ExplainAble Recommendation and Search (EARS)

Yongfeng Zhang  
Rutgers University

Jiaxin Mao  
Tsinghua University

Qingyao Ai  
UMass Amherst

Xu Chen  
Tsinghua University
Outline of the Tutorial

• Why Explainable Recommendation and Search
• A Unified View of Search, Recommendation, and Explainability
• Part 1: Explainable Recommendation
• Part 2: Explainable Search
• Summary
Why Explainable Recommendation and Search
Explainable AI on the Web

- Recent research on explainable recommendation and search is related to Explainable AI
The “Black-Box” Learning Problem

- State-of-the-art Web intelligent systems rely on advanced machine learning (deep learning) models.

- We don’t always understand what happens in the box.
  - Difficult to provide explanations for the machine outputs.
Sometimes Explanations are Important

<table>
<thead>
<tr>
<th>For Users</th>
<th>For System Designers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why did you show this result to me?</td>
<td>Why does my system give this output?</td>
</tr>
<tr>
<td>*A recommended item</td>
<td>How to conduct system diagnostics?</td>
</tr>
<tr>
<td>*A search result</td>
<td>Which component of system is wrong?</td>
</tr>
<tr>
<td>*Especially when result is <strong>personalized</strong></td>
<td>How to tune the system performance?</td>
</tr>
<tr>
<td>Why should I trust the result?</td>
<td>How to increase system robustness?</td>
</tr>
<tr>
<td>How should I take actions?</td>
<td></td>
</tr>
</tbody>
</table>
Broader Impacts of Explanations

- **Fairness Perspectives of AI Systems**
  - Asymmetric information creates unfairness
  - Users deserve reliable explanations of AI decisions to take fair actions

- **Social Justice Perspectives**
  - Sometimes absolutely fair solutions do not exist
  - At least explain to users what happens in the systems

- **New Human-Computer Interaction Paradigms**
  - Give machine an opportunity to explain itself
  - May change human behaviors in CHI, e.g., in conversational AI
  - Feed back from machine, more efficient human-machine interaction
AI Policy Perspectives

• EU General Data Protection Regulation (GDPR)
  – Article 5.2: a data controller “must be able to demonstrate that personal data are processed in a transparent manner in relation to the data subject”
  – Article 12 provides general rules on transparency, which apply to the provision of information (Articles 13-14), communications with data subjects concerning their rights (Articles 15-22), and in relation to data breaches (Article 34).

• Implications of the regulation is still to be clarified in legal practice
• Should we have AI Regulations? – A debatable problem

• Not the key focus of today’s tutorial.
Technical Perspectives

- Is it possible to develop explainable AI systems?
- Is it possible to provide accountable explanations to users (i.e., data subjects, as required in GDPR)?
- What are the technical responses to such regulations?

Widely deployed AI systems on the Web, influence nearly every Web user’s daily life. They are very good platforms to develop, verify and test explainable AI algorithms. Explainable Recommendation and Search.
A Unified View of Search, Recommendation, and Explainability
An Overview of Search Systems

- From query to documents *and explanations*
  - User information need is explicitly represented by the search query
    - Search keywords, questions, etc.

Query → Search Engine → Information Storage

- Web pages, images, videos, User Generated Contents (UGC), etc.
An Overview of Search Systems

- From query to documents and explanations
  - User information need is explicitly represented by the search query
    - Search keywords, questions, etc.

When the search algorithm is explainable
- e.g., search snippets

Diagram:
- Query -> Search Engine -> Information Storage
- Search Results
- Explanations

Search engine example:
- Web search engine - Wikipedia
  - A web search engine or Internet search engine is a software system that is designed to carry out web search (Internet search), which means to search the World...
An Overview of Search Systems

• From query to documents *and explanations*
  – User information need is explicitly represented by the search query
    • Search keywords, questions, etc.

![Diagram](image)

When the search algorithm is not quite explainable..
An Overview of Recommendation Systems

• From user to items and *explanations*
  – User information need is implicitly represented by the user profile
    • User content information, interaction history, etc.

![Diagram of Recommendation System]

- User
- Recommendation Engine
- Candidate Items
  - Products, movies, music, videos, news, friends, twits, actions, etc.
- Recommended Items
An Overview of Recommendation Systems

• From user to items and *explanations*
  – User information need is implicitly represented by the user profile
    • User content information, interaction history, etc.

When the recommendation algorithm is explainable
  e.g., Most popular recommendation
  Other explainable recommendation algorithms
An Overview of Recommendation Systems

- From user to items and *explanations*
  - User information need is implicitly represented by the user profile
    - User content information, interaction history, etc.

When the recommendation algorithm is not quite explainable...
Usually generate post-hoc explanations
Unified View of Search and Recommendation

- [Belkin and Croft, 1992] [Garcia-Molina et. al., 2011]

<table>
<thead>
<tr>
<th></th>
<th>Search</th>
<th>Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delivery Mode</td>
<td>Pull</td>
<td>Push or Pull</td>
</tr>
<tr>
<td>Beneficiary</td>
<td>User</td>
<td>User and provider</td>
</tr>
<tr>
<td>Unexpected good?</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Collective knowledge</td>
<td>Maybe</td>
<td>Maybe</td>
</tr>
<tr>
<td>Query available</td>
<td>Yes</td>
<td>Maybe</td>
</tr>
<tr>
<td>Context dependent</td>
<td>Maybe</td>
<td>Maybe</td>
</tr>
</tbody>
</table>

Courtesy Table from [2]
Unified View of Search and Recommendation

User Information
- Content: documents, products, news, etc.

Explainable Search/Recommendation Algorithm

User and context information
- Query
- User History
- User Location
- User Location

Search Results
- Recommendations

Explanations

Figure adapted from [2]
Unified View of Search and Recommendation

Query | User History
User Location | …

User and context information

Search/Recommendation Algorithm

Search Results
Recommendations

Explanations

User Information

Content: documents, products, news, etc.

Information

Figure adapted from [2]
About this tutorial

**Explainable**
Search and Recommendation Algorithms

**Non-Explainable**
Search and Recommendation Algorithms

(Post-hoc) Explanation Algorithms
Outline of the Tutorial

• Why Explainable Recommendation and Search
• A Unified View of Search, Recommendation, and Explainability
• Part 1: Explainable Recommendation
  – History Overview
  – Explainable Recommendation Methods
  – Challenges and Open Directions
• Part 2: Explainable Search
• Summary
References


Explainable Recommendation
Recommendation Systems – The 5W

- **What** to recommend: the fundamental problem of all recommendation systems.
- **When** to recommend: the research task of Time-aware recommendation
- **Where** to recommend: the research task of Location-based recommendation
- **Who** to recommend: the research task of Social recommendation
- **Why** to recommend: the research task of Explainable Recommendation
A Brief Historical Overview – How the Problem Origins

• Early approaches to recommendation were highly explainable
  – User-based Collaborative Filtering [Resnick et al. CSCW’1994]
  – Item-based Collaborative Filtering [Sarwar et al. WWW’2001]

<table>
<thead>
<tr>
<th>Title</th>
<th>Genre</th>
<th>Author</th>
<th>Type</th>
<th>Price</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Lace Reader</td>
<td>Fiction, Mystery</td>
<td>Brunonia Barry</td>
<td>Hardcover</td>
<td>49.90</td>
<td>American contemporary fiction, detective, historical</td>
</tr>
<tr>
<td>Into the Fire</td>
<td>Romance, Suspense</td>
<td>Suzanne Brockmann</td>
<td>Hardcover</td>
<td>45.90</td>
<td>American fiction, murder, neo-Nazism</td>
</tr>
</tbody>
</table>
Content-based Recommendation and Explanation

- **Item attributes**

<table>
<thead>
<tr>
<th>Title</th>
<th>Genre</th>
<th>Author</th>
<th>Type</th>
<th>Price</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Lace Reader</td>
<td>Fiction, Mystery</td>
<td>Brunonia Barry</td>
<td>Hardcover</td>
<td>49.90</td>
<td>American contemporary fiction, detective, historical</td>
</tr>
<tr>
<td>Into the Fire</td>
<td>Romance, Suspense</td>
<td>Suzanne Brockmann</td>
<td>Hardcover</td>
<td>45.90</td>
<td>American fiction, murder, neo-Nazism</td>
</tr>
</tbody>
</table>

- **User profile**

<table>
<thead>
<tr>
<th>Title</th>
<th>Genre</th>
<th>Author</th>
<th>Type</th>
<th>Price</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>Fiction</td>
<td>Brunonia, Barry, Ken Follett</td>
<td>Paperback</td>
<td>25.65</td>
<td>Detective, murder, New York</td>
</tr>
</tbody>
</table>

Simple approach

Compute the similarity of an unseen item with the user profile based on the **keyword overlap** (e.g. using Jaccard similarity)

\[
\frac{|\text{keywords}(b_i) \cap \text{keywords}(b_j)|}{|\text{keywords}(b_i) \cup \text{keywords}(b_j)|}
\]

Explanation can be naturally provided based on content information
User-based Collaborative Filtering and Explanation

- A matrix of ratings of the current user, Alice, and some other users is given

<table>
<thead>
<tr>
<th></th>
<th>Item1</th>
<th>Item2</th>
<th>Item3</th>
<th>Item4</th>
<th>Item5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>?</td>
</tr>
<tr>
<td>User1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>User2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>User3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>User4</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

- Consider each row as a user vector
- Find top-K similar users (i.e., k-nearest neighbor) based on similarity measure
  - E.g., Adjusted Cosine Similarity
    \[
    \text{sim}(a, b) = \frac{\sum_{p \in P} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P} (r_{b,p} - \bar{r}_b)^2}}
    \]
- Average similar users’ rating on the target item as prediction, recommend if a high rating

**Explanation:** Users who have similar ratings with you highly rated this item
Item-based Collaborative Filtering and Explanation

• A matrix of ratings of the current user, Alice, and some other users is given

<table>
<thead>
<tr>
<th></th>
<th>Item1</th>
<th>Item2</th>
<th>Item3</th>
<th>Item4</th>
<th>Item5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>?</td>
</tr>
<tr>
<td>User1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>User2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>User3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>User4</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

• Consider each column as an item vector
• Find top-K similar items (i.e., k-nearest item) based on similarity measure
  – E.g., Adjusted Cosine Similarity
  \[
  \text{sim}(\bar{a}, \bar{b}) = \frac{\sum_{u \in U} (r_{u,a} - \bar{r}_u)(r_{u,b} - \bar{r}_u)}{\sqrt{\sum_{u \in U} (r_{u,a} - \bar{r}_u)^2} \sqrt{\sum_{u \in U} (r_{u,b} - \bar{r}_u)^2}}
  \]
• Average similar items’ rating on the target user as prediction, recommend if a high rating

Explanation: You have highly rated items that are similar to this item

The commonly seen “based on your view history” explanation in movie review and EC
Validate Explanations based on User Surveys

- Explaining Collaborative Filtering Recommendations
  - [Herlocker et al. CSCW’2000]

21 different explanation interfaces, 78 users on MovieLens website, each user was provided with 21 recommendations, each with a different explanation.

Ask users to rate on a scale of 1-7 how likely they would go and see the movie.
The most effective explanation based on Neighbors' ratings.

User-based CF: Users who have similar ratings with you highly rated this item.

Shaded rows indicate explanations with a mean response significantly different from the base cases (two-tailed $\alpha = 0.05$).

Explanation 11 and 12 represent the base case of no additional information (focus on system: we recommend, focus on user: people are watching).
Machine Learning vs Non-Machine Learning

• Most of them are non-machine learning approaches
  – Highly explainable, but sometimes less effective in rating prediction accuracy

• Rise of Machine Learning Approaches
  – The Netflix Prize, 2006-2009
  – Netflix provided a training data
  – 100,480,507 ratings, 480,189 users, 17,770 movies
  – US$1,000,000 prize to teams that are 10%+ better than Netflix’s own algorithm for rating prediction on RMSE

https://www.netflixprize.com/
Machine Learning for Recommendation

- Why not directly minimize the rating prediction error?

