ExplainAble Recommendation and Search (EARS)

Yongfeng Zhang
Rutgers University

Qingyao Ai
UMass Amherst

Jiaxin Mao
Tsinghua University

Xu Chen
Tsinghua University

Outline of the Tutorial

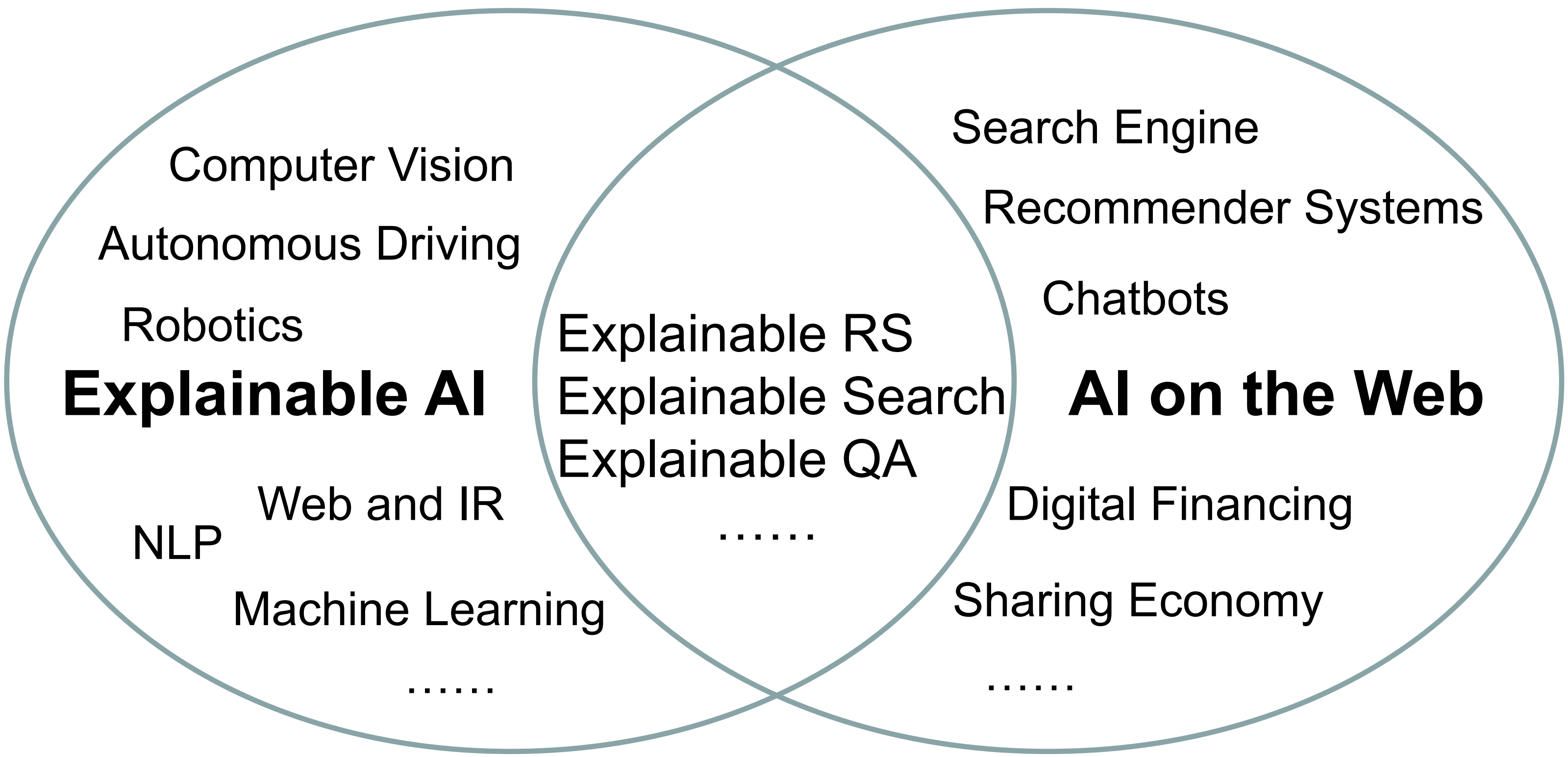
- Why Explainable Recommendation and Search
- A Unified View of Search, Recommendation, and Explainability
- Part 1: Explainable Recommendation
- Part 2: Explainable Search
- Summary

Rutgers · 清華大學 Tsinghua University · UMass Amherst

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

For Users	**For System Designers**
Why did you show this result to me? *A recommended item *A search result *Especially when result is **personalized**	Why does my system give this output?
	How to conduct system diagnostics?
Why should I trust the result?	Which component of system is wrong?
How should I take actions?	How to tune the system performance?
	How to increase system robustness?

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

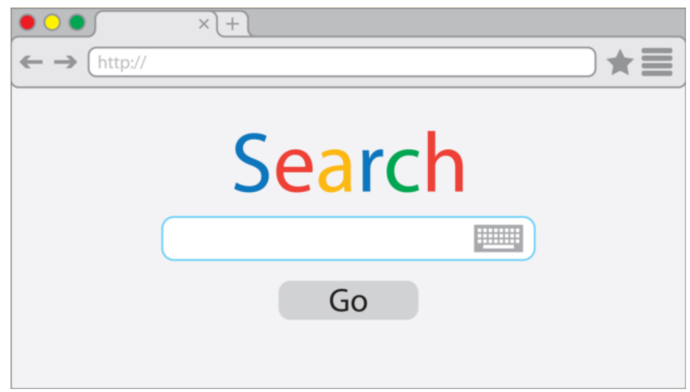
- EU General Data Protection Regulation (GDPR)
  - Article 5.2: a data controller “must be able to demonstrate that personal data are processed in a transparent manner in relation to the data subject”
  - Article 12 provides general rules on transparency, which apply to the provision of information (Articles 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.

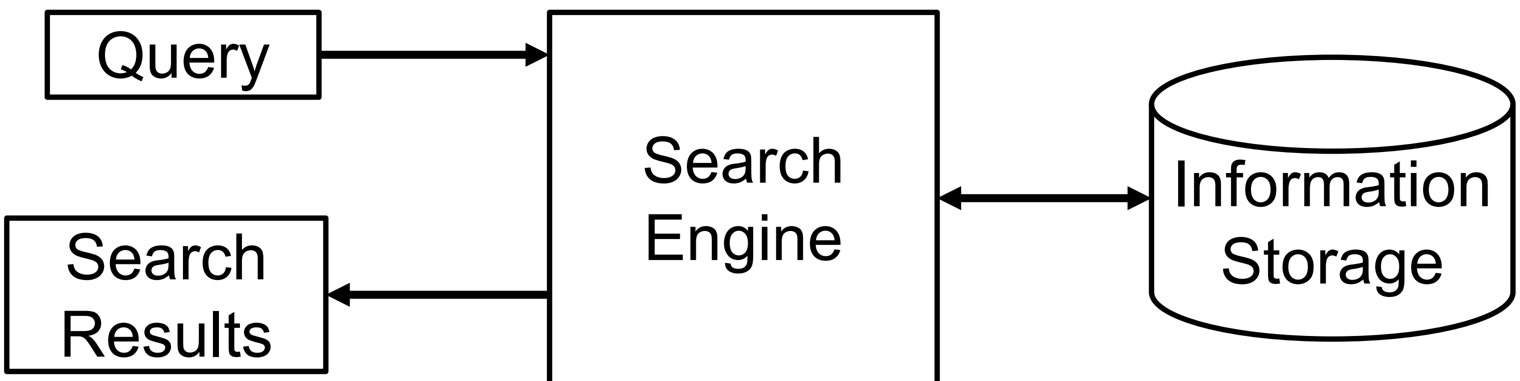
They are very good platforms to develop, verify and test explainable AI algorithms.

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.

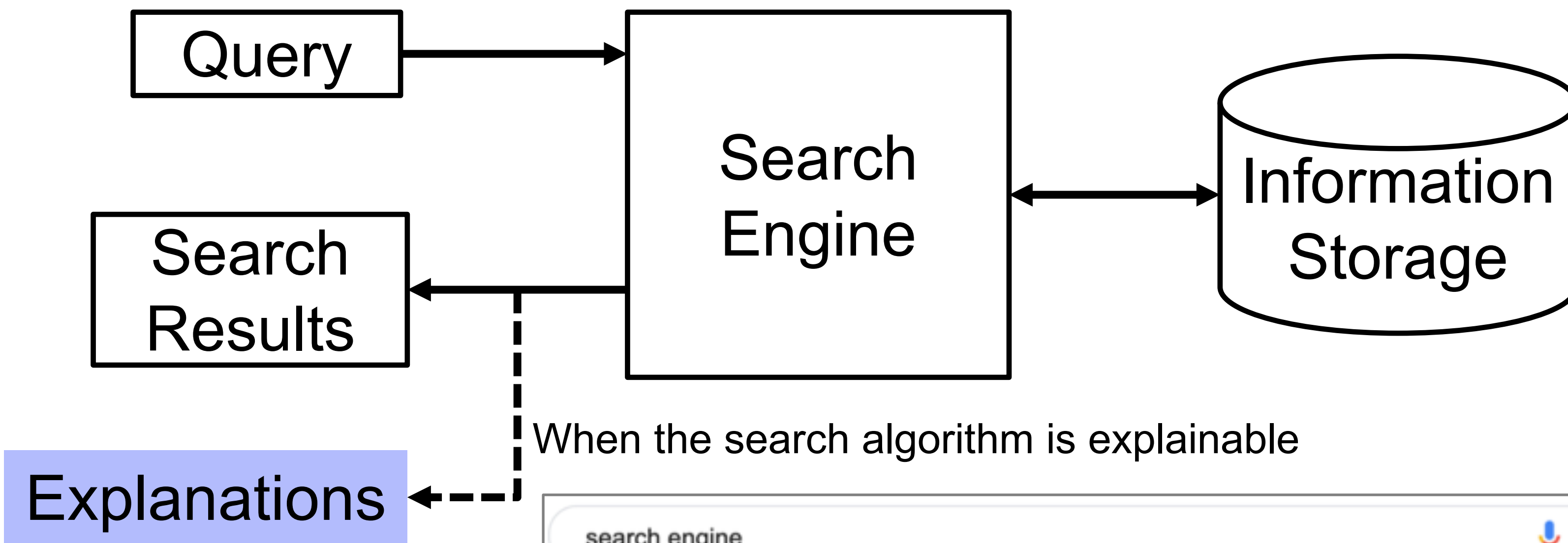


Web pages, answers, etc.

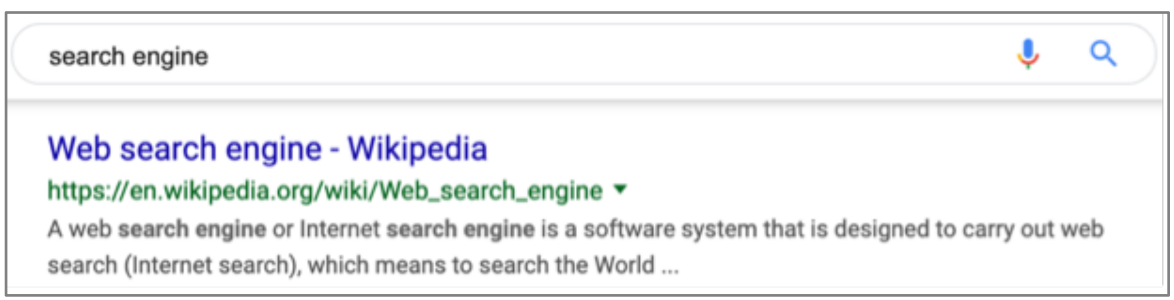
Web pages, images, videos, User Generated Contents (UGC), etc.

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.

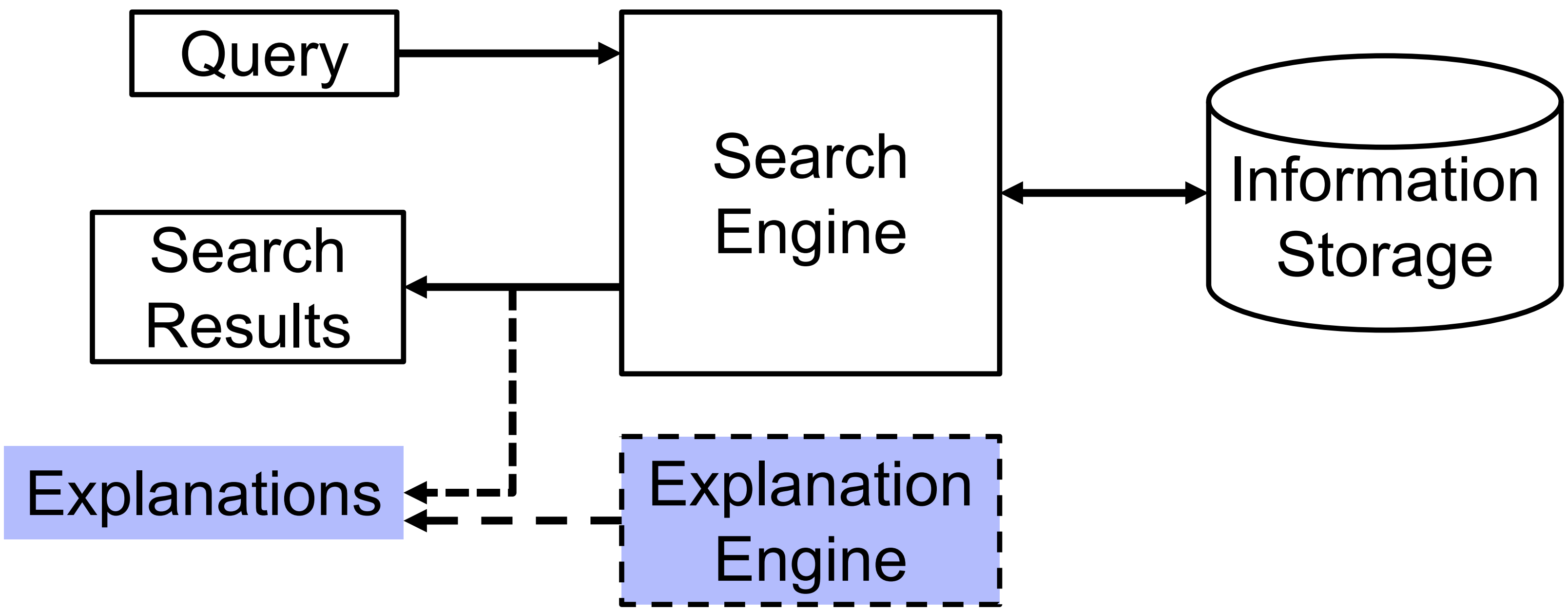


e.g., search snippets



An Overview of Search Systems

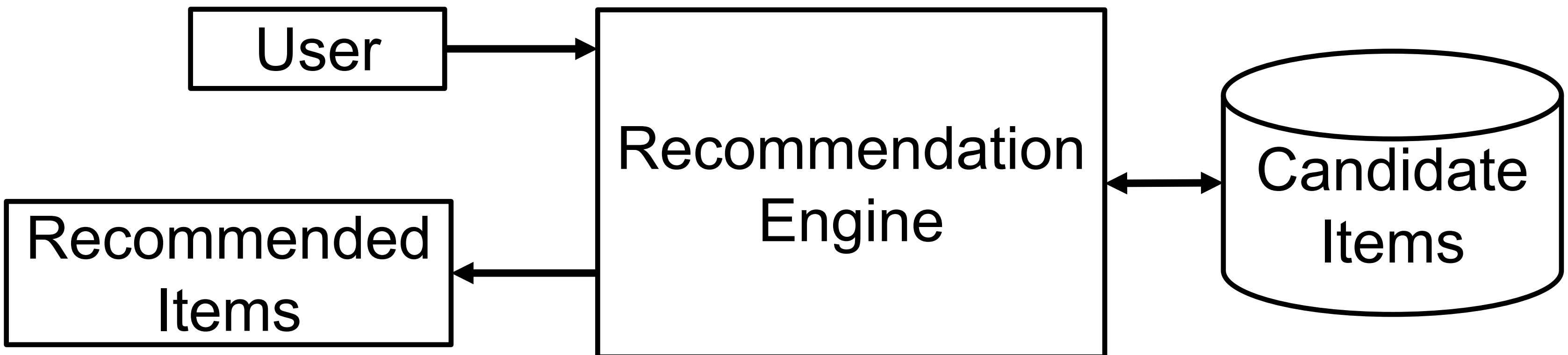
- From query to documents *and explanations*
  - User information need is explicitly represented by the search query
    - Search keywords, questions, etc.



When the search algorithm is not quite explainable..

An Overview of Recommendation Systems

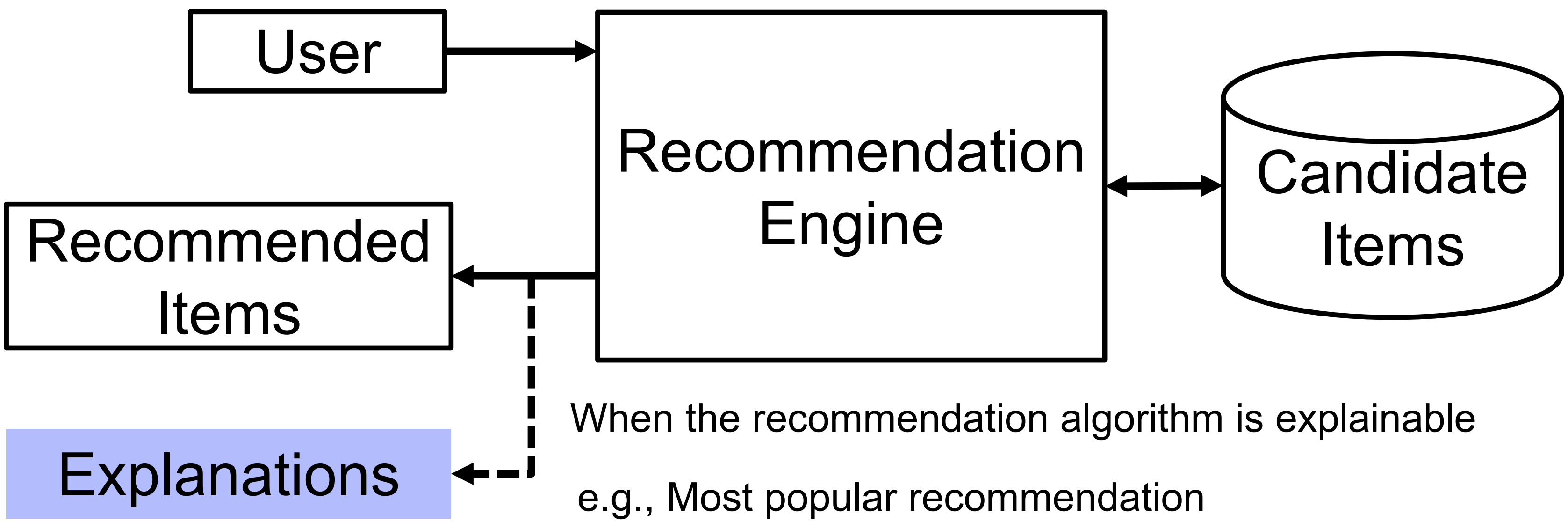
- From user to items and *explanations*
  - User information need is implicitly represented by the user profile
    - User content information, interaction history, etc.



Products, movies, music, videos, news, friends, twits, actions, etc.

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.

