DYNAMIC SEARCH MODELS AND APPLICATIONS

Jiyun Luo

Sep 2012 — May 2018
Department of Computer Science
Georgetown University
**Dynamic Search**

E.g. A user wants to know old US coins. Relevant information includes value of old US coins, types of old US coins, old US silver dollar, how to start collecting old US coins, how to sell old US coins and how to buy them, where to buy those coins.

[TREC’13 session 9]

- $q_1$ – old US coins
- $q_2$ – collecting old US coins
- $q_3$ – selling old US coins
- $q_4$ – selling old “USA coins”

**Dynamic search**

- aim to find relevant documents for a session of multiple queries
Dynamic Search

E.g. A user wants to know old US coins. Relevant information includes value of old US coins, types of old US coins, old US silver dollar, how to start collecting old US coins, how to sell old US coins and how to buy them, where to buy those coins.

[TREC’13 session 9]

- Dynamic search
  - aim to find relevant documents for a session of multiple queries

- The information need
  - complex or vague
    - usually contains multiple subtopics

- $q_1$ – old US coins
- $q_2$ – collecting old US coins
- $q_3$ – selling old US coins
- $q_4$ – selling old “USA coins”
**Dynamic Search**

E.g. A user wants to know old US coins. Relevant information includes value of old US coins, types of old US coins, old US silver dollar, how to start collecting old US coins, how to sell old US coins and how to buy them, where to buy those coins.

[TREC’13 session 9]

- Dynamic search
  - aim to find relevant documents for a session of multiple queries

- The information need
  - complex or vague
  - may change and evolve during the search process

$q_1$ – old US coins

$q_2$ – collecting old US coins

$q_3$ – selling old US coins

$q_4$ – selling old “USA coins”

How to collect old US coins?

How to sell them?
**Dynamic Search**

E.g. A user wants to know old US coins. Relevant information includes value of old US coins, types of old US coins, old US silver dollar, how to start collecting old US coins, how to sell old US coins and how to buy them, where to buy those coins.

- $q_1$ – old US coins
  - Coin Dealer : Midwestmint.com...
  - Rare Coins & Old US Coins
- $q_2$ – collecting old US coins
  - The United States Mint
- $q_3$ – selling old US coins
- $q_4$ – selling old “USA coins”

- **Trial-and-error**
  - through repeated, varied attempts until
    - success or is stopped by the user

- **Temporal dependency between search iterations**
**Dynamic Search**

E.g. A user wants to know old US coins. Relevant information includes value of old US coins, types of old US coins, old US silver dollar, how to start collecting old US coins, how to sell old US coins and how to buy them, where to buy those coins.

- $q_1$ – old US coins  
  - Coin Dealer : Midwestmint.com...
  - Rare Coins & Old US Coins

- $q_2$ – collecting old US coins  
  - The United States Mint

- $q_3$ – selling old US coins

- $q_4$ – selling old “USA coins”

- Rich interactions
  - Query formulation
  - Document clicks
  - Document examination
  - Mouse movements
  - etc.

[TREC’13 session 9]
TREC Session Tracks 2011 - 2014

<session num="9" starttime="0">
  <topic>
    <interaction num="1" starttime="37.628318">
      <query>old us coins</query>
      <results>
        <result rank="1">
          <url>http://www.midwestmint.com/</url>
          <title>Coin Dealer : Midwestmint.com : ...</title>
          <snippet>... Coin Appraisal We are coin buyers! ...</snippet>
        </result>
        ...
      </results>
      <clicked>
        <click num="1" starttime="130.840147" endtime="170.664415">
          <rank>1</rank>
        </click>
        ...
      </clicked>
    </interaction>
    <interaction num="2" starttime="310.98016">
      <query>collecting old us coins</query>
      ...
    </interaction>
  </topic>
  <currentquery starttime="458.023971">
    <query>selling old "usa coins"
  </currentquery>
</session>
EXISTING APPROACHES FOR DYNAMIC SEARCH

- **Whole session relevance feedback**
  - utilize whole session query reformulation and user clicks to enhance relevance
    - Levine et al. SIGIR’17
    - Guan et al. SIGIR’13

- **Diversification**
  - increase result diversity to cover multiple subtopics
    - Zhang et al. TREC DD Track’17
    - Moraes et al. ICTIR’17

- **Novelty**
  - punish pre-viewed documents to enhance result novelty
    - Jiang et al. TREC Session Track’12
An ideal Dynamic Search model

- Model information need hidden behind user queries and other interactions and its dynamic changes
- Handle the temporal dependency
- Model rich user interactions
- Set up a reward mechanism to guide the trial-and-error search process
REINFORCEMENT LEARNING (RL)

- Designed for modeling trial-and-error settings
- They are to learn from repeated, varied attempts which are continued until success.
- The learner (also known as agent) learns from its dynamic interactions with the world
  - rather than from a labeled dataset as in supervised learning.
- They assume that the system's current state depends on the previous state and acts in a non-deterministic manner
CHALLENGES OF RL IN IR

○ Formulation of the Problem
  ● What are the states?
    ◦ What changes with each time step?
  ● Who are the agents and what are their actions?
    ◦ Who changes the state and how do they change it?
  ● What are the rewards?
    ◦ How can we measure feedback or effectiveness?

○ Efficiency
  ● RL training is computational expensive.
    ◦ how to improve its efficiency without losing much effectiveness?

○ How to better infer searcher inner states?
Solutions

- Formulation of the Problem
  - dynamic search as a dual-agent stochastic game
    - two agents: user agent and search engine agent work together
      - maximize the long term reward — fulfill the information need

- Improve efficiency
  - another RL framework: direct policy learning

- How to better infer searcher inner states?
  - user struggle
Research Statement

- I proposed effective and efficient retrieval models for dynamic search
  - based on reinforcement learning frameworks

- I studied RL elements from an IR perspective
  - including states, actions, rewards, and policies
  - focus on providing solutions to the challenges when modeling dynamic search
OUTLINE

- A POMDP framework — Win-Win Search
- A direct policy learning framework
- Detecting user struggle state
- Implementation of a dynamic search tool
PARTIALLY OBSERVABLE MARKOV DECISION PROCESS (POMDP)

\[ s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \ldots \]

\[ a_0, o_1, r_0 \rightarrow a_1, o_2, r_1 \rightarrow a_2, o_3, r_2 \]

\(^1\)R. D. Smallwood et. al., ‘73
PARTIALLY OBSERVABLE MARKOV DECISION PROCESS (POMDP)

- Elements
  - Hidden states

\[ \text{R. D. Smallwood et. al., ’73} \]
PARTIALLY OBSERVABLE MARKOV DECISION PROCESS (POMDP)

- **Elements**
  - Hidden states
  - **Beliefs**
    - e.g.
      - \( b(s_0) = 0.2, \ b(s_1) = 0.4, \ b(s_2) = 0.2, \ b(s_3) = 0.2 \)

\(^1\)R. D. Smallwood et. al., ‘73
**Partially Observable Markov Decision Process (POMDP)**

- **Elements**
  - Hidden states
  - Beliefs
  - Actions

---

^1 R. D. Smallwood et. al., ‘73
PARTIALLY OBSERVABLE MARKOV DECISION PROCESS (POMDP)

Elements
- Hidden states
- Beliefs
- Actions
- Rewards

\[ s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \ldots \]
- \( a_0 \) to \( r_0 \)
- \( a_1 \) to \( r_1 \)
- \( a_2 \) to \( r_2 \)

R. D. Smallwood et. al., ‘73
PARTIALLY OBSERVABLE MARKOV DECISION PROCESS (POMDP)

○ Elements
  • Hidden states
  • Beliefs
  • Actions
  • Rewards
  • Observations

1R. D. Smallwood et. al., '73
PARTIALLY OBSERVABLE MARKOV DECISION PROCESS (POMDP)

- **Elements**
  - Hidden states
  - Beliefs
  - Actions
  - Rewards
  - Observations

- **Markov Chain**

\[ s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{\ldots} \]

\[ o_1 \quad o_2 \quad o_3 \quad r_0 \quad r_1 \quad r_2 \]

\[^1\text{R. D. Smallwood et. al., '73}\]
**Partially Observable Markov Decision Process (POMDP)**

- **Elements**
  - Hidden states
  - Beliefs
  - Actions
  - Rewards
  - Observations

- **Markov Chain**
- **Goal**
  - Long Term Optimization

---

1R. D. Smallwood et. al., ‘73
**Partially Observable Markov Decision Process (POMDP)**

- **Elements**
  - Hidden states
  - Beliefs
  - Actions
  - Rewards
  - Observations

- **Markov Chain**
- **Goal**
  - Long Term Optimization

- **Policy** $\pi(s) \rightarrow a$

---

1. R. D. Smallwood et. al., ‘73
PARTIALLY OBSERVABLE MARKOV DECISION PROCESS (POMDP)

- Elements
  - Hidden states
  - Beliefs
  - Actions
  - Rewards
  - Observations

- Markov Chain
- Goal
  - Long Term Optimization
- Policy \( \pi(s) \rightarrow a \)
- Value Function
  - Given a policy, outputs long term reward \( v(s) \)
What drives search Process forward?

- E.g. A user wants to know information about old US coins. How to start collecting old US coins? How to sell old US coins? ...

![Diagram showing POMDP Components: States Design](Attachment)
**States**

- **Two dimensions**
  - “relevance dimension” → whether the user finds relevant content in the returned documents
  - “exploration dimension” → whether the user would like to explore another subtopic.

- **Four States**
  - relevant and explorative
    - e.g. scooter price → scooter stores
  - relevant and exploiting
    - e.g. Hartford visitors → Hartford Connecticut tourism
  - non-relevant and exploiting
    - e.g. Philadelphia NYC travel → Philadelphia NYC train
  - non-relevant and explorative
    - e.g. distance New York Boston → maps.bing.com
**Actions**

- **User actions**
  - **query changes**
    - Added query terms $+\Delta q_t = q_t \setminus q_{t-1}$
    - Removed query terms $-\Delta q_t = q_{t-1} \setminus q_t$
    - Theme terms $q_{\text{theme}} = \text{Longest Common Subsequence}(q_t, q_{t-1})$
  - **clicks**
    - SAT clicks (dwell time $\geq 30$ sec)
    - **clicks**
- **Search engine actions**
  - different retrieval strategies
    - increasing, decreasing and maintaining term weights
    - adjusting parameters of different search technologies
The Observation function $O(s_t, a_t, \omega_t)$, defined as $P(\omega_t \mid s_t, a_t)$, is the probability of observing $\omega_t \in \Omega$ when agents take action $a_t$ and land on state $s_t$.

- Relevant or Non-relevant
  - $O(s_t=\text{Relevant}, \Sigma_u, \omega_t=\text{Relevant})$
    - given user action $\Sigma_u$ and the landing state is “Relevant”, the probability of observing “Relevant” (observing a SAT Click)
The Observation function \( O(s_t, a_t, \omega_t) \), defined as \( P(\omega_t | s_t, a_t) \), is the probability of observing \( \omega_t \in \Omega \) when agents take action \( a_t \) and land on state \( s_t \).

- **Relevant or Non-relevant**
  - \( O(s_t=\text{Relevant}, \Sigma_u, \omega_t=\text{Relevant}) \)
    - given user action \( \Sigma_u \) and the landing state is “Relevant”, the probability of observing “Relevant” (observing a SAT Click)

\[ \propto P(s_t=\text{Relevant} | \omega_t=\text{Relevant}) P(\omega_t=\text{Relevant} | \Sigma_u) \]

\[ \Rightarrow P(s_t=\text{Relevant} | \omega_t=\text{Relevant}) = \frac{\# \text{ of observed true relevant}}{\# \text{ of observed relevant}} \]
**Observation Function**

- The Observation function $O(s_t, a_t, \omega_t)$, defined as $P(\omega_t | s_t, a_t)$, is the probability of observing $\omega_t \in \Omega$ when agents take action $a_t$ and land on state $s_t$.