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th></th>
<th></th>
<th>3</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Predict the Missing Ratings

A key task: Predict the missing ratings
Matrix Factorization for Recommendation

- One key idea of winning solutions
  - Also called Latent Factor Models
  - [Koren et al. Computer’2009]

\[
\hat{r}_{ui} = p_u^T q_i
\]

\[
\min\sum_{(u,i) \in R} (r_{ui} - p_u^T q_i)^2 + \lambda_1 \sum_u \|p_u\|^2 + \lambda_2 \sum_i \|q_i\|^2
\]

Goodness of fit

Regularization

Latent Factors
Explainability vs Accuracy

- Latent factor models
  - More accurate (directly minimize prediction error)
  - But less explainable (due to the “latent” factors)
From Shallow to Deep: More Explainability Problems

- MF is a shallow network
  - Each latent factor is a neuron
- More explainability problems from Shallow to Deep
  - No explicit meaning of the neurons, non-linearity
Explainable Recommendation

• From Know How to Know Why
  – Can we develop algorithms that are both accurate and explainable?

• Explainable Recommendation Approaches
  – Explainable Recommendation based on Matrix Factorization
  – Explainable Recommendation based on Deep Learning
  – Knowledge Graph Reasoning Approaches
  – Post-hoc and Model-agnostic Approaches
  – Others
Factorization-based Approaches

- **From latent factors to explicit factors**
  - **EFM**: Explicit factor models for explainable recommendation [Zhang et al. SIGIR’2014]
  - **L2RF**: Learning to rank features for recommendation over multiple categories [Chen et al. SIGIR’2016]
  - **MTER**: Explainable recommendation via multi-task learning in opinionated text data [Wang et al. SIGIR’2018]
Explicit Factor Model

- Explicit factor models for explainable recommendation [Zhang et al. SIGIR’2014]
  - Formally introduced the Explainable Recommendation problem

- Basic idea: To recommend an item that performs well on the features that a user concerns.
**Explicit Factor Model**

- **User-Feature Attention Matrix**

\[
X_{ij} = \begin{cases} 
0, & \text{if user } u_i \text{ did not mention feature } F_j \\
1 + (N - 1)\left(\frac{2}{1 + e^{-t_{ij}}} - 1\right), & \text{else}
\end{cases}
\]

\( t_{ij} \) is the frequency that user \( j \) mentions feature \( j \)
Explicit Factor Model

- Item-Feature Quality Matrix

\[
Y_{ij} = \begin{cases} 
0, & \text{if item } p_i \text{ is not reviewed on feature } F_j \\
\frac{N - 1}{1 + e^{-k \cdot s_{ij}}}, & \text{else}
\end{cases}
\]

- \( k \) is the frequency feature \( j \) is mentioned on item \( i \)
- \( s_{ij} \) is the average sentiment of these mentions

Sparse
Explicit Factor Model

- Integrating the Explicit and Implicit Features

\[
\begin{align*}
\text{minimize} & \quad \left\{ \|PQ^T - A\|^2_F + \lambda_x \|U_1 V^T - X\|^2_F + \lambda_y \|U_2 V^T - Y\|^2_F \\
& + \lambda_u (\|U_1\|^2_F + \|U_2\|^2_F) + \lambda_h (\|H_1\|^2_F + \|H_2\|^2_F) + \lambda_v \|V\|^2_F \right\} \\
\text{subject to} & \quad P = [U_1 \ H_1], \quad Q = [U_2 \ H_2]
\end{align*}
\]
Explicit Factor Model

- Generating recommendation list

User-based feature selection: select the top-k most cared features for vector multiplication.

For each user $i$, rank the items with the ranking score:

$$R_{ij} = \alpha \cdot \frac{\sum_{c \in C_i} \tilde{X}_{ic} \tilde{Y}_{jc}}{kN} + (1 - \alpha) \tilde{A}_{ij}$$
Explicit Factor Model

Feature-level explanation for a recommended item

You might be interested in [feature], on which this product performs well.
Learning to Rank Features

- Learning to rank features for recommendation over multiple categories [Chen et al. SIGIR’2016]
  - Generalize EFM:
    - From User-Feature and Item-Feature matrix factorization to User-Item-Feature tensor factorization: user may only like a feature over a certain item instead of globally
    - From point-wise prediction to pair-wise learning to rank: improves ranking performance

\[
\hat{T}_{uf} = \sum_{k=0}^{K-1} R_{uk}^U \cdot R_{fk}^{UF} + \sum_{k=0}^{K-1} R_{ik}^I \cdot R_{fk}^{IF} + \sum_{k=0}^{K-1} R_{uk}^U \cdot R_{ik}^I
\]

Pair-wise learning to rank over features

\[
\hat{T}_{uf_{AB}} = \hat{T}_{uf_A} - \hat{T}_{uf_B}
\]

Tensor factorization

\[
\min_{\Theta} \sum_{u \in U} \sum_{i \in I} (A_{ui} - (R_{ui}^U)^T \cdot R_{ui}^I)^2 - \lambda \sum_{u \in U} \sum_{i \in I} \sum_{f_{A \in F_{ui}^+}} \sum_{f_{B \in F_{ui}^-}} \ln \sigma(\hat{T}_{uf_{AB}}) + \lambda \|\Theta\|_F^2
\]

\(T_{uf}\) directly give us feature-level explanation (selected feature is item-specific)
Multi-Task Learning for Explainable Recommendation

- Explainable Recommendation via Multi-Task Learning in Opinionated Text Data [Wang et al. SIGIR’2018]
  - Two tasks: 1. User preference modeling for recommendation
  - 2. Opinionated content modeling for explanation
Multi-Task Learning for Explainable Recommendation

- Explainable Recommendation via Multi-Task Learning in Opinionated Text Data [Wang et al. SIGIR’2018]

- Task relatedness is captured by sharing latent factors of $U$, $I$, $F$, $O$ across the tensors.

- Improve performance of each task by multi-task learning.

- Also helps alleviate sparsity problem.
Experimental Evaluation

- Recommendation Performance [Wang et al. SIGIR’2018]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Amazon (Cellphones &amp; Accessories)</th>
<th>Yelp</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Methods</strong></td>
<td>Point-wise Learning Methods</td>
<td>Pair-wise Learning</td>
</tr>
<tr>
<td>NDCG@K</td>
<td>Most Pop</td>
<td>NMF</td>
</tr>
<tr>
<td>10</td>
<td>0.0930</td>
<td>0.1879</td>
</tr>
<tr>
<td>20</td>
<td>0.1278</td>
<td>0.0829</td>
</tr>
<tr>
<td>50</td>
<td>0.1879</td>
<td>0.1614</td>
</tr>
<tr>
<td><strong>Dataset</strong></td>
<td>Yelp</td>
<td></td>
</tr>
<tr>
<td><strong>Methods</strong></td>
<td>Point-wise Learning Methods</td>
<td>Pair-wise Learning</td>
</tr>
<tr>
<td>NDCG@K</td>
<td>Most Pop</td>
<td>NMF</td>
</tr>
<tr>
<td>10</td>
<td>0.1031</td>
<td>0.0581</td>
</tr>
<tr>
<td>20</td>
<td>0.1359</td>
<td>0.0812</td>
</tr>
<tr>
<td>50</td>
<td>0.1917</td>
<td>0.1366</td>
</tr>
</tbody>
</table>

Explainable recommendation methods are comparable to or better than traditional (non-explainable) recommendation methods.
Experimental Evaluation

- Explanation Performance [Zhang et al. SIGIR’2014]
- 3 user groups
  - A (experimental group): Receive personalized explanations
  - B (comparison group): Receive the ‘people also viewed’ explanation
  - C (control group): Receive no explanation

<table>
<thead>
<tr>
<th>User Set</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Records</td>
<td>#Record</td>
<td>#Click</td>
<td>#Record</td>
</tr>
<tr>
<td></td>
<td>15,933</td>
<td>691</td>
<td>11,483</td>
</tr>
<tr>
<td>CTR</td>
<td>4.34%</td>
<td>3.22%</td>
<td>3.20%</td>
</tr>
</tbody>
</table>

Providing explanations improve the persuasiveness of system decisions.
Experimental Evaluation

- Explanation Performance [Wang et al. SIGIR’2018]
  - Effectiveness of different explanations may be different

<table>
<thead>
<tr>
<th>Amazon Dataset</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean Value</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BPR</td>
<td>3.540</td>
<td>3.447</td>
<td>-</td>
<td>3.333</td>
<td>-</td>
</tr>
<tr>
<td>MTER</td>
<td><strong>3.767</strong></td>
<td><strong>3.660</strong></td>
<td><strong>3.707</strong></td>
<td><strong>3.727</strong></td>
<td><strong>3.620</strong></td>
</tr>
<tr>
<td><strong>Paired t-test</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTER vs. BPR</td>
<td>0.0142</td>
<td>0.0273</td>
<td>-</td>
<td>0.0001</td>
<td>-</td>
</tr>
<tr>
<td>MTER vs. EFM</td>
<td>0.0001</td>
<td>0.0027</td>
<td>0</td>
<td>0</td>
<td>0.0004</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Yelp Dataset</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean Value</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BPR</td>
<td>3.400</td>
<td>3.387</td>
<td>-</td>
<td>3.180</td>
<td>-</td>
</tr>
<tr>
<td>EFM</td>
<td><strong>3.540</strong></td>
<td><strong>3.473</strong></td>
<td><strong>3.287</strong></td>
<td><strong>3.200</strong></td>
<td><strong>3.200</strong></td>
</tr>
<tr>
<td>MTER</td>
<td>3.500</td>
<td><strong>3.713</strong></td>
<td><strong>3.540</strong></td>
<td><strong>3.520</strong></td>
<td><strong>3.360</strong></td>
</tr>
<tr>
<td><strong>Paired t-test</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTER vs. BPR</td>
<td>0.1774</td>
<td>0.0015</td>
<td>-</td>
<td>0.0013</td>
<td>-</td>
</tr>
<tr>
<td>MTER vs. EFM</td>
<td>0.3450</td>
<td>0.0128</td>
<td>0.0108</td>
<td>0.0015</td>
<td>0.0775</td>
</tr>
</tbody>
</table>

Five Survey questions for users:

Q1: Generally, are you satisfied with this recommendation?
Q2: Do you think you get some idea about recommended item?
Q3: Does the explanation help you know more about the item?
Q4: Do you think you gain some insight of why we recommend this to you?
Q5: Do you think explanations help you better understand our system, e.g., based on what we made the recommendation?

