policy

- Here: task-oriented dialogue in a restricted "slot-filling" sense
  - A *domain ontology* describes the specific intents, slots, and entities that can be talked about
  - The user can specifying their constraints in terms of *informable slots* and requesting information on *requestable slots*
  - Appropriate for modeling user goals in some scenarios (e.g., item recommendation), while others (e.g., exploratory search) are open research problems
- Dialogue is represented as a sequence dialogue acts by the system (a_i^s) and the user (a_i^u) as they take turns: $a_0^s \rightarrow a_0^u \rightarrow a_1^s \rightarrow a_1^u \rightarrow \cdots \rightarrow a_{t-1}^s \rightarrow a_t^s$
- The policy π determines what action a_{t+1}^u the user should take next, given the dialogue history

Statistical User Models: N-grams Models

- Next response based on the dialogue history (resembling the estimation of language models):

$$\pi(s_t) = P(a^u_{t+1}|a^s_t, a^u_t, a^s_{t-1}, a^u_{t-1}, \ldots, a^u_0, a^s_0)$$

- Strong simplifying assumption to condition the next user action exclusively on the preceding system action:

$$\pi(s_t) = P(a^u_{t+1}|a^s_t)$$

- Conditional probabilities estimated from an annotated dialogue corpus
- No information about the user's goal, no constraints on the simulated user behaviour $\Rightarrow$ fails to produce realistic dialogues
  - Placing constraints on the dialogue flow yields somewhat more realistic dialogues, but the consistency between user responses across the dialogue is still not guaranteed

Statistical User Models: Goal-directed User Model with Memory

- Explicit representation of the user goal as a sequence of slot-value pairs with priority: $G = \langle (slot_1, value_1, prior_1), \ldots, (slot_n, value_n, prior_n) \rangle$
  - When the user is prompted for the relaxation of some attribute, slot-value pairs with a higher priority are less likely to be relaxed
- Dialogue history at time t is represented as a vector $h_t = \langle c_1, \ldots, c_n \rangle$
  - c_i is the count of the occurrences a value is provided for the corresponding $slot_i$
  - Enables the simulator to disclose new information to the system if mixed initiative is supported
- Allows for automatic evaluation in terms of full or partial task completion (given how goals are represented)

Statistical User Models: Agenda-based Simulator

- Factors the user state into an agenda and a goal $s_t = (A_t, G_t)$
- Agenda A_t is a stack-like structure, representing the pending intentions of the user
- Goal is a tuple $G_t = (C_t, R_t)$, where
  - C_t is a set of domain-specific constraints the user wants to impose on the dialogue
  - R_t specify requests, i.e., slots whose values are initially unknown to the user and will need to be filled out during the conversation
- For example (restaurant recommendation): looking for the name, address, and phone number of a centrally located bar serving beer:

$$C_0 = \begin{bmatrix} \text{type} & = & \text{bar} \\ \text{drinks} & = & \text{beer} \\ \text{area} & = & \text{central} \end{bmatrix} \qquad R_0 = \begin{bmatrix} \text{name} & = \\ \text{addr} & = \\ \text{phone} & = \end{bmatrix}$$

Statistical User Models: Agenda-based Simulator

- Agenda initialization
  - All goal constraints set to `INFORM` acts and all goal requests set to `REQUEST` acts
  - `BYE` added at the bottom of the agenda to close the dialogue
- As the conversation progresses, the agenda and goal are dynamically updated
  - Next user action simplifies to popping items from the top of the agenda
  - Agenda updates are push operations, where dialogue acts get added on top of the agenda

$$C_0 = \begin{bmatrix} type = bar \\ drinks = beer \\ area = central \end{bmatrix}$$

$$R_0 = \begin{bmatrix} name = \\ addr = \\ phone = \end{bmatrix}$$

Sys 0 Hello, how may I help you?

$$A_1 = \begin{bmatrix} inform(type = bar) \\ inform(drinks = beer) \\ inform(area = central) \\ request(name) \\ request(addr) \\ request(phone) \\ bye() \end{bmatrix}$$

Usr 1 I'm looking for a nice bar serving beer.

Sys 1 Ok, a wine bar. What pricerange?

$$A_2 = \begin{bmatrix} negate(drinks = beer) \\ inform(pricerange = cheap) \\ inform(area = central) \\ request(name) \\ request(addr) \\ request(phone) \\ bye() \end{bmatrix}$$

Usr 2 No, beer please!