- Relevant or Non-relevant
  - $O(s_t=\text{Relevant}, \Sigma_u, \omega_t=\text{Relevant})$
    - given user action $\Sigma_u$ and the landing state is “Relevant”, the probability of observing “Relevant” (observing a SAT Click)
    
    $\propto P(s_t=\text{Relevant} | \omega_t=\text{Relevant})P(\omega_t=\text{Relevant} | \Sigma_u)$

    $\Rightarrow P(s_t=\text{Relevant} | \omega_t=\text{Relevant}) = \frac{\# \text{of observed true relevant}}{\# \text{of observed relevant}}$

  - $s_t$ is likely to be $\begin{cases} \text{Relevant} & \text{if a SAT Click in } D_{t-1} \\ \text{Non-Relevant} & \text{otherwise.} \end{cases}$
**Observation Function**

- The Observation function
  - Exploration or Exploitation
    - $O(s_t = \text{Exploration}, a_u = \Delta q_t, \Sigma_se = D_{t-1}, \omega_t = \text{Exploration})$
    - $\propto P(s_t = \text{Exploration}|\omega_t = \text{Exploration})$
    - $\times P(\omega_t = \text{Exploration}|\Delta q_t, D_{t-1})$

- $s_t$ is likely to be
  - Exploitation if (newly added term is from $D_{t-1}$)
    - or (no term change from $q_{t-1}$)
  - Exploration if (added term is from nowhere)
    - or (no added term, but removed term from $q_{t-1}$)
The Observation function

- Exploration or Exploitation
  - $O(s_t = \text{Exploration}, a_u = \Delta q_t, \Sigma_{se} = D_{t-1}, \omega_t = \text{Exploration})$
  - $\propto P(s_t = \text{Exploration}|\omega_t = \text{Exploration})$
  - $\times P(\omega_t = \text{Exploration}|\Delta q_t, D_{t-1})$

- $s_t$ is likely to be
  - Exploitation if (newly added term is from $D_{t-1}$)
  - or (no term change from $q_{t-1}$)
  - Exploration if (added term is from nowhere)
  - or (no added term, but removed term from $q_{t-1}$)

The final observation function

- the product of the two observation functions for relevant and exploration
  - assume relevant and exploration are independent variables
Belief updates

- At every search iteration the belief state $b$ is updated when a new observation is obtained.
  - based on the observation function

The belief update function is

$$b_{t+1}(s_j) = \frac{P(s_j | \omega_t, a_t, b_t) \sum_{s_i \in S} P(s_j | s_i, a_t, b_t) b_t(s_i)}{P(\omega_t | a_t, b_t)}$$
REWARDS

- The reward
  - clicks, relevance score, etc.
    - it measures the gain received by the user from examining $D_t$
Rewards

- The reward
  - clicks, relevance score, etc.
    - it measures the gain received by the user from examining $D_t$

- The long term reward function for the search engine agent

$$Q_{se}(b, a) = \sum_{s \in S} b(s)R(s, a) + \gamma \sum_{\omega \in \Omega} P(\omega | b, a, \Sigma_{se})P(\omega | b, \Sigma_u)a \max_{a} Q_{se}(b', a)$$

The reward

Discounted future reward
EXPERIMENTS

- **Tasks**: TREC 2011 to 2014 Session Tracks

<table>
<thead>
<tr>
<th>TREC Session Dataset Statistics.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TREC 2011</td>
</tr>
<tr>
<td>#Sessions</td>
</tr>
<tr>
<td>#Search topics</td>
</tr>
<tr>
<td>#Queries</td>
</tr>
<tr>
<td>Avg. session length</td>
</tr>
<tr>
<td>Max session length</td>
</tr>
<tr>
<td>Avg. #sessions per topic</td>
</tr>
<tr>
<td>#Sessions w/ length &lt;4</td>
</tr>
<tr>
<td>#Sessions w/ length 4~10</td>
</tr>
<tr>
<td>#Sessions w/ length &gt;10</td>
</tr>
</tbody>
</table>
Baselines

- Lemur (language model + Dirichlet smoothing)
- PRF (pseudo relevance feedback with top 20 ranked docs)
- Rocchio (relevance feedback using top 10 ranked docs)
- Rocchio-CLK (relevance feedback using previously clicked docs)
- Rocchio-SAT (relevance feedback using previously SAT-clicked docs)
- QCM+DUP [Guan et al. SIGIR’ 13]
- QCM+SAT [Zhang et al. SIGIR’ 13]
- TREC median and best (scores across all runs)
- The original run
  - the retrieval list provided by TREC log data.
- Win-Win (proposed algorithm)
METRICS

- **Search Accuracy scores**
  - Normalized Discounted Cumulative Gain (nDCG)
  - nDCG@10
  - Mean Average Precision (MAP)
  - normalized Expected Reciprocal Rank (nERR@10)

- **Immediate Search Accuracy**
  - nDCG@10 and nERR@10 at each search iteration i

- **Bigger values of these metrics mean better retrieval performance**
## Search Accuracy

<table>
<thead>
<tr>
<th>Approach</th>
<th>nDCG@10</th>
<th>nDCG</th>
<th>MAP</th>
<th>nERR@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lemur</td>
<td>0.3642</td>
<td>0.2873</td>
<td>0.1202</td>
<td>0.4555</td>
</tr>
<tr>
<td>TREC median</td>
<td>0.3407</td>
<td>0.2577</td>
<td>0.1198</td>
<td>0.4155</td>
</tr>
<tr>
<td>TREC best</td>
<td>0.4540</td>
<td>0.4118</td>
<td>0.1909</td>
<td>0.5519</td>
</tr>
<tr>
<td>PRF</td>
<td>0.3547</td>
<td>0.2753</td>
<td>0.1184</td>
<td>0.4158</td>
</tr>
<tr>
<td>Rocchio</td>
<td>0.3681</td>
<td>0.2900</td>
<td>0.1270</td>
<td>0.4338</td>
</tr>
<tr>
<td>Rocchio-CLK</td>
<td>0.4126</td>
<td>0.3221</td>
<td>0.1559</td>
<td>0.4980</td>
</tr>
<tr>
<td>Rocchio-SAT</td>
<td>0.3921</td>
<td>0.3052</td>
<td>0.1395</td>
<td>0.4781</td>
</tr>
<tr>
<td>QCM+DUP</td>
<td>0.4161</td>
<td>0.3118</td>
<td>0.1451</td>
<td>0.4865</td>
</tr>
<tr>
<td>QCM SAT</td>
<td>0.4230</td>
<td>0.3175</td>
<td>0.1531</td>
<td>0.4928</td>
</tr>
<tr>
<td><strong>Win-Win</strong></td>
<td>0.5252†</td>
<td>0.3651†</td>
<td>0.1715†</td>
<td>0.6272†</td>
</tr>
</tbody>
</table>

### TREC 2011 Session Track

<table>
<thead>
<tr>
<th>Approach</th>
<th>nDCG@10</th>
<th>nDCG</th>
<th>MAP</th>
<th>nERR@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lemur</td>
<td>0.1043</td>
<td>0.1102</td>
<td>0.0398</td>
<td>0.1409</td>
</tr>
<tr>
<td>TREC median</td>
<td>0.1531</td>
<td>0.1357</td>
<td>0.0574</td>
<td>0.2049</td>
</tr>
<tr>
<td>TREC best</td>
<td>0.1706</td>
<td>0.2445</td>
<td>0.1008</td>
<td>0.2259</td>
</tr>
<tr>
<td>PRF</td>
<td>0.0764</td>
<td>0.0971</td>
<td>0.0339</td>
<td>0.1045</td>
</tr>
<tr>
<td>Rocchio</td>
<td>0.0783</td>
<td>0.1000</td>
<td>0.0348</td>
<td>0.1130</td>
</tr>
<tr>
<td>Rocchio-CLK</td>
<td>0.1161</td>
<td>0.1253</td>
<td>0.0430</td>
<td>0.1697</td>
</tr>
<tr>
<td>Rocchio-SAT</td>
<td>0.1002</td>
<td>0.1100</td>
<td>0.0409</td>
<td>0.1311</td>
</tr>
<tr>
<td>QCM+DUP</td>
<td>0.1555†</td>
<td>0.1380†</td>
<td>0.0568†</td>
<td>0.2229†</td>
</tr>
<tr>
<td>QCM SAT</td>
<td>0.1440†</td>
<td>0.1212</td>
<td>0.0494</td>
<td>0.2111†</td>
</tr>
<tr>
<td><strong>Win-Win</strong></td>
<td>0.2137†</td>
<td>0.1639†</td>
<td>0.0632†</td>
<td>0.3147†</td>
</tr>
</tbody>
</table>

### TREC 2012 Session Track

<table>
<thead>
<tr>
<th>Approach</th>
<th>nDCG@10</th>
<th>nDCG</th>
<th>MAP</th>
<th>nERR@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lemur</td>
<td>0.2622</td>
<td>0.2745</td>
<td>0.1342</td>
<td>0.3004</td>
</tr>
<tr>
<td>TREC median</td>
<td>0.2344</td>
<td>0.2241</td>
<td>0.1214</td>
<td>0.2831</td>
</tr>
<tr>
<td>TREC best</td>
<td>0.3153</td>
<td>0.2613</td>
<td>0.1557</td>
<td>0.3576</td>
</tr>
<tr>
<td>PRF</td>
<td>0.2614</td>
<td>0.2634</td>
<td>0.1293</td>
<td>0.2971</td>
</tr>
<tr>
<td>Rocchio</td>
<td>0.2489</td>
<td>0.2561</td>
<td>0.1226</td>
<td>0.2924</td>
</tr>
<tr>
<td>Rocchio-CLK</td>
<td>0.3439</td>
<td>0.3120</td>
<td>0.1610</td>
<td>0.3832</td>
</tr>
<tr>
<td>Rocchio-SAT</td>
<td>0.2864</td>
<td>0.2836</td>
<td>0.1440</td>
<td>0.3265</td>
</tr>
<tr>
<td>QCM+DUP</td>
<td>0.3368</td>
<td>0.3052</td>
<td>0.1536</td>
<td>0.3880</td>
</tr>
<tr>
<td>QCM SAT</td>
<td>0.3351</td>
<td>0.3047</td>
<td>0.1533</td>
<td>0.3867</td>
</tr>
<tr>
<td><strong>Win-Win</strong></td>
<td>0.3631†</td>
<td>0.3316†</td>
<td>0.1651†</td>
<td>0.4222†</td>
</tr>
</tbody>
</table>

### TREC 2013 Session Track

<table>
<thead>
<tr>
<th>Approach</th>
<th>nDCG@10</th>
<th>nDCG</th>
<th>MAP</th>
<th>nERR@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lemur</td>
<td>0.2169</td>
<td>0.1837</td>
<td>0.0227</td>
<td>0.2730</td>
</tr>
<tr>
<td>TREC median</td>
<td>0.1881</td>
<td>0.1620</td>
<td>0.0261</td>
<td>0.2183</td>
</tr>
<tr>
<td>TREC best</td>
<td>0.2482</td>
<td>0.2218</td>
<td>0.0366</td>
<td>0.3187</td>
</tr>
<tr>
<td>PRF</td>
<td>0.2310</td>
<td>0.1890</td>
<td>0.0249</td>
<td>0.2880</td>
</tr>
<tr>
<td>Rocchio</td>
<td>0.2351</td>
<td>0.1925</td>
<td>0.0248</td>
<td>0.2968</td>
</tr>
<tr>
<td>Rocchio-CLK</td>
<td>0.2407</td>
<td>0.1960</td>
<td>0.0248</td>
<td>0.3128</td>
</tr>
<tr>
<td>Rocchio-SAT</td>
<td>0.2051</td>
<td>0.1749</td>
<td>0.0230</td>
<td>0.2546</td>
</tr>
<tr>
<td>QCM+DUP</td>
<td>0.2422</td>
<td>0.1881</td>
<td>0.0254</td>
<td>0.3112</td>
</tr>
<tr>
<td>QCM SAT</td>
<td>0.2419</td>
<td>0.1884</td>
<td>0.0254</td>
<td>0.3116</td>
</tr>
<tr>
<td><strong>Win-Win</strong></td>
<td>0.3076†</td>
<td>0.2339†</td>
<td>0.0282</td>
<td>0.3916†</td>
</tr>
</tbody>
</table>