Users do have different feelings of different explanations. Providing good explanation is important.
Short Summery

• A series of work on making latent factor models explainable
• Key idea: assign “explicit” meanings to the “latent” factors
• Better recommendation performance, better explainability
References


Explainable Recommendation with Deep Models

- **Explainable Deep Models over Text**
  - Based on **Attention Mechanism**
    - Word-level Attention [Seo et al. RecSys’2017]
    - Review-level Attention [Chen et al. WWW’2018]
    - Item-level Attention [Chen et al. WSDM’2018]
  - Based on **Textual Explanation Generation**
    - Sequence-to-Sequence Models with LSTM [Li et al. SIGIR’2017]
    - Generative Adversarial Networks (GAN) [Lu et al. RecSys’2018]

- **Explainable Deep Models over Image**
  - Based on **Attention Mechanism**
    - Image Region-of-Interest Explanation [Chen et al. SIGIR’2019]
Word-level Attentive Explanation

- Interpretable Convolutional Neural Networks with Dual Local and Global Attention [Seo et al. RecSys’2017]

\[ \hat{r}_{ui} \]

\begin{align*}
\text{L-Attn} & : \text{Local attention, learns which words are more informative in a local window of words.} \\
\text{G-Attn} & : \text{Global attention, learns which words are informative in the entire text.}
\end{align*}

User document: reviews of the user  
Item document: reviews of the item
Word-level Attentive Explanation

• Highlighted explanation words by local and global attention

<table>
<thead>
<tr>
<th>Yelp (user), D-Attn model: local attention</th>
</tr>
</thead>
<tbody>
<tr>
<td>They carry some <strong>rare</strong> things that you can’t find anywhere else. The staff is <strong>pretty damn</strong> <strong>cool</strong> <strong>too best</strong> in Arizona. I prefer ma-and-pa. They <strong>treat</strong> you the <strong>best</strong> and they <strong>value</strong> your business <strong>extreme</strong>. They are good people <strong>great atmosphere</strong> and music. I <strong>definitely believe</strong> that Lux has the <strong>best</strong> coffee I’ve ever had at this point. Screw all my previous reviews. This place has coffee down, they make <strong>damn</strong> <strong>good toast too</strong>.</td>
</tr>
</tbody>
</table>

Local attention: highlighted words are **important** words (i.e., words that have **high attention**)

Observation: Local attention helps to **select informative words** for prediction and as explanation.

<table>
<thead>
<tr>
<th>Yelp (user), D-Attn model: global attention</th>
</tr>
</thead>
<tbody>
<tr>
<td>They carry some <strong>rare</strong> things that you can’t find anywhere else. The staff is <strong>pretty damn</strong> <strong>cool</strong> <strong>too best</strong> in Arizona. I <strong>prefer ma-and-pa.</strong> They <strong>treat</strong> you the <strong>best</strong> and they <strong>value</strong> your business <strong>extreme</strong>. They are <strong>good people</strong> <strong>great atmosphere</strong> and <strong>music</strong>. I <strong>definitely believe</strong> that Lux has the <strong>best</strong> coffee I’ve ever had at this point. Screw all my previous reviews. <strong>This place has coffee down, they make damn good toast too</strong>.</td>
</tr>
</tbody>
</table>

Global attention: highlighted words are **unimportant** words (i.e., words that have **low attention**)

Observation: Global attention helps to **eliminate unimportant words** for better prediction.
Review-Level Attentive Explanation

- Attentively select useful reviews as explanation [Chen et al. WWW’2018]

Attention mechanism learns the importance of each review

\[
a^{*}_i = h^T \text{ReLU}(W_O O_i + W_u u_i + b_1) + b_2
\]

\[
a_{il} = \frac{\exp(a^{*}_{il})}{\sum_{l=0}^{k} \exp(a^{*}_{il})}
\]

\[
O_i = \sum_{l=1 \ldots k} a_{il} O_{il}
\]
## Review-Level Attentive Explanation

- Provide selected useful reviews as explanations

<table>
<thead>
<tr>
<th>Item 1</th>
<th>(a(\alpha_{ij}=0.1932))</th>
<th>These brushes are great quality for children’s art work. They seem to last well and the bristles stay in place very well even with tough use.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(b(\alpha_{ij}=0.0161))</td>
<td>I bought it for my daughter as a gift.</td>
</tr>
<tr>
<td>Item 2</td>
<td>(a(\alpha_{ij}=0.2143))</td>
<td>From beginning to end this book is a joy to read. Full of mystery, mayhem, and a bit of magic for good measure. Perfect flow with excellent writing and editing.</td>
</tr>
<tr>
<td></td>
<td>(b(\alpha_{ij}=0.0319))</td>
<td>I like reading in my spare time, and I think this book is very suitable for me.</td>
</tr>
</tbody>
</table>

Examples of the high-weight and low-weight reviews selected by the model (Item1 from Amazon Toys_and_Games, Item2 from Amazon Kindle_Store)
Review-Level Attentive Explanation

• Crowd-sourcing based Usefulness Evaluation of the Explanations

Most of the selected review explanations are rated “useful” by users.

Group A: top-5 algorithm selected reviews.
Group B: top-5 reviews rated helpful in Amazon.
In 67% of the cases, selected reviews are equal to or better than Amazon user rated reviews.
Item-Level Attentive Explanation

- Sequential Recommendation with Memory Networks [Chen et al. WSDM’2018]
  - Which previous item(s) influence the recommended item?

Attentive selection over the latest $K$ (e.g., 5) interacted items of the user through memory network.
Attention weighted show which previous item(s) highly influence the recommendation.
Item-Level Attentive Explanation

- Two types of influence patterns

One-to-Multiple: an item consistently influence subsequent user behaviors.

One-to-One: previous item influences the current item, and current item influence the next item…
LSTM-based Textual Explanation Generation

- Sequence-to-Sequence Models with LSTM [Li et al. SIGIR’2017]

Rating prediction based on learned latent user and item embeddings.

An LSTM generator to predict the ground-truth tips of the user item pair, personalized by the user-item embeddings and rating.
## LSTM-based Textual Explanation Generation

- **Sequence-to-Sequence Models with LSTM** [Li et al. SIGIR’2017]

<table>
<thead>
<tr>
<th>Rating</th>
<th>Tips</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4.64</strong></td>
<td>This is a great product for a great price. Great product at a great price.</td>
</tr>
<tr>
<td><strong>4.87</strong></td>
<td>I purchased this as a replacement and it is a perfect fit and the sound is excellent. Amazing sound.</td>
</tr>
<tr>
<td><strong>4.69</strong></td>
<td>I have been using these for a couple of months. Plenty of wire gets signals and power to my amp just fine quality wise.</td>
</tr>
<tr>
<td><strong>4.87</strong></td>
<td>One of my favorite movies. This is a movie that is not to be missed.</td>
</tr>
<tr>
<td><strong>2.25</strong></td>
<td>Not as good as i expected. Jack of all trades master of none.</td>
</tr>
<tr>
<td><strong>1.46</strong></td>
<td>What a waste of time and money. The coen brothers are two sick bastards.</td>
</tr>
<tr>
<td><strong>4.34</strong></td>
<td>Not bad for the price. Ended up altering it to get rid of ripples.</td>
</tr>
</tbody>
</table>

**Bold line:** Predicted ratings and generated tips.  
**Second line:** ground truth tips.

Sampled tips on Yelp
Explanation Generation with GANs

- Generative Adversarial Networks (GAN) [Lu et al. RecSys’2018]

Regularizers force user/item features to approximate each other
Explanation Generation with GANs

- The learned generator generates personalized user-item pair explanations.
  - Concatenate user and item textual features and feed into the review decoder.

<table>
<thead>
<tr>
<th>Model</th>
<th>Y13</th>
<th>Y14</th>
<th>AE</th>
<th>AV</th>
<th>AG</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-gram</td>
<td>0.007</td>
<td>0.009</td>
<td>0.005</td>
<td>0.009</td>
<td>0.011</td>
</tr>
<tr>
<td>Skip-gram</td>
<td>0.009</td>
<td>0.014</td>
<td>0.007</td>
<td>0.011</td>
<td>0.015</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.019</td>
<td>0.022</td>
<td>0.014</td>
<td>0.017</td>
<td>0.019</td>
</tr>
<tr>
<td>Opinionised</td>
<td>0.029</td>
<td>0.031</td>
<td>0.029</td>
<td>0.025</td>
<td>0.027</td>
</tr>
<tr>
<td>MT-U</td>
<td>0.048</td>
<td>0.043</td>
<td>0.042</td>
<td>0.044</td>
<td>0.047</td>
</tr>
<tr>
<td>MT-I</td>
<td>0.051</td>
<td>0.045</td>
<td>0.046</td>
<td>0.049</td>
<td>0.049</td>
</tr>
<tr>
<td>MT-P</td>
<td><strong>0.053</strong></td>
<td><strong>0.052</strong></td>
<td><strong>0.049</strong></td>
<td><strong>0.042</strong></td>
<td><strong>0.051</strong></td>
</tr>
</tbody>
</table>

- Explanation performance in terms of tf-idf (both ground-truth and generated review are represented as a tf-idf vector of vocabulary size, cosine similarity between ground-truth and generated review explanation is reported)

- Explanations should be relevant to both user and item.
Attentive Visual Explanation over Images

• Visual Explanation based on Image Region-of-Interest [Chen et al. SIGIR’2019]

1. **Image feature extraction**: divide image by 14*14, each region is fed into VGG network to generate a 512-dim vector.

2. **Attention mechanism** learns the importance of each region.

   \[ a_{ijk} = E_2[\text{ReLU}(E_1[(W_u p_i) \odot (W_f f^k_j)])] \]

   \[ \alpha_{ijk} = \frac{\exp(a_{ijk})}{\sum_{k'=1}^{h} \exp(a_{ijk'})} \]

   \[ I_{ij} = f_j \alpha_{ij} = \sum_{k=1}^{h} \alpha_{ijk} \cdot f^k_j \]

3. **Aggregated user, item, and image embedding** used to predict the user review based on GRU.
Attentive Visual Explanation over Images

• Visual Explanation based on Image Region-of-Interest [Chen et al. SIGIR’2019]

<table>
<thead>
<tr>
<th>Method</th>
<th>Random</th>
<th>VECTF(-rev)</th>
<th>VECTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>M=1</td>
<td>0.777</td>
<td>1.220</td>
<td>2.273  (86.3%↑)</td>
</tr>
<tr>
<td>M=2</td>
<td>1.430</td>
<td>2.012</td>
<td>3.180  (58.1%↑)</td>
</tr>
<tr>
<td>M=3</td>
<td>1.968</td>
<td>2.516</td>
<td>4.513  (79.4%↑)</td>
</tr>
<tr>
<td>M=4</td>
<td>2.281</td>
<td>2.857</td>
<td>4.514  (58.0%↑)</td>
</tr>
<tr>
<td>M=5</td>
<td>2.749</td>
<td>3.350</td>
<td>4.774  (42.5%↑)</td>
</tr>
</tbody>
</table>

VECF(-rev): remove the GRU review prediction component.

Observation: Including reviews is much better. i.e., there exist useful correlation signals between image and reviews, e.g., user comment the image features in reviews.

For each image, the correct top-5 explanation regions are labeled using crowd-sourcing.