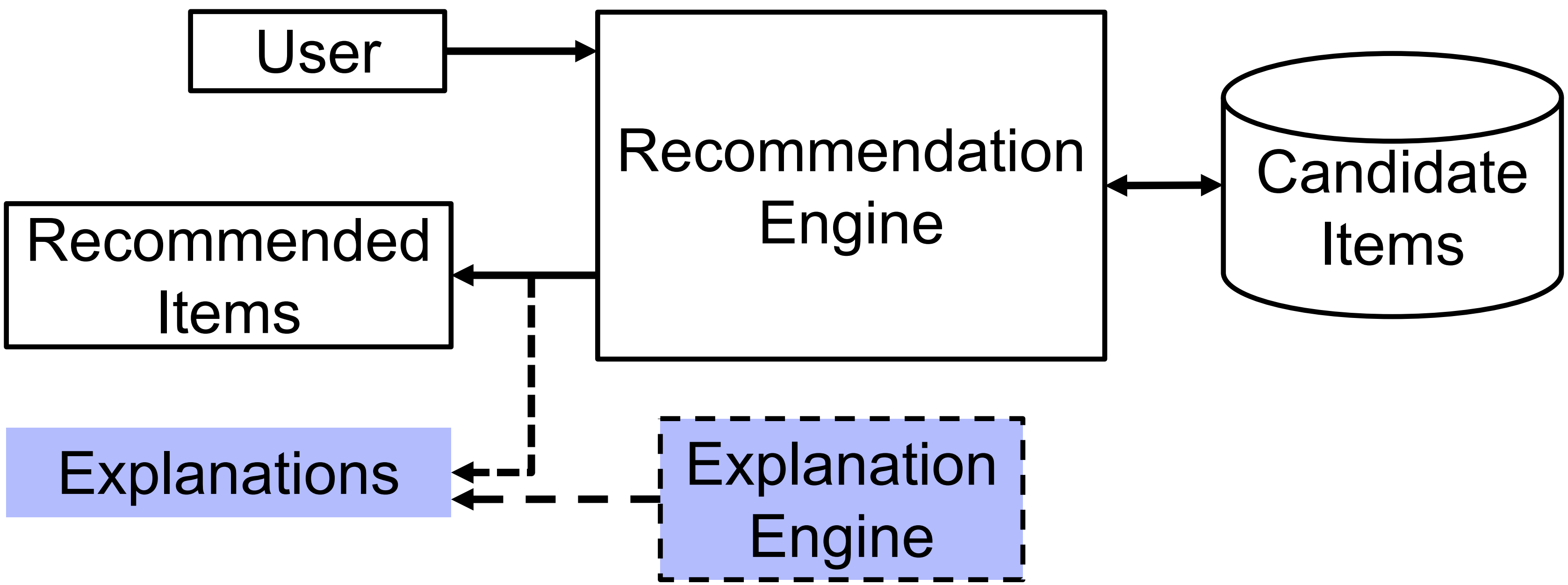


When the recommendation algorithm is explainable

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]

	Search	Recommendations
Delivery Mode	Pull	Push or Pull
Beneficiary	User	User and provider
Unexpected good?	No	Yes
Collective knowledge	Maybe	Maybe
Query available	Yes	Maybe
Context dependent	Maybe	Maybe

Courtesy Table from [2]

Unified View of Search and Recommendation

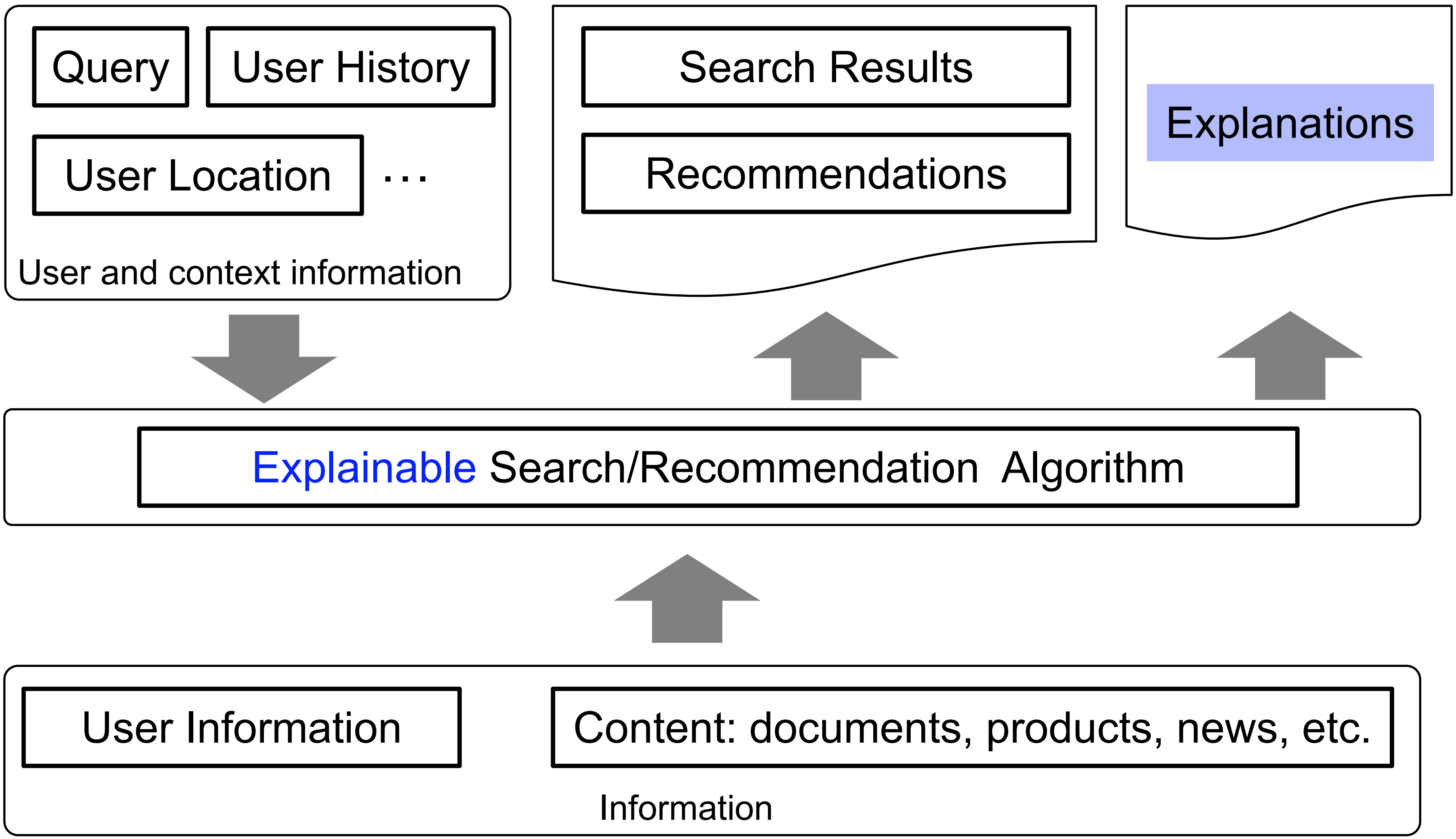


Figure adapted from [2]

Unified View of Search and Recommendation

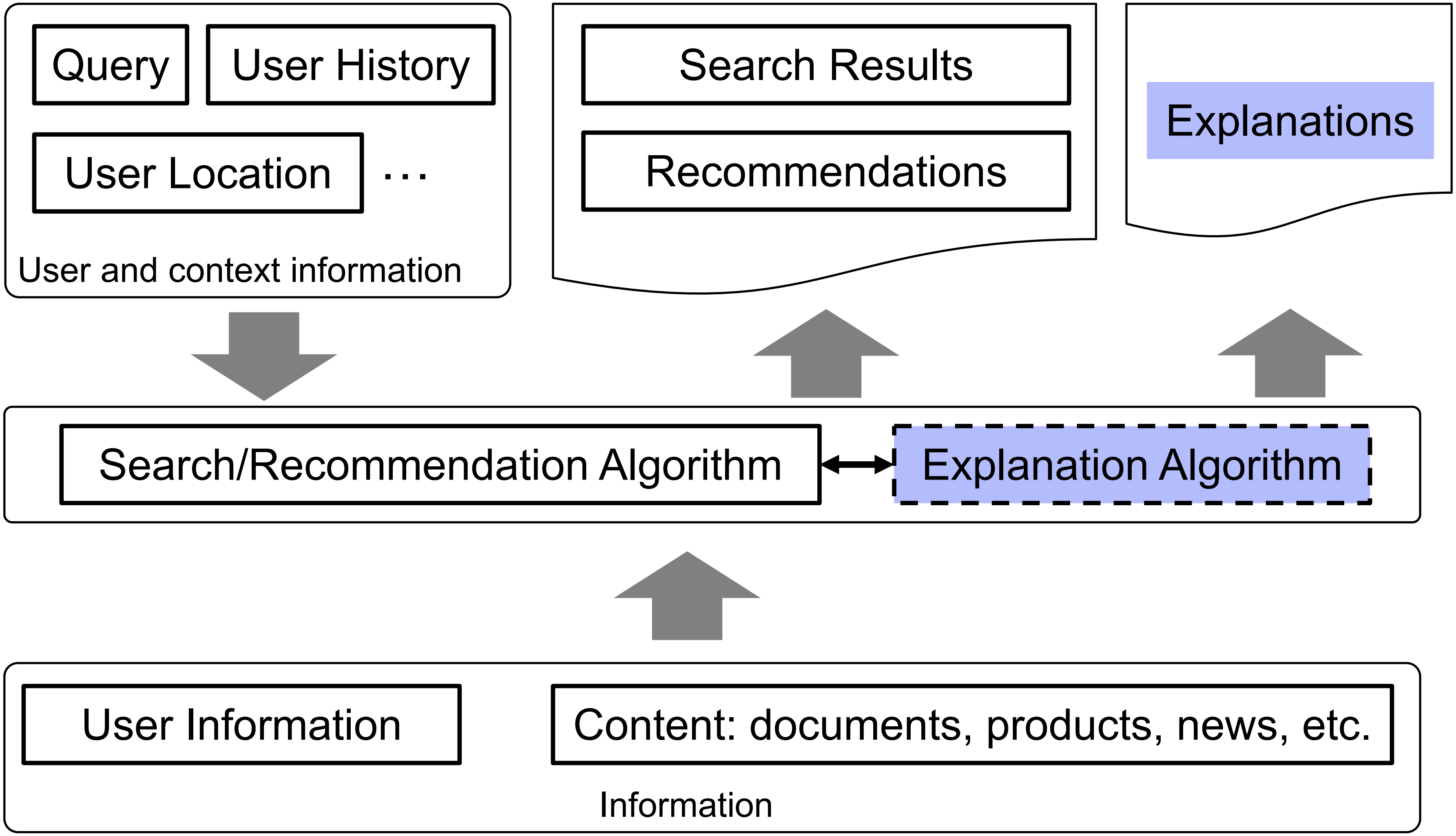
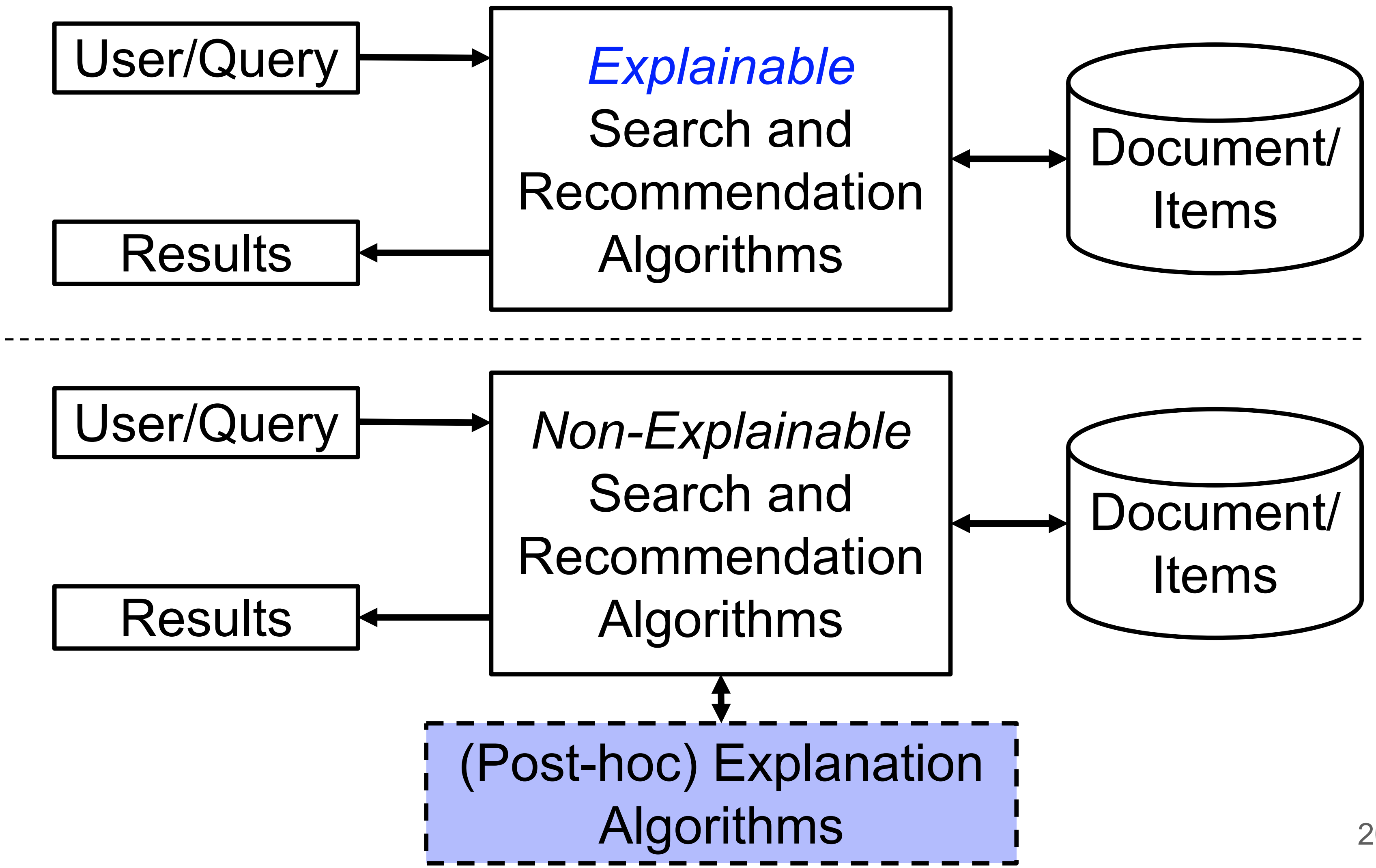


Figure adapted from [2]

About this tutorial



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

- [1] Nicholas J. Belkin, and W. Bruce Croft. "Information filtering and information retrieval: two sides of the same coin." In *Communications of the ACM*. 1992.
- [2] Hector Garcia-Molina, Georgia Koutrika, and Aditya Parameswaran. "Information seeking: convergence of search, recommendations and advertising." *Communications of the ACM* (2011).

Explainable Recommendation

Recommendation Systems – The 5W

- Recommendation system research can be broadly classified into 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
  - Content-based Recommendation [Balabanović et al. CACM'1997, Pazzani et al. AdapWeb'2007]
  - User-based Collaborative Filtering [Resnick et al. CSCW'1994]
  - Item-based Collaborative Filtering [Sarwar et al. WWW'2001]

Item Attributes

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
Into the Fire	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism

Content-based Recommendation and Explanation

- Item attributes

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
Into the Fire	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism

- User profile

Title	Genre	Author	Type	Price	Keywords
…	Fiction	Brunonia, Barry, Ken Follett	Paperback	25.65	Detective, murder, New York

Simple approach

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

$$\frac{|keywords(b_i) \cap keywords(b_j)|}{|keywords(b_i) \cup keywords(b_j)|}$$

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

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	5	3	3	4	4

- 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

$$\boldsymbol{sim(a,b)} = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2}}$$

- 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

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	5	3	3	4	4

- 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

$$sim(\vec{a}, \vec{b}) = \frac{\sum_{u \in U}(r_{u,a} - \overline{r_u})(r_{u,b} - \overline{r_u})}{\sqrt{\sum_{u \in U}(r_{u,a} - \overline{r_u})^2} \sqrt{\sum_{u \in U}(r_{u,b} - \overline{r_u})^2}}$$

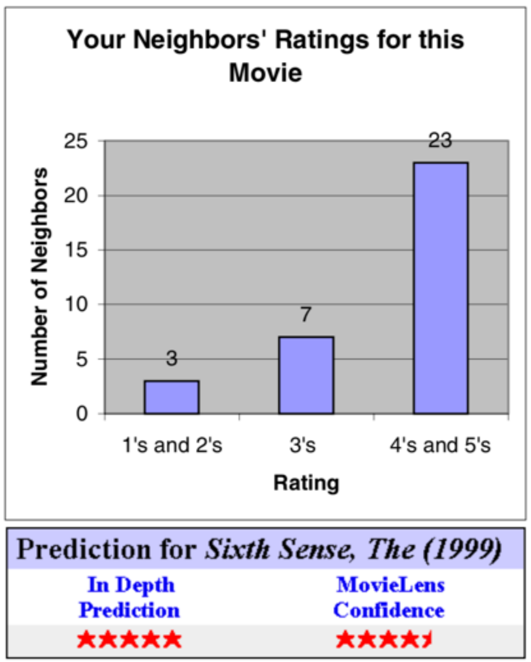
- Average similar items' rating on the target user as prediction, recommend if a high rating

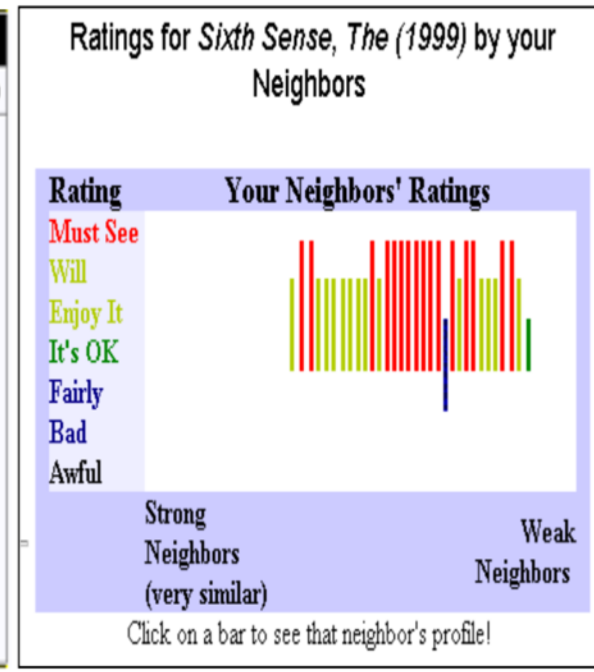
Explanation: You have highly rated items that are similar to this item

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]

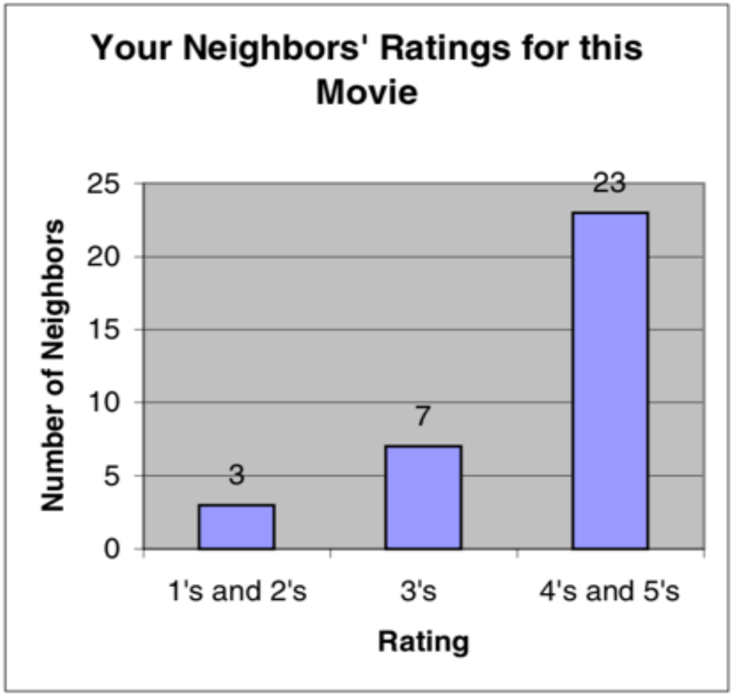




21 different explanation interfaces, 78 users on MovieLens website, each user was provided with 21 recommendations, each with a different explanation.

Ask users to rate on a scale of 1-7 how likely they would go and see the movie.