### TREC 2014 Session Track

<table>
<thead>
<tr>
<th>Approach</th>
<th>nDCG@10</th>
<th>nDCG</th>
<th>MAP</th>
<th>nERR@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lemur</td>
<td>0.3076</td>
<td>0.2339</td>
<td>0.0282</td>
<td>0.3916</td>
</tr>
</tbody>
</table>

- In TREC 2011 to 2014, Win-Win search is better than all other systems in terms of all metrics.
Immediate Search Accuracy

- In TREC 2011 to 2014, Win-Win search
  - immediate nDCG@10 score > the original run in almost every iteration.
  - immediate nERR@10 score > the original run in almost every iteration.
- In TREC 2011 to 2014, Win-Win Search quickly converges to its optimal whole session performance.
  - Users can fulfill the information need quicker when using Win-Win.
Design POMDP Elements is an Art

- **State design options**
  - fixed or varying number of states
    - four states in Win-Win
    - queries as states

- **Action design options**
  - technology selection
  - all possible ranking lists
  - etc.

- **Reward design options**
  - explicit or implicit feedback

- Compare different design options in terms of effectiveness and efficiency
  - Luo et al. ECIR’15
OUTLINE

- A POMDP framework — Win-Win Search
- A direct policy learning framework
- Detecting user struggle state
- Implementation of a dynamic search tool
**DIRECT POLICY LEARNING**

<table>
<thead>
<tr>
<th>Approach</th>
<th>Wall Clock</th>
<th>CPU Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lemur</td>
<td>$2.6 \times 10^3$ s</td>
<td>$7.0 \times 10^{12}$</td>
</tr>
<tr>
<td>QCM+DUP [Guan et al. SIGIR’13]</td>
<td>$5.6 \times 10^3$ s</td>
<td>$1.5 \times 10^{13}$</td>
</tr>
<tr>
<td>Win-Win</td>
<td>$2.5 \times 10^4$ s</td>
<td>$6.8 \times 10^{13}$</td>
</tr>
</tbody>
</table>

- Addresses high complexity of RL in IR
  - directly learns mappings from observations to actions
    - skips states, beliefs
  - flatten the model structure (a more down-to-the-earth model)
    - ... but, still complex enough to be interesting
  - less model complexity leads to higher efficiency
A DIRECT POLICY LEARNING FRAMEWORK

- At each search iteration, the search engine maximizes long-term rewards (value function)

\[
V_\theta(s_0) = E\left( \sum_{t=0}^{\infty} \gamma^t r(t) | s_0 \right)
\]

- Learns a direct mapping from observations to actions

\[
V_\theta(s_0) = \sum_{t=0}^{\infty} \gamma^t \sum_{h \in H} P(h|\theta)r(t, h)
\]
**Defining a History**

- History: the record of a session from the search iteration 0 to the current iteration \( t \)

- A chain of events happening in a session
  - the dynamic changes of states, actions, observations, and rewards in a session

\[
h_t = [h_{t-1}, C_t, T_t, q_t, \Delta q_t, D_t]
\]

- three phases in each iteration
  - a browse phase, a query phase, a rank phase
Direct Policy Learning for Dynamic Search

1: \textbf{procedure} DPL(H, \epsilon)
2: \hspace{1em} \triangleright H is the training history set. \epsilon is a threshold.
3: \hspace{1em} \theta_3 \leftarrow \text{random}(0, 1)
4: 
5: \textbf{repeat}
6: \hspace{1em} Sample history \( h \) from \( H \):
7: \hspace{2em} q_0, D_0, C_0, T_0 \leftarrow \emptyset, r(1, h) \leftarrow 0, \nabla \theta_3 \leftarrow 0
8: 
9: \hspace{1em} \textbf{for} \( t = 1 \) to \( \text{len}(h) \) \textbf{do}
10: \hspace{2em} o_{rank} \leftarrow D_{t-1}, n_1 \leftarrow o_{rank}
11: 
12: \hspace{2em} \triangleright a_{browse} \text{ is performed by the user}
13: \hspace{2em} o_{browse} \leftarrow (C_t, T_t), n_2 \leftarrow o_{browse}
14: 
15: \hspace{2em} r(t, h) \leftarrow \text{CalculateReward}(D'_t, o_{browse}, h)
16: 
17: \hspace{2em} \triangleright a_{query} \text{ is performed by the user}
18: \hspace{2em} o_{query} \leftarrow \text{GetQueryChange}(q_t, q_{t-1})
19: \hspace{2em} n_3 \leftarrow (o_{rank}, o_{browse}, o_{query})
20: 
21: \hspace{2em} Sample a search engine action \( a_{rank} \)
22: \hspace{2em} \sim P(a_{rank} | n_3, \theta_3)
23: \hspace{2em} D'_t \leftarrow \text{DocRanking}(a_{rank})
24: 
25: \hspace{2em} \nabla \theta_3 \leftarrow \text{UpdateGradient}(r(t, h), D'_t, n_3, \theta_3)
26: \hspace{2em} \theta_3 \leftarrow \theta_3 + \nabla \theta_3
27: \hspace{1em} \textbf{end for}
28: \hspace{1em} \textbf{until} \ \nabla \theta_3 < \epsilon \ \text{or} \ \ H \ \text{is running out}
procedure DPL(H, ε)
  \( \triangleright H \) is the training history set. \( \epsilon \) is a threshold.
  \( \theta_3 \leftarrow \text{random}(0, 1) \)

 repeat
  Sample history \( h \) from \( H \):
  \( q_0, D_0, C_0, T_0 \leftarrow \emptyset, r(1, h) \leftarrow 0, \triangledown \theta_3 \leftarrow 0 \)

  for \( t = 1 \) to \( \text{len}(h) \) do
    \( o_{\text{rank}} \leftarrow D_{t-1}, n_1 \leftarrow o_{\text{rank}} \)

    \( \triangleright \) \( a_{\text{browse}} \) is performed by the user
    \( o_{\text{browse}} \leftarrow (C_t, T_t), n_2 \leftarrow o_{\text{browse}} \)

    \( r(t, h) \leftarrow \text{CalculateReward}(D'_t, o_{\text{browse}}, h) \)

    \( \triangleright \) \( a_{\text{query}} \) is performed by the user
    \( o_{\text{query}} \leftarrow \text{GetQueryChange}(q_t, q_{t-1}) \)
    \( n_3 \leftarrow (o_{\text{rank}}, o_{\text{browse}}, o_{\text{query}}) \)

    Sample a search engine action \( a_{\text{rank}} \)
    \( \sim P(a_{\text{rank}}|n_3, \theta_3) \)
    \( D'_t \leftarrow \text{DocRanking}(a_{\text{rank}}) \)

    \( \triangledown \theta_3 \leftarrow \text{UpdateGradient}(r(t, h), D'_t, n_3, \theta_3) \)
    \( \theta_3 \leftarrow \theta_3 + \triangledown \theta_3 \)

  end for
  until \( \triangledown \theta_3 < \epsilon \) or \( H \) is running out
Direct Policy Learning for Dynamic Search

1: **procedure** DPL($H$, $\epsilon$)  
2: $\triangleright$ $H$ is the training history set. $\epsilon$ is a threshold.  
3: $\theta_3 \leftarrow \text{random}(0, 1)$  
4:  
5: **repeat**  
6: Sample history $h$ from $H$:  
7: $q_0, D_0, C_0, T_0 \leftarrow \emptyset, r(1, h) \leftarrow 0, \nabla \theta_3 \leftarrow 0$  
8:  
9: **for** $t = 1$ to len($h$) **do**  
10: $o_{\text{rank}} \leftarrow D_{t-1}, n_1 \leftarrow o_{\text{rank}}$  
11:  
12: $\triangleright$ a$_{\text{browse}}$ is performed by the user  
13: $o_{\text{browse}} \leftarrow (C_t, T_t), n_2 \leftarrow o_{\text{browse}}$  
14:  
15: $r(t, h) \leftarrow \text{CalculateReward}(D'_t, o_{\text{browse}}, h)$  
16:  
17: $\triangleright$ a$_{\text{query}}$ is performed by the user  
18: $o_{\text{query}} \leftarrow \text{GetQueryChange}(q_t, q_{t-1})$  
19: $n_3 \leftarrow (o_{\text{rank}}, o_{\text{browse}}, o_{\text{query}})$  
20:  
21: Sample a search engine action a$_{\text{rank}}$  
22: $\sim P(a_{\text{rank}}|n_3, \theta_3)$  
23: $D'_t \leftarrow \text{DocRanking}(a_{\text{rank}})$  
24:  
25: $\nabla \theta_3 \leftarrow \text{UpdateGradient}(r(t, h), D'_t, n_3, \theta_3)$  
26: $\theta_3 \leftarrow \theta_3 + \nabla \theta_3$  
27: **end for**  
28: **until** $\nabla \theta_3 < \epsilon$ or $H$ is running out
Direct Policy Learning for Dynamic Search

1: \textbf{procedure} DPL(\(H, \epsilon\))
2: \hskip 1em \text{\(\triangleright\) \(H\) is the training history set. \(\epsilon\) is a threshold.}
3: \hskip 1em \(\theta_3 \leftarrow random(0, 1)\)
4: \hskip 1em 
5: \hskip 1em repeat
6: \hskip 2em Sample history \(h\) from \(H\):
7: \hskip 2em \(q_0, D_0, C_0, T_0 \leftarrow \emptyset, r(1, h) \leftarrow 0, \nabla \theta_3 \leftarrow 0\)
8: \hskip 1em 
9: \hskip 2em \textbf{for} \(t = 1\) to \(len(h)\) \textbf{do}
10: \hskip 3em \(o_{\text{rank}} \leftarrow D_{t-1}, n_1 \leftarrow o_{\text{rank}}\)
11: \hskip 1em 
12: \hskip 2em \text{\(\triangleright\) \(a_{\text{browse}}\) is performed by the user}
13: \hskip 2em \(o_{\text{browse}} \leftarrow (C_t, T_t), n_2 \leftarrow o_{\text{browse}}\)
14: \hskip 1em 
15: \hskip 2em \(r(t, h) \leftarrow \text{CalculateReward}(D'_t, o_{\text{browse}}, h)\)
16: \hskip 1em 
17: \hskip 2em \text{\(\triangleright\) \(a_{\text{query}}\) is performed by the user}
18: \hskip 2em \(o_{\text{query}} \leftarrow \text{GetQueryChange}(q_t, q_{t-1})\)
19: \hskip 2em \(n_3 \leftarrow (o_{\text{rank}}, o_{\text{browse}}, o_{\text{query}})\)
20: \hskip 1em 
21: \hskip 2em \text{Sample a search engine action \(a_{\text{rank}}\)}
22: \hskip 2em \(\sim P(a_{\text{rank}}|n_3, \theta_3)\)
23: \hskip 2em \(D'_t \leftarrow \text{DocRanking}(a_{\text{rank}})\)
24: \hskip 1em 
25: \hskip 2em \(\nabla \theta_3 \leftarrow \text{UpdateGradient}(r(t, h), D'_t, n_3, \theta_3)\)
26: \hskip 2em \(\theta_3 \leftarrow \theta_3 + \nabla \theta_3\)
27: \hskip 1em \textbf{end for}
28: \hskip 1em \textbf{until} \(\nabla \theta_3 < \epsilon\) or \(H\) is running out
Direct Policy Learning for Dynamic Search