Algorithm predicts the top-M region of interest.
All numbers are % numbers
**Attentive Visual Explanation over Images**

- Visual Explanation based on Image Region-of-Interest [Chen et al. SIGIR’2019]

<table>
<thead>
<tr>
<th>#</th>
<th>Target Item</th>
<th>Historical Records</th>
<th>Textual Review</th>
<th>Visual Explanation</th>
<th>VECF</th>
<th>Re-VECF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><img src="image1" alt="Watch" /></td>
<td><img src="image2" alt="Watch" /> <img src="image3" alt="Watch" /> <img src="image4" alt="Watch" /></td>
<td><em>this is a large watch... nearly as large as my suunto but due to its articulated strap it fits on the wrist very well.</em></td>
<td><img src="image5" alt="Visual Explanation" /> <img src="image6" alt="Visual Explanation" /></td>
<td><img src="image7" alt="VECF" /> <img src="image8" alt="Re-VECF" /></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><img src="image9" alt="T-shirt" /></td>
<td><img src="image10" alt="T-shirt" /> <img src="image11" alt="T-shirt" /> <img src="image12" alt="T-shirt" /></td>
<td>*this is a really comfortable v-neck. I found that the size and location of the v are just right for me. I'm 5'8 &amp; #34, but 200 lbs (and dropping :))</td>
<td><img src="image13" alt="Visual Explanation" /></td>
<td><img src="image14" alt="VECF" /> <img src="image15" alt="Re-VECF" /></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><img src="image16" alt="Pants" /></td>
<td><img src="image17" alt="Pants" /> <img src="image18" alt="Pants" /> <img src="image19" alt="Pants" /></td>
<td><em>Great leggings. Perfect for fly fishing or hunting or running. Just perfect anytime you are cold!</em></td>
<td><img src="image20" alt="Visual Explanation" /></td>
<td><img src="image21" alt="VECF" /> <img src="image22" alt="Re-VECF" /></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><img src="image23" alt="Shoes" /></td>
<td><img src="image24" alt="Shoes" /> <img src="image25" alt="Shoes" /> <img src="image26" alt="Shoes" /></td>
<td><em>The socks on the shoes are a perfect fit for me. First time with a shoe with the speed laces and I like them a lot</em></td>
<td><img src="image27" alt="Visual Explanation" /></td>
<td><img src="image28" alt="VECF" /> <img src="image29" alt="Re-VECF" /></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><img src="image30" alt="Socks" /></td>
<td><img src="image31" alt="Socks" /> <img src="image32" alt="Socks" /> <img src="image33" alt="Socks" /></td>
<td><em>Really like these socks! They are really thick woolen socks and are good for cold days. They cover a good portion of your feet as they go a little (halfway) above the calf muscle area.</em></td>
<td><img src="image34" alt="Visual Explanation" /></td>
<td><img src="image35" alt="VECF" /> <img src="image36" alt="Re-VECF" /></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><img src="image37" alt="Shirts" /></td>
<td><img src="image38" alt="Shirts" /> <img src="image39" alt="Shirts" /> <img src="image40" alt="Shirts" /></td>
<td><em>I like the front pocket~! Very cool!</em></td>
<td><img src="image41" alt="Visual Explanation" /></td>
<td><img src="image42" alt="VECF" /> <img src="image43" alt="Re-VECF" /></td>
<td></td>
</tr>
</tbody>
</table>
Short Summary

• Explainable recommendation based on both Text and Image
• Most methods are based on attention mechanism
  – Learning “weights” as explanations, similar to what did in simple linear regression.
• Generating natural language explanations
References

Explainable Recommendation based on KGs

• KG is a Flexible Structure
  – Easy to integrate various heterogeneous information

• Bridge Symbolic Reasoning and Neural Modeling
  – Unite GOFAI (Good-Old Fashioned AI, dominate AI approach before 1980s) and machine learning/deep learning (dominate AI approach after 1980s)
  – Improves both Explainability and Accuracy
Explainable Recommendation based on KGs

- Mostly based on Explanation Path between User and Item Entities
- Embedding Learning Approaches
  - Learn some kind of user and item representations from KG
  - Recommendation based on the similarity between user-item entity
    - Translational KG Embedding for Rec and Explanation [Ai et al. Alg’2018]
    - Propagating User Preferences on the Knowledge Graph [Wang et al. CIKM’2018]
    - Learning Path Embedding for Recommendation [Wang et al. AAAI’2019]
    - Jointly Learning Explainable Rules for Recommendation [Ma et al. WWW’2019]
- Symbolic Reasoning Approaches
  - Recommendation based on path reasoning beginning from user entity
    - Reinforcement KG Reasoning for Explainable Recommendation [Xian et al. SIGIR’2019]
Embedding Learning Approach

- Recommendation based on the similarity between user-item entity
- Reasoning using hard-rules over KG is inefficient and difficult to generalize
- KG embedding makes it easier to calculate the similarity between entities

TransE: translation-based embedding

\[ h + \ell \approx t \quad d(h + \ell, t) \]

Minimize the hinge-loss to learn entity and relation embeddings

\[ \mathcal{L} = \sum_{(h, \ell, t) \in S} \sum_{(h', \ell, t') \in S'} \left[ \gamma + d(h + \ell, t) - d(h' + \ell, t') \right]_+ \]
Translational KG Embedding for Recommendation

- Learning heterogeneous knowledge base embeddings for explainable recommendation [Ai et al. Alg’2018]

\[ e_t = \text{trans}(e_h, r) = e_h + r \]

\[ P(e_t | \text{trans}(e_h, r)) = \frac{\exp(e_t \cdot \text{trans}(e_h, r))}{\sum_{e'_t \in E_t} \exp(e'_t \cdot \text{trans}(e_h, r))} \]

\[ \mathcal{L}(S) = \sum_{(e_h, e_t, r) \in S} \log \sigma(e_t \cdot \text{trans}(e_h, r)) + k \cdot \mathbb{E}_{e'_t \sim P_t} [\log \sigma(-e'_t \cdot \text{trans}(e_h, r))] \]

Recommendation:
Calculate \( e_{\text{user}} + r_{\text{purchase}} \)
Find top-K nearest item entity
Translational KG Embedding for Recommendation

- Learning heterogeneous knowledge base embeddings for explainable recommendation [Ai et al. Alg’2018]

Find an intermediate entity $e_x$

$$e_u + \sum_{\alpha=1}^{m} r_\alpha = e_i + \sum_{\beta=1}^{n} r_\beta$$

Calculate connectivity of the path:

$$P(e_x|\text{trans}(e_u, R_\alpha)) = \frac{\exp(e_x \cdot \text{trans}(e_u, R_\alpha))}{\sum_{e' \in E^r_m} \exp(e' \cdot \text{trans}(e_u, R_\alpha))}$$

$$P(e_x|e_u, R_\alpha, e_i, R_\beta) = P(e_x|\text{trans}(e_u, R_\alpha))P(e_x|\text{trans}(e_i, R_\beta))$$

Rank explanation paths based on connectivity.

Post-hoc explanation by finding a path between user and the (already) recommended item.
Propagating User Preferences on KG

- RippleNet: Propagating user preferences on the knowledge graph for recommender systems. [Wang et al. CIKM’2018]

Seeds: user interacted entities

1-hop related entities

2-hop related entities

H-hop related entities

Attentively select entities based on memory network

Calculate user-item similarity for recommendation
Propagating User Preferences on KG

- RippleNet: Propagating user preferences on the knowledge graph for recommender systems. [Wang et al. CIKM’2018]

Explanation path constructed by selecting the most significant entity in each hop.

'user' -> Forrest Gump -> directed by Robert Zemeckis -> Back to the Future
Learning Path Embedding for Recommendation

- Explainable Reasoning over Knowledge Graphs for Recommendation [Wang et al. AAAI’2019]

Each path represented as a path embedding using LSTM (input: entity embedding + entity type embedding + relation embedding)
Jointly Learning Explainable Rules for Recommendation

- Extract rules from knowledge graph for recommendation [Ma et al. WWW’2019]

**Rule Learning Module**

**Rule Selection**

<table>
<thead>
<tr>
<th>Rules</th>
<th>$w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\text{phones.manufacturer, accessories.manufacturer}^{-1}) \rightarrow \text{Buy Together}$</td>
<td>0.12</td>
</tr>
<tr>
<td>$(\text{phones.manufacturer, rivals, phones.manufacturer}^{-1}) \rightarrow \text{Also View}$</td>
<td>0.21</td>
</tr>
<tr>
<td>$(\text{phones.manufacturer, rivals, laptops.manufacturer}^{-1}) \rightarrow \text{Also View}$</td>
<td>0.03</td>
</tr>
<tr>
<td>$(\text{phones.manufacturer, accessories.earsets.manufacturer}^{-1}) \rightarrow \text{Buy Also}$</td>
<td>0.14</td>
</tr>
</tbody>
</table>

$w$: Importance of a rule learned by feature selection

$$\sum_{\text{allpairs} \in A} \sum_{i=0}^{\left| x_{(a,b)} \right|} (w_i \cdot x_{(a,b)}(i) + b - y_{a,b|A})^2$$

$x_{(a,b)}(i) = P(b|a, R_i)$: probability that $a$ and $b$ are linked by $R_i$

$y_{a,b|A} = 1$ if $a$, $b$ are truly linked by relation $A$ (e.g., buy together)

**Rule**: a sequence of relation types: e.g., $r_1-r_2-r_3$

Connection strength between items $a$ & $b$ through rule $R$

$$P(b|a, R) = \sum_{e \in N(a, R')} P(e|a, R') \cdot P(b|e, r_k).$$
Jointly Learning Explainable Rules for Recommendation

- Extract rules from knowledge graph for recommendation [Ma et al. WWW’2019]

**Rule Selection**

<table>
<thead>
<tr>
<th>Rules</th>
<th>(W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(phones.manufacturer, accessories.manufacturer(^{-1})) → Buy Together</td>
<td>0.12</td>
</tr>
<tr>
<td>(phones.manufacturer, rivals, phones.manufacturer(^{-1})) → Also View</td>
<td>0.21</td>
</tr>
<tr>
<td>(phones.manufacturer, rivals, laptops.manufacturer(^{-1})) → Also View</td>
<td>0.03</td>
</tr>
<tr>
<td>(phones.manufacturer, accessories.earsets..manufacturer(^{-1})) → Buy Also</td>
<td>0.14</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Recommendation Module**

Recommendation provided by user history and rule importance

For a candidate item \(i\), ranking score is calculated based on the rules between \(i\) and each of the user’s history items, weighted by rule importance.

The most important rule serves as the recommendation explanation.

**Rule:** a sequence of relation types: e.g., \(r_1\)-\(r_2\)-\(r_1\)-\(r_3\)

**Connection strength between items \(a\) & \(b\) through rule \(R\)**

\[
P(b|a, R) = \sum_{e \in N(a, R')} P(e|a, R') \cdot P(b|e, r_k).
\]
Reinforcement KG Reasoning

- Reinforcement Knowledge Graph Reasoning for Explainable Recommendation [Xian et al. SIGIR'2019]
- Paradigm of previous methods: for each user, for each candidate item, calculate ranking score based on path info between this user-item pair.
- Too many candidate items: Can we avoid enumerating all candidate items?

**KG Reasoning**: train an agent, which starts from a user and walks over the graph, and reach a “good” item node with high probability.

**RL-based training**: reach positive item – high reward, reach negative item - low reward.
Reinforcement KG Reasoning

- Reinforcement Knowledge Graph Reasoning for Explainable Recommendation [Xian et al. SIGIR’2019]

The reasoning path (how the agent reached the item from the user) naturally serve as the explanation.
Short Summary

• Explainable Recommendation based on KGs
  – Mostly based on Explanation Path between User and Item Entities

• Embedding Learning Approaches
  – Learn some kind of user and item representations from KG for recommendation

• Symbolic Reasoning Approaches
  – Recommendation based on path reasoning beginning from user entity and reach a good item entity
References

Post-hoc and Model-Agnostic Explanation

• Provide explanation for a (possibly unexplainable) model

• Mining-based Approach

• Learning-based Approach
  – A Reinforcement Learning Framework for Explainable Recommendation [Wang et al. ICDM’2018]
Post-hoc Explanation based on Association Rule Mining


Recommendation list by a black-box model (e.g., latent factor model)

“Unexplainable Items”

Extract associate rules $X \rightarrow Y$ based on the completed matrix $R$. (For each user, take top-D highly predicted items as a transaction)

$X$ in training data, $Y$ not in training data. Rank items according to some interestingness score (support/confidence/lift).

“Explainable Items” (because you liked $X$)
Post-hoc Explanation based on Association Rule Mining
• Evaluate explainability based on Model Fidelity

\[
\text{Model Fidelity} = \frac{|\text{MF recommended items} \cap \text{AR retrieved items}|}{|\text{MF recommended items}|} = \frac{|\text{explainable items} \cap \text{recommended items}|}{|\text{recommended items}|}
\]

<table>
<thead>
<tr>
<th>Rules</th>
<th>K</th>
<th>Interestingness</th>
<th>Model Fidelity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>N</td>
<td>Support</td>
<td>0.522369</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Confidence</td>
<td>0.568602</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lift</td>
<td>0.423591</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>Support</td>
<td>0.828272</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Confidence</td>
<td><strong>0.842889</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lift</td>
<td>0.412679</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Support</td>
<td>0.791095</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>Confidence</td>
<td>0.817715</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lift</td>
<td>0.452805</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>Confidence</td>
<td>0.799536</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lift</td>
<td>0.44886</td>
</tr>
</tbody>
</table>

Global rules: association rules are mined with all users, each user is a transaction.