#		N	Mean Response	Std Dev
1	Histogram with grouping	76	5.25	1.29
2	Past performance	77	5.19	1.16
3	Neighbor ratings histogram	78	5.09	1.22
4	Table of neighbors ratings	78	4.97	1.29
5	Similarity to other movies rated	77	4.97	1.50
6	Favorite actor or actress	76	4.92	1.73
7	MovieLens percent confidence in prediction	77	4.71	1.02
8	Won awards	76	4.67	1.49
9	Detailed process description	77	4.64	1.40
10	# neighbors	75	4.60	1.29
11	No extra data – focus on system	75	4.53	1.20
12	No extra data – focus on users	78	4.51	1.35
13	MovieLens confidence in prediction	77	4.51	1.20
14	Good profile	77	4.45	1.53
15	Overall percent rated 4+	75	4.37	1.26
16	Complex graph: count, ratings, similarity	74	4.36	1.47
17	Recommended by movie critics	76	4.21	1.47
18	Rating and %agreement of closest neighbor	77	4.21	1.20
19	# neighbors with std. deviation	78	4.19	1.45
20	# neighbors with avg correlation	76	4.08	1.46
21	Overall average rating	77	3.94	1.22



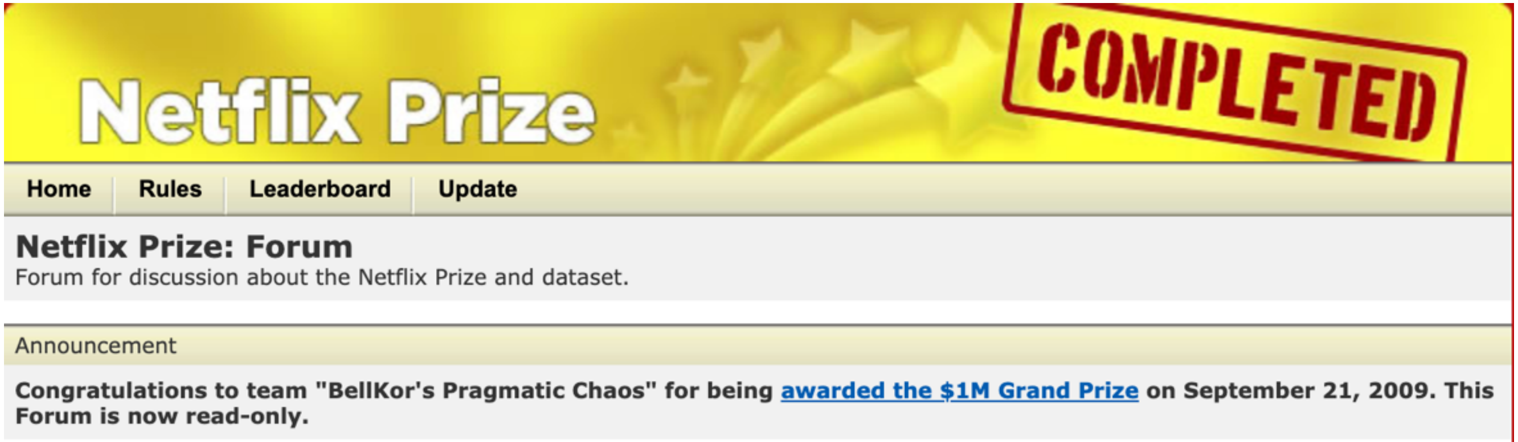
The most effective explanation based on Neighbors' ratings.
User-based CF: Users who have similar ratings with you highly rated this item

Shaded rows indicate explanations with a mean response significantly different from the base cases (two-tailed α = 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?

?	4	?	?	3	?
5	?	?	?	?	2
?	3	?	5	?	?
?	?	1	?	?	3
4	?	?	?	?	2

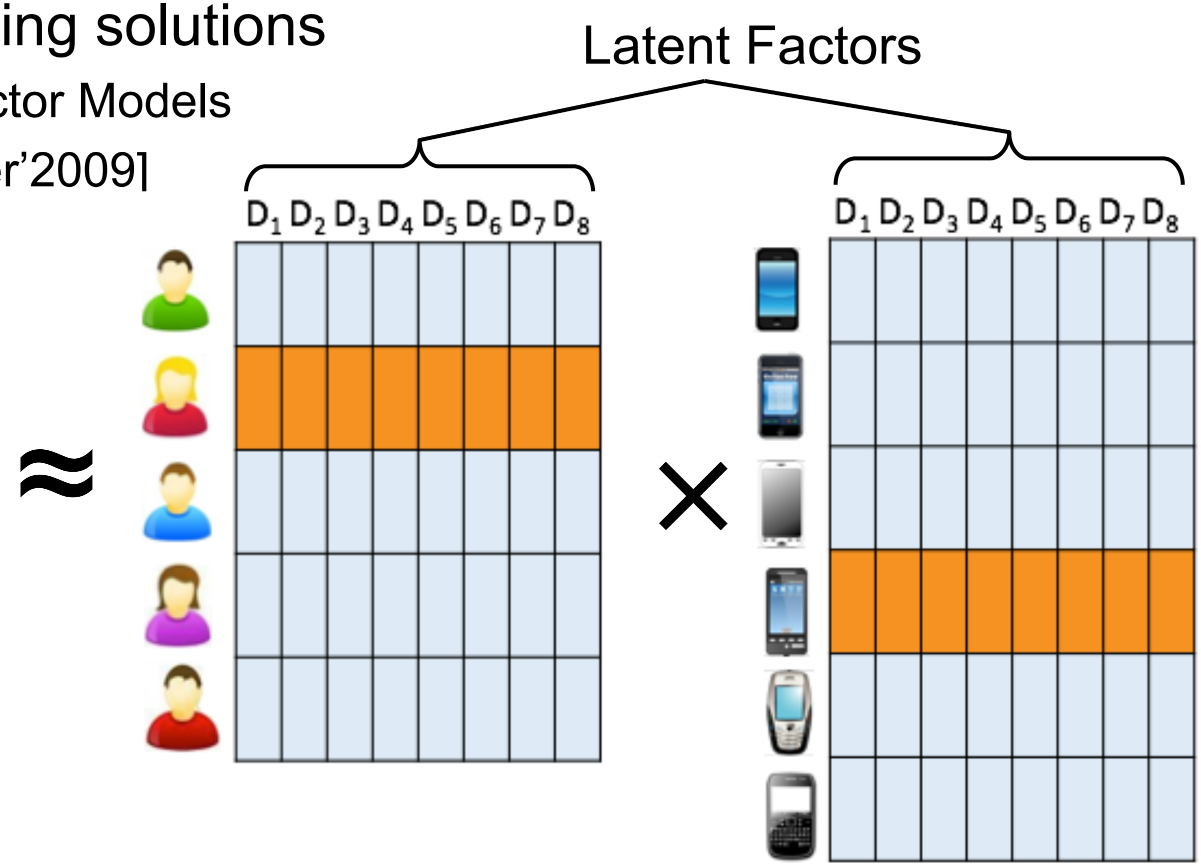
A key task:
Predict the missing ratings

Predict the Missing Ratings

Matrix Factorization for Recommendation

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

Original Matrix



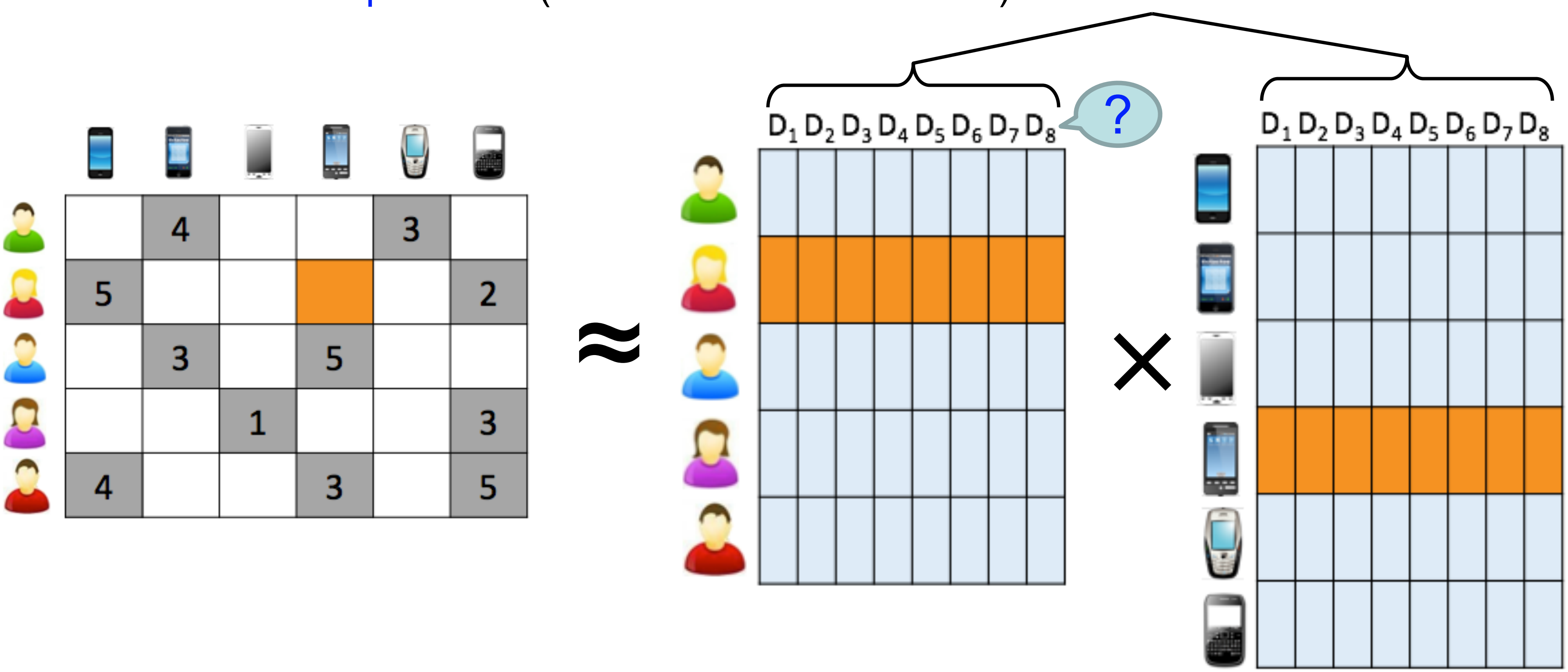
$$\hat{r}_{ui} = p_u^T q_i$$

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

Goodness of fit   Regularization

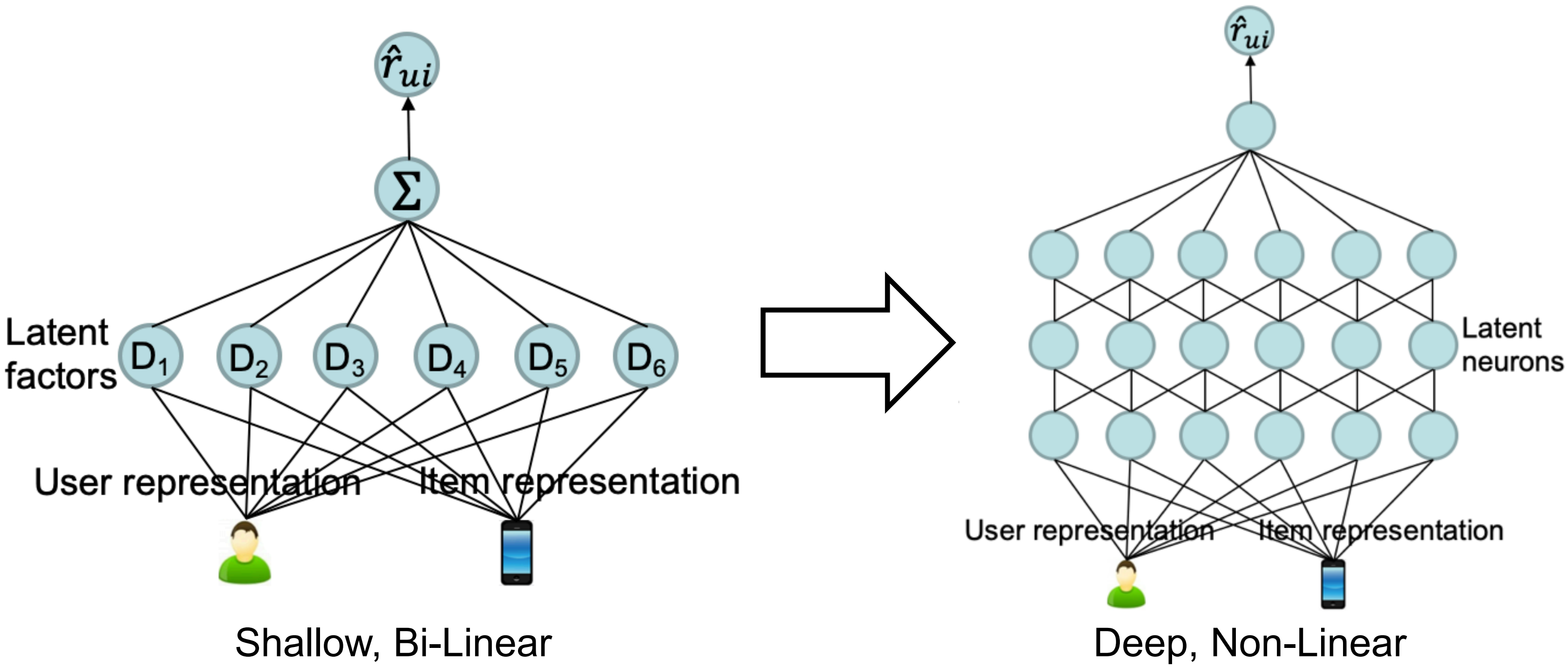
Explainability vs Accuracy

- 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

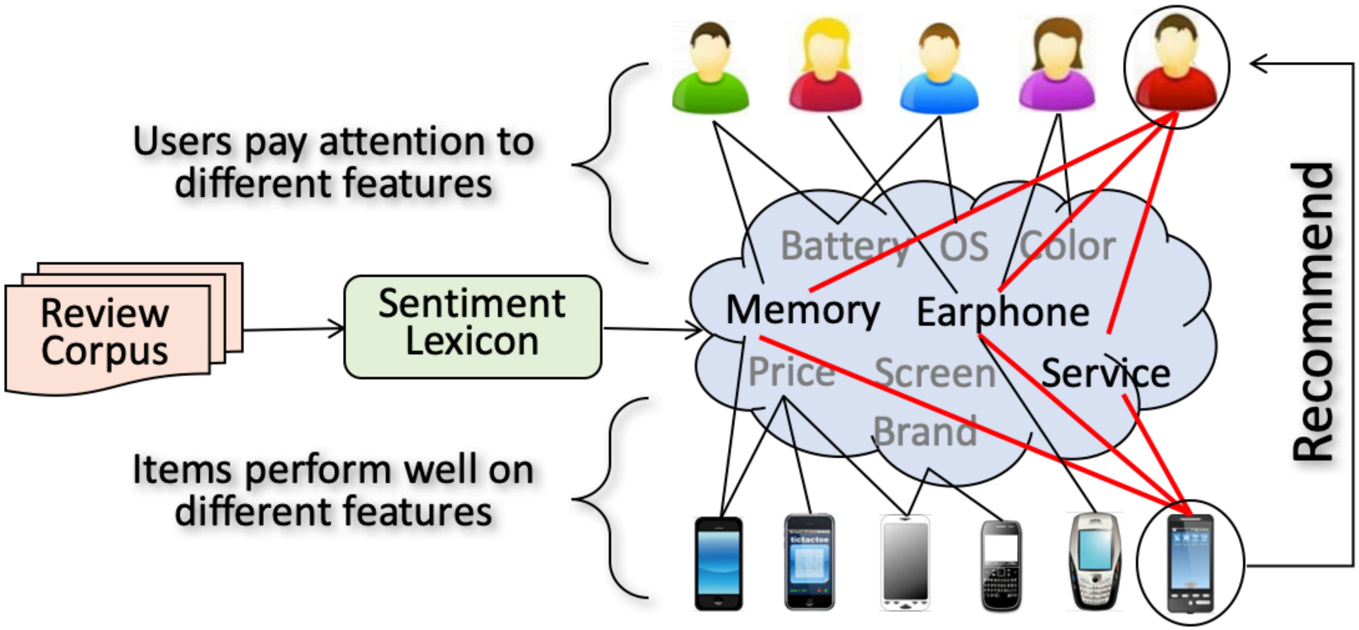
Factorization-based Approaches

- 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

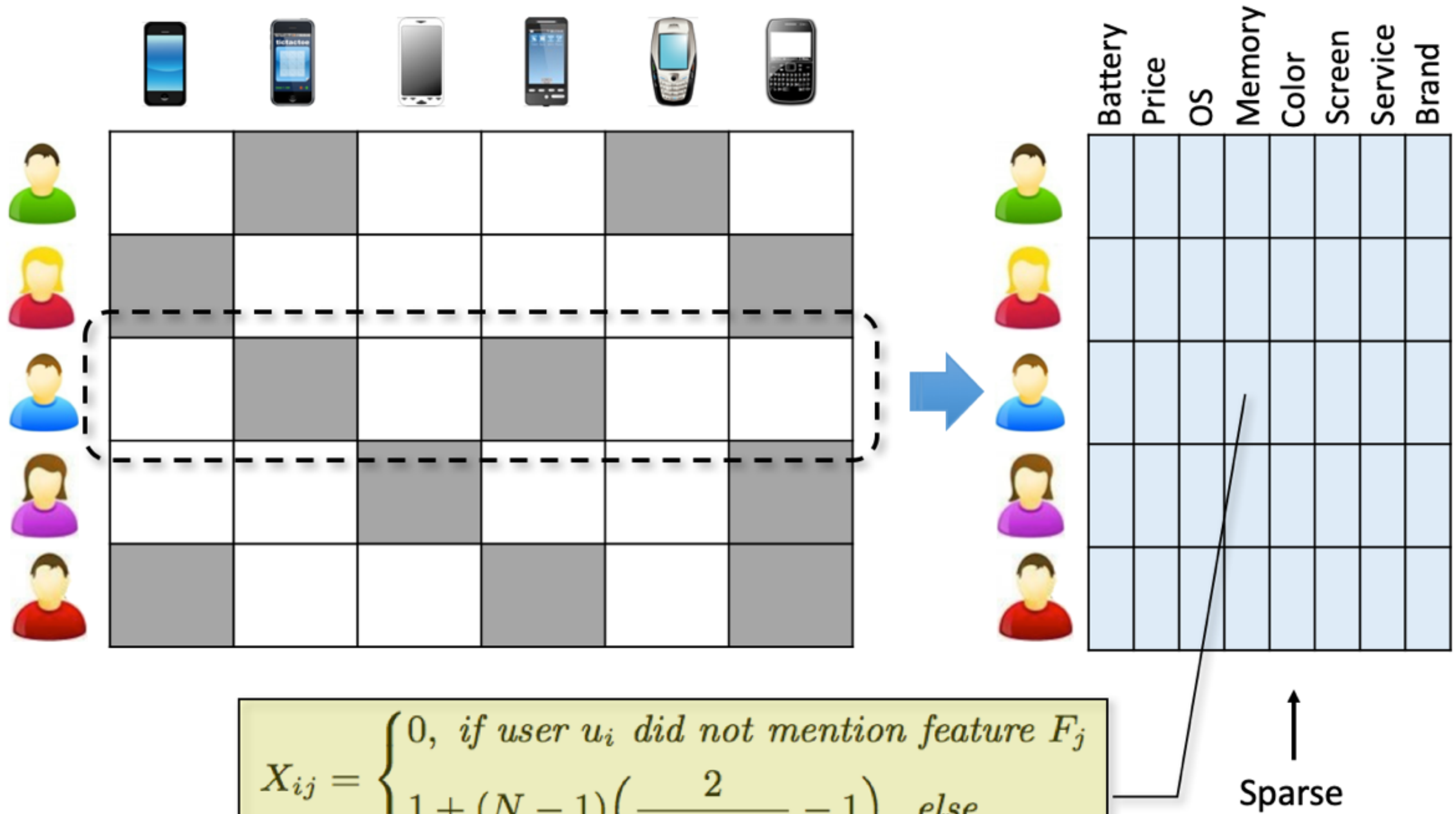
- Explicit factor models for explainable recommendation [Zhang et al. SIGIR'2014]
  - Formally introduced the Explainable Recommendation problem

– Basic idea: To recommend an item that performs well on the features that a user concerns.