1: \textbf{procedure} DPL(H, \epsilon)
2: \triangleright H is the training history set. \epsilon is a threshold.
3: \theta_3 \leftarrow \text{random}(0, 1)
4: 
5: \textbf{repeat}
6: \quad \text{Sample history } h \text{ from } H:
7: \quad q_0, D_0, C_0, T_0 \leftarrow \emptyset, r(1, h) \leftarrow 0, \nabla \theta_3 \leftarrow 0
8: 
9: \quad \textbf{for} t = 1 \text{ to } \text{len}(h) \textbf{ do}
10: \quad \quad o_{\text{rank}} \leftarrow D_{t-1}, n_1 \leftarrow o_{\text{rank}}
11: \quad \textbf{Browse}
12: \quad \quad \triangleright \text{a}_{\text{browse}} \text{ is performed by the user}
13: \quad \quad o_{\text{browse}} \leftarrow (C_t, T_t), n_2 \leftarrow o_{\text{browse}}
14: \quad \quad r(t, h) \leftarrow \text{CalculateReward}(D'_t, o_{\text{browse}}, h)
15: \quad \textbf{Query}
16: \quad \quad \triangleright \text{a}_{\text{query}} \text{ is performed by the user}
17: \quad \quad o_{\text{query}} \leftarrow \text{GetQueryChange}(q_t, q_{t-1})
18: \quad \quad n_3 \leftarrow (o_{\text{rank}}, o_{\text{browse}}, o_{\text{query}})
19: \quad \quad \text{Sample a search engine action } a_{\text{rank}}
20: \quad \quad \sim P(a_{\text{rank}} \mid n_3, \theta_3)
21: \quad \quad D'_t \leftarrow \text{DocRanking}(a_{\text{rank}})
22: \quad \quad \nabla \theta_3 \leftarrow \text{UpdateGradient}(r(t, h), D'_t, n_3, \theta_3)
23: \quad \quad \theta_3 \leftarrow \theta_3 + \nabla \theta_3
24: \quad \textbf{end for}
25: \textbf{until} \nabla \theta_3 < \epsilon \text{ or } H \text{ is running out}
Direct Policy Learning for Dynamic Search

1: \textbf{procedure} \text{DPL}(H, \epsilon)
2: \quad \triangleright H \text{ is the training history set. } \epsilon \text{ is a threshold.}
3: \quad \theta_3 \leftarrow \text{random}(0, 1)
4: 
5: \quad \textbf{repeat}
6: \quad \quad \text{Sample history } h \text{ from } H:
7: \quad \quad q_0, D_0, C_0, T_0 \leftarrow \emptyset, r(1, h) \leftarrow 0, \nabla\theta_3 \leftarrow 0
8: 
9: \quad \quad \textbf{for} t = 1 \text{ to } \text{len}(h) \text{ do}
10: \quad \quad \quad o_{\text{rank}} \leftarrow D_{t-1}, n_1 \leftarrow o_{\text{rank}}
11: 
12: \quad \quad \quad \triangleright a_{\text{browse}} \text{ is performed by the user}
13: \quad \quad \quad o_{\text{browse}} \leftarrow (C_t, T_t), n_2 \leftarrow o_{\text{browse}}
14: 
15: \quad \quad \quad r(t, h) \leftarrow \text{CalculateReward}(D'_t, o_{\text{browse}}, h)
16: 
17: \quad \quad \quad \triangleright a_{\text{query}} \text{ is performed by the user}
18: \quad \quad \quad o_{\text{query}} \leftarrow \text{GetQueryChange}(q_t, q_{t-1})
19: \quad \quad \quad n_3 \leftarrow (o_{\text{rank}}, o_{\text{browse}}, o_{\text{query}})
20: 
21: \quad \quad \text{Sample a search engine action } a_{\text{rank}}
22: \quad \quad \quad \sim P(a_{\text{rank}}|n_3, \theta_3)
23: \quad \quad \quad D'_t \leftarrow \text{DocRanking}(a_{\text{rank}})
24: \quad \quad \quad \nabla\theta_3 \leftarrow \text{UpdateGradient}(r(t, h), D'_t, n_3, \theta_3)
25: \quad \quad \quad \theta_3 \leftarrow \theta_3 + \nabla\theta_3
26: \quad \quad \textbf{end for}
27: \quad \textbf{until } \nabla\theta_3 < \epsilon \text{ or } H \text{ is running out}
Direct Policy Learning for Dynamic Search

1: **procedure** DPL($H$, $\epsilon$)
2: \hspace{1em} $\triangleright$ $H$ is the training history set. $\epsilon$ is a threshold.
3: \hspace{1em} $\theta_3 \leftarrow \text{random}(0, 1)$
4:
5: \hspace{1em} **repeat**
6: \hspace{2em} Sample history $h$ from $H$:
7: \hspace{2em} $q_0, D_0, C_0, T_0 \leftarrow \emptyset$, $r(1, h) \leftarrow 0$, $\nabla \theta_3 \leftarrow 0$
8: \hspace{1em} **for** $t = 1$ to \text{len}(h) **do**
9: \hspace{2em} $o_{\text{rank}} \leftarrow D_{t-1}$, $n_1 \leftarrow o_{\text{rank}}$
10: \hspace{2em} $\triangleright$ $a_{\text{browse}}$ is performed by the user
11: \hspace{2em} $o_{\text{browse}} \leftarrow (C_t, T_t)$, $n_2 \leftarrow o_{\text{browse}}$
12: \hspace{2em} $r(t, h) \leftarrow \text{CalculateReward}(D'_t, o_{\text{browse}}, h)$
13: \hspace{1em} $\triangleright$ $a_{\text{query}}$ is performed by the user
14: \hspace{2em} $o_{\text{query}} \leftarrow \text{GetQueryChange}(q_t, q_{t-1})$
15: \hspace{2em} $n_3 \leftarrow (o_{\text{rank}}, o_{\text{browse}}, o_{\text{query}})$
16: \hspace{1em} Sample a search engine action $a_{\text{rank}}$
17: \hspace{2em} $\sim P(a_{\text{rank}}|n_3, \theta_3)$
18: \hspace{2em} $D'_t \leftarrow \text{DocRanking}(a_{\text{rank}})$
19: \hspace{2em} $\nabla \theta_3 \leftarrow \text{UpdateGradient}(r(t, h), D'_t, n_3, \theta_3)$
20: \hspace{2em} $\theta_3 \leftarrow \theta_3 + \nabla \theta_3$
21: \hspace{1em} **end for**
22: \hspace{1em} **until** $\nabla \theta_3 < \epsilon$ or $H$ is running out

Feature function: $\phi(a_{\text{rank}}, n_3)$

- **Query Features**
  - Test if a search term $w \in q_t$ and $w \in q_{t-1}$
  - # of times that a term $w$ occurs in $q_1, q_2, \ldots, q_t$

- **Query-Document Features**
  - Test if a search term $w \in +\Delta q_t$ and $w \in D_{t-1}$
  - Test if a document $d$ contains a term $w \in -\Delta q_t$
  - tf idf score of a document $d$ to $q_t$

- **Click Features**
  - Test if there are SAT-Clicks in $D_{t-1}$
  - # of times a document being clicked in the current session
  - # of seconds a document being viewed and reviewed in the current session

- **Query-Document-Click Features**
  - Test if $q_i$ leads to SAT-Clicks in $D_i$, where $i = 0, \ldots, t-1$

- **Session Features**
  - position at the current session
EXPERIMENT: EFFICIENCY

<table>
<thead>
<tr>
<th>Approach</th>
<th>Wall Clock</th>
<th>CPU Cycle</th>
<th>Speed Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win-Win</td>
<td>$2.5 \times 10^4s$</td>
<td>$6.8 \times 10^{13}$</td>
<td>1.00</td>
</tr>
<tr>
<td>Win-Win-CLK</td>
<td>$2.5 \times 10^4s$</td>
<td>$6.8 \times 10^{13}$</td>
<td>1.00</td>
</tr>
<tr>
<td>QCM+DUP</td>
<td>$5.6 \times 10^3s$</td>
<td>$1.5 \times 10^{13}$</td>
<td>4.46</td>
</tr>
<tr>
<td>DPL-Upper</td>
<td>$3.2 \times 10^3s$</td>
<td>$8.7 \times 10^{12}$</td>
<td>7.81</td>
</tr>
<tr>
<td>DPL</td>
<td>$3.1 \times 10^3s$</td>
<td>$8.4 \times 10^{12}$</td>
<td>8.06</td>
</tr>
<tr>
<td>Lemur</td>
<td>$2.6 \times 10^3s$</td>
<td>$7.0 \times 10^{12}$</td>
<td>9.62</td>
</tr>
</tbody>
</table>

- Lemur > DPL > QCM+DUP > Win-Win in Wall Clock and CPU Cycles
  - DPL-Upper: the upper bounder of DPL, which uses nDCG as reward
### TREC 2014 Session Track

<table>
<thead>
<tr>
<th>Model</th>
<th>nDCG@10</th>
<th>$\sigma^2$</th>
<th>MAP</th>
<th>$\sigma^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lemur</td>
<td>0.2169</td>
<td>0.0470</td>
<td>0.0227</td>
<td>0.0009</td>
</tr>
<tr>
<td>Win-Win-CLK</td>
<td>0.2241</td>
<td>0.0420</td>
<td>0.0248</td>
<td>0.0010</td>
</tr>
<tr>
<td>QCM+DUP (baseline)</td>
<td>0.2422</td>
<td>0.0451</td>
<td>0.0254</td>
<td>0.0010</td>
</tr>
<tr>
<td>DPL</td>
<td>0.2608†</td>
<td>0.0435</td>
<td>0.0269</td>
<td>0.0010</td>
</tr>
<tr>
<td>Win-Win</td>
<td>0.3076†</td>
<td>0.0549</td>
<td>0.0282†</td>
<td>0.0011</td>
</tr>
<tr>
<td>DPL-Upper</td>
<td>0.3266†</td>
<td>0.0606</td>
<td>0.0134</td>
<td>0.0002</td>
</tr>
</tbody>
</table>

- **DPL** > QCM+DUP, but < Win-Win
- **DPL-Upper** achieves the best nDCG@10 score
- **DPL** achieves a good balance between accuracy and efficiency
OUTLINE

- A POMDP framework — Win-Win Search
- A direct policy learning framework
- Detecting user struggle state
- Implementation of a dynamic search tool
SEARCHER STRUGGLE

- A user experiences difficulty in finding information at some moment during the search
  - an emotional feeling
  - does not directly related to search failure
  - e.g.

22:07:40  Query: delete myspace
22:07:43  Click: www.tech-recipes.com/rx/1649/myspace_how...
22:07:48  Click: myspace.com/delete
22:13:19  Click: www.askdavetaylor.com/how_can_i_delete__...
22:15:58  Query: permanently delete myspace
22:16:16  Query: how to permanently delete myspace account
22:16:45  Click: www.accountcleaner.com/white-list/how-to-...
22:40:12  Query: delete myspace photos
22:40:23  Click: answers.yahoo.com/question/index?qid=200...
Why does modeling user struggle matter?