Local rules: each user’s association rules are mined with this user’s top-K similar user, each user is a transaction.

With appropriate nearest neighbor and interestingness selection, 80%+ of the recommendations can be post-hoc explained.
Model-Agnostic Explanation based on RL
• A Reinforcement Learning Framework for Explainable Recommendation [Wang et al. IDCM’2018]

Agent 1 selects explanations $z = [z_1, z_2, \ldots, z_m]$. $z_i=1$: i-th interpretable component is selected.

Each item has several interpretable components. E.g., a description sentence of a book: “As of Feb 2018, the books have sold more than 500 million copies worldwide, making them the best-selling book series in history”

Agent 2 predict the rating based on agent 1’s selected explanations.

Reward:
$$r = \mathcal{L}(f(u, v), y) + \Omega(z)$$
Model-Agnostic Explanation based on RL

- Evaluate explanation based on consistency $M_c$ and explainability $M_e$

$$M_c = \frac{\sum_{(u,v) \in T} (\phi(u, v) - \bar{\phi})(f(u, v) - \bar{f})}{\sqrt{\sum_{(u,v) \in T} (\phi(u, v) - \bar{\phi})^2} \sqrt{\sum_{(u,v) \in T} (f(u, v) - \bar{f})^2}}$$

$$M_e = - \sum_{(u,v) \in T} (y - f(u, v))^2$$

Pearson correlation between the sentiment of selected explanation sentences and the output rating of the recommendation model.

Closeness between the ratings of the explanation agent and the recommendation model to be explained.

<table>
<thead>
<tr>
<th></th>
<th>$M_c$</th>
<th></th>
<th>$M_e$</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NMF</td>
<td>PMF</td>
<td>SVD++</td>
<td>CDL</td>
<td>GT</td>
<td>NMF</td>
</tr>
<tr>
<td>Random</td>
<td>-0.030</td>
<td>-0.030</td>
<td>-0.031</td>
<td>0.012</td>
<td>0.007</td>
<td>-0.478</td>
</tr>
<tr>
<td>NARRE</td>
<td>-0.015</td>
<td>-0.000</td>
<td>0.018</td>
<td>0.031</td>
<td>0.038</td>
<td>-0.448</td>
</tr>
<tr>
<td>Ours</td>
<td>0.018</td>
<td>0.037</td>
<td>0.041</td>
<td>0.227</td>
<td>0.168</td>
<td>-0.421</td>
</tr>
</tbody>
</table>

Results on Yelp dataset, NMF, PMF, SVD++, CDL are models to be explained. GT is the ground-truth score.
Short Summary

• Post-hoc and Model-Agnostic Explanation
  – Provide explanation for a (possibly unexplainable) model

• Mining-based Approach
  – Extract association rules as post-hoc explanations

• Learning-based Approach
  – Learn an explainable model to approximate the unexplainable model
References


Challenges and Directions

• Explainable Recommendation + NLP
  – Generating Natural Language Explanations
  – Explainable Conversational Systems: Answering the why in conversations

• Offline evaluation of explainability
  – Current evaluation
    • online evaluation with users (sometimes expensive and inefficient)
    • case studies (only covers a small amount of cases)
    • model dependent measures (depends on the model)
  – Can we develop a general “explainability” measure?

• Explanation beyond persuasiveness
  – Explanations are not (or should not) be used to just attract user click/purchase
  – Should help users to make better decisions, improve user well-being, social justice, and sustainability of the Web.
Explainable Recommendation and Search

Part II
Outline

• Background and motivation
  – What is explainable search?
  – Why do we need explainable search?

• Existing work on explainable search
  – How can we make search models more explainable?
    • Building Interpretable search models
    • Using structured knowledge
    • Post-hoc explanation methods for search
    • Axiomatic analysis of search models

• Wrap up
Outline

• Background and motivation
  – What is explainable search?
  – Why do we need explainable search?

• Existing work on explainable search
  – How can we make search models more explainable?
    • Building Interpretable search models
    • Using structured knowledge
    • Post-hoc explanation methods for search
    • Axiomatic analysis of search models

• Wrap up
Explainable AI on the Web

- Recent research on explainable recommendation and search is related to Explainable AI
Background

- What is **explainable search**?
  - Search: one of the most important AI application on the Web
  - In a narrow sense:
    - How to build an **interpretable** search model
  - In a broad sense:
    - Re-examine the search problem from the **explainable AI/ML** perspective
Motivation

• Why do we need explainable search?
• Give explanations to whom?
  – Search users
  – System designers
Motivation: Why do we need explainable search?

• To search users:
  – A search engine is an interactive tool to access a huge information repository
Motivation: Why do we need explainable search?

• To search users:
  – The user must have a correct **mental model** of the system about:
    • The **capability** and **limitation** of the system
      – e.g. Can the search engine answer natural language questions?
      – Can the image search engine find pictures similar/identical to a queried picture?
    • When to **trust** the search system
      – Are those top-ranked results **good enough**?
      – Are they **trustworthy**?
      – Are they **biased**?
    • How to **intervene** when the results are not satisfactory
      – Query reformulation
      – Search strategies and expertise
  
• **Better explanation** may help the user **build** better mental models for search
Motivation: Why do we need explainable search?

- Examples of explanations to users: search snippet
  - Query-centric, with keywords highlighted
  - Explain why a webpage is retrieved
Motivation: Why do we need explainable search?

- Investigating the interpretability of search result summaries in a user study (Mi an Jiang, 2019)
Motivation: Why do we need explainable search?

• To system designers:
  – Objective: to estimate the relevance of each query-doc pair and use it to rank the document when given the query
    • Retrieve model: $f_M(q, d)$
  – The ranking performance can be evaluated by a range of evaluation metrics.
    • Offline evaluation metrics based on relevance labels e.g. MAP, nDCG…
    • Online evaluation metrics: CTR, A/B test, SAT clicks…
  – But evaluation metrics are still incomplete descriptions of the search tasks
Motivation: Why do we need explainable search?

• To system designers:
  – Interpretability of search models can help with:
    • understanding relevance itself (i.e. why a document is relevant to a query)
      – Keyword match?
      – Topically/semantically related?
      – Usefulness?
    • comprehensive analysis and evaluation of search models at the global level
      – Why the model works (better than other models)?
      – Does the model overfit the test set?
      – Fairness, Accountability, Credibility, Transparency, Privacy
  • diagnosing and debugging the model at the local level
    – Why the model fails for some queries?
    – How to handle bad cases?
Motivation: Why do we need explainable search?

• To system designers:
  – Ranking models are becoming more and more sophisticated:
    • Retrieval models:
      – TF-IDF, BM25, query likelihood model…
    • Learning-to-rank models:
      – RankSVM, LambdaMart…
    • Neural IR models:
      – DSSM, DRMM, KRM…
  – A trade-off between the ranking performance and interpretability
  – Understanding how these more powerful but more complex ranking models work has become a new challenge
Outline

• Background and motivation
  – What is explainable search?
  – Why do we need explainable search?

• Existing work on explainable search
  – How can we make search models more explainable?
    • Building interpretable search models
    • Using structured knowledge
    • Post-hoc explanation methods for search
    • Axiomatic analysis of search models

• Wrap up
Interpretability of retrieval model

- Existing retrieval models are quite explainable, for example:
  - TF-IDF model:
    - $f_{tfidf}(q, d) = \sum_{w \in q \cap d} tf(w) \cdot idf(w)$
  - Based on exact match between query and document terms
  - Modeling the importance of query term with inverse document frequency: $idf(w) = \log \frac{N}{n_w}$
  - Allows diverse matching patterns
    - ignores the order and positions of matching terms

- It is easy to understand how and why the TF-IDF model works because it is designed in this way
Integrate Interpretable Structure

- We can also design and integrate interpretable components into the neural models to address these interpretable factors:
  - Exact matching signals
  - Query term importance
  - Allow diverse matching patterns

- A deep relevance matching model for ad-hoc retrieval (Guo et al. 2016)
Model Architecture

1. Matching Histogram Mapping
2. Feed Forward Matching Network
3. Term Gating Network

Score Aggregation

Matching Score

q

d

(gu et al. 2016)
Model Architecture

• 1. Matching Histogram Mapping
  – map the varied-size interactions into a fixed-length representation
  – Groups local interactions according to different strength levels
  – position-free but strength-focused representation

\[ z_i^{(0)} = h\left( w_i^{(q)} \otimes d \right), \quad i = 1, \ldots, M \]

cosine similarity

– Different mappings \( h() \):
  • Count-based histogram: frequency
  • Normalized histogram: relative frequency
  • LogCount-based histogram: logarithm
Model Architecture

2. Feed forward Matching Network

- Extract hierarchical matching patterns from different levels of interaction signals

\[
\begin{align*}
    z_i^{(0)} &= h \left( w_i^{(q)} \otimes d \right), & i = 1, \ldots, M \\
    z_i^{(l)} &= \tanh \left( W^{(l)} z_i^{(l-1)} + b^{(l)} \right), & i = 1, \ldots, M, l = 1, \ldots, L
\end{align*}
\]
Model Architecture

3. Term Gating Network

- Modeling term importance by control how much relevance score on each query term contribute to the final relevance score

\[
    s = \sum_{i=1}^{M} g_i z_i^{(L)}
\]

\[
    g_i = \frac{\exp(w_g x_i^{(q)})}{\sum_{j=1}^{M} \exp(w_g x_j^{(q)})}, \quad i = 1, \ldots, M
\]

- Input:
  - Term vector
  - Inverse document frequency
Experimental Settings:

• Dataset:
  – Robust04: news collection
  – ClueWeb09-Cat-B: Web collection

• Evaluation Methodology:
  – 5-fold cross validation
  – Tuned towards MAP
  – Evaluated by MAP, nDCG@20, P@20

<table>
<thead>
<tr>
<th></th>
<th>Robust04</th>
<th>ClueWeb09-Cat-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vocabulary</td>
<td>0.6M</td>
<td>38M</td>
</tr>
<tr>
<td>Document Count</td>
<td>0.5M</td>
<td>34M</td>
</tr>
<tr>
<td>Collection Length</td>
<td>252M</td>
<td>26B</td>
</tr>
<tr>
<td>Query Count</td>
<td>250</td>
<td>150</td>
</tr>
</tbody>
</table>

The ClueWeb-09-Cat-B collection has been filtered to the set of documents in the 60th percentile of spam scores.
## Retrieval Performance on Robust-04

| Model Type                              | Model Name | Topic Titles | | Topic Descriptions | |
|-----------------------------------------|------------|--------------|-----------------|-------------------|-----------------|----------|----------|
|                                        |            | MAP | nDCG@20 | P@20 | MAP | nDCG@20 | P@20 |
| **Traditional Retrieval Baselines**     | QL         | 0.253 | 0.415 | 0.369 | 0.246 | 0.391 | 0.334 |
|                                        | BM25       | 0.255 | 0.418 | 0.370 | 0.241 | 0.399 | 0.337 |
| **Representation-Focused Matching Baselines** | DSSM_D     | 0.095 | 0.201 | 0.171 | 0.078 | 0.169 | 0.145 |
|                                        | CDSSM_D    | 0.067 | 0.146 | 0.125 | 0.050 | 0.113 | 0.093 |
|                                        | ARC-I      | 0.041 | 0.066 | 0.065 | 0.030 | 0.047 | 0.045 |
| **Interaction-Focused Matching Baselines** | ARC-II     | 0.067 | 0.147 | 0.128 | 0.042 | 0.086 | 0.074 |
|                                        | MP_IND     | 0.169 | 0.319 | 0.281 | 0.067 | 0.142 | 0.118 |
|                                        | MP_COS     | 0.189 | 0.330 | 0.290 | 0.094 | 0.190 | 0.162 |
|                                        | MP_DOT     | 0.083 | 0.159 | 0.155 | 0.047 | 0.104 | 0.092 |
| **Our Approach**                        | DRMM_{CHXTV} | 0.253 | 0.407 | 0.357 | 0.247 | 0.404 | 0.341 |
|                                        | DRMM_{NHXTV} | 0.160 | 0.293 | 0.258 | 0.132 | 0.217 | 0.186 |
|                                        | DRMM_{LCHXTV} | 0.268 | 0.423 | 0.381 | 0.265 | 0.423 | 0.360 |
|                                        | DRMM_{CHXIDF} | 0.259 | 0.412 | 0.362 | 0.255 | 0.410 | 0.344 |
|                                        | DRMM_{NHXIDF} | 0.187 | 0.312 | 0.282 | 0.145 | 0.243 | 0.199 |
|                                        | DRMM_{LCHXIDF} | 0.279 | 0.431 | 0.382 | 0.275 | 0.437 | 0.371 |
## Retrieval Performance on ClueWeb-09-Cat-B