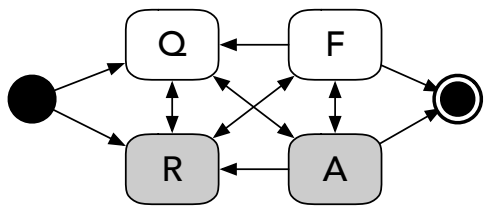
Learning User Simulators Fully Data-driven

- Operating on the semantic level of dialogue acts vs. text utterances directly
- From manual feature engineering to progressively adopting end-to-end approaches
  - Interpretability diminishes, limited control over the behaviour of the simulated user
  - Effectively, only indirect control through the input training data provided

User Simulation for Conversational Search

Two main types of user utterances considered:

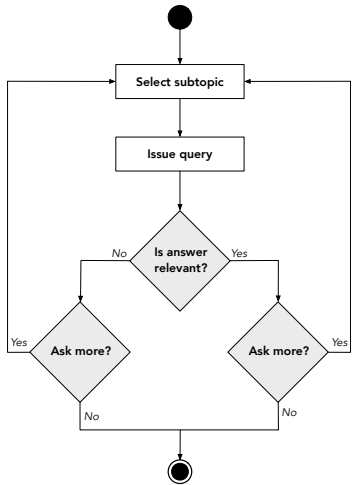
- User-initiated questions (Query)
- Responses to system-initiated questions (Feedback)



QRFA: generic model of conversational information seeking processes

Simulating User Questions (Lipani et al., 2021)

- It is assumed that the user's goal is to learn about a set of subtopics by interacting with the system
- Both user queries and system responses are represented as *subtopics*
- At each dialogue turn the user asks about a particular subtopic
- Based on the relevance of the system's response and the user's *persistence*, the user will ask further questions (about the same subtopic or a different one) or stop querying



Simulating Answers to Clarifying Questions (Salle et al., 2021)

- Simulating how a user would respond to clarifying questions that are in the form: "Are you looking for *[facet]*?"
- *User intent model*: represents the user's information need and estimates whether the clarifying question matches the user's intent
  - Implemented by fine-tuning a BERT model for binary classification
- *Persona model*: specifies personal user characteristics
  - *Cooperativeness* ($\in [0, 1]$): the user's willingness to help the system by giving an informative answer (e.g., "No, I'm looking for *[intent]*") vs. simply "Yes" or "No")
  - *Patience*: maximum effort (number of turns) the user is willing to spend interacting with the system

Simulating Answers to Clarifying Questions (Sekulic et al., 2022)

- Fine-tuning a transformer-based large language model (LLM) for the task of answering clarifying questions
- DoubleHead GPT-2 with language modeling and classification losses
- Training input part 1 is given as the sequence `in[SEP]q[SEP]cq[bos]a[eos]`
  - in: textual description of the user's information need
  - q: user's query
  - cq: clarifying question asked by the system
  - a: answer given by the user
  - `[bos]` and `[eos]` are special tokens indicating the beginning and end of a sequence
  - `[SEP]` is a separation token
- Training input part 2: distractor answer and a binary label indicating which of the answers is preferable
  - Distractor answers are sampled from the training dataset heuristically
  - E.g., if the answer starts with "Yes" then the distractor answer starts with "No"
- At inference time, the above input sequence is given without the answer segment, which will be generated by the LLM

User Simulation for Conversational Search (Owoicho et al., 2023)

- Generating a variety of utterances by few-shot prompting a ChatGPT model:
  - Queries to seek information
  - Answers to clarifying questions
  - Feedback to system responses
- Note: LLM-based approaches generate answers that are fluent and natural-sounding, they work much like black boxes
  - The behaviour of the simulated user can be controlled only indirectly and only to a certain extent via training examples

Validating Simulators

- *Individual utterances*: commonly, human raters evaluate the generated responses along different dimensions (e.g., naturalness, usefulness, grammar)
- *Individual dialogues*: side-by-side human evaluation
  - Assessors are given transcripts of two conversations, in random order
  - They have to guess which of the two is the generated by a human
- *A collection of generated dialogues*:
  - *High-level dialogue features*: avg. dialogue length, ratio of user vs. system actions, etc.
  - *Dialogue style*: distribution of dialogue acts, user cooperativeness (proportion of slot values provided when requested), etc.
  - *Dialogue efficiency*: success (or task completion) rate, reward, completion time, etc.

Summary

- User simulation offers a sound and scalable means of automatic evaluation of information access systems
- Traditional evaluation measures may be viewed as naive simulators
- Simulating various interactions with search systems (queries, scanning, clicks, document processing, stopping)
- Validating user simulators
- Simulating interactions with conversational assistants

Reading

- Balog and Zhai. User Simulation for Evaluating Information Access Systems. *Foundations and Trends in Information Retrieval*, 2024. `https://arxiv.org/abs/2306.08550`