- Understand whether users are under the struggling state can:
  - help estimate the confidence of the user queries
  - help search engine decide search strategies
    - query recommendation
    - document recommendation
EXISTING APPROACHES

- Use effort-based features to predict user struggle
  - number of queries
    - Aula et al. CHI’10, Feild et al. SIGIR’10
  - number of clicked URLs, dwell time
    - Feild et al. SIGIR’10, Hassan et al. WSDM’14
  - diversity of queries and clicked URLs
    - Hassan et al. WSDM’14
- Assume that the struggle is linked to spending a great amount of effort to search
- However, the same features are also used to predict engagement (opposite feeling to struggle)
  - using effort-based features alone is problematic
    - Edwards and Kelly SIGIR’17
THE REVERSAL THEORY

- What causes the inconsistency?
  - one cause may be different motivations
    - e.g. when searching for entertainment, don’t mind effort

The Reversal Theory (Gerrig et al. Psychology and life. 2015)

<table>
<thead>
<tr>
<th>Means–ends</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal-oriented</td>
<td>Non-goal-oriented</td>
</tr>
<tr>
<td>• Serious. Focus on future goals and achievement. Tend to avoid arousal, risk &amp; anxiety.</td>
<td>• Playful, passion and fun. Focus on current moment. Seek excitement and entertainment.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rules</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-explorative</td>
<td>Explorative</td>
</tr>
<tr>
<td>• Conforming. Value rules and tradition. Tend to operate within rules and expectations.</td>
<td>• Rebellious. Value innovation and changes. Like to explore new possibilities.</td>
</tr>
</tbody>
</table>
EXAMPLES OF EFFORT-BASED FEATURES

- Effort to Query
  - number of queries/unique queries in the current session
  - avg. number of terms/characters per query

- Effort in Query Re-formulation
  - avg. cosine similarity between all queries and the first query
  - avg. number of terms added/deleted/substituted from the previous query
  - avg. edit distance per adjacent query pair

- Effort to Click
  - total and avg. number of clicks/SAT clicks in the session
  - percentage of abandoned queries

- Search Topic
  - the topic of the search session according to the ODP labels

- Effort in Click Diversity
  - percentage of unique URLs/domains among all clicked URLs/domains
  - number and percentage of revisited clicks

- Effort to Read
  - log(1 + time passed until the first SAT click)
  - log(1 + avg. dwell time per click/query)

- Effort to Scroll
  - mobile/PC screen size
  - total and avg. number of actions at the bottom of a mobile/PC screen

- Effort in Rare Query & Rare Clicks
  - log(1 + avg. query occurrences in an external log)
  - avg. query clickthrough rate/SAT clickthrough rate in an external log
User effort in non-goal-oriented tasks are significantly bigger than effort spent in goal-oriented tasks.

Explorative users averagely spend more effort than non-explorative users.
**REMOVE TASK AND USER BIASES**

- Assume the user effort spent in a goal-oriented task is
  \[ X \sim \mathcal{N}(\mu_1, \sigma_1^2) \]
- Assume the user effort spent in a non-goal-oriented task is
  \[ X' \sim \mathcal{N}(\mu_2, \sigma_2^2) \]
- Implicitly assume that \( X' \) can be linearly transformed from \( X \) using
  \[ X' = \frac{\sigma_2}{\sigma_1} X + \mu_2 - \frac{\sigma_2}{\sigma_1} \mu_1 \]

- Normalize feature value of a non-goal-oriented task using
  \[ \text{normalized value}(x) = \sigma_1 \times \frac{\text{original value}(x') - \mu_2}{\sigma_2} + \mu_1 \]

- User Biases are removed using similar approach
EXPERIMENTS

- Predict user struggle before and after removing task and user biases
  - dataset: Yahoo! search log data

- ZeroRule: label all instances using the majority label of the datasets
- LM: Logistical Regression
- SVM
- MART_SUB: MART model trained on features in [Hassan et al. WSDM’14]
- MART
- NeuralNetwork: a three-layer neural network model
  - 10-fold Cross-validation for all models

- Metrics
  - Classification Accuracy
  - Precision and Recall for the positive label (“Struggle”)
  - Precision and Recall for the negative label (“Non-Struggle”)
## BEFORE AND AFTER BIASES REMOVED

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Positive Precision</th>
<th>Positive Recall</th>
<th>Negative Precision</th>
<th>Negative Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZeroRule</td>
<td>0.7341</td>
<td>—</td>
<td>0.0000</td>
<td>0.7341</td>
<td>1.0000</td>
</tr>
<tr>
<td>LM</td>
<td>0.8418</td>
<td>0.7320</td>
<td>0.6665†</td>
<td>0.8799</td>
<td>0.9098</td>
</tr>
<tr>
<td>MART_SUB</td>
<td>0.8483</td>
<td>0.7718</td>
<td>0.6408</td>
<td>0.8733</td>
<td>0.9277</td>
</tr>
<tr>
<td>NeuralNetwork</td>
<td>0.8564</td>
<td>0.7404</td>
<td>0.7085†</td>
<td>0.8961†</td>
<td>0.9011</td>
</tr>
<tr>
<td>SVM</td>
<td>0.8596</td>
<td>0.7948†</td>
<td>0.6606†</td>
<td>0.8813</td>
<td>0.9100</td>
</tr>
<tr>
<td>MART</td>
<td>0.8727†</td>
<td>0.7949†</td>
<td>0.7273†</td>
<td>0.9001†</td>
<td>0.9284</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM</td>
<td>0.8598 (2.1%↑)</td>
<td>0.9057 (0.5%↓)</td>
</tr>
<tr>
<td>MART_SUB</td>
<td>0.8753 (3.2%↑)</td>
<td>0.9480 (2.2%↑)</td>
</tr>
<tr>
<td>NeuralNetwork</td>
<td>0.8825 (3.0%↑)</td>
<td>0.9400↑ (4.3%↑)</td>
</tr>
<tr>
<td>SVM</td>
<td>0.8883 (3.3%↑)</td>
<td>0.9504↑ (4.4%↑)</td>
</tr>
<tr>
<td>MART</td>
<td><strong>0.9037↑</strong> (3.6%↑)</td>
<td><strong>0.9532↑</strong> (2.7%↑)</td>
</tr>
</tbody>
</table>

- Among above models, MART is the most effective predicting model.
- † indicates statistically significant improvement over baseline MART_SUB.
### Before and After Biases Removed

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Positive Precision</th>
<th>Positive Recall</th>
<th>Negative Precision</th>
<th>Negative Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZeroRule</td>
<td>0.7341</td>
<td>—</td>
<td>0.0000</td>
<td>0.7341</td>
<td>1.0000</td>
</tr>
<tr>
<td>LM</td>
<td>0.8418</td>
<td>0.7320</td>
<td>0.6665†</td>
<td>0.8799</td>
<td>0.9098</td>
</tr>
<tr>
<td>MART_SUB</td>
<td>0.8483</td>
<td>0.7718</td>
<td>0.6408</td>
<td>0.8733</td>
<td>0.9277</td>
</tr>
<tr>
<td>NeuralNetwork</td>
<td>0.8564</td>
<td>0.7404</td>
<td>0.7085†</td>
<td>0.8961†</td>
<td>0.9011</td>
</tr>
<tr>
<td>SVM</td>
<td>0.8596</td>
<td>0.7948†</td>
<td>0.6606†</td>
<td>0.8813</td>
<td>0.9100</td>
</tr>
<tr>
<td>MART</td>
<td>0.8727†</td>
<td>0.7949†</td>
<td>0.7273†</td>
<td>0.9001†</td>
<td>0.9284</td>
</tr>
<tr>
<td>LM</td>
<td>0.8598†</td>
<td>0.7508</td>
<td>0.7560†</td>
<td>0.9079</td>
<td>0.9057†</td>
</tr>
<tr>
<td>MART_SUB</td>
<td>0.8753</td>
<td>0.8312†</td>
<td>0.6866†</td>
<td>0.8894</td>
<td>0.9480</td>
</tr>
<tr>
<td>NeuralNetwork</td>
<td>0.8825†</td>
<td>0.8148†</td>
<td>0.7239</td>
<td>0.9038</td>
<td>0.9400†</td>
</tr>
<tr>
<td>SVM</td>
<td>0.8883†</td>
<td>0.8428†</td>
<td>0.7255†</td>
<td>0.9026</td>
<td>0.9504†</td>
</tr>
<tr>
<td>MART</td>
<td>0.9037†</td>
<td>0.8599†</td>
<td>0.7797†</td>
<td>0.9183†</td>
<td>0.9532†</td>
</tr>
</tbody>
</table>

- After biases are removed, all models performance increases.
  - accuracy achieves 0.9037
OUTLINE

- A POMDP framework — Win-Win Search
- A direct policy learning framework
- Detecting user struggle state
- Implementation of a dynamic search tool
Provide 2 dynamic search functions

1) Win-Win

2) Win-Win-Subtopic
IMPLEMENTING DETAILS

○ States
  ● three dimensions ($2 \times 2 \times 2 = 8$ states)
    ◦ 1) relevance 2) exploration 3) struggle
  ● simplify the inference method for the struggling state

$$P(\text{struggle}) = \begin{cases} \frac{n}{N} & \text{if } n < N \\ 1 & \text{otherwise} \end{cases}$$

○ n: # of search trials without relevance judgment received
○ N: # of search functions ($N = 6$)
  ◦ the closer n is to N, the more likely the user is struggling right now
IMPLEMENTING DETAILS

- Search Engine Actions
  - actions as in Win-Win Search
    - ad-hoc retrieval strategy and the brute-force strategy
  - new actions
    - CurrentSubtopic: searching using the current subtopic name as the query
    - AllSubtopics: combining all subtopic names as a query
    - CurrentPassage: using the most recently judged relevant passage to form a query
    - AllPassages: combining all relevant passages under the current topic as a query
IMPLEMENTING DETAILS

- **Rewards**
  - 1) nDCG
  - 2) weighted $\alpha$-nDCG

\[
\text{weighted } \alpha nDCG = (1 - P(\text{struggle})) \times nDCG + P(\text{struggle}) \times \alpha nDCG
\]

- $P(\text{struggle})$: the probability that the user is currently under the struggling state
EXPERIMENTS

- Compare the relevant documents and passages founded by each retrieval functions
  - default retrieval algorithms in
    - Lemur, Solr, and Terrier
  - Rocchio
  - Win-Win
  - Win-Win with the Struggling State
    - reward function: weighted $\alpha$-nDCG
  - Win-Win-Subtopic
    - Win-Win Search at sub topic level
## Experiments

- † indicates statistically significant improvement over Rocchio

<table>
<thead>
<tr>
<th></th>
<th>Recall</th>
<th>Precision@10</th>
<th>MAP</th>
<th>nDCG@10</th>
<th>α-nDCG@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solr</td>
<td>0.42</td>
<td>0.36</td>
<td>0.22</td>
<td>0.20</td>
<td>0.41</td>
</tr>
<tr>
<td>Lemur</td>
<td>0.47</td>
<td>0.35</td>
<td>0.24</td>
<td>0.25</td>
<td>0.40</td>
</tr>
<tr>
<td>Terrier</td>
<td>0.64</td>
<td>0.52</td>
<td>0.34</td>
<td>0.33</td>
<td>0.57</td>
</tr>
<tr>
<td>Rocchio</td>
<td>0.75</td>
<td>0.69</td>
<td>0.41</td>
<td>0.43</td>
<td>0.68</td>
</tr>
<tr>
<td>Win-Win-Subtopic</td>
<td>0.81†</td>
<td>0.73†</td>
<td>0.45†</td>
<td>0.39</td>
<td>0.64</td>
</tr>
<tr>
<td>Win-Win</td>
<td>0.89†</td>
<td>0.79†</td>
<td>0.57†</td>
<td>0.58†</td>
<td>0.77†</td>
</tr>
<tr>
<td>Win-Win-Struggle</td>
<td>0.89†</td>
<td>0.83†</td>
<td>0.59†</td>
<td>0.74†</td>
<td>0.80†</td>
</tr>
</tbody>
</table>
CONCLUSIONS

○ In my PhD study, I proposed novel RL frameworks to model dynamic search effectively and efficiently.