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Model Name</th>
<th>Topic Titles</th>
<th>Topic Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MAP</td>
<td>nDCG@20</td>
</tr>
<tr>
<td><strong>Traditional Retrieval Baselines</strong></td>
<td>QL</td>
<td>0.100</td>
<td>0.224</td>
</tr>
<tr>
<td></td>
<td>BM25</td>
<td>0.101</td>
<td>0.225</td>
</tr>
<tr>
<td><strong>Representation-Focused Matching Baselines</strong></td>
<td>DSSM&lt;sub&gt;T&lt;/sub&gt;</td>
<td>0.054</td>
<td>0.132</td>
</tr>
<tr>
<td></td>
<td>DSSM&lt;sub&gt;D&lt;/sub&gt;</td>
<td>0.039</td>
<td>0.099</td>
</tr>
<tr>
<td></td>
<td>CDSSM&lt;sub&gt;T&lt;/sub&gt;</td>
<td>0.064</td>
<td>0.253</td>
</tr>
<tr>
<td></td>
<td>CDSSM&lt;sub&gt;D&lt;/sub&gt;</td>
<td>0.054</td>
<td>0.134</td>
</tr>
<tr>
<td></td>
<td>ARC-I</td>
<td>0.024</td>
<td>0.073</td>
</tr>
<tr>
<td><strong>Interaction-Focused Matching Baselines</strong></td>
<td>ARC-II</td>
<td>0.033</td>
<td>0.087</td>
</tr>
<tr>
<td></td>
<td>MP&lt;sub&gt;IND&lt;/sub&gt;</td>
<td>0.056</td>
<td>0.139</td>
</tr>
<tr>
<td></td>
<td>MP&lt;sub&gt;COS&lt;/sub&gt;</td>
<td>0.066</td>
<td>0.158</td>
</tr>
<tr>
<td></td>
<td>MP&lt;sub&gt;DOT&lt;/sub&gt;</td>
<td>0.044</td>
<td>0.109</td>
</tr>
<tr>
<td><strong>Our Approach</strong></td>
<td>DRMM&lt;sub&gt;CHXTV&lt;/sub&gt;</td>
<td>0.103</td>
<td>0.245</td>
</tr>
<tr>
<td></td>
<td>DRMM&lt;sub&gt;NHXTV&lt;/sub&gt;</td>
<td>0.065</td>
<td>0.151</td>
</tr>
<tr>
<td></td>
<td>DRMM&lt;sub&gt;LCHXTV&lt;/sub&gt;</td>
<td>0.111</td>
<td>0.250</td>
</tr>
<tr>
<td></td>
<td>DRMM&lt;sub&gt;CHXIDF&lt;/sub&gt;</td>
<td>0.104</td>
<td>0.252</td>
</tr>
<tr>
<td></td>
<td>DRMM&lt;sub&gt;NHXIDF&lt;/sub&gt;</td>
<td>0.066</td>
<td>0.151</td>
</tr>
<tr>
<td></td>
<td>DRMM&lt;sub&gt;LCHXIDF&lt;/sub&gt;</td>
<td>0.113</td>
<td>0.258</td>
</tr>
</tbody>
</table>
## Retrieval Performance

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Model Name</th>
<th>Topic Titles</th>
<th>Topic Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MAP</td>
<td>nDCG@2</td>
</tr>
<tr>
<td>Our Approach</td>
<td>DRMM&lt;sub&gt;CHXTV&lt;/sub&gt;</td>
<td>0.253</td>
<td>0.407</td>
</tr>
<tr>
<td></td>
<td>DRMM&lt;sub&gt;NHXTV&lt;/sub&gt;</td>
<td>0.160</td>
<td>0.293</td>
</tr>
<tr>
<td></td>
<td>DRMM&lt;sub&gt;LCHXTV&lt;/sub&gt;</td>
<td>0.268</td>
<td>0.423</td>
</tr>
<tr>
<td></td>
<td>DRMM&lt;sub&gt;CHXIDF&lt;/sub&gt;</td>
<td>0.259</td>
<td>0.412</td>
</tr>
<tr>
<td></td>
<td>DRMM&lt;sub&gt;NHXIDF&lt;/sub&gt;</td>
<td>0.187</td>
<td>0.312</td>
</tr>
<tr>
<td></td>
<td>DRMM&lt;sub&gt;LCHXIDF&lt;/sub&gt;</td>
<td><strong>0.279</strong></td>
<td><strong>0.431</strong></td>
</tr>
</tbody>
</table>

- **LCH-based histogram > CH-based histogram > NH-based histogram**
  - CH-based > NH-based: Document length information is important in ad-hoc retrieval
  - LCH-based best: input signals with reduced range, and non-linear transformation useful for learning multiplicative relationships

- **IDF-based Term Gating > Term vector-based Gating**
  - Term vectors do not contain sufficient information
  - Model using term vectors introduces too many parameters to be learned sufficiently
Leverage structured knowledge

- Explainable Product Search with Knowledge Base Embedding
Knowledge Base Embedding

• Reasoning is a form of explanation
• Reasoning using hard-rules over knowledge graph is inefficient and difficult to generalize
• Knowledge graph embedding makes it easier to calculate the similarity between any pair of entity

transE: translation-based embedding

\[ h + \ell \approx t \]
\[ d(h + \ell, t) \]

Minimize the hinge-loss to learn entity and relation embeddings

\[ \mathcal{L} = \sum_{(h, \ell, t) \in S} \sum_{(h', \ell, t') \in S'} \left[ \gamma + d(h + \ell, t) - d(h' + \ell, t') \right]_+ \]
User-Product Knowledge Graph

- Include both user interactions on products, and our knowledge about the products

$$
\mathcal{L} = \sum_{(u,v,i) \in D(u,v,i)} \log P(i|u,v) + \sum_{(x,y) \in S(x,y)} \log P(y|x,r)
$$

$$
v = f(q) = f(\{w_q|w_q \in q\})
$$

$$
f(q) = \tanh(W \cdot \frac{\sum_{w_q \in q} w_q}{|q|} + b)
$$
Generating Search Results

• Given user embedding \( u \), query embedding \( f(q) \), and candidate item embedding \( i \), rank \( i \)'s by similarity between \( u + f(q) \) and \( i \)
Generating Search Explanations

- Finding path on the knowledge graph

  “we retrieve this dress for Alice because she often writes about fashion in her reviews and fashion is frequently used to describe the dress by other users”
# Amazon Product Datasets

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Electronics</th>
<th>Kindle Store</th>
<th>CDs &amp; Vinyl</th>
<th>Cell Phones &amp; Accessories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of reviews</td>
<td>1,689,188</td>
<td>982,618</td>
<td>1,097,591</td>
<td>194,439</td>
</tr>
<tr>
<td>Number of users</td>
<td>192,403</td>
<td>68,223</td>
<td>75,258</td>
<td>27,879</td>
</tr>
<tr>
<td>Number of items</td>
<td>63,001</td>
<td>61,934</td>
<td>64,443</td>
<td>10,429</td>
</tr>
<tr>
<td>Number of brands</td>
<td>3,525</td>
<td>1</td>
<td>1,414</td>
<td>955</td>
</tr>
<tr>
<td>Number of categories</td>
<td>983</td>
<td>2,523</td>
<td>770</td>
<td>206</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relationships</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Write per user</td>
<td>777.23±1748.6</td>
<td>1174.23±3682.39</td>
<td>1846.88±7667.51</td>
<td>500.01±979.78</td>
</tr>
<tr>
<td>Write per item</td>
<td>2373.62±5860.33</td>
<td>1293.47±1916.72</td>
<td>2156.83±4024.15</td>
<td>1336.64±2698.30</td>
</tr>
<tr>
<td>Also_bought per item</td>
<td>36.70±38.56</td>
<td>82.56±29.92</td>
<td>57.28±39.22</td>
<td>56.53±35.82</td>
</tr>
<tr>
<td>Also_viewed per item</td>
<td>4.36±9.44</td>
<td>0.16±1.66</td>
<td>0.27±1.86</td>
<td>1.24±4.29</td>
</tr>
<tr>
<td>Bought_together per item</td>
<td>0.59±0.72</td>
<td>0.00±0.04</td>
<td>0.68±0.80</td>
<td>0.81±0.77</td>
</tr>
<tr>
<td>Brand per item</td>
<td>0.47±0.50</td>
<td>0.00±0.00</td>
<td>0.21±0.41</td>
<td>0.52±0.50</td>
</tr>
<tr>
<td>Category per item</td>
<td>4.39±0.95</td>
<td>9.85±2.61</td>
<td>7.25±3.13</td>
<td>3.49±1.08</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Train/Test</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of reviews</td>
<td>1,275,432/413,756</td>
<td>720,006/262,612</td>
<td>804,090/293,501</td>
<td>150,048/44,391</td>
</tr>
<tr>
<td>Number of queries</td>
<td>904/85</td>
<td>3313/1290</td>
<td>534/160</td>
<td>134/31</td>
</tr>
<tr>
<td>Number of user-query pairs</td>
<td>1,204,928/5,505</td>
<td>1,490,349/232,668</td>
<td>1,287,214/45,490</td>
<td>114,177/665</td>
</tr>
<tr>
<td>Relevant items per pair</td>
<td>1.12±0.48/1.01±0.09</td>
<td>1.87±3.30/1.48±1.94</td>
<td>2.57±6.59/1.30±1.19</td>
<td>1.52±1.13/1.00±0.05</td>
</tr>
</tbody>
</table>
Experimental setup

• We adopt the 3-step approach (Van Gysel et al. 2016) to construct the query
  – Extract the multi-level category information of item a purchased item $v_j$
  – Concatenate the terms as a topic string
  – Remove stopwords and duplicate words

• Baselines:
  – Query likelihood (QL)
  – BM25
  – LambdaMART
  – Latent Space Embedding (LSE) (Van Gysel et al. 2016)
  – Hierarchical Embedding Model (Ai et al. 2017)
Search Performance

- Better than baselines (query likelihood, Latent Semantic Entity model (LSE), Hierarchical Embedding Model (HEM))
- Using more relation types (i.e., more knowledge) is better