Explicit Factor Model

- User-Feature Attention Matrix

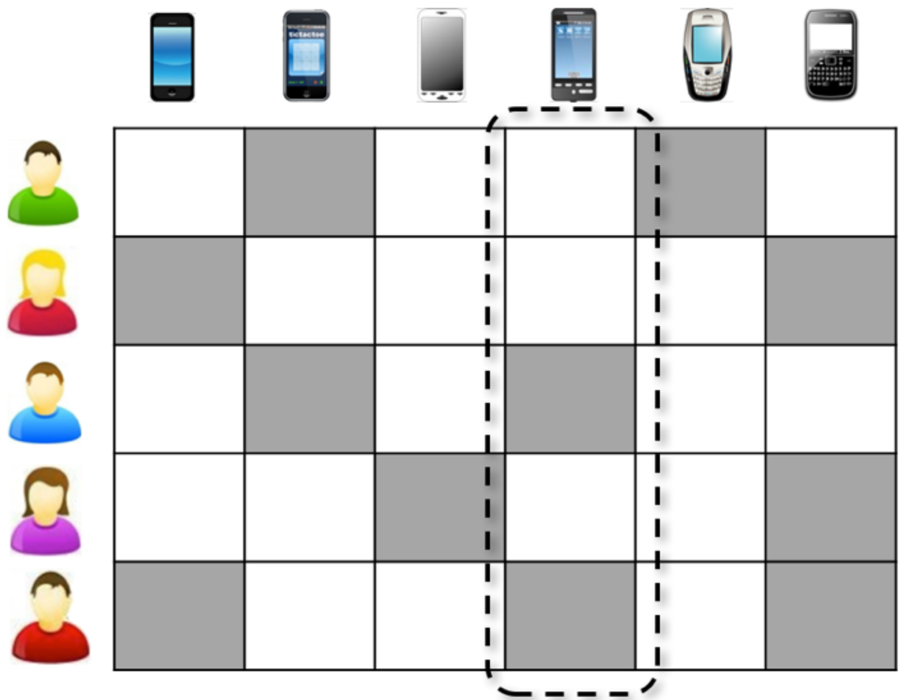


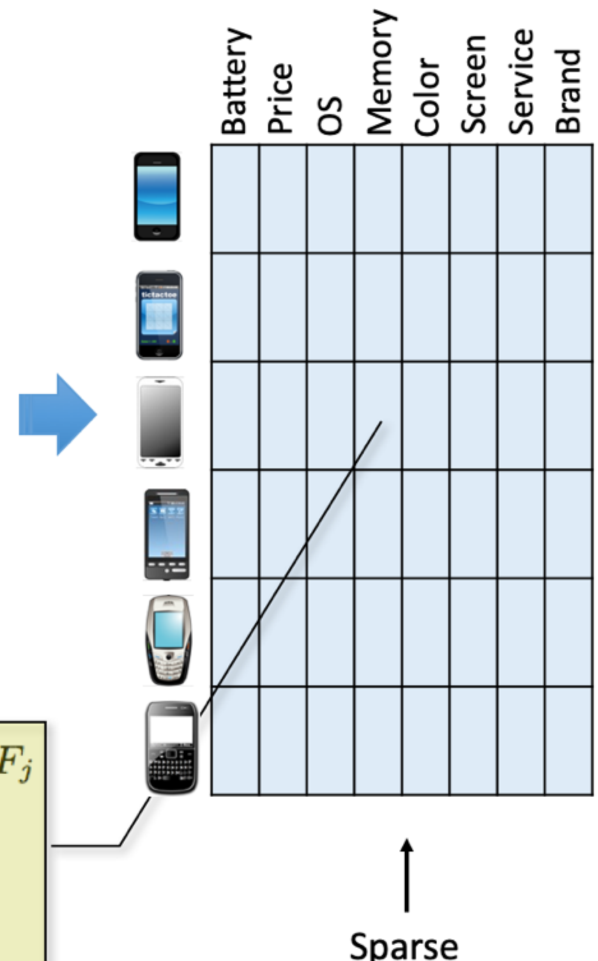
$$X_{ij} = \begin{cases} 0, & \textit{if user } u_i \textit{ did not mention feature } F_j \\ 1 + (N-1)\left(\frac{2}{1+e^{-t_{ij}}} - 1\right), & \textit{else} \end{cases}$$

t_{ij} is the frequency that user i mentions feature j

Explicit Factor Model

- Item-Feature Quality Matrix





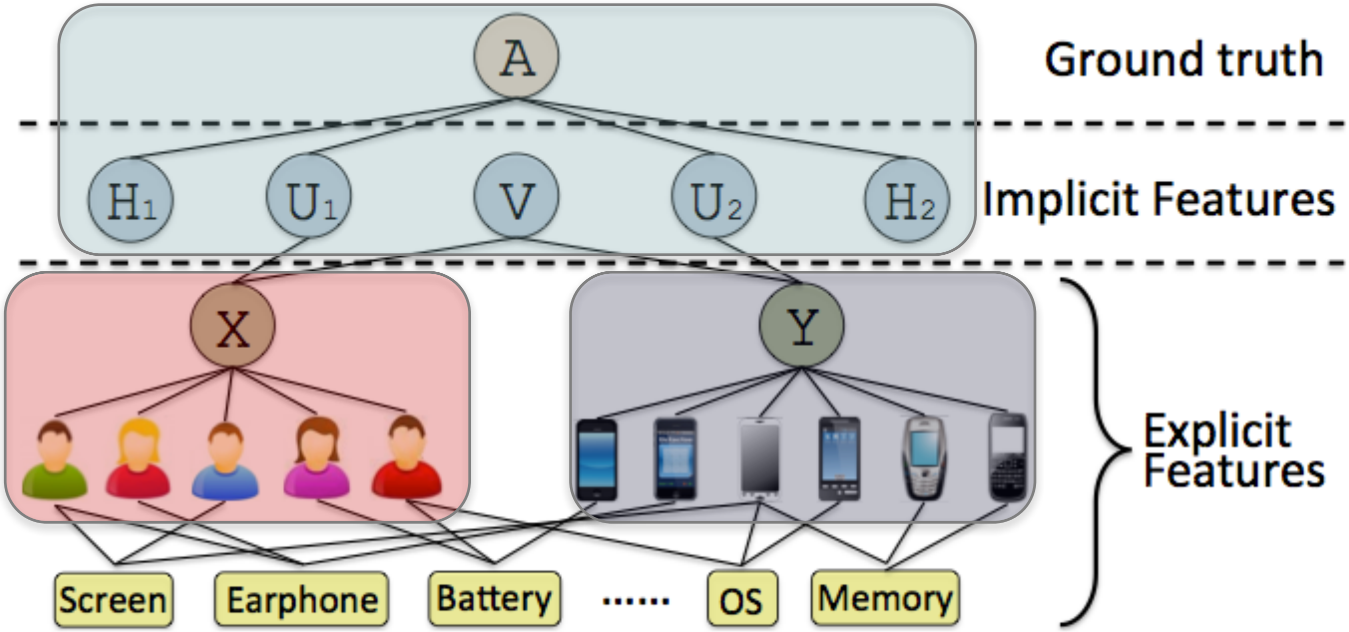
$$Y_{ij} = \begin{cases} 0, & \textit{if item } p_i \textit{ is not reviewed on feature } F_j \\ 1 + \dfrac{N-1}{1+e^{-k \cdot s_{ij}}}, & \textit{else} \end{cases}$$

k is the frequency feature j is mentioned on item i

s_{ij} is the average sentiment of these mentions

Explicit Factor Model

- Integrating the Explicit and Implicit Features

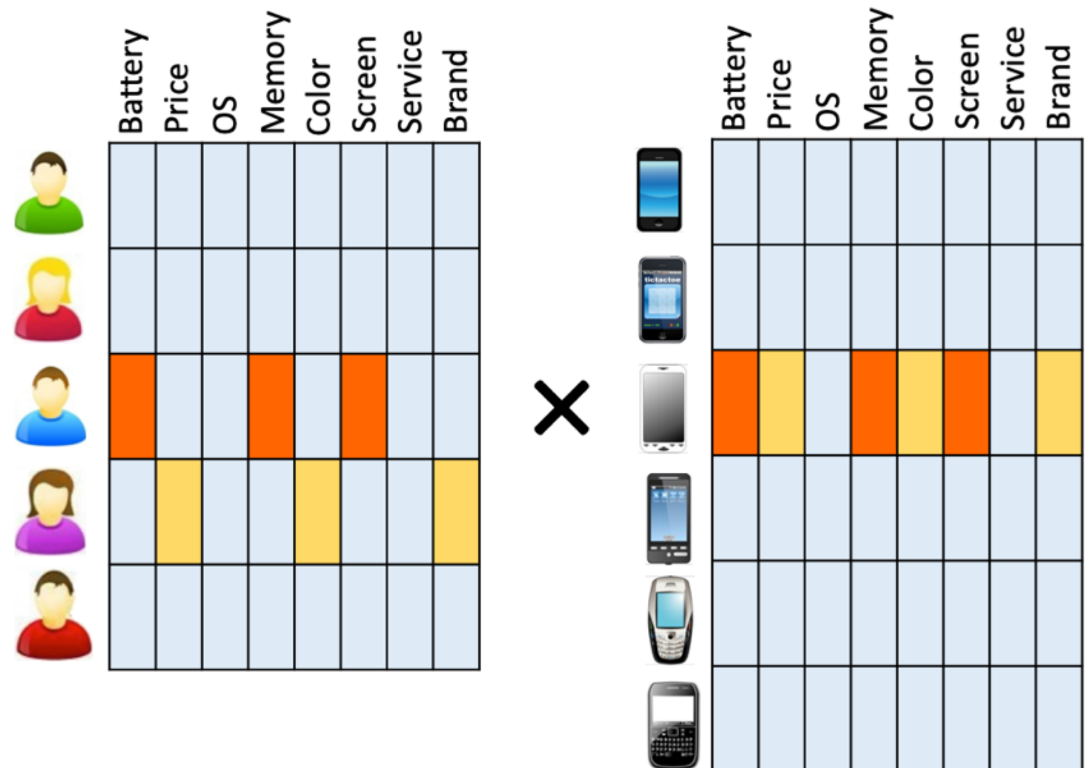


$$\underset{U_1,U_2,V,H_1,H_2}{\text{minimize}} \left\{ \|PQ^T - A\|_F^2 + \lambda_x \|U_1 V^T - X\|_F^2 + \lambda_y \|U_2 V^T - Y\|_F^2 + \lambda_u (\|U_1\|_F^2 + \|U_2\|_F^2) + \lambda_h (\|H_1\|_F^2 + \|H_2\|_F^2) + \lambda_v \|V\|_F^2 \right\}$$

$$P = [U_1 \ H_1], \ Q = [U_2 \ H_2]$$

Explicit Factor Model

- Generating recommendation list



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

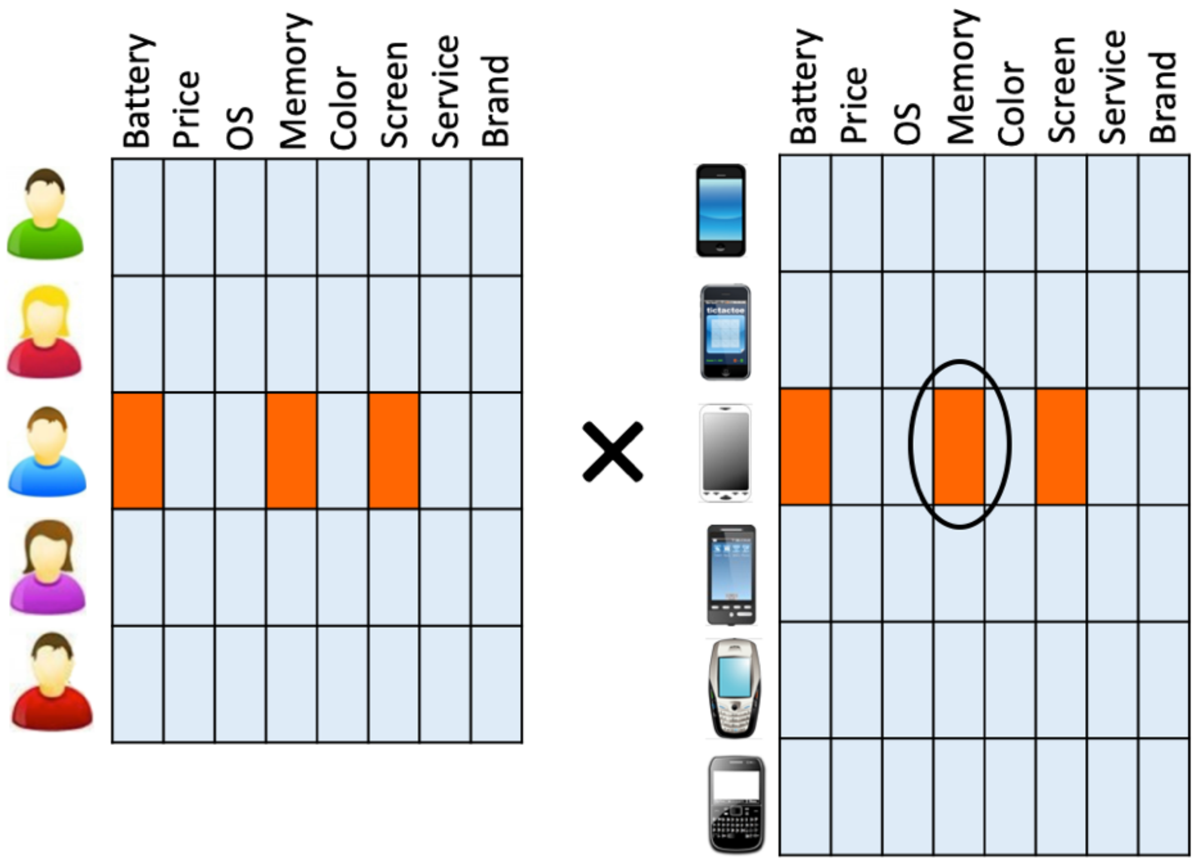
For each user i, rank the items with the ranking score:

$$R_{ij} = \alpha \cdot \boxed{\frac{\sum_{c \in \mathcal{C}_i} \tilde{X}_{ic} \tilde{Y}_{jc}}{kN}} + (1-\alpha)\tilde{A}_{ij}$$

Explicit Factor Model

Feature-level explanation for a recommended item

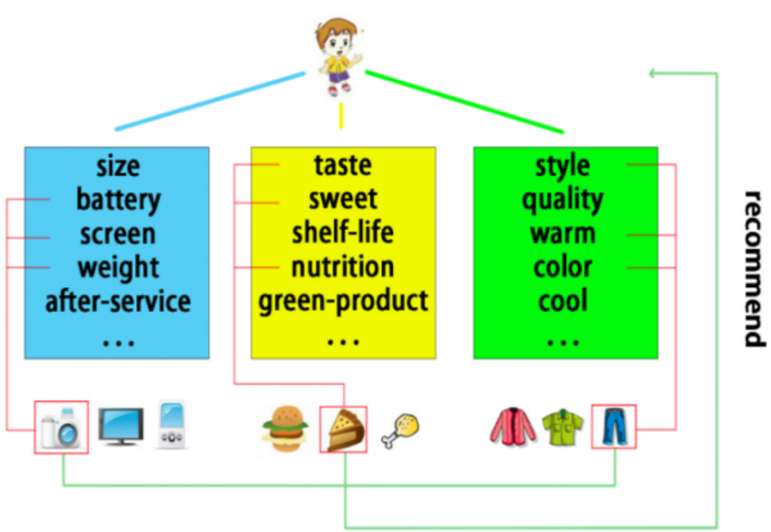
```
You might be interested in [feature],
on which this product performs well.
```



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



User-Item-Feature interaction

$$\hat{T}_{uif} = \sum_{k=0}^{K-1} R^U_{uk} \cdot R^{UF}_{fk} + \sum_{k=0}^{K-1} R^I_{ik} \cdot R^{IF}_{fk} + \sum_{k=0}^{K-1} R^U_{uk} \cdot R^I_{ik}$$

Pair-wise learning to rank over features

$$\hat{T}_{uif_Af_B} = \hat{T}_{uif_A} - \hat{T}_{uif_B}$$

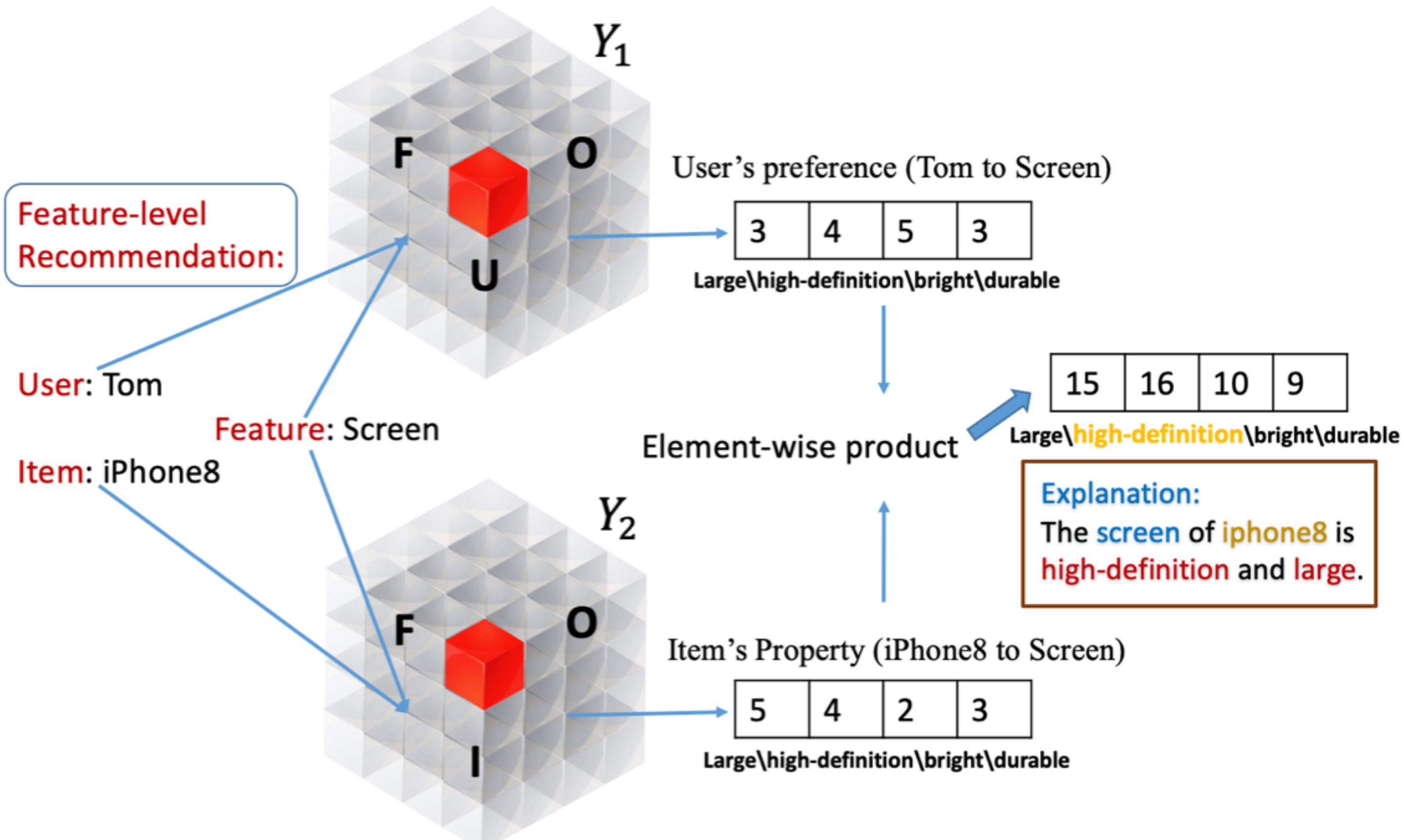
Tensor factorization

$$\min_{\Theta} \sum_{u\in U}\sum_{i\in I}(A_{ui} - (R^U_{u\cdot})^T \cdot R^I_{i\cdot})^2 - \lambda \sum_{u\in U}\sum_{i\in I}\sum_{f_A\in F^+_{ui}}\sum_{f_B\in F^-_{ui}} ln\sigma(\hat{T}_{uif_Af_B}) + \lambda_\Theta \|\Theta\|_F^2$$

T_{uif} directly give us feature-level explanation (selected feature is item-specific)

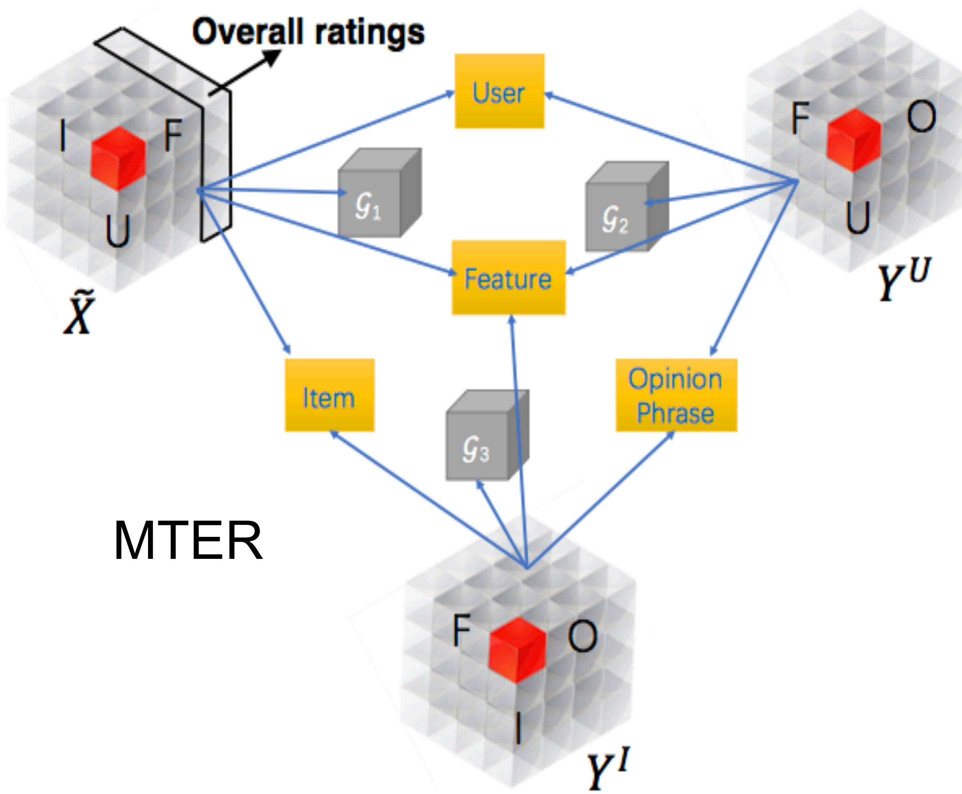
Multi-Task Learning for Explainable Recommendation

- 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]



MTER

- 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]

Dataset	Amazon (Cellphones & Accessories)				
Methods	Point-wise Learning Methods			Pair-wise Learning	
NDCG@K	Most Pop	NMF	EFM	BPRMF	MTER
10	0.0930	0.1879	0.1137	0.1182	**0.1362**
20	0.1278	0.0829	**0.1465**	0.1518	**0.1681**
50	0.1879	0.1614	**0.2062**	0.2070	**0.2268**

Dataset	Yelp				
Methods	Point-wise Learning Methods			Pair-wise Learning	
NDCG@K	Most Pop	NMF	EFM	BPRMF	MTER
10	0.1031	0.0581	**0.1056**	0.1244	**0.1384**
20	0.1359	0.0812	**0.1366**	0.1634	**0.1812**
50	0.1917	0.1366	0.1916	0.2213	**0.2369**

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

User Set	A		B		C	
Records	#Record	#Click	#Record	#Click	#Record	#Click
	15,933	691	11,483	370	17,265	552
CTR	4.34%		3.22%		3.20%	

Providing explanations improve the *persuasiveness* of system decisions.