○ Provided novel methods to infer searcher decision making states.

○ Implemented a dynamic search application to support a real life dynamic search task.
TAKE HOME MESSAGE

Reinforcement Learning frameworks are suitable for modeling dynamic search, however the design of its elements from IR perspective is an art and requires careful thinking.
THANK YOU

Q & A

Jiyun Luo

Department of Computer Science
Georgetown University
Washington, DC, USA
jl1749@georgetown.edu
Future Work

- Include other decision making states
  - user desire of revisiting webpages
  - personal interest and knowledge background
  - etc.

- Include more forms of user feedback
  - implicit/explicit feedback
  - feedback from physical sensors

- Examine other utility functions besides clicks and nDCG as reward functions
  - optimize other aspects of search
    - e.g. maximizing matching between retrieval content and user reading proficiency
Introduction
**Dynamic Search Workshops**

**TREC Session Track Task**

- Corpus: ClueWeb09, ClueWeb12
  - TREC 2011 and 2012
    - ClueWeb09 CatB
      - 50 million English webpages crawled in 2009
  - TREC 2013 and 2014
    - ClueWeb12 CatB
      - 50 million English webpages crawled in 2012
Win-Win
**The Win-win Search Framework**

- a dual-agent stochastic game, $<S, A_u, A_{se}, \Sigma_u, \Sigma_{se}, \Omega_u, \Omega_{se}, O, B, T, R>$.  

<table>
<thead>
<tr>
<th>Name</th>
<th>symbol</th>
<th>Meanings</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>$S$</td>
<td>the four hidden decision states</td>
</tr>
<tr>
<td>user action</td>
<td>$A_u$</td>
<td>add/remove/keep query terms</td>
</tr>
<tr>
<td>search engine action</td>
<td>$A_{se}$</td>
<td>increase/decrease/keep term weights, adjust search techniques, etc.</td>
</tr>
<tr>
<td>message from user to search engine</td>
<td>$\Sigma_u$</td>
<td>clicked and SAT clicked documents</td>
</tr>
<tr>
<td>message from search engine to user</td>
<td>$\Sigma_{se}$</td>
<td>top k returned documents</td>
</tr>
<tr>
<td>user's observation</td>
<td>$\Omega_u$</td>
<td>observations that the user makes from the world</td>
</tr>
<tr>
<td>search engine's observation</td>
<td>$\Omega_{se}$</td>
<td>observations that the search engine makes from the world and from the user</td>
</tr>
<tr>
<td>user reward</td>
<td>$R_u$</td>
<td>relevant information the user gains from reading the documents</td>
</tr>
<tr>
<td>search engine reward</td>
<td>$R_{se}$</td>
<td>nDCG that the search gains by returning documents</td>
</tr>
<tr>
<td>belief state</td>
<td>$B$</td>
<td>belief states generated from the belief updater and shared by both agents</td>
</tr>
</tbody>
</table>

---

**Win-Search Framework**

○ a dual-agent stochastic game, $<S, A_u, A_{se}, \Sigma_u, \Sigma_{se}, \Omega_u, \Omega_{se}, O, B, T, R>$.  

**Table**

<table>
<thead>
<tr>
<th>Name</th>
<th>symbol</th>
<th>Meanings</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>$S$</td>
<td>the four hidden decision states</td>
</tr>
<tr>
<td>user action</td>
<td>$A_u$</td>
<td>add/remove/keep query terms</td>
</tr>
<tr>
<td>search engine action</td>
<td>$A_{se}$</td>
<td>increase/decrease/keep term weights, adjust search techniques, etc.</td>
</tr>
<tr>
<td>message from user to search engine</td>
<td>$\Sigma_u$</td>
<td>clicked and SAT clicked documents</td>
</tr>
<tr>
<td>message from search engine to user</td>
<td>$\Sigma_{se}$</td>
<td>top k returned documents</td>
</tr>
<tr>
<td>user's observation</td>
<td>$\Omega_u$</td>
<td>observations that the user makes from the world</td>
</tr>
<tr>
<td>search engine's observation</td>
<td>$\Omega_{se}$</td>
<td>observations that the search engine makes from the world and from the user</td>
</tr>
<tr>
<td>user reward</td>
<td>$R_u$</td>
<td>relevant information the user gains from reading the documents</td>
</tr>
<tr>
<td>search engine reward</td>
<td>$R_{se}$</td>
<td>nDCG that the search gains by returning documents</td>
</tr>
<tr>
<td>belief state</td>
<td>$B$</td>
<td>belief states generated from the belief updater and shared by both agents</td>
</tr>
</tbody>
</table>
POMDP COMPONENTS

- Observations
  the observation function $O(s_j, a_t, \omega_t)$, defined as $P(\omega_t | s_j, a_t)$, is the probability of observing $\omega_t \in \Omega$ when agents take action $a_t$ and land on state $s_j$.

- Relevant or Non-relevant
  - $O(s_t = \text{Rel}, \Sigma_u, \omega_t = \text{Rel}) \propto P(s_t = \text{Rel} | \omega_t = \text{Rel})P(\omega_t = \text{Rel} | \Sigma_u)$
  - $s_t$ is likely to be $\begin{cases} \text{Relevant} & \text{if } \exists \ d \in D_{t-1} \text{ and } d \text{ is SATClicked} \\ \text{Non-Relevant} & \text{otherwise} \end{cases}$
  - $P(s_t = \text{Rel} | \omega_t = \text{Rel}) = \frac{\# \text{ of observed true relevant}}{\# \text{ of observed relevant}}$
    - “# of observed true relevant” is the number of times where the previously returned document set $D_{t-1}$ contain at least one SAT clicks and those SAT clicked documents are indeed relevant documents in the ground truth.
    - “# of observed relevant” is the number of times where $D_{t-1}$ contains at least one SAT clicks.
POMDP COMPONENTS

- Observations
  - Exploration or Exploitation

\[ O(s_t = \text{Exploration}, a_u = \Delta q_t, \Sigma_{se} = D_{t-1}, \omega_t = \text{Exploration}) \]
\[ \propto P(s_t = \text{Exploration} | \omega_t = \text{Exploration}) \times P(\omega_t = \text{Exploration} | \Delta q_t, D_{t-1}) \]

- \( s_t \) is likely to be

\[ \begin{align*}
\text{Exploration} & \quad \text{if} \left( +\Delta q_t \neq \emptyset \text{ and } +\Delta q_t \notin D_{t-1} \right) \\
& \quad \text{or} \left( +\Delta q_t = \emptyset \text{ and } -\Delta q_t \neq \emptyset \right) \\
\text{Exploitation} & \quad \text{if} \left( +\Delta q_t \neq \emptyset \text{ and } +\Delta q_t \in D_{t-1} \right) \\
& \quad \text{or} \left( +\Delta q_t = \emptyset \text{ and } -\Delta q_t = \emptyset \right)
\end{align*} \]

\[ P(s_t = \text{Exploration} | \omega_t = \text{Exploration}) = \]
\[ \frac{\text{# of observed true explorations}}{\text{# of observed explorations}} \]

\[ P(\omega_t = \text{Exploration} | a = \text{'delete\_term'}, \Delta q_t, D_{t-1}) = \]
\[ \frac{\text{# of observed explorations due to deleting terms}}{\text{# of observations of deleting terms}} \]
BELIEF UPDATES

○ At every search iteration the belief state $b$ is updated when a new observation is obtained.

The belief update function is

$$b_{t+1}(s_j) = P(s_j|\omega_t, a_t, b_t)$$

$$= \frac{P(\omega_t|s_j, a_t, b_t) \sum_{s_i \in S} P(s_j|s_i, a_t, b_t) b_t(s_i)}{P(\omega_t|a_t, b_t)}$$

$O(s_j, a_t, \omega_t)$ $T(s_i, a_t, s_j)$

$T(s_i, a_t, s_j) = \frac{\#\text{Transition}(s_i, a_t, s_j)}{\#\text{Transition}(s_i, a_t, s^*)}$

○ $\#\text{Transition}(s_i, a_t, s_j)$ is the number of all transitions that starts at state $s_i$, takes action $a_t$, and lands at state $s_j$.

○ $\#\text{Transition}(s_i, a_t, s^*)$ is the number of all transitions that starts at state $s_i$ and lands at any state by action $a_t$. 
JOINT OPTIMIZATION AND RETRIEVAL

- The long term reward function for the search engine agent

\[ Q_{se}(b, a) = \sum_{s \in S} b(s)R(s, a) + \gamma \sum_{\omega \in \Omega} P(\omega \mid b, a_u, \Sigma_{se})P(\omega \mid b, \Sigma_u) \max_a Q_{se}(b', a) \]

- The long term reward function for the user agent

\[ Q_u(b, a_u) = R(s, a_u) + \gamma \sum a_u T(s_t \mid s_{t-1}, D_{t-1}) \max_{s_{t-1}} Q_u(s_{t-1}, a_u) \]

- Joint optimization

\[ a_{se} = \arg\max_a (Q_{se}(b, a) + Q_u(b, a_u)) \]
IMMEDIATE SEARCH ACCURACY

- In TREC 2011 and 2012, win-win search
  - immediate nDCG@10 score > the original run from Iteration 2.
  - immediate nERR@10 score > the original run from Iteration 2.
Table 3.7: Number of Selections of Each Search Technology by Win-Win, TREC 2011–2014.

<table>
<thead>
<tr>
<th></th>
<th>Increase weights</th>
<th>Decrease weights</th>
<th>QCM+DUP</th>
<th>PRF</th>
<th>Ad-hoc</th>
<th>All Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>TREC 2011</td>
<td>11</td>
<td>7</td>
<td>46</td>
<td>11</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>TREC 2012</td>
<td>19</td>
<td>10</td>
<td>59</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TREC 2013</td>
<td>11</td>
<td>10</td>
<td>53</td>
<td>12</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>TREC 2014</td>
<td>15</td>
<td>7</td>
<td>55</td>
<td>10</td>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>
POMDP Design
Design POMDP elements is an art

- MDP in dynamic search [SIGIR’13, Guan et al.]
- States: Queries

“old US coins”

- Actions:
  - User actions:
    - Add/remove/unchange the query terms
  - Search Engine actions:
    - Increase/ decrease /remain term weights
Design POMDP elements is an art

- POMDP in re-ranking multi-page search result (WWW’13, Jin et al.)
- States: a document’s true relevance distribution
  - # of states: number of existed documents
- Actions: each possible ranking list as one action
  - # of actions: factorial of the number of retrieved documents
What types of design choices are available?

Which one of them are the best option for dynamic search?
STATE DESIGN OPTIONS

○ (S1) Fixed number of states
  ● use two binary relevance states
    ◦ “Relevant” or “Non-Relevant” (Zhang et al. SIGIR’13)
  ● use four states
    ◦ Win-Win Search

○ (S2) Varying number of states
  ● model queries as states, n queries -> n states
    ◦ SIGIR’13, Guan et al.
  ● infinity states
    ◦ document relevance score distribution as state
      ◦ WWW’13, Jin et al.
ACTION DESIGN OPTIONS

- (A1) Technology Selection
  - a meta-level modeling of actions
    - implement multiple search methods, and select the best methods for each query
    - select the best parameters for each method
      - Win-Win Search

- (A2) Term Weight Adjustment
  - adjusted term weights
    - SIGIR’13, Guan et al.