<table>
<thead>
<tr>
<th></th>
<th>Electronics</th>
<th>Kindle Store</th>
<th></th>
<th>CDs &amp; Vinyl</th>
<th>Cell Phones &amp; Accessories</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAP</td>
<td>MRR</td>
<td>NDCG</td>
<td>MAP</td>
<td>MRR</td>
</tr>
<tr>
<td>QL</td>
<td>0.289</td>
<td>0.289</td>
<td>0.316</td>
<td>0.011</td>
<td>0.012</td>
</tr>
<tr>
<td>BM25</td>
<td>0.283</td>
<td>0.280</td>
<td>0.304</td>
<td>0.021</td>
<td>0.013</td>
</tr>
<tr>
<td>LambdaMART</td>
<td>0.180</td>
<td>0.181</td>
<td>0.237</td>
<td>0.028</td>
<td>0.029</td>
</tr>
<tr>
<td>LSE</td>
<td>0.233</td>
<td>0.234</td>
<td>0.239</td>
<td>0.006</td>
<td>0.007</td>
</tr>
<tr>
<td>HEM</td>
<td>0.308++</td>
<td>0.309++</td>
<td>0.329++</td>
<td>0.029</td>
<td>0.035†</td>
</tr>
<tr>
<td>DREMNoMeta</td>
<td>0.291</td>
<td>0.291</td>
<td>0.319</td>
<td>0.036*</td>
<td>0.044*</td>
</tr>
<tr>
<td>DREMAB</td>
<td>0.283</td>
<td>0.283</td>
<td>0.312</td>
<td>0.043++</td>
<td>0.052++</td>
</tr>
<tr>
<td>DREMAV</td>
<td>0.318++</td>
<td>0.319++</td>
<td>0.349++</td>
<td>0.035*</td>
<td>0.043*</td>
</tr>
<tr>
<td>DREMBT</td>
<td>0.320++</td>
<td>0.321++</td>
<td>0.346++</td>
<td>0.037*</td>
<td>0.045*</td>
</tr>
<tr>
<td>DREM_Bnd</td>
<td>0.314++</td>
<td>0.315++</td>
<td>0.340++</td>
<td>0.037*</td>
<td>0.044*</td>
</tr>
<tr>
<td>DREM_Cat</td>
<td>0.299*</td>
<td>0.300+</td>
<td>0.360++</td>
<td>0.048++</td>
<td>0.056++</td>
</tr>
<tr>
<td>DREMAll</td>
<td>0.366++†</td>
<td>0.367++†</td>
<td>0.408+++†</td>
<td>0.057+++†</td>
<td>0.067+++†</td>
</tr>
</tbody>
</table>
Generating Explanations

Generating Explanations

Query: “sports outdoors accessory electronics gadget fitness track”

- $u+\hat{S}P+B \rightarrow \text{Pebble Technology} \leftarrow i_p + \hat{B} (5.81\%):$
  “Based on your profile and query, you may like to see some-things by Pebble Technology, and Pebble Smartwatch by Pebble Technology is a top product of this brand.”

- $u+\hat{S}P+C \rightarrow \text{Clothing, Shoes, Jewelry} \leftarrow i_p + \hat{C} (3.17\%):$
  “Based on your profile and query, you may like to see some-things in Clothing, Shoes, Jewelry, and Pebble Smartwatch by Pebble Technology is a top product in this category.”

- $u+\hat{S}P+C \rightarrow \text{Health&Personal Care} \leftarrow i_j + \hat{C} (0.184\%):$
  “Based on your profile and query, you may like to see some-things in Health&Personal Care, and Up 24 Activity Tracker by Jawbone is a top product in this category.”

- $u+\hat{S}P+C \rightarrow \text{Sports&Outdoors} \leftarrow i_j + \hat{C} (2.32\%):$
  “Based on your profile and query, you may like to see some-things in Sports&Outdoors, and Up 24 Activity Tracker by Jawbone is a top product in this category.”
Post-hoc explanation methods for search

- Post-hoc explanation
  - Construct a second model to interpret the trained model
  - Usually model agnostic (i.e. works for any trained model)

- EXS: Explainable Search Using Local Model Agnostic Interpretability (J. Singh and A. Anand 2019)

- Primary goal: aid users in answering the following questions:
  - Why is this document relevant to the query?
  - Why is this document ranked higher than the other?
  - What is the intent of the query according to the ranker?

- Basic Idea: Adapt LIME to search task
LIME (Ribeiro et al. 2016)

- **Local Interpretable Model-agnostic Explanations**
- For a trained model $f$ and an instance $x$, the explanation by LIME is obtained by:
  \[
  \hat{x}(x) = \text{argmin}_{g \in G} L(f, g, \pi_x) + \Omega(g)
  \]
  - $G$: a class of interpretable models (e.g. sparse linear models)
  - $\pi_x(z)$: a proximity measure between an instance $z$ to $x$, so as to define locality around $x$ (e.g. $\pi_x(z) = \exp\left(- \frac{D(x,z)^2}{\sigma^2}\right)$)
  - $L(f, g, \pi_x)$: a measure of how unfaithful $g$ is in approximating $f$ in the locality defined by $\pi_x$ (e.g. $L(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) \left(f(z) - g(z')\right)^2$)
Adapt LIME to search task

• For a trained binary model $B$ and a doc $d$, train a simple linear SVM $M_d$ on a feature space of words that minimize:
  
  $L(B, M_d, \pi_d)$

  - $L(B, M_d, \pi_d)$: difference between predictions of $M_d$ and $B$ for all $d' \in \pi_d$
  - $d' \in \pi_d$ is created by removing random words from random positions in $d$

• How to convert a ranker $R$ into a classifier $B$:
  - Estimate $P(X = \text{relevant} \mid q, d', R)$
  - Top-k Binary: $P(X = \text{relevant} \mid q, d', R) = 1$ if $R(q, d') > R(q, d_k')$
  - Score based: $P(X = \text{relevant} \mid q, d', R) = 1 - \frac{R(q, d_1) - R(q, d')}{{R(q, d_1)}}$
  - Rank based: $P(X = \text{relevant} \mid q, d', R) = 1 - \frac{\text{rank}(d')}{k}$
Visualizing Explanations

• Why is this document $d$ relevant to the query?
  – Show the sign and magnitude of learned coefficients of $M_d$ along with the associated words

Figure 1: The EXS User Interface. The top bar of the application houses the retrieval model selector (A), Score-to-Probability converter for LIME (B), search box (C) and the Explain Intent button in that order. To the right is the rank depth input box (E) and the corpus selector (D). The left pane shows the search results for the query 'Rail Strikes' according to DRMM. The right pane shows the output of clicking on the 'Explain' button corresponding to the top result. The bar chart on the right shows the words in the document that make it relevant and irrelevant according to DRMM. The green bar indicates the strength of a word for the relevant class and red for the irrelevant class. EXS can be found at http://bit.ly/exs-search

(J.Singh and A.Anand 2019)
Visualizing Explanations

- Why is this document $d_A$ ranked higher than another $d_B$?
  - Set $k = \text{rank}(d_B)$ and $d_k = d_B$. $M_{d_A}$ now tells us which words in $d_A$ are strong indictors when compared to the threshold set by $d_B$
  - Only show the positive words

![Bar chart showing word importance](attachment:chart.png)

*Figure 3: Explanation for AP890710-0178 vs AP890713-0045 for the query 'Rail Strikes' when using DRMM*

(J.Singh and A.Anand 2019)
Visualizing Explanations

• What is the intent of the query according to the ranker?
  – Aggregating $m_d$ for all $d \in D_q^k$
  – add the coefficients of each word $w \in m_d$ for all $m_d$ and show top words and coefficients to users

![Graph showing word importance](image)

Figure 2: Intent explanation for the query 'Rail Strikes' when using DRMM to rank documents from a news collection.

(J.Singh and A.Anand 2019)
Axiomatic analysis of search models

• Seek a set of desirable properties of retrieval models as formal constraints (or axioms)
• Analyze and diagnose retrieval models with formal constraints
• Provide theoretical guidance on how to optimize a retrieval model and how to design novel retrieval models
Axiomatic analysis of search models

• A Formal Study of Information Retrieval Heuristics (Fang et al. 2004)
  – Define 7 formal constraints on retrieval models
  – Analytically examine three representative retrieval models with these constraints
    • Pivoted model, Okapi Mode, Dirichlet Prior Method
  – Empirically show that the satisfaction of the constraints is correlated with good ranking performance
    • The violation of the constraints often indicates non-optimality of the retrieval model
    • Constraints analysis reveals optimal ranges of parameters
# Seven Relevance Constraints

<table>
<thead>
<tr>
<th>Constraints</th>
<th>Intuitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFC1</td>
<td>To favor a document with more occurrences of a query term</td>
</tr>
<tr>
<td>TFC2</td>
<td>To ensure that the amount of increase in score due to adding a query term repeatedly must decrease as more terms are added</td>
</tr>
<tr>
<td>TFC3</td>
<td>To favor a document matching more distinct query terms</td>
</tr>
<tr>
<td>TDC</td>
<td>To penalize the words popular in the collection and assign higher weights to discriminative terms</td>
</tr>
<tr>
<td>LNC1</td>
<td>To penalize a long document (assuming equal TF)</td>
</tr>
<tr>
<td>LNC2, TF-LNC</td>
<td>To avoid over-penalizing a long document</td>
</tr>
<tr>
<td>TF-LNC</td>
<td>To regulate the interaction of TF and document length</td>
</tr>
</tbody>
</table>

(Fang et al. 2004)(Fang et al. 2011)
Term Frequency Constraints

• TFC1
  – Intuition: give a higher score to a document with more occurrences of a query term
  – Let $Q$ be a query and $D$ be a document
  – If $q \in Q$ and $t \notin Q$, then $S(Q, D \cup \{q\}) > S(Q, D \cup \{t\})$

\[ S(Q, D_1) > S(Q, D_2) \]  \hspace{1cm} (Fang and Zhai, 2014)
Term Frequency Constraints

- **TFC2**
  - Intuition: require that the amount of increase in the score due to adding a query term must decrease as we add more terms.
  - Let $Q$ be a query with only one term $q$
  - Let $D$ be a document,
    then $S(Q, D \cup \{q\}) - S(Q, D) > S(Q, D \cup \{q\} \cup \{q\}) - S(Q, D \cup \{q\})$

\[\begin{align*}
Q: & \quad q \\
D_1: & \quad \quad q \\
D_2: & \quad q \quad q \\
D_3: & \quad q \quad q
\end{align*}\]

\[S(D_2, Q) - S(D_1, Q) > S(D_3, Q) - S(D_2, Q)\]

(Fang and Zhai, 2014)
Term Frequency Constraints

- **TFC3**
  - Intuition: favor a document with more distinct query terms
  - Let \( q \) be a query and \( w_1, w_2 \) be two query terms.

Assume \( idf(w_1) = idf(w_2) \) and \( |d_1| = |d_2| \)

if \( c(w_1, d_2) = c(w_1, d_1) + c(w_2, d_1) \)
and \( c(w_2, d_2) = 0, c(w_1, d_1) \neq 0, c(w_2, d_1) \neq 0 \)
then \( S(q, d_1) > S(q, d_2) \)

\[
S(d_1, q) > S(d_2, q) \quad \text{(Fang and Zhai, 2014)}
\]
Term Discrimination Constraint

- **TDC**
  - Intuition: to penalize the words popular in the collection and assign higher weights to discriminative terms
  - Let $Q = \{q_1, q_2\}$ Assume $|D_1| = |D_2|$ and $c(q_1, D_1) + c(q_2, D_1) = c(q_1, D_2) + c(q_2, D_2)$. If $idf(q_1) \geq idf(q_2)$ and $c(q_1, D_1) \geq c(q_1, D_2)$, we have $S(q, D_1) \geq S(q, D_2)$
Length Normalization Constraints

• **LNC1**
  – Intuition: penalize long documents
  – Let $Q$ be a query and $D$ be a document.
  – If $t$ is a non-query term, then $S(Q, D \cup \{t\}) < S(Q, D)$

• **LNC2**
  – Intuition: avoid over-penalizing long documents
  – Let $Q$ be a query and $D$ be a document.
  – If $D \cap Q \neq \emptyset$, and $D_k$ is constructed by concatenating $D$ with itself $k$ times, then $S(Q, D_k) \geq S(Q, D)$
Analyze Neural IR Models with Formal Constraints

• For traditional IR models, the satisfaction of the constraints is correlated with good empirical performance (Fang et al. 2004)

• The formal constraints should also be useful in analyzing and optimizing the neural IR models

• Some recent work on this direction:
  – An Axiomatic Approach to Diagnosing Neural IR Models (Rennnings et al. 2019)
  – An Axiomatic Approach to Regularizing Neural Ranking Models (Rosset et al. 2019)
  – Teach Machine How to Read: Reading Behavior Inspired Relevance Estimation (Li et al. 2019)
An Axiomatic Approach to Diagnosing Neural IR Models (Rennnings et al. 2019)

- Create diagnostic datasets based on the relevance constraints including:
  - TFC1, TFC2, M-TDC, LNC2
  - With necessary extensions and relaxations
  - By sampling queries and documents pairs/triplets that match the condition of the axioms (do not require relevance labels!)