Experimental Evaluation

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

Amazon Dataset		Q1	Q2	Q3	Q4	Q5
Mean Value	BPR	3.540	3.447	-	3.333	-
	EFM	3.367	3.360	3.173	3.240	3.227
	MTER	**3.767**	**3.660**	**3.707**	**3.727**	**3.620**
Paired t-test	MTER vs. BPR	0.0142	0.0273	-	0.0001	-
	MTER vs. EFM	0.0001	0.0027	0	0	0.0004
Yelp Dataset		*Q1*	*Q2*	*Q3*	*Q4*	*Q5*
Mean Value	BPR	3.400	3.387	-	3.180	-
	EFM	**3.540**	3.473	3.287	3.200	3.200
	MTER	3.500	**3.713**	**3.540**	**3.520**	**3.360**
Paired t-test	MTER vs. BPR	0.1774	0.0015	-	0.0013	-
	MTER vs. EFM	0.3450	0.0128	0.0108	0.0015	0.0775

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

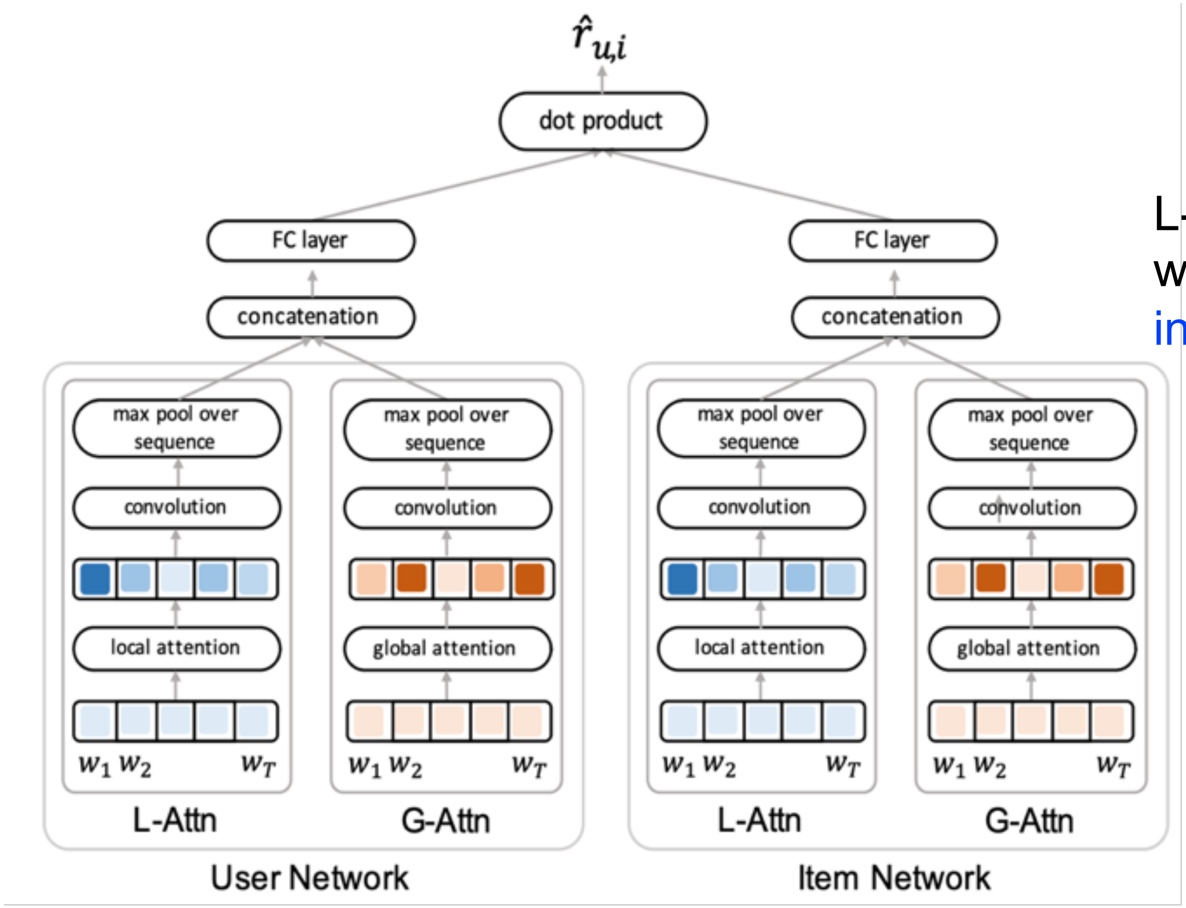
- [1] Balabanović, Marko, and Yoav Shoham. "Fab: content-based, collaborative recommendation." Communications of the ACM 40, no. 3 (1997): 66-72.
- [2] Pazzani, Michael J., and Daniel Billsus. "Content-based recommendation systems." In *The adaptive web*, pp. 325-341. Springer, Berlin, Heidelberg, 2007.
- [3] Resnick, Paul, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. "GroupLens: an open architecture for collaborative filtering of netnews." In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp. 175-186. ACM, 1994.
- [4] Sarwar, Badrul Munir, George Karypis, Joseph A. Konstan, and John Riedl. "Item-based collaborative filtering recommendation algorithms." *WWW* 1 (2001): 285-295.
- [5] Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." *Computer* (2009): 30-37.
- [6] Herlocker, Jonathan L., Joseph A. Konstan, and John Riedl. "Explaining collaborative filtering recommendations." In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pp. 241-250. ACM, 2000.
- [7] Zhang, Yongfeng, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis." In SIGIR, pp. 83-92. ACM, 2014.
- [8] Chen, Xu, Zheng Qin, Yongfeng Zhang, and Tao Xu. "Learning to rank features for recommendation over multiple categories." In SIGIR, pp. 305-314. ACM, 2016.
- [9] Wang, Nan, Hongning Wang, Yiling Jia, and Yue Yin. "Explainable recommendation via multi-task learning in opinionated text data." In SIGIR, pp. 165-174. ACM, 2018.

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]



L-Attn: Local attention, learns which words are more informative in a local window of words.

G-Attn: Global attention, learns which words are informative in the entire text.

User document: reviews of the user Item document: reviews of the item

Word-level Attentive Explanation

- Highlighted explanation words by local and global attention

Yelp (user), D-Attn model: local attention

They carry some rare things that you can't find anywhere else. The staff is pretty damn cool too best in Arizona. I prefer ma-and-pa. They treat you the best and they value your business extreme. They are good people great atmosphere and music. I definitely believe that Lux has the best coffee I've ever had at this point. Screw all my previous reviews. This place has coffee down, they make damn good toast too.

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.

Yelp (user), D-Attn model: global attention

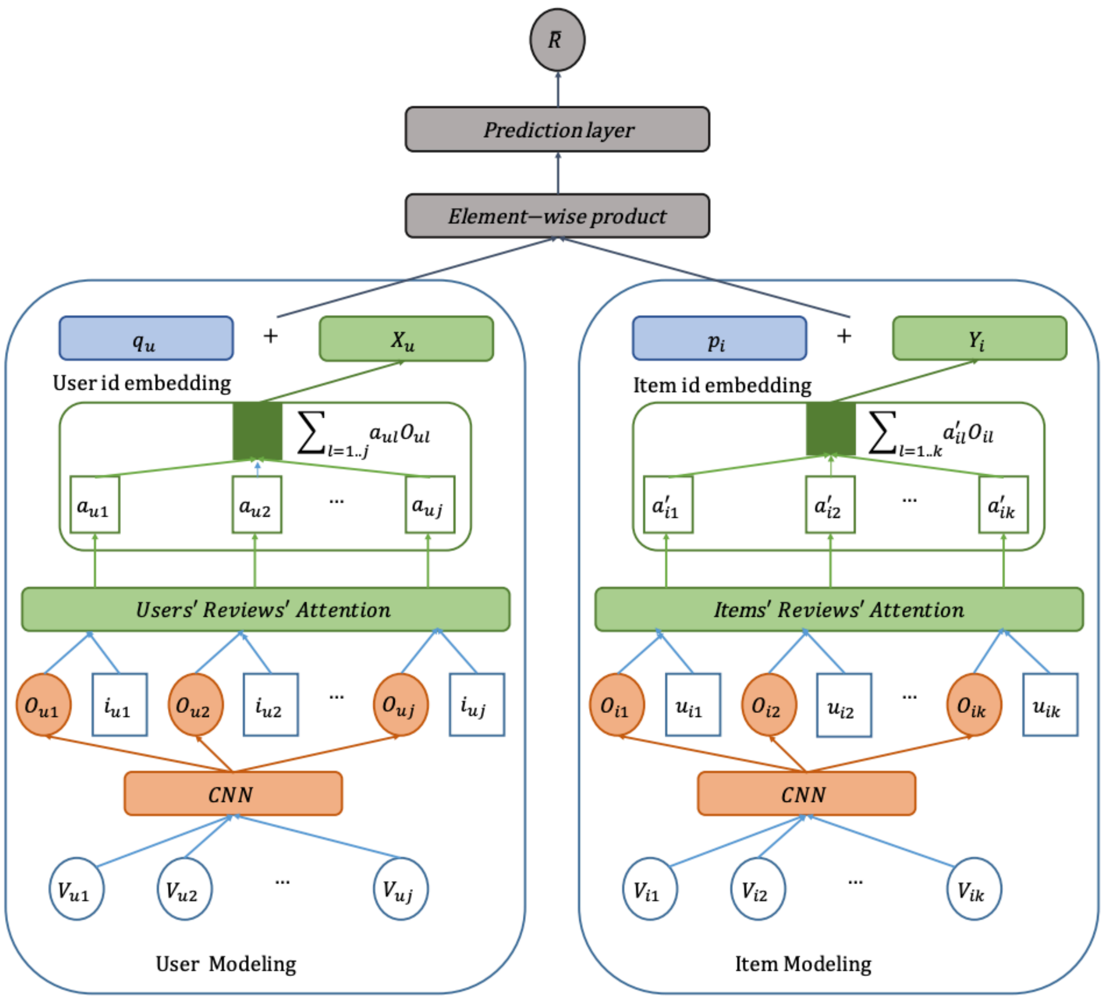
They carry some rare things that you can't find anywhere else. The staff is pretty damn cool too best in Arizona. I prefer ma-and-pa. They treat you the best and they value your business extreme. They are good people great atmosphere and music. I definitely believe that Lux has the best coffee I've ever had at this point. Screw all my previous reviews. This place has coffee down, they make damn good toast too.

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]



Reviews written by the user

Reviews of the item

$$a^*_{il} = h^T ReLU(W_O O_{il} + W_u u_{il} + b_1) + b_2$$

$$a_{il} = \frac{exp(a^*_{il})}{\sum_{l=0}^{k} exp(a^*_{il})} \qquad O_i = \sum_{l=1,\ldots k} a_{il} O_{il}$$

Attention mechanism learns the importance of each review

Review-Level Attentive Explanation

- Provide selected useful reviews as explanations

Item 1	a (a_{ij}=0.1932)	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.
	b (a_{ij}=0.0161)	I bought it for my daughter as a gift.
Item 2	a (a_{ij}=0.2143)	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.
	b (a_{ij}=0.0319)	I like reading in my spare time, and I think this book is very suitable for me.

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

RUTGERS | 清華大學 Tsinghua University | UMASS AMHERST

Review-Level Attentive Explanation

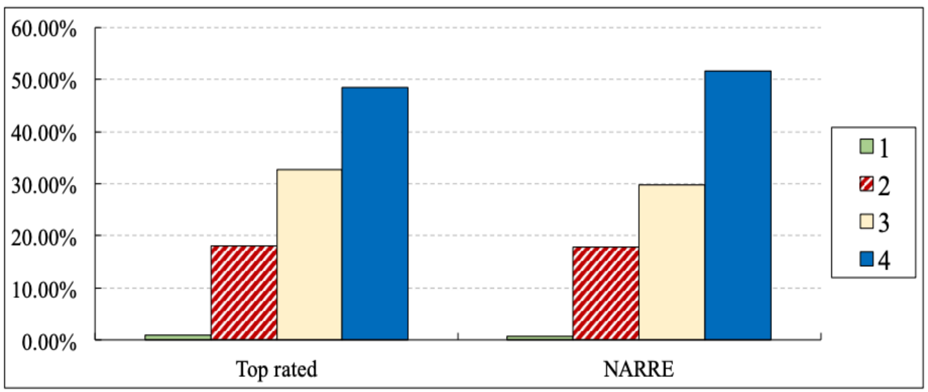
- Crowd-sourcing based Usefulness Evaluation of the Explanations

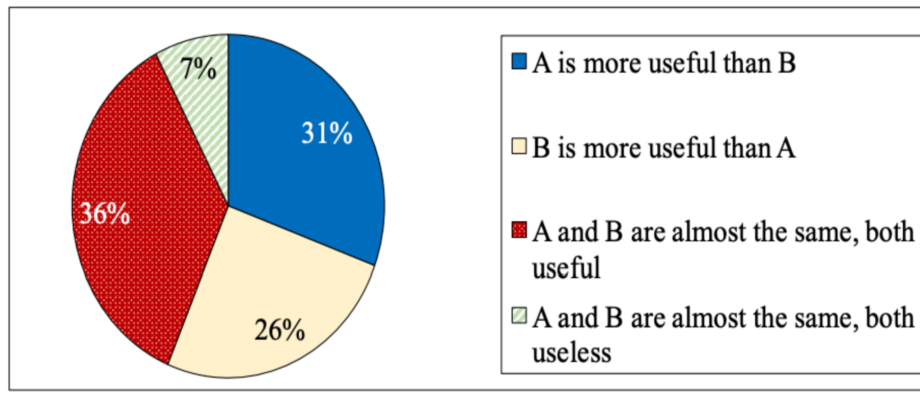
Annotation Instructions 1:
Background: You are going to buy an item, so you want to refer to the reviews written by previous consumers to know more about this item.
Task1: You need to browse each of the reviews below and then determine whether it is useful for your purchasing.
The review can be classified as follows:

- **1 star**: Not useful at all.
- **2 stars**: Somewhat useful.
- **3 stars**: Fairly useful.
- **4 stars**: Very useful.

Annotation Instructions 2:
Task2: You will see two groups of reviews, and each group contains 5 reviews. You need to browse each group and annotate pairwise usefulness between Group A and Group B.

- A is more useful than B.
- B is more useful than A.
- A and B are almost the same, both useful.
- A and B are almost the same, both useless.



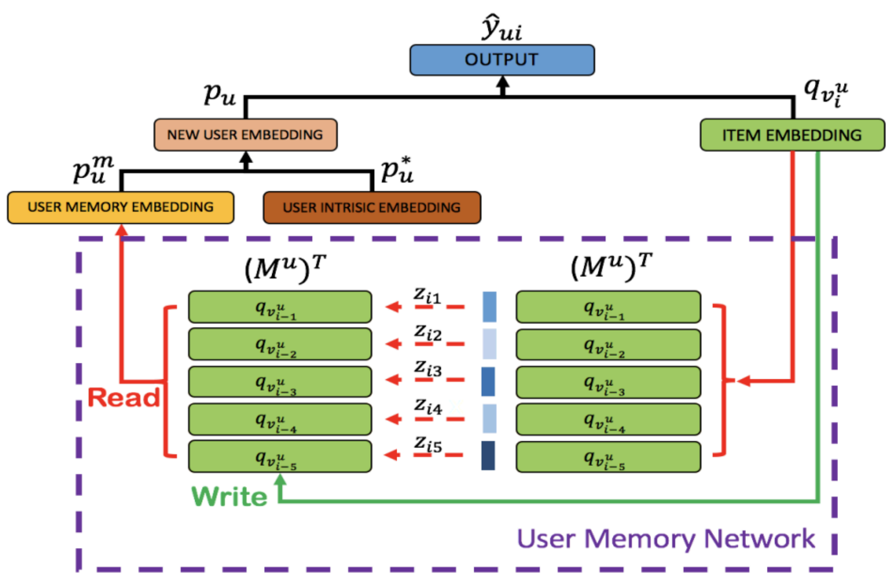


Most of the selected review explanations are rated “useful” by users.