- (A3) Ranked List
  - one possible ranking of a list of documents is one single action
    - WWW’13, Jin et al.
**Reward Function Design Options**

- **(R1) Explicit Feedback**
  - Rewards generated from user’s relevance assessments.
    - nDCG, MAP, etc

- **(R2) Implicit Feedback**
  - Use implicit feedback obtained from user behavior
    - Clicks, SAT clicks
EXPERIMENTS

- Systems under comparing (8 out of 12 combinations)
  - Win-Win, $S_1A_1R_2$, and $S_2A_1R_1$
    - Win-Win Search and its two variants
  - $S_1A_3R_2$
    - reimplementation of Zhang et al. SIGIR’14
  - QCM+DUP, and $S_1A_2R_1$
    - reimplementation of QCM (Guan et al. SIGIR’13) and its variant
  - UCAIR
    - reimplementation of Shen et al. CIKM’05
  - IES
    - reimplementation of Jin et al. WWW’13
ACCURACY VS. EFFICIENCY

Accuracy (nDCG@10) vs. Efficiency (Wall Clock Time in Seconds)

(TREC 2011)

(TREC 2012)
Accuracy vs. Efficiency

- Accuracy tends to increase when efficiency decreases
Accuracy vs. Efficiency

- Accuracy tends to increase when efficiency decreases
- UCAIR strikes a good balance between accuracy and efficiency
Accuracy vs. Efficiency

- Accuracy tends to increase when efficiency decreases.
- UCAIR strikes a good balance between accuracy and efficiency.
- Win-Win gives impressive accuracy with a fair degree of efficiency.
Conclusions are consistent on TREC 2013 and 2014 Session Tracks
DESIGN CHOICE RECOMMENDATION

- If focus on accuracy
DESIGN CHOICE RECOMMENDATION

- If focus on accuracy
- If time limit is within one hour
**DESIGN CHOICE RECOMMENDATION**

- If focus on accuracy
- If time limit is within one hour
- If want the balance of accuracy and efficiency
## Search Engine Agent’s Actions

<table>
<thead>
<tr>
<th>∈ D_{i-1}</th>
<th>action</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_{\text{theme}}$</td>
<td>Y</td>
<td>increase</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>increase</td>
</tr>
<tr>
<td>$+\Delta q$</td>
<td>Y</td>
<td>decrease</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>increase</td>
</tr>
<tr>
<td>$-\Delta q$</td>
<td>Y</td>
<td>decrease</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>No change</td>
</tr>
</tbody>
</table>

[Guan, Zhang, and Yang SIGIR 2013]
**Query Change Retrieval Model (QCM)**

Bellman Equation gives the optimal value for an MDP:

$$V^*(s) = \max_a R(s,a) + \gamma \sum_{s'} P(s' | s,a) V^*(s')$$

The reward function is used as the document relevance score function and is tweaked backwards from Bellman equation:

$$\text{Score}(q_i, d) = P(q_i | d) + \gamma \sum_a P(q_i | q_{i-1}, D_{i-1}, a) \max_{D_{i-1}} P(q_{i-1} | D_{i-1})$$

- Document relevant score
- Current reward/relevance score
- Query Transition model
- Maximum past relevance
Calculating the Transition Model

- According to Query Change and Search Engine Actions

[Guan, Zhang, and Yang SIGIR 2013]

\[
\text{Score}(q_i, d) = \log P(q_i \mid d) + \alpha \sum_{t \in \text{theme}} [1 - P(t \mid d_{i-1}^*)] \log P(t \mid d) \\
- \beta \sum_{\substack{t \in +\Delta q \not\in d_{i-1}^*}} P(t \mid d_{i-1}^*) \log P(t \mid d) + \varepsilon \sum_{\substack{t \in +\Delta q \not\in d_{i-1}^*}} \text{idf}(t) \log P(t \mid d) \\
- \delta \sum_{t \in -\Delta q} P(t \mid d_{i-1}^*) \log P(t \mid d)
\]
DPL
Decompose a history

• First level: iteration by iteration

• Second level: break down an iteration into
  • browse phase
  • query phase
  • retrieval phase
Browse Phase

- Actor: the user

- It happens
  - after the search results are shown to the user
  - before the user starts to write the next query

- Records how the user perceives and examines the (previously retrieved) search results

\[ s(t) \]

\[ o_{\text{rank}}(t) \]

\[ n_1(t) \]

\[ a_{\text{browse}}(t) \]
Query Phase

- Actor: the user
- It happens
  - when the user writes a query
- Assuming the query is created based on
  - what has been seen in the browse phase
  - the information need
Rank Phase

- Actor: the search engine
- It happens
  - after the query is entered
  - before the search results are returned
- It is where the search algorithm takes place
Our objective function:

\[
V_\theta(s_0) = \sum_{t=0}^{\infty} \gamma^t \sum_{h \in H} P(h|\theta)r(t,h)
\]

where

\[
P(h|\theta) = \prod_{t=1}^{\text{len}(h)} P(o_{\text{rank}}(t), a_{\text{browse}}(t), o_{\text{browse}}(t), a_{\text{query}}(t), o_{\text{query}}(t), a_{\text{rank}}(t)|h_{t-1}, \theta)
\]

\[
\propto \prod_{t=1}^{\text{len}(h)} P(a_{\text{browse}}(t)|o_{\text{rank}}(t), \theta_1)
\]

\[
\times P(a_{\text{query}}(t)|o_{\text{browse}}(t), \theta_2)
\]

\[
\times P(a_{\text{rank}}(t)|o_{\text{browse}}(t), o_{\text{query}}(t), o_{\text{rank}}(t), \theta_3)
\]

\[
\propto \prod_{t=1}^{\text{len}(h)} \prod_{i \in \{1,2,3\}} P(a^i(t)|n_i(t), \theta_i)
\]
Action Selection Distribution

Softmax Function

\[ P(a^i|n_i, \theta_i) = \frac{e^{\theta_i \cdot \phi(a^i, n_i)}}{\sum_{a'^i} e^{\theta_i \cdot \phi(a'^i, n_i)}} \]

Gradient

\[ \frac{\partial V_\theta(s_0)}{\partial \theta_k} = \sum_{t=1}^{\infty} \gamma^t \sum_{h \in H} r(t, h) \frac{\partial P(h|\theta)}{\partial \theta_k} \]

\[ = \sum_{t=1}^{\infty} \gamma^t \sum_{h \in H} r(t, h) P(h|\theta) \]

\[ \times \sum_{i=0}^{t} \frac{\partial \ln[P(a_{\text{browse}}|n_1, \theta_1)P(a_{\text{query}}|n_2, \theta_2)P(a_{\text{rank}}|n_3, \theta_3)]}{\partial \theta_k} \]
Ranking Function

\[ P(a_{rank} | n_3, \theta_3) = \frac{e^{\theta_3 \cdot \phi(a_{rank}, n_3)}}{\sum_{a'_{rank}} e^{\theta_3 \cdot \phi(a'_{rank}, n_3)}} \]

- It originally presents the probability of selecting a (ranking) action

- In our context, the probability of selecting d to be put at the top of a ranked list under \( n_3 \) and \( \theta_3 \) at the \( t^{th} \) iteration

- Then we sort the documents by it to generate the document list
Direct Policy Learning for Dynamic Search

\begin{algorithm}
\begin{algorithmic}
\Procedure{DPL}{H, \epsilon}
\Comment{H is the training history set. \epsilon is a threshold.}
\State \theta_3 \leftarrow \text{random}(0, 1)
\Repeat
\State Sample history \( h \) from \( H \):
\State \( q_0, D_0, C_0, T_0 \leftarrow \emptyset, r(1, h) \leftarrow 0, \nabla \theta_3 \leftarrow 0 \)
\For {t = 1 to len(h)}
\State \( o_{\text{rank}} \leftarrow D_{t-1}, n_1 \leftarrow o_{\text{rank}} \)
\Comment{a_{\text{browse}} is performed by the user}
\State \( o_{\text{browse}} \leftarrow (C_t, T_t), n_2 \leftarrow o_{\text{browse}} \)
\EndFor
\State \( r(t, h) \leftarrow \text{CalculateReward}(D'_t, o_{\text{browse}}, h) \)
\Comment{a_{\text{query}} is performed by the user}
\State \( o_{\text{query}} \leftarrow \text{GetQueryChange}(q_t, q_{t-1}) \)
\State \( n_3 \leftarrow (o_{\text{rank}}, o_{\text{browse}}, o_{\text{query}}) \)
\EndRepeat
\State Sample a search engine action \( a_{\text{rank}} \)
\State \( \sim P(a_{\text{rank}}|n_3, \theta_3) \)
\State \( D'_t \leftarrow \text{DocRanking}(a_{\text{rank}}) \)
\State \( \nabla \theta_3 \leftarrow \text{UpdateGradient}(r(t, h), D'_t, n_3, \theta_3) \)
\State \( \theta_3 \leftarrow \theta_3 + \nabla \theta_3 \)
\Until {\nabla \theta_3 < \epsilon or H is running out}
\end{algorithmic}
\end{algorithm}

Updates:
\[
\nabla \theta_3 = \sum_{h \in H} \sum_{t=1}^{\text{len}(h)} \gamma^t r(t, h) \times \sum_{i=1}^{t} \left[ \phi(a_{\text{rank}}, n_3) - \phi(a'_{\text{rank}}, n_3) P(a'_{\text{rank}}|n_3, \theta_3) \right]
\]

Feature function: \( \phi(a_{\text{rank}}, n_3) \)

Query Features
\begin{itemize}
\item Test if a search term \( w \in q_t \) and \( w \in q_{t-1} \)
\item # of times that a term \( w \) occurs in \( q_1, q_2, \ldots, q_t \)
\end{itemize}

Query-Document Features
\begin{itemize}
\item Test if a search term \( w \in +\Delta q_t \) and \( w \in D_{t-1} \)
\item Test if a document \( d \) contains a term \( w \in -\Delta q_t \)
\end{itemize}

Click Features
\begin{itemize}
\item Test if there are SAT-Clicks in \( D_{t-1} \)
\item # of times a document being clicked in the current session
\item # of seconds a document being viewed and reviewed in the current session
\end{itemize}

Query-Document-Click Features
\begin{itemize}
\item Test if \( q_i \) leads to SAT-Clicks in \( D_i \), where \( i = 0, \ldots, t-1 \)
\end{itemize}

Session Features
\begin{itemize}
\item position at the current session
\end{itemize}
User Struggle Detection
Data Prepare: Extract Search Tasks

- extract cross-session search tasks from one-week-long Yahoo! search logs
  - for all query pairs \((q_i, q_j)\) generated by the same user
    - a logistic regression model to predict whether they are topic-coherent [Han et al. CIKM’ 16]
  - search task clustering
    - given \((q_i, q_j)\) are topic-coherent, if \(q_i/q_j\) belongs to a task cluster, then merge them into the same task cluster
- sampling search tasks for annotation
DATA ANNOTATION

- Two assessors annotate each task independently
- A third assessor joins to solve disagreement
- Fleiss’ Kappa score: 0.61
- Also check whether the search task contains multiple search topics
  - if yes, filter out

<table>
<thead>
<tr>
<th></th>
<th>Duration</th>
<th>#Sessions</th>
<th>#Struggle</th>
<th>#NonStruggle</th>
<th>#Query/Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile</td>
<td>Mar. 9 ~</td>
<td>1,226</td>
<td>326</td>
<td>900</td>
<td>5.58</td>
</tr>
<tr>
<td>PC</td>
<td>15, 2015</td>
<td>365</td>
<td>142</td>
<td>223</td>
<td>5.06</td>
</tr>
</tbody>
</table>
The annotating procedure is assigned to the annotation department in Yahoo!

- **Introduction:** “We have a set of user search sessions from year 2015. Each session consists of a time-stamped query followed by a click or maybe a re-query. Our goal is to just look at the search activities in a search session and levy a judgment on whether or not the searcher was struggling to find information.”