---

**Fig. 1:** Overview of the diagnostic dataset creation pipeline. In *italics*, we show an example for the TFC2 axiom as extracted from question 1317 on passages from Wikipedia document 283 in the WikiPassageQA dataset, and refer to appended documents as an example of artificial data (for LNC2).
An Axiomatic Approach to Diagnosing Neural IR Models (Rennnings et al. 2019)

- Use the diagnostic dataset to test whether neural IR models’ output is consistent with the axioms

<table>
<thead>
<tr>
<th>Retrieval effectiveness</th>
<th>Performance per axiom</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>MRR</td>
</tr>
<tr>
<td>BM25</td>
<td>0.52\textsuperscript{3,4}</td>
</tr>
<tr>
<td>QL</td>
<td>0.54\textsuperscript{1,3,4}</td>
</tr>
<tr>
<td>Duet</td>
<td>0.25</td>
</tr>
<tr>
<td>MatchPyramid</td>
<td>0.44\textsuperscript{3}</td>
</tr>
<tr>
<td>DRMM</td>
<td>0.55\textsuperscript{1,2,3,4}</td>
</tr>
<tr>
<td>aNMM</td>
<td>0.57\textsuperscript{1,2,3,4}</td>
</tr>
</tbody>
</table>

- Find a positive but not significant correlation (0.44) between MAP and the average fraction of fulfilled axiom instances
An Axiomatic Approach to Regularizing Neural Ranking Models (Rosset et al. 2019)

- Use IR axioms to augment the labeled data for training neural ranking models
  - For each document $d$ and constraint $\Delta_i$, generate a perturbed document $d^{(i)}$ to regularize the pairwise hinge loss function (i.e. increase the loss if the ranking model fails to satisfy constraint $\Delta_i$ on the pair $d$ and $d^{(i)}$)

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Perturbation</th>
<th>Expected result</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFC1-A</td>
<td>Sample a query term from query $q$ and insert it at a random position in $d$</td>
<td>$d^{(i)} &gt;_q d$</td>
</tr>
<tr>
<td>TFC1-D</td>
<td>Sample a query term from query $q$ and delete it in $d$</td>
<td>$d^{(i)} &lt;_q d$</td>
</tr>
<tr>
<td>TFC3</td>
<td>Sample a query term not present in $d$, and insert it in $d$.</td>
<td>$d^{(i)} &gt;_q d$</td>
</tr>
<tr>
<td>LNC</td>
<td>Sample $k$ terms and insert them at random positions in $d$</td>
<td>$d^{(i)} &lt;_q d$</td>
</tr>
</tbody>
</table>
An Axiomatic Approach to Regularizing Neural Ranking Models (Rosset et al. 2019)

- Experiment on MS-MARCO ranking dataset
  - Neural ranking model: CKNRM (Dai et al. 2018)
  - Axiomatic Regularization can improve the ranking performance, especially when the size of training data is limited

![MRR on Different Data Sizes](image)

**Figure 1:** MRR results of training CKNRM and its axiomatic variant on datasets with 100, 1k, 10k, 100k, and all MS-MARCO queries on the dev set. Each point represents the ensemble of four independently trained models.

**Table 2:** An add-one-in ablation study of each of the axiomatic losses; the last row shows all axioms.

<table>
<thead>
<tr>
<th>Ablation on 10k Queries</th>
<th>MAP</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CKNRM</td>
<td>15.13</td>
<td>15.36</td>
</tr>
<tr>
<td>+ TFC1-A</td>
<td>19.33</td>
<td>19.56</td>
</tr>
<tr>
<td>+ TFC1-D</td>
<td>18.16</td>
<td>18.38</td>
</tr>
<tr>
<td>+ TFC3</td>
<td>19.05</td>
<td>19.28</td>
</tr>
<tr>
<td>+ LNC</td>
<td>11.42</td>
<td>11.47</td>
</tr>
<tr>
<td>+ All Axioms</td>
<td>19.70</td>
<td>19.95</td>
</tr>
</tbody>
</table>
Teach Machine How to Read: Reading Behavior Inspired Relevance Estimation (Li et al. 2019)

- Retrieval models try to approximate users’ relevance judgment of a query-doc pair
- By investigating how the user makes relevance judgment, we may be able to find some human-inspired heuristic constraints that are useful for improving retrieval models
How does a human make relevance judgment?

- Conduct an **eye-tracking study** to log users’ eye-fixations during relevance judgment task (Li et al. 2018)
  - A two-stage relevance judgment process
    - Stage 1: preliminary relevance judgment
    - Stage 2: reading with preliminary relevance judgment
Heuristic Constraints from Reading Behavior

- Define six reading heuristic

<table>
<thead>
<tr>
<th>#</th>
<th>Heuristic</th>
<th>Description</th>
<th>Implication for retrieval models</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Sequential reading</td>
<td>Reading direction is from top to bottom</td>
<td>The presented order of the content may affect relevance</td>
</tr>
<tr>
<td>b</td>
<td>Vertical decaying attention</td>
<td>Reading attention is decaying vertically</td>
<td>Retrieval model should assign more weights to the text at the beginning of documents</td>
</tr>
<tr>
<td>c</td>
<td>Query centric guidance</td>
<td>Reading attention is higher in the contexts around query terms</td>
<td>Retrieval models should follow IR heuristics [7] and capture the interactions between query and document</td>
</tr>
<tr>
<td>d</td>
<td>Context-aware reading</td>
<td>Reading behavior is influenced by the relevance perception from previously read text</td>
<td>The local relevance of the text should also depend on its surrounding context</td>
</tr>
<tr>
<td>e</td>
<td>Selective attention</td>
<td>Users will skip some seemingly irrelevant text during relevance judgement</td>
<td>Retrieval models should ignore the text that has no or little influence on relevance</td>
</tr>
<tr>
<td>f</td>
<td>Early stop reading</td>
<td>Users will stop reading once the read text is enough to make relevance judgement</td>
<td>Retrieval models should be able to estimate the relevance without processing the whole document</td>
</tr>
</tbody>
</table>

- Analyze whether existing neural IR models satisfy these heuristics

<table>
<thead>
<tr>
<th>Models</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARC-I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARC-II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>DRMM</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Match Pyramid</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>KNRM</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PACRR</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DeepRank</td>
<td></td>
<td>√</td>
<td></td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HiNT</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Incorporating Reading Heuristics

• Design a novel Reading Inspired Model (RIM)
  – Satisfy the proposed reading heuristics
  – Use reinforcement learning method to incorporate the selective attention and early stop reading heuristics into a neural retrieval model
Incorporating Reading Heuristics

- Incorporating the reading heuristic constraints do improve the ranking performance

<table>
<thead>
<tr>
<th></th>
<th>Test-SAME (PSCM)</th>
<th>Test-DIFF (UBM)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NDCG@1</td>
<td>NDCG@3</td>
</tr>
<tr>
<td>BM25</td>
<td>0.7048*</td>
<td>0.7202*</td>
</tr>
<tr>
<td>ARC-I</td>
<td>0.7583*</td>
<td>0.7647*</td>
</tr>
<tr>
<td>ARC-II</td>
<td>0.7239*</td>
<td>0.7347*</td>
</tr>
<tr>
<td>DRMM</td>
<td>0.6958*</td>
<td>0.7141*</td>
</tr>
<tr>
<td>MatchPyramid</td>
<td>0.6851*</td>
<td>0.7028*</td>
</tr>
<tr>
<td>KNRM</td>
<td>0.6997*</td>
<td>0.7121*</td>
</tr>
<tr>
<td>PACRR</td>
<td>0.7072*</td>
<td>0.7219*</td>
</tr>
<tr>
<td>DeepRank</td>
<td>0.7058*</td>
<td>0.7227*</td>
</tr>
<tr>
<td>HiNT</td>
<td>0.7550*</td>
<td>0.7592*</td>
</tr>
<tr>
<td>RIM</td>
<td>0.7746*</td>
<td>0.7705*</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Test-NTCIR-13</th>
<th>Test-NTCIR-14</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NDCG@1</td>
<td>NDCG@3</td>
</tr>
<tr>
<td>BM25</td>
<td>0.6099*</td>
<td>0.6194*</td>
</tr>
<tr>
<td>ARC-I</td>
<td>0.5933*</td>
<td>0.6153*</td>
</tr>
<tr>
<td>ARC-II</td>
<td>0.6466*</td>
<td>0.6649*</td>
</tr>
<tr>
<td>DRMM</td>
<td>0.6866*</td>
<td>0.6490*</td>
</tr>
<tr>
<td>MatchPyramid</td>
<td>0.6866*</td>
<td>0.6507*</td>
</tr>
<tr>
<td>KNRM</td>
<td>0.6700*</td>
<td>0.6564*</td>
</tr>
<tr>
<td>PACRR</td>
<td>0.6700*</td>
<td>0.6661*</td>
</tr>
<tr>
<td>DeepRank</td>
<td>0.6750*</td>
<td>0.6606*</td>
</tr>
<tr>
<td>HiNT</td>
<td>0.6566*</td>
<td>0.6599*</td>
</tr>
<tr>
<td>RIM</td>
<td><strong>0.7050</strong></td>
<td><strong>0.6797</strong></td>
</tr>
</tbody>
</table>
Outline

• Background and motivation
  – What is explainable search?
  – Why do we need explainable search?

• Existing work on explainable search
  – How can we make search models more explainable?
    • Building Interpretable search models
    • Using structured knowledge
    • Post-hoc explanation methods for search
    • Axiomatic analysis of search models

• Wrap up
Wrap up

• **What** is explainable search **about**?
  – In narrow sense:
    • How to build an *interpretable* search model
  – In broad sense:
    • Re-examine the search problem from the *explainable AI/ML* perspective

• **Why** do we need explainable search?
  – For search users, to build better *mental models* for search
  – For system designers, to better deal with more powerful but *more complex* search systems
Wrap up

- **How can we make search models more explainable?**
  - We introduce some recent work which covers two dimensions of interpretability

<table>
<thead>
<tr>
<th>Method</th>
<th>Global vs Local</th>
<th>Intrinsic vs Post-hoc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building Interpretable search models</td>
<td>Global</td>
<td>Intrinsic</td>
</tr>
<tr>
<td>(Guo et al. 2016)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Using structured knowledge</td>
<td>Local</td>
<td>Intrinsic</td>
</tr>
<tr>
<td>(Explainable Product Search with Knowledge Base Embedding)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Post-hoc explanation methods for search</td>
<td>Local</td>
<td>Post-hoc</td>
</tr>
<tr>
<td>(J.Singh and A.Anand 2019)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Axiomatic analysis of search models</td>
<td>Global</td>
<td>Post-hoc</td>
</tr>
</tbody>
</table>
Reference


Reference


Reference

• (Li et al. 2019) Xiangsheng Li, Jiaxin Mao, Chao Wang, Yiqun Liu, Min Zhang and Shaoping Ma. “Teach Machine How to Read: Reading Behavior Inspired Relevance Estimation.” to appear in SIGIR 2019