Group A: top-5 algorithm selected reviews.
Gropu 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?



$$l_{RUM} = \log \prod_{(u,i)} (\hat{y}_{ui})^{y_{ui}} (1 - \hat{y}_{ui})^{1-y_{ui}} - \lambda ||\Theta||_F^2$$

$$= \sum_u \sum_{i \in I_u^+} \log \hat{y}_{ui} + \sum_u \sum_{i \in I / I_u^+} \log(1 - \hat{y}_{ui}) - \lambda ||\Theta||_F^2$$

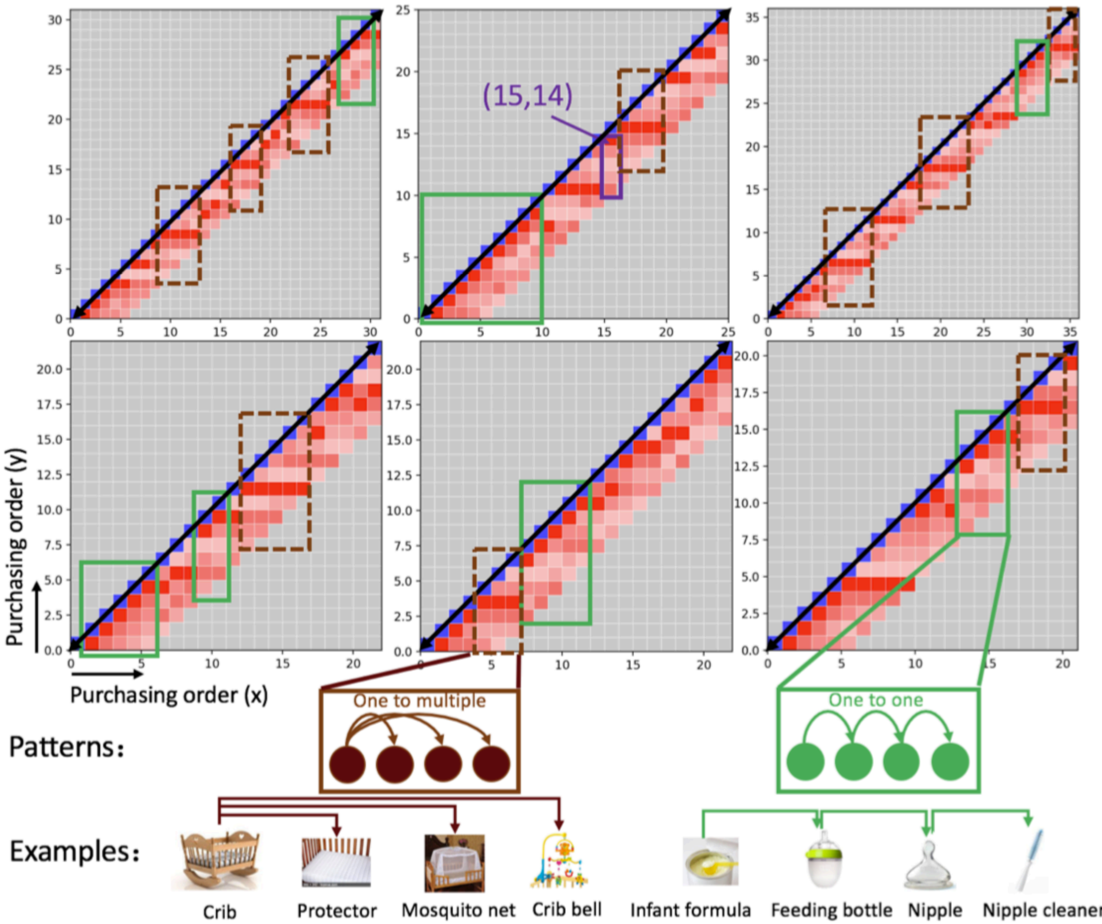
$$w_{ik} = (\boldsymbol{q}_{v_i^u})^T \cdot \boldsymbol{m}_k^u, \; z_{ik} = \frac{\exp(\beta w_{ik})}{\sum_j \exp(\beta w_{ij})}, \; \forall k = 1, 2, \cdots, K$$

$$\boldsymbol{p}_u^m = \sum_{k=1}^{K} z_{ik} \cdot \boldsymbol{m}_k^u$$

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

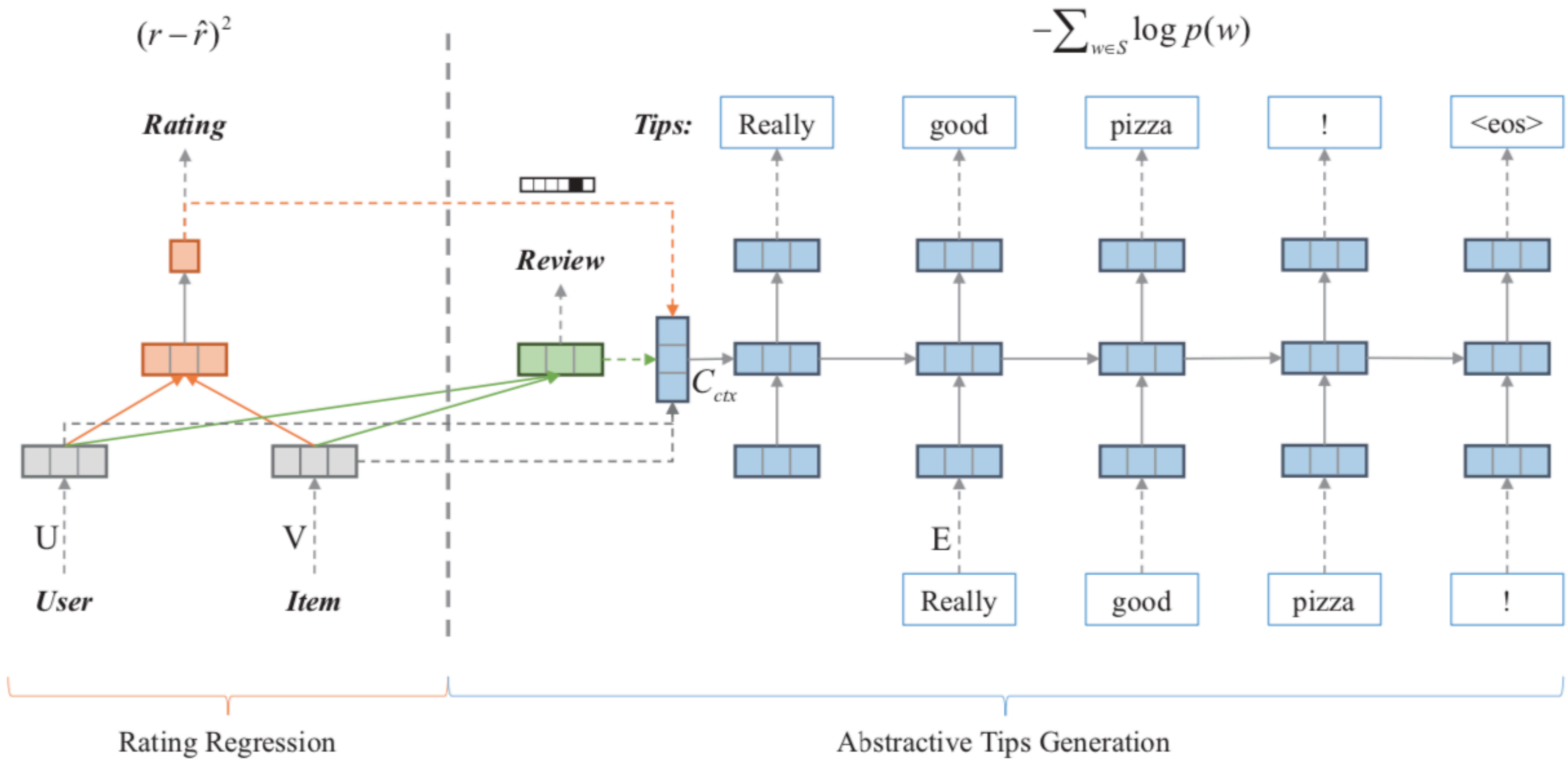


One-to-Multiple: an item consistently influence subsequent user behaviors.

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.

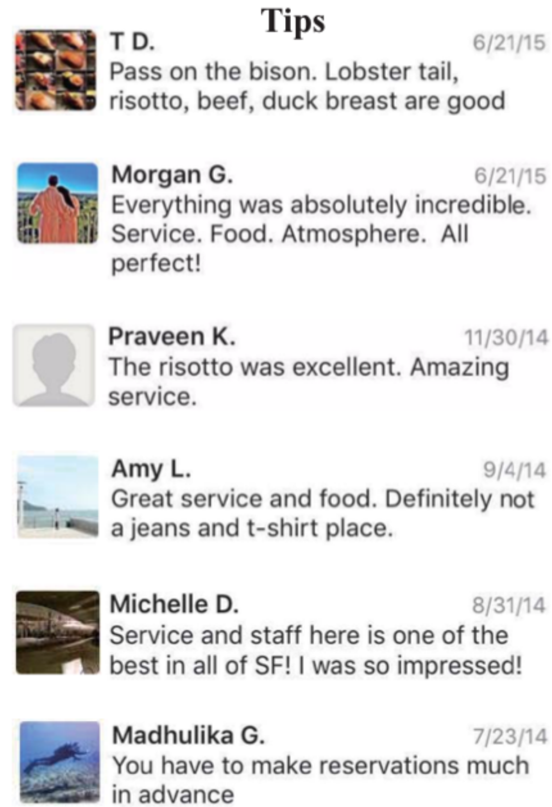
An LSTM generator to predict the ground-truth tips of the user item pair, personalized by the user-item embeddings and rating.

LSTM-based Textual Explanation Generation

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

Rating	Tips
4.64	***This is a great product for a great price.***
5	Great product at a great price.
4.87	***I purchased this as a replacement and it is a perfect fit and the sound is excellent.***
5	Amazing sound.
4.69	***I have been using these for a couple of months.***
4	Plenty of wire gets signals and power to my amp just fine quality wise.
4.87	***One of my favorite movies.***
5	This is a movie that is not to be missed.
4.07	***Why do people hate this film.***
4	Universal why didnt your company release this edition in 1999.
2.25	***Not as good as i expected.***
5	Jack of all trades master of none.
1.46	***What a waste of time and money.***
1	The coen brothers are two sick bastards.
4.34	***Not bad for the price.***
3	Ended up altering it to get rid of ripples.

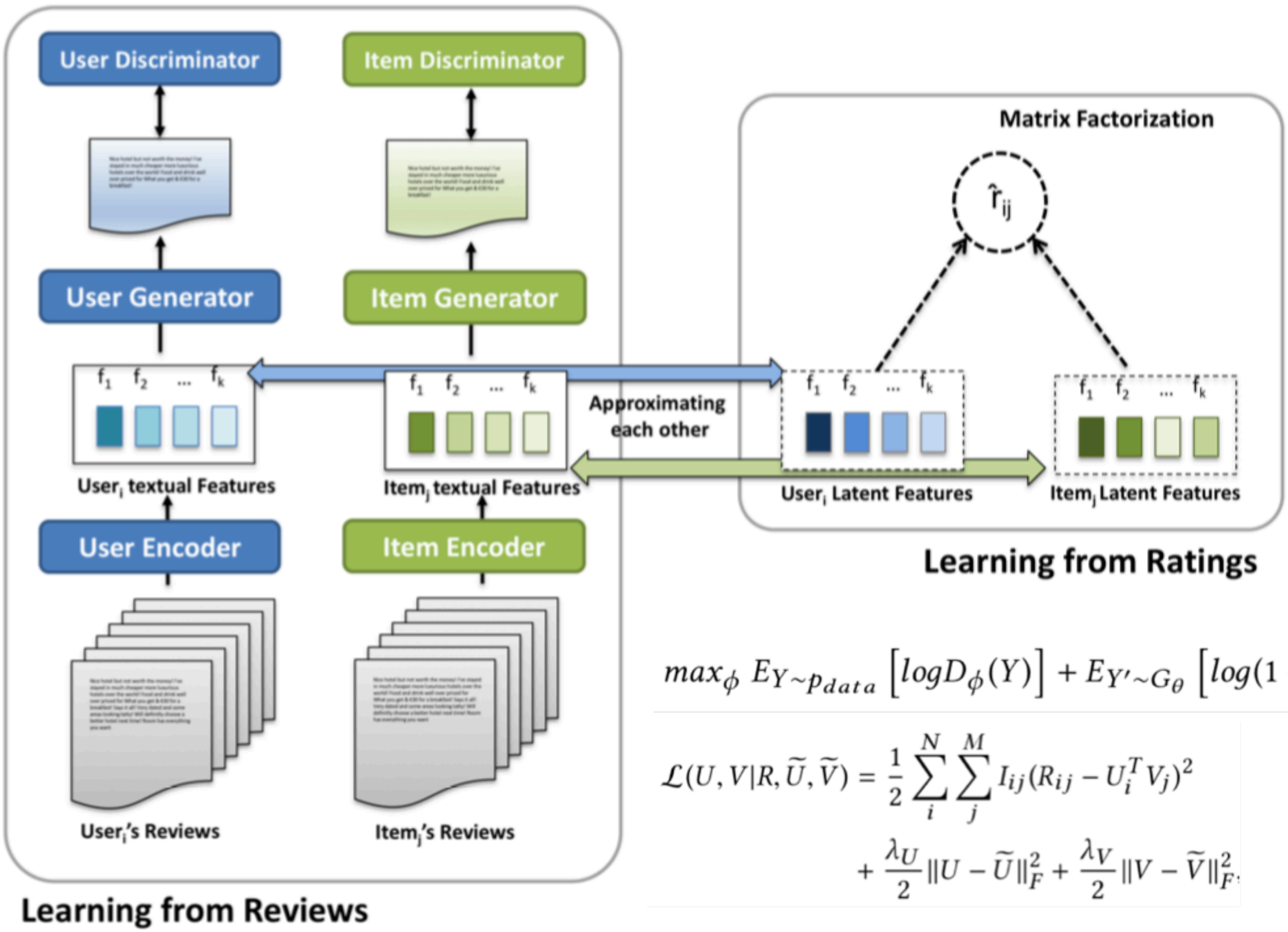
Bold line: Predicted ratings and generated tips.
Second line: ground truth tips.



Sampled tips on Yelp

Explanation Generation with GANs

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



$$max_\phi \ E_{Y \sim p_{data}} \left[log D_\phi(Y) \right] + E_{Y' \sim G_\theta} \left[log(1 - D_\phi(Y')) \right]$$

$$\mathcal{L}(U, V | R, \widetilde{U}, \widetilde{V}) = \frac{1}{2} \sum_i^N \sum_j^M I_{ij} (R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2} \|U - \widetilde{U}\|_F^2 + \frac{\lambda_V}{2} \|V - \widetilde{V}\|_F^2.$$

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.

Model	Y13	Y14	AE	AV	AG
N-gram	0.007	0.009	0.005	0.009	0.011
Skip-gram	0.009	0.014	0.007	0.011	0.015
LSTM	0.019	0.022	0.014	0.017	0.019
Opinosis	0.029	0.031	0.029	0.025	0.027
MT-U	0.048	0.043	0.042	0.044	0.047
MT-I	0.051	0.045	0.046	0.049	0.049
MT-P	**0.053**	**0.052**	**0.049**	**0.042**	**0.051**

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)

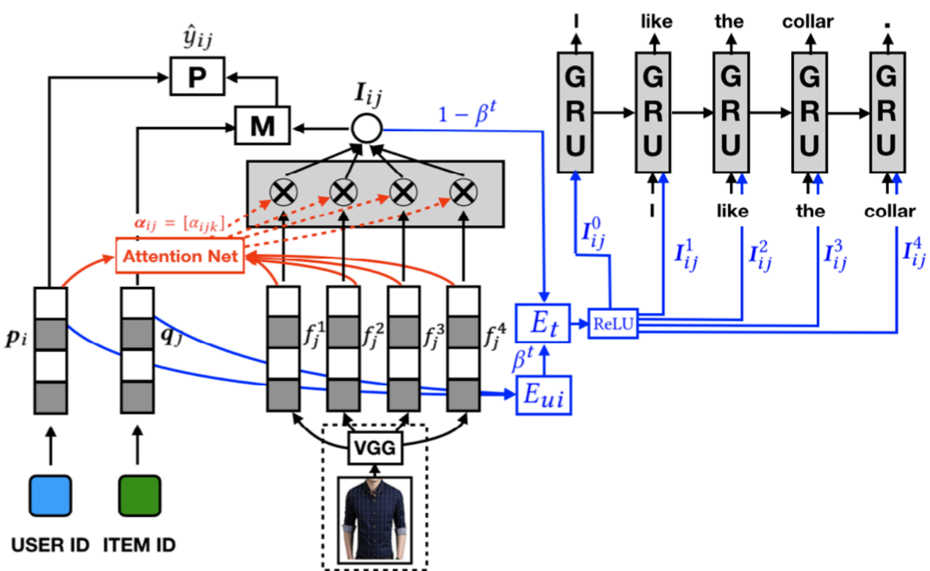
Y13: Yelp 2013 dataset
Y14: Yelp 2014 dataset
AE: Amazon Electronics
AV: Amazon Video Games
AG: Amazon Grocery

MT-U: user-level explanation
MT-I: item-level explanation
MT-P: user-item pair-level explanation

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.

$$a_{ijk} = E_2[\mathrm{ReLU}(E_1[(W_u p_i) \odot (W_f f_j^k)])]$$

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

$$I_{ij} = F_j \alpha_{ij} = \sum_{k=1}^{h} \alpha_{ijk} \cdot f_j^k$$

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]

Method		Random	VECF(-rev)	VECF
F_1	**M**=1	0.777	1.220	**2.273** (*86.3%* ↑)
	M=2	1.430	2.012	**3.180** (*58.1%* ↑)
	M=3	1.968	2.516	**4.513** (*79.4%* ↑)
	M=4	2.281	2.857	**4.514** (*58.0%* ↑)
	M=5	2.749	3.350	**4.774** (*42.5%* ↑)
NDCG	**M**=1	2.975	4.348	**7.551** (*73.7%* ↑)
	M=2	2.975	4.436	**6.666** (*50.3%* ↑)
	M=3	3.458	4.254	**7.089** (*66.6%* ↑)
	M=4	2.882	4.039	**6.320** (*56.5%* ↑)
	M=5	3.501	4.284	**6.455** (*50.7%* ↑)

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]

#	Target Item	Historical Records	Textual Review	Visual Explanation VECF	Visual Explanation Re-VECF
1			this is a large watch... nearly as large as my suunto but due to ***its articulated strap it fits on the wrist very well.***		
2			***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 & #34, but 200 lbs (and dropping :))		
3			***Great leggings. perfect for fly fishing or hunting or running.*** just perfect anytime you are cold!		
4			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***		
5			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.***		
6			***I like the front pocket~!*** Very cool!		

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

- [1] Seo, Sungyong, Jing Huang, Hao Yang, and Yan Liu. "Interpretable convolutional neural networks with dual local and global attention for review rating prediction." In RecSys, 2017.
- [2] Chen, Chong, Min Zhang, Yiqun Liu, and Shaoping Ma. "Neural attentional rating regression with review-level explanations." In WWW, pp. 1583-1592. 2018.
- [3] Chen, Xu, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. "Sequential recommendation with user memory networks." In WSDM, pp. 108-116, 2018.
- [4] Li, Piji, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. "Neural rating regression with abstractive tips generation for recommendation." In SIGIR, pp. 345-354. ACM, 2017.
- [5] Lu, Yichao, Ruihai Dong, and Barry Smyth. "Why I like it: multi-task learning for recommendation and explanation." In RecSys, pp. 4-12. ACM, 2018.
- [6] Kang, Wang-Cheng, Chen Fang, Zhaowen Wang, and Julian McAuley. "Visually-aware fashion recommendation and design with generative image models." In ICDM, pp. 207-216. IEEE, 2017.
- [7] Chen, Xu, Yongfeng Zhang, Zheng Qin. “Dynamic Explainable Recommendation based on Neural Attentive Models.” In AAAI, 2019.
- [8] Gao, Jingyue, Xiting Wang, Yasha Wang, Xing Xie. “Explainable Recommendation Through Attentive Multi-View Learning.” In AAAI, 2019.

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

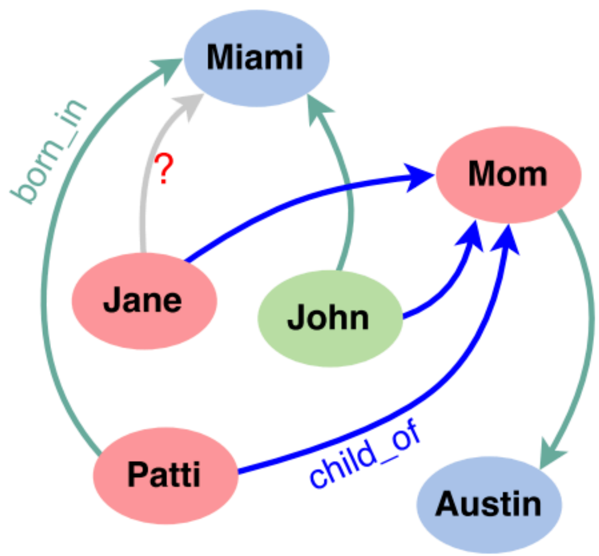
Explainable Recommendation based on KGs

- 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. AIg'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

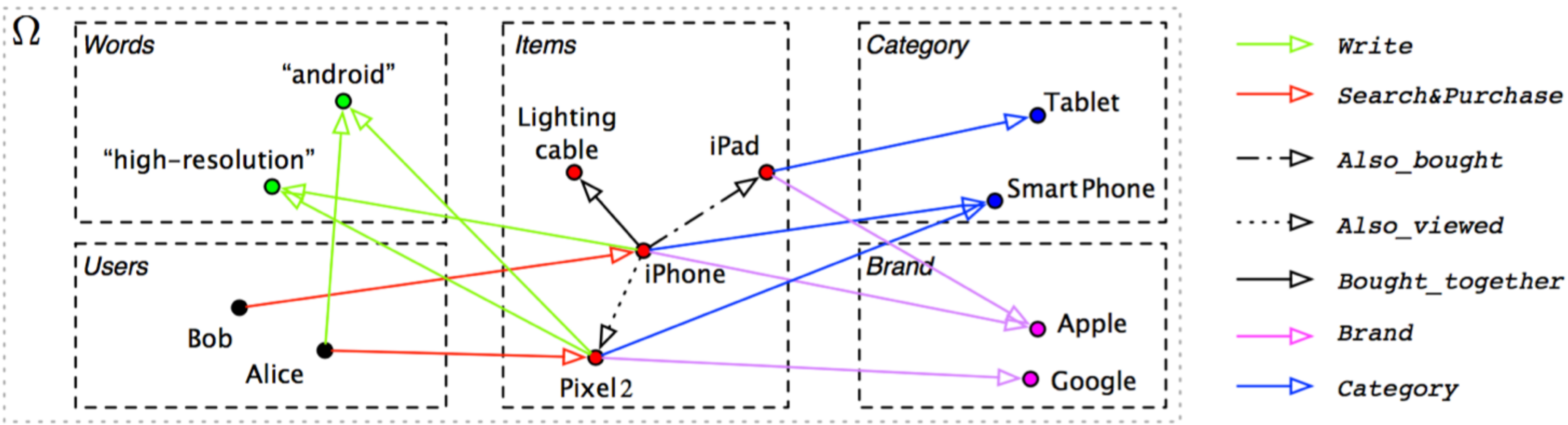
$$\boldsymbol{h} + \boldsymbol{\ell} \approx \boldsymbol{t} \qquad d(\boldsymbol{h} + \boldsymbol{\ell}, \boldsymbol{t})$$

Minimize the hinge-loss to learn entity and relation embeddings

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

Translational KG Embedding for Recommendation

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



$$\boldsymbol{e_t} = trans(e_h, r) = \boldsymbol{e_h} + \boldsymbol{r}$$

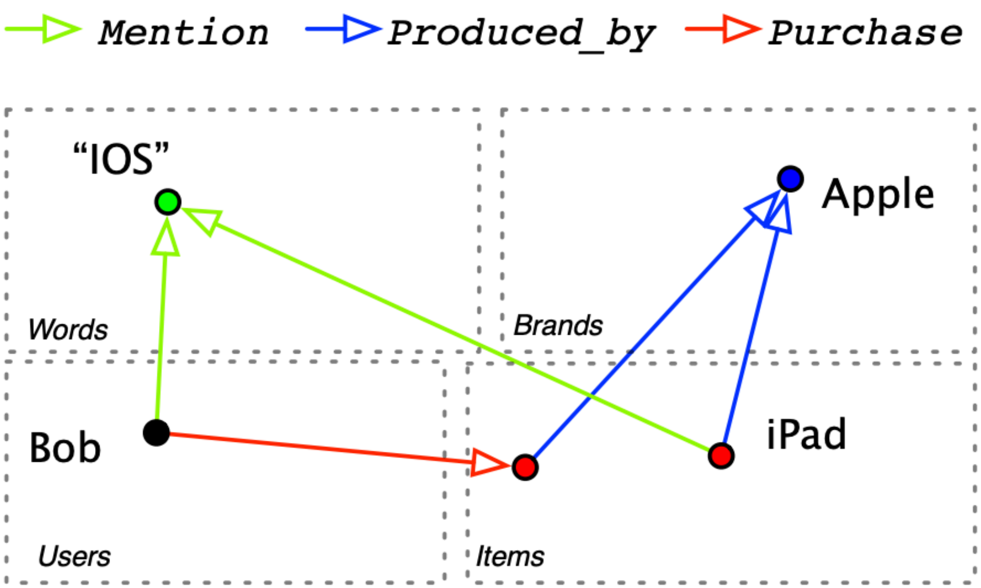
$$P(e_t|trans(e_h,r)) = \frac{\exp(\boldsymbol{e_t} \cdot trans(e_h,r))}{\sum_{e'_t \in E_t} \exp(\boldsymbol{e'_t} \cdot trans(e_h,r))}$$

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

Recommendation:
Calculate $e_{user}+r_{purchase}$
Find top-K nearest item entity

Translational KG Embedding for Recommendation

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



Post-hoc explanation by finding a path between user and the (already) recommended item.

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:

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

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

Rank explanation paths based on connectivity.

Propagating User Preferences on KG

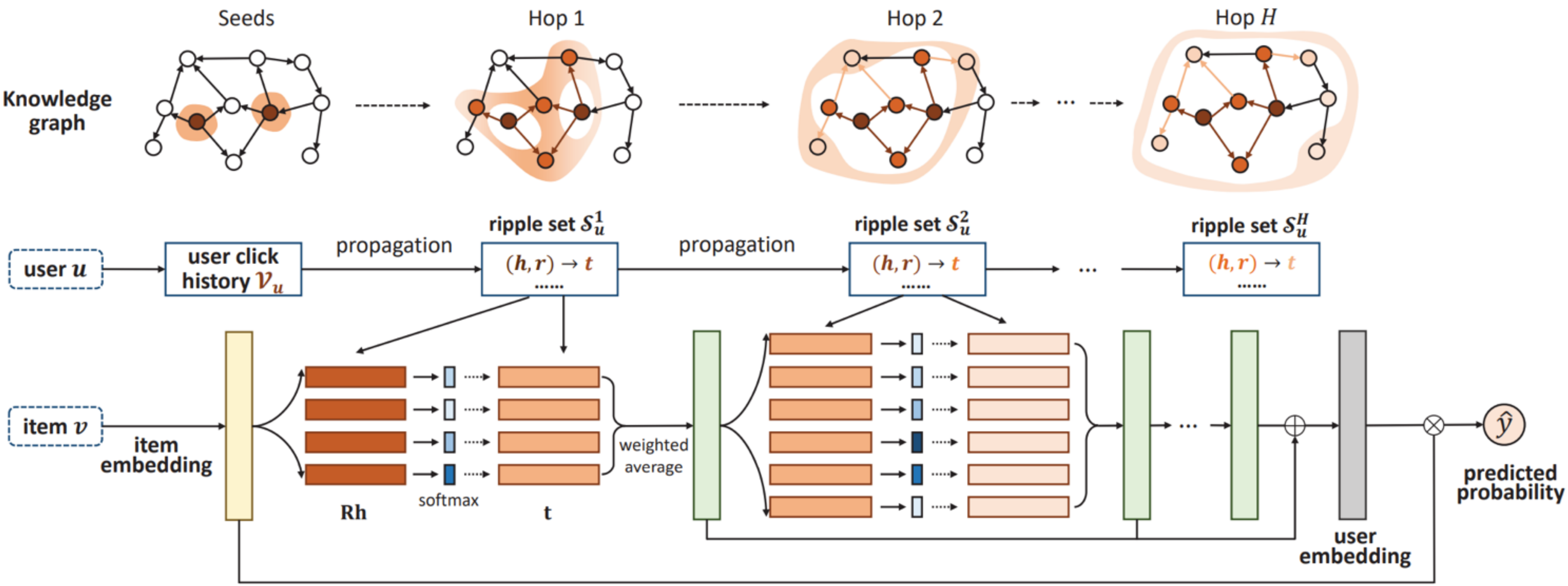
- RippleNet: Propagating user preferences on the knowledge graph for recommender systems. [Wang et al. CIKM'2018]

Seeds: user interacted entities

1-hop related entities

2-hop related entities

H-hop related entities

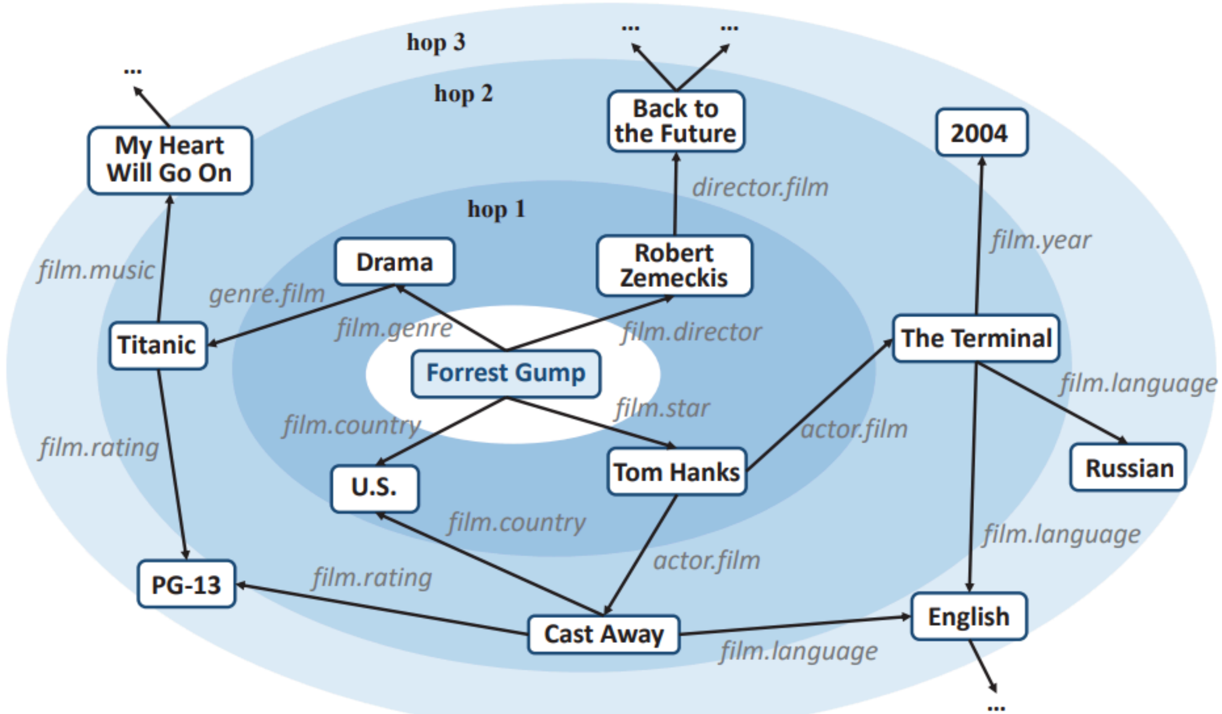


Attentively select entities based on memory network

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]

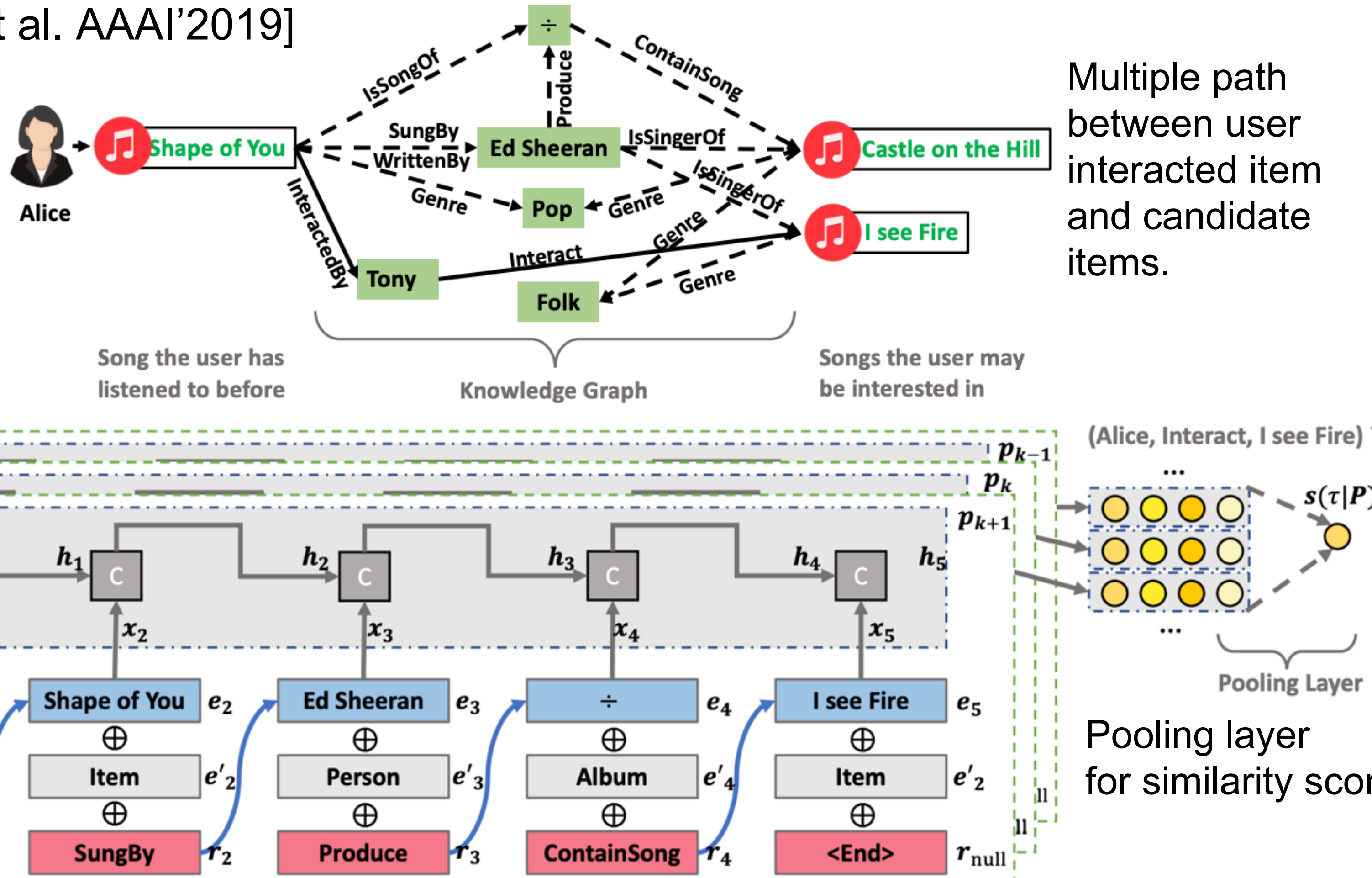


Explanation path constructed by selecting the most significant entity in each hop.

$$\text{'}user \xrightarrow{watched} Forrest\ Gump \xrightarrow{directed\ by} Robert\ Zemeckis \xrightarrow{directs} Back\ to\ the\ Future\text{'}$$

Learning Path Embedding for Recommendation

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



Multiple path between user interacted item and candidate items.

Pooling layer for similarity score

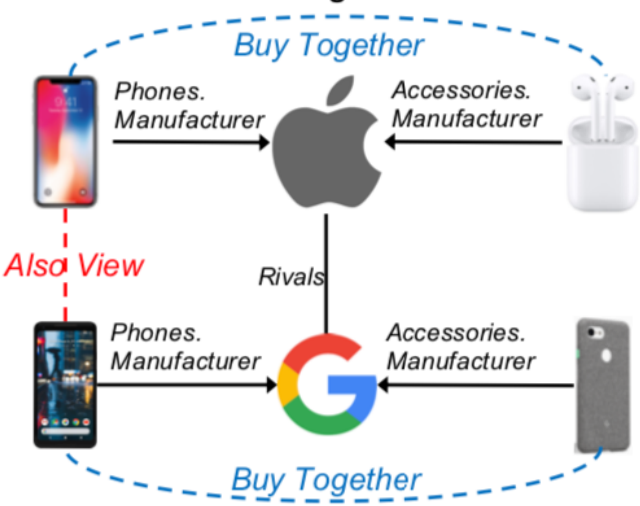
Each path represented as a path embedding using LSTM (input: entity embedding + entity type embedding + relation embedding)

Jointly Learning Explainable Rules for Recommendation

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

Rule Learning Module

Learning Rules



Black edges: Relations
Colored edges: Item associations

Rule Selection

Rules	w
(phones.manufacturer, accessories.manufacturer^{-1}) → Buy Together	0.12
(phones.manufacturer, rivals, phones.manufacturer^{-1}) → Also View	0.21
(phones.manufacturer, rivals, laptaops.manufacturer^{-1}) → Also View	0.03
(phones.manufacturer, accessories.earsets..manufacturer^{-1}) → Buy Also	0.14
...	...

w: Importance of a rule

learned by feature selection

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

$x_{(a,b)}(i)=P(b|a, R_i)$: 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)

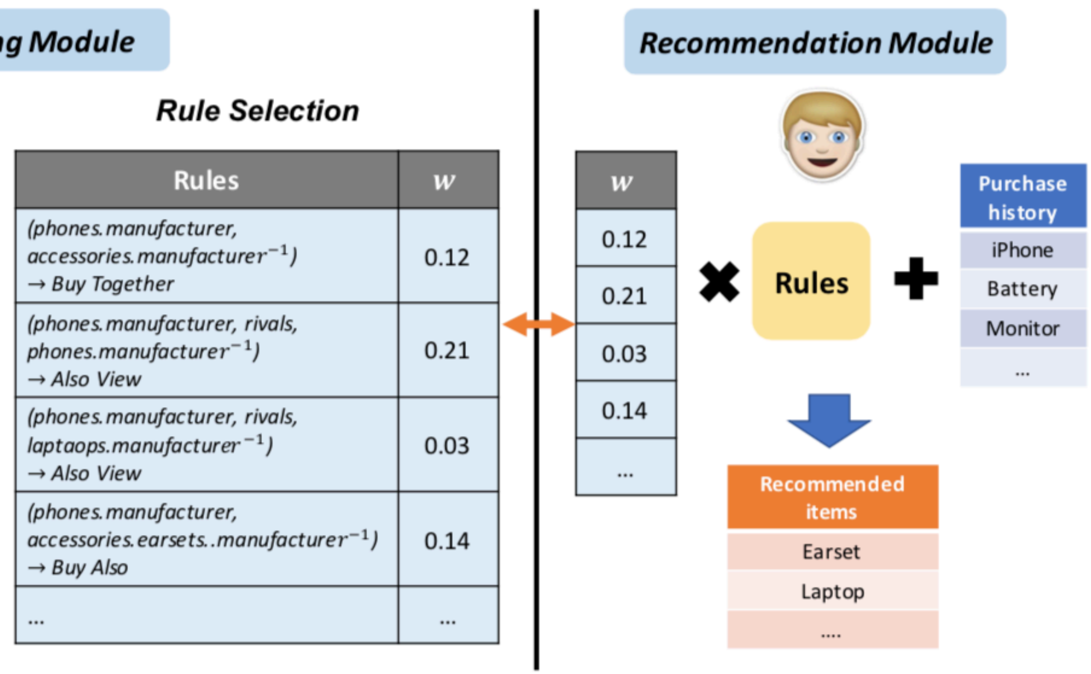
Rule: a sequence of relation types: e.g., r_1-r_2-r_1-r_3

Connection strength between items a & b through rule R

$$P(b|a,R)=\sum_{e\in N(a,R')}P(e|a,R')\cdot P(b|e,r_k).$$

Jointly Learning Explainable Rules for Recommendation

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



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.

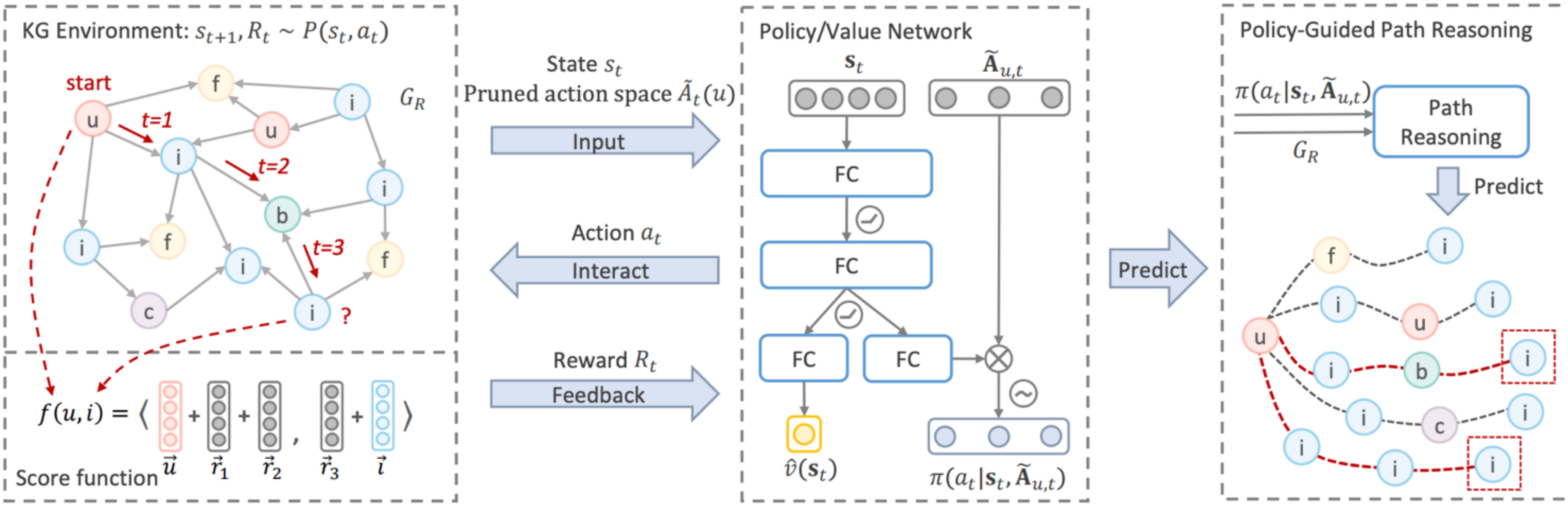
Rule: a sequence of relation types: e.g., r_1-r_2-r_1-r_3

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

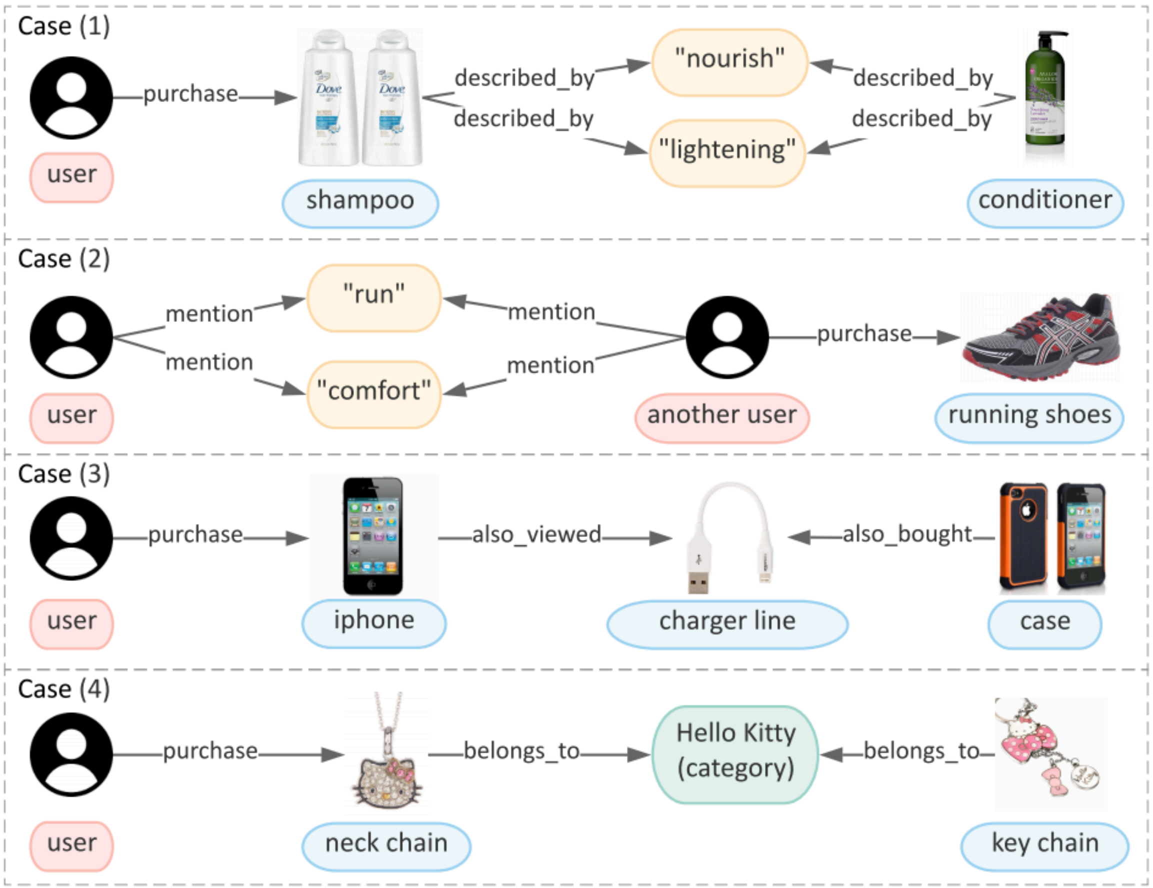
- Reinforcement Knowledge Graph Reasoning for Explainable Recommendation [Xian et al. SIGIR'2019]
- Paradigm of previous methods: for each user, for each candidate item, calculate ranking score based on path info between this user-item pair.
- Too many candidate items: Can we avoid enumerating all candidate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

- [1] Ai, Qingyao, Vahid Azizi, Xu Chen, and Yongfeng Zhang. "Learning heterogeneous knowledge base embeddings for explainable recommendation." *Algorithms* 11, no. 9 (2018): 137.
- [2] Wang, Hongwei, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. "RippleNet: Propagating user preferences on the knowledge graph for recommender systems." In CIKM, pp. 417-426. ACM, 2018.
- [3] Wang, Xiang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. "Explainable Reasoning over Knowledge Graphs for Recommendation." In AAAI, 2019.
- [4] Cao, Yixin, Xiang Wang, Xiangnan He, and Tat-Seng Chua. "Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preferences." In WWW, 2019.
- [5] Ma, Weizhi, Min Zhang, Yue Cao, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren. "Jointly Learning Explainable Rules for Recommendation with Knowledge Graph." In WWW, 2019.
- [6] Yikun Xian, Zuohui Fu, Shan Muthukrishnan, Gerard de Melo and Yongfeng Zhang. “Reinforcement Knowledge Graph Reasoning for Explainable Recommendation.” In SIGIR, 2019.

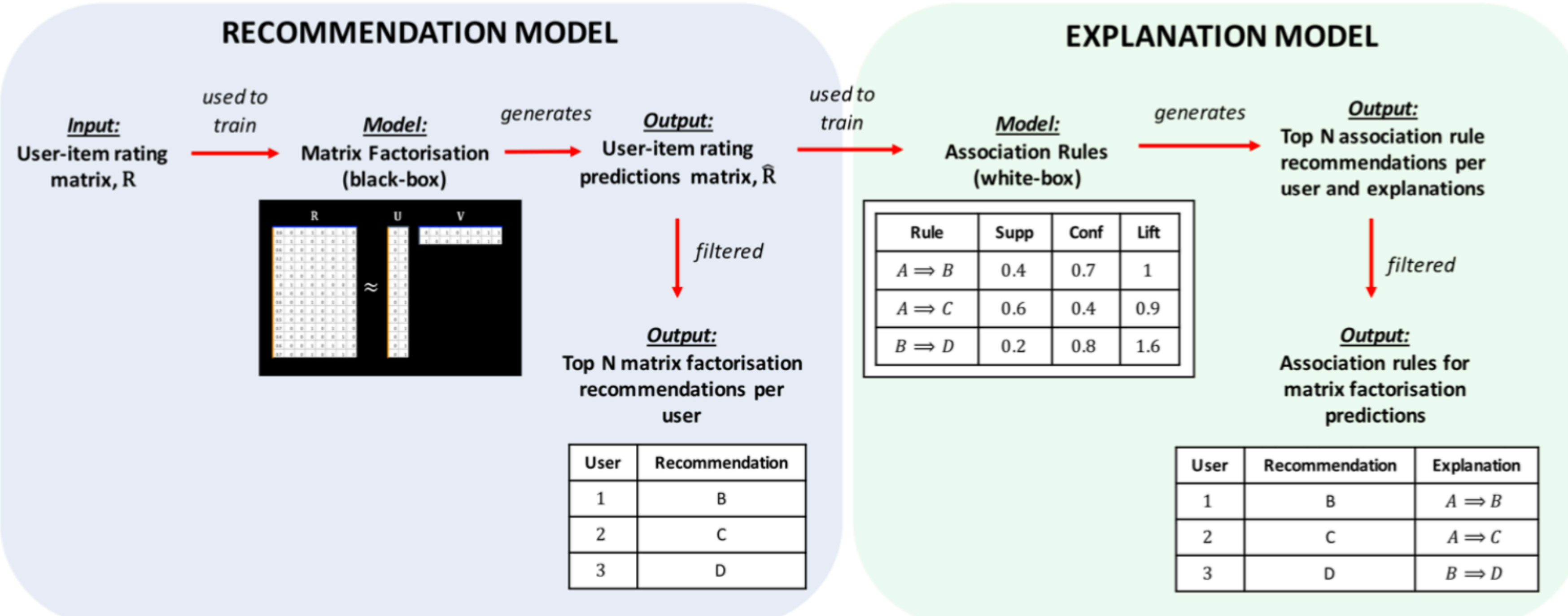
Post-hoc and Model-Agnostic Explanation

- Provide explanation for a (possibly unexplainable) model

- Mining-based Approach
  - Explanation Mining: Post Hoc Interpretability of Latent Factor Models for Recommendation Systems [Peake et al. KDD'2018]
- Learning-based Approach
  - A Reinforcement Learning Framework for Explainable Recommendation [Wang et al. ICDM'2018]

Post-hoc Explanation based on Association Rule Mining

- Explanation Mining: Post Hoc Interpretability of Latent Factor Models for Recommendation Systems [Peake et al. KDD’2018]



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

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

“Unexplainable Items”

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)

RUTGERS 清華大學 Tsinghua University UMASS AMHERST

Post-hoc Explanation based on Association Rule Mining

- Evaluate explainability based on Model Fidelity

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

Rules	K	Interestingness	Model Fidelity
Global	N	Support	0.522369
		Confidence	0.568602
		Lift	0.423591
Local	10	Support	0.828272
		Confidence	$\boxed{0.842889}$
		Lift	0.412679
	50	Support	0.791095
		Confidence	0.817715
		Lift	0.452805
	100	Support	0.770759
		Confidence	0.799536
		Lift	0.44886

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.

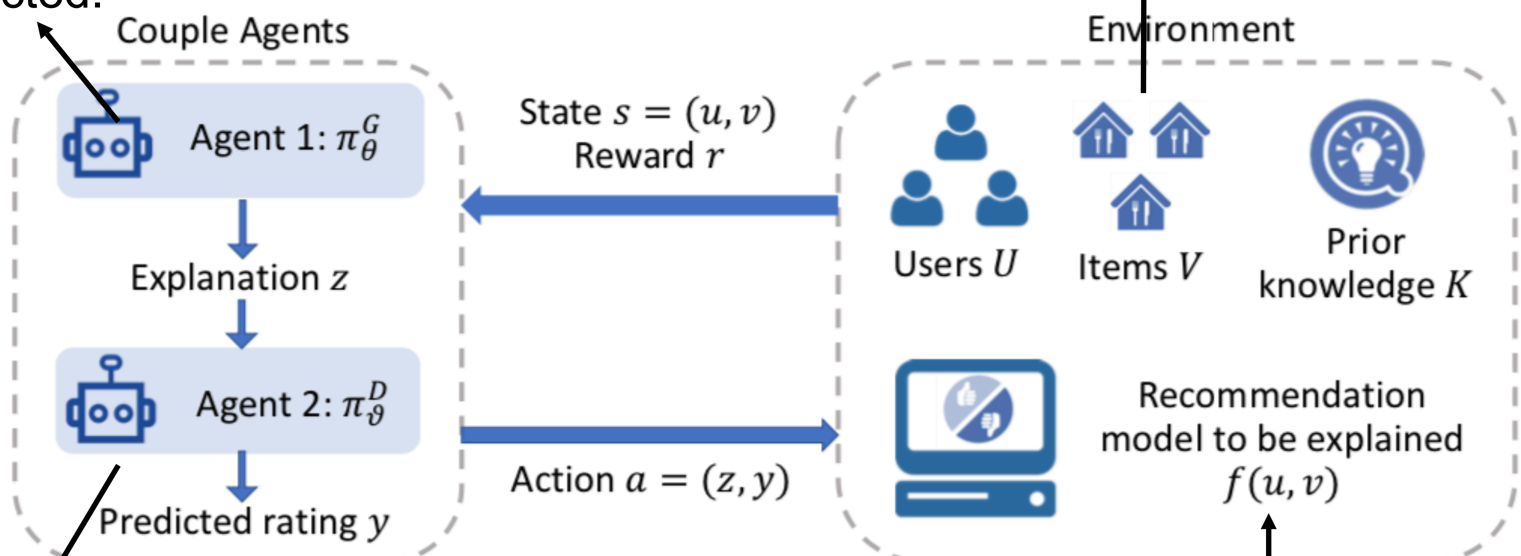
With appropriate nearest neighbor and interestingness selection, 80%+ of the recommendations can be post-hoc explained.

Model-Agnostic Explanation based on RL

- A Reinforcement Learning Framework for Explainable Recommendation [Wang et al. IDCM’2018]

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



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

Reward:

$$r = \mathcal{L}(f(\boldsymbol{u}, \boldsymbol{v}), y) + \Omega(\boldsymbol{z})$$

A recommendation model to be explained

Model-Agnostic Explanation based on RL

- Evaluate explanation based on consistency M_c and explainability M_e

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

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

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

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

	M_c					M_e				
	NMF	PMF	SVD++	CDL	GT	NMF	PMF	SVD++	CDL	GT
Random	-0.030	-0.030	-0.031	0.012	0.007	-0.478	-0.287	-0.266	-0.517	-1.488
NARRE	-0.015	-0.000	0.018	0.031	0.038	-0.448	-0.266	-0.239	-0.482	-1.424
Ours	**0.018**	**0.037**	**0.041**	**0.227**	**0.168**	**-0.421**	**-0.258**	**-0.232**	**-0.460**	**-1.380**

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

RUTGERS 清華大學 Tsinghua University UMASS AMHERST

References

- [1] Peake, Georgina, and Jun Wang. "Explanation Mining: Post Hoc Interpretability of Latent Factor Models for Recommendation Systems." In KDD, pp. 2060-2069. ACM, 2018.
- [2] Wang, Xiting, Yiru Chen, Jie Yang, Le Wu, Zhengtao Wu, and Xing Xie. "A Reinforcement Learning Framework for Explainable Recommendation." In *ICDM*, pp. 587-596. IEEE, 2018.

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

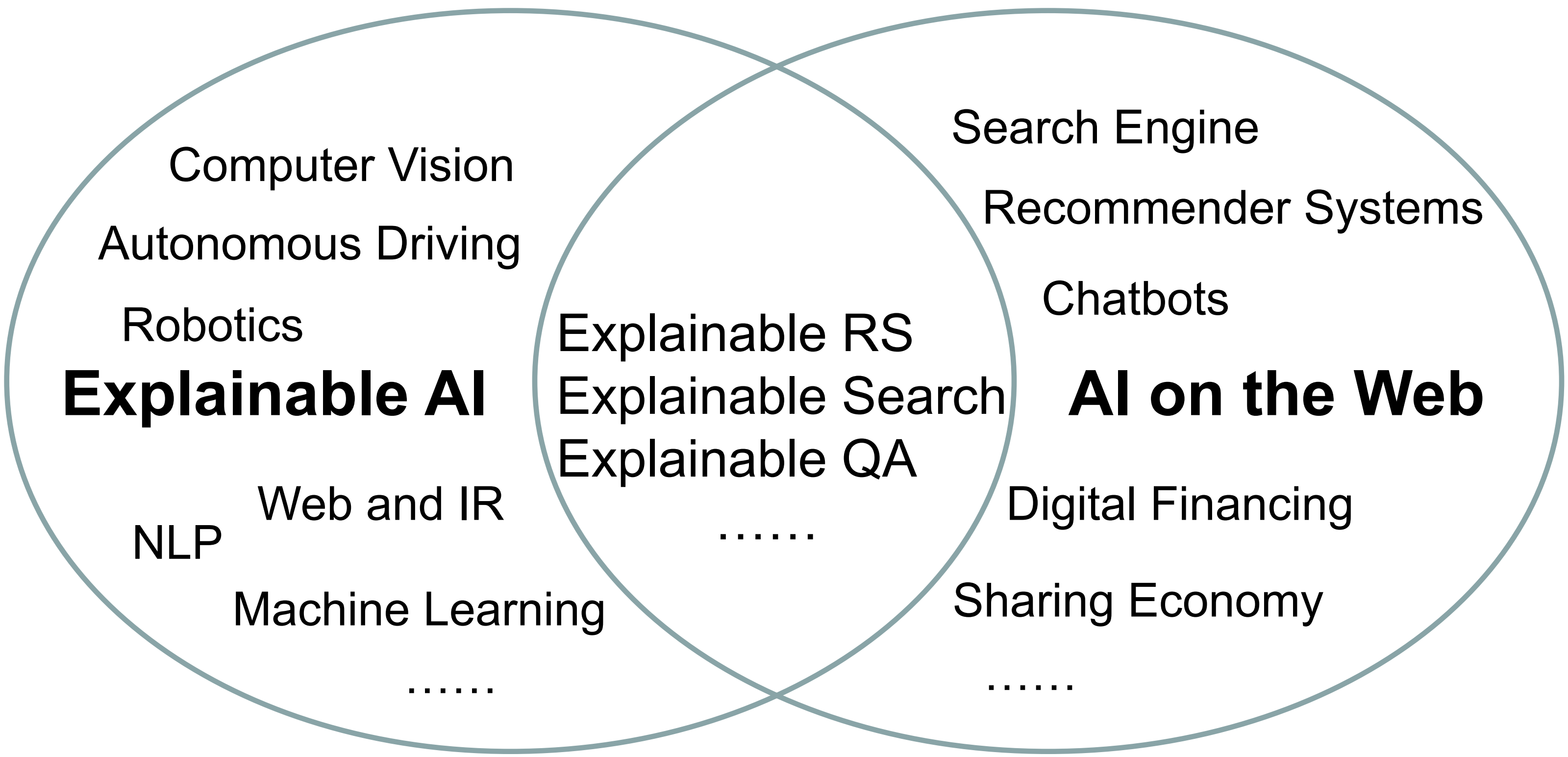
- Background and motivation
  - What is explainable search?
  - Why do we need explainable search?
- Existing work on explainable search
  - How can we make search models more explainable?
    - Building Interpretable search models
    - Using structured knowledge
    - Post-hoc explanation methods for 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

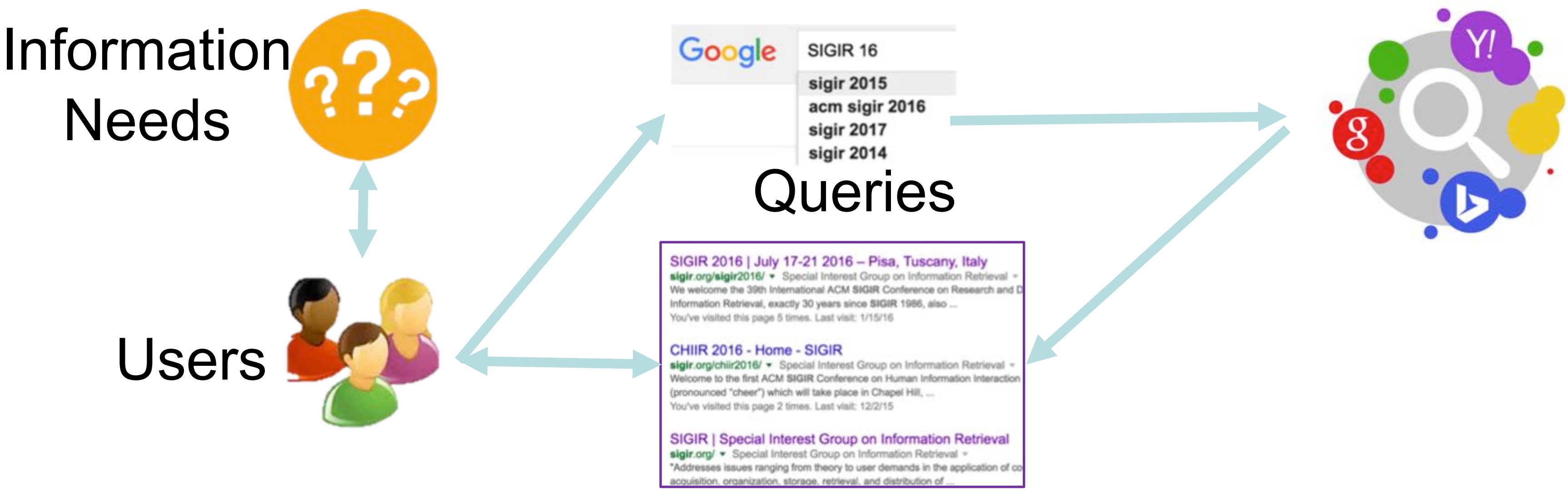
- What is explainable search?
  - Search: one of the most important AI application on the Web
  - In a narrow sense:
    - How to build an interpretable search model
  - In a broad sense:
    - Re-examine the search problem from the explainable AI/ML perspective

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

Explainable artificial intelligence - Wikipedia
https://en.wikipedia.org/wiki/Explainable_artificial_intelligence ▾
Explainable AI (**XAI**), Interpretable AI, or Transparent AI refer to techniques in artificial intelligence (AI) which can be trusted and easily understood by humans. It contrasts with the concept of the "black box" in machine learning where even their designers cannot explain why the AI arrived at a specific decision.
Goals · History and methods · Regulation
You visited this page on 5/5/19.

Explainable Artificial Intelligence - Darpa
https://www.darpa.mil › Program Information ▾
Figure 1. The Need for Explainable AI. Dramatic success in machine learning has led to a torrent of Artificial Intelligence (AI) applications. Continued advances ...

Explainable Artificial Intelligence - KDnuggets
https://www.kdnuggets.com/2019/01/explainable-ai.html ▾
We outline the necessity of explainable AI, discuss some of the methods in academia, take a look at explainability vs accuracy, investigate use cases, and more.

Should AI explain itself? or should we design Explainable AI so that it ...
https://towardsdatascience.com/should-ai-explain-itself-or-should-we-design-explaina... ▾
Mar 4, 2019 - Explainable AI (XAI)is NOT an AI that can explain itself, it is a design decision by developers. It is AI that is transparent enough so that the ...

An Explainable AI (XAI) or Transparent AI is an **artificial intelligence (AI)** whose actions can be easily understood by humans.

Explainable Artificial Intelligence - Wikipedia
en.wikipedia.org/wiki/Explainable_Artificial_Intelligence

Is this answer helpful?

Explainable AI: Making machines understandable for humans ...
https://**explainableai**.com ▾
There is no denying the fact that artificial intelligence is the future. From the security forces to the military applications, **AI** has spread out its wings to encompass our daily lives as well