- **Demonstration:** “Sometimes I click on a link to understand what the person saw. In case that the links are dead, I will rely on the text in the URL to give me a sense of the story. After judging hundreds of these, my thoughts are: A person is not struggling when 1) she uses search engine as a bookmark, for example a user searched ‘hancock bank’ and clicked www.hancockbank.com 2) she is doing research on a topic, e.g. ‘how many chromosomes are present in interphase of meiosis?’”
The annotating procedure is assigned to the annotation department in Yahoo!

- **Demonstration:** “3) she is just looking up information, such as stock ticker prices 4) she is just checking the same thing over and over to check Facebook or email, or monitor sports results, or see if there are new Craigslist listings. 5) sometimes the initial query shows up twice with minor spelling correction. Then she clicked on a URL that seems to answer the story. Then there’s no other action. I think she found what she’s looking for, hence there is no struggle here. A person is struggling when 1) they’re not finding what they want in that initial query. We see this a lot on ambiguous queries and people’s names. 2) A person is probably struggling when they try multiple variations of a query or click into different URLs and then re-query. 3) Clicking on an ad and then re-querying also makes me think they’re struggling. 4) Then there are things like: ‘What is the search topic? Is someone just having fun and trying to find the story behind the movie? Why continue re-querying on that “true story” angle and still hitting the Wikipedia article?’ This to me feels like struggle, but I’d be hard-pressed to explain why beyond ‘This is my gut feeling’. Hence I label them as ‘Weak Struggle’.”
The annotating procedure is assigned to the annotation department in Yahoo!

- **Demonstration:** “Sometimes I can’t decide and go with the ‘Uncertain’ decision. A session of two identical queries with no click tells me nothing (unless the query is a stock name etc.). Also I can’t do anything with sessions that aren’t in English.”
Table 6.4: Mobile: Results of Detecting Struggle.
(† means a statistically significant improvement over MART_SUB [71] (t-test, p<0.05)).

<table>
<thead>
<tr>
<th></th>
<th>accuracy</th>
<th>pos. precision</th>
<th>pos. recall</th>
<th>neg. precision</th>
<th>neg. recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoClick</td>
<td>0.7276</td>
<td>0.4756</td>
<td>0.2393</td>
<td>0.7665</td>
<td>0.9044</td>
</tr>
<tr>
<td>LM</td>
<td>0.8418</td>
<td>0.7320</td>
<td>0.6665†</td>
<td>0.8799</td>
<td>0.9098</td>
</tr>
<tr>
<td>MART_SUB</td>
<td>0.8483</td>
<td>0.7718</td>
<td>0.6408</td>
<td>0.8733</td>
<td>0.9277</td>
</tr>
<tr>
<td>NeuralNetwork</td>
<td>0.8564</td>
<td>0.7404</td>
<td>0.7085†</td>
<td>0.8961†</td>
<td>0.9011</td>
</tr>
<tr>
<td>SVM</td>
<td>0.8596†</td>
<td>0.7948†</td>
<td>0.6606†</td>
<td>0.8813</td>
<td>0.9100</td>
</tr>
<tr>
<td>MART</td>
<td>0.8727†</td>
<td>0.7949†</td>
<td>0.7273†</td>
<td>0.9001†</td>
<td>0.9284</td>
</tr>
</tbody>
</table>

Table 6.5: PC: Results of Detecting Struggle.
(† means a statistically significant improvement over MART_SUB (t-test, p < 0.05)).

<table>
<thead>
<tr>
<th></th>
<th>accuracy</th>
<th>pos. precision</th>
<th>pos. recall</th>
<th>neg. precision</th>
<th>neg. recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoClick</td>
<td>0.6795</td>
<td>0.7273</td>
<td>0.2817</td>
<td>0.6710</td>
<td>0.9327</td>
</tr>
<tr>
<td>LM</td>
<td>0.8388</td>
<td>0.7681</td>
<td>0.8278</td>
<td>0.8907</td>
<td>0.8422</td>
</tr>
<tr>
<td>MART_SUB</td>
<td>0.8406</td>
<td>0.7640</td>
<td>0.8308</td>
<td>0.8939</td>
<td>0.8434</td>
</tr>
<tr>
<td>NeuralNetwork</td>
<td>0.8356</td>
<td>0.7697</td>
<td>0.8239</td>
<td>0.8826</td>
<td>0.8430</td>
</tr>
<tr>
<td>SVM</td>
<td>0.8600†</td>
<td>0.7855†</td>
<td>0.8751†</td>
<td>0.9205†</td>
<td>0.8487</td>
</tr>
<tr>
<td>MART</td>
<td>0.8732†</td>
<td>0.8139†</td>
<td>0.8615†</td>
<td>0.9124†</td>
<td>0.8788†</td>
</tr>
</tbody>
</table>
Table 6.7: PC: Results of Detecting Struggle with Biases Removed.
(† means a statistically significant improvement over the original runs without biases removed (t-test, p < 0.05)).

<table>
<thead>
<tr>
<th></th>
<th>accuracy</th>
<th>pos. precision</th>
<th>pos. recall</th>
<th>neg. precision</th>
<th>neg. recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoClick</td>
<td>0.6877</td>
<td>0.7800†</td>
<td>0.2746</td>
<td>0.6730</td>
<td>0.9507</td>
</tr>
<tr>
<td></td>
<td>(1.2%†)</td>
<td>(7.2%†)</td>
<td>(2.5%↓)</td>
<td>(0.3%†)</td>
<td>(1.9%↑)</td>
</tr>
<tr>
<td>LM</td>
<td>0.8743†</td>
<td>0.8357†</td>
<td>0.8496</td>
<td>0.9025</td>
<td>0.8912†</td>
</tr>
<tr>
<td></td>
<td>(4.2%†)</td>
<td>(8.8%†)</td>
<td>(2.6%↑)</td>
<td>(1.3%↑)</td>
<td>(5.8%↑)</td>
</tr>
<tr>
<td>MART_SUB</td>
<td>0.9014†</td>
<td>0.8641†</td>
<td>0.8692†</td>
<td>0.9208</td>
<td>0.9201†</td>
</tr>
<tr>
<td></td>
<td>(7.2%†)</td>
<td>(13.1%↑)</td>
<td>(4.6%↑)</td>
<td>(3.0%↑)</td>
<td>(9.1%↑)</td>
</tr>
<tr>
<td>NeuralNetwork</td>
<td>0.8959†</td>
<td>0.8562†</td>
<td>0.8803†</td>
<td>0.9224†</td>
<td>0.9058†</td>
</tr>
<tr>
<td></td>
<td>(7.2%↑)</td>
<td>(11.2%↑)</td>
<td>(6.8%↑)</td>
<td>(4.5%↑)</td>
<td>(7.4%↑)</td>
</tr>
<tr>
<td>SVM</td>
<td>0.9121†</td>
<td>0.8752†</td>
<td>0.8954</td>
<td>0.9374</td>
<td>0.9197†</td>
</tr>
<tr>
<td></td>
<td>(6.1%†)</td>
<td>(11.4%↑)</td>
<td>(2.3%↑)</td>
<td>(1.8%↑)</td>
<td>(8.4%↑)</td>
</tr>
<tr>
<td>MART</td>
<td>0.9260†</td>
<td>0.8926†</td>
<td>0.9156†</td>
<td>0.9462†</td>
<td>0.9336†</td>
</tr>
<tr>
<td></td>
<td>(6.0%†)</td>
<td>(9.7%↑)</td>
<td>(6.3%↑)</td>
<td>(3.7%↑)</td>
<td>(6.2%↑)</td>
</tr>
</tbody>
</table>
**Before and After Biases Removed (PC)**

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Positive Precision</th>
<th>Positive Recall</th>
<th>Negative Precision</th>
<th>Negative Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZeroRule</td>
<td>0.6110</td>
<td>—</td>
<td>0.0000</td>
<td>0.6110</td>
<td>1.0000</td>
</tr>
<tr>
<td>NeuralNetwork</td>
<td>0.8356</td>
<td>0.7697</td>
<td>0.8239</td>
<td>0.8826</td>
<td>0.8430</td>
</tr>
<tr>
<td>LM</td>
<td>0.8388</td>
<td>0.7681</td>
<td>0.8278</td>
<td>0.8907</td>
<td>0.8422</td>
</tr>
<tr>
<td>MART_SUB</td>
<td>0.8406</td>
<td>0.7640</td>
<td>0.8308</td>
<td>0.8939</td>
<td>0.8434</td>
</tr>
<tr>
<td>SVM</td>
<td>0.8600†</td>
<td>0.7855†</td>
<td>0.8751†</td>
<td>0.9205†</td>
<td>0.8487</td>
</tr>
<tr>
<td>MART</td>
<td>0.8732†</td>
<td>0.8139†</td>
<td>0.8615†</td>
<td>0.9124†</td>
<td>0.8788†</td>
</tr>
<tr>
<td>LM</td>
<td>0.8743†</td>
<td>0.8357†</td>
<td>0.8496</td>
<td>0.9025</td>
<td>0.8912†</td>
</tr>
<tr>
<td>MART_SUB</td>
<td>0.8959†</td>
<td>0.8562†</td>
<td>0.8803†</td>
<td>0.9224†</td>
<td>0.9058†</td>
</tr>
<tr>
<td>NeuralNetwork</td>
<td>0.9014†</td>
<td>0.8641†</td>
<td>0.8692†</td>
<td>0.9208</td>
<td>0.9201†</td>
</tr>
<tr>
<td>SVM</td>
<td>0.9121†</td>
<td>0.8752†</td>
<td>0.8954</td>
<td>0.9374</td>
<td>0.9197†</td>
</tr>
<tr>
<td>MART</td>
<td>0.9260†</td>
<td>0.8926†</td>
<td>0.9156†</td>
<td>0.9462†</td>
<td>0.9336†</td>
</tr>
</tbody>
</table>

Before

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Positive Precision</th>
<th>Positive Recall</th>
<th>Negative Precision</th>
<th>Negative Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM</td>
<td>0.8743†</td>
<td>0.8357†</td>
<td>0.8496</td>
<td>0.9025</td>
<td>0.8912†</td>
</tr>
<tr>
<td>MART_SUB</td>
<td>0.8959†</td>
<td>0.8562†</td>
<td>0.8803†</td>
<td>0.9224†</td>
<td>0.9058†</td>
</tr>
<tr>
<td>NeuralNetwork</td>
<td>0.9014†</td>
<td>0.8641†</td>
<td>0.8692†</td>
<td>0.9208</td>
<td>0.9201†</td>
</tr>
<tr>
<td>SVM</td>
<td>0.9121†</td>
<td>0.8752†</td>
<td>0.8954</td>
<td>0.9374</td>
<td>0.9197†</td>
</tr>
<tr>
<td>MART</td>
<td>0.9260†</td>
<td>0.8926†</td>
<td>0.9156†</td>
<td>0.9462†</td>
<td>0.9336†</td>
</tr>
</tbody>
</table>

The observations on PC platforms are consistent with those on mobile platforms.
Dynamic Search Tool Implementation
TREC Dynamic Domain Tracks 2015-2017

Task:

- involves several rounds of interactions between a user and a participating search engine system
- the user is a simulated user (called Jig) provided by the Track organizers
TREC Dynamic Domain Tracks 2015-2017

- In each task:
  - Jig initiates a query to the participating system
    - the query is the task’s topic name
  - the system retrieves 5 documents and sends them to Jig
TREC Dynamic Domain Tracks 2015-2017

- In each task:
  - Jig provides passage level relevance judgement about the 5 documents to the system
TREC Dynamic Domain Tracks 2015-2017

- In each task:
  - the system decides how to use the feedback
  - and decide either
    - retrieve 5 more documents
    - or stop the search task for the sake of saving user effort
The ground truth is essential
• for Jig to provide online relevance judgment
• and for evaluating participating systems afterwards

It is generated by annotation
Publications

Journal

Conference

Workshop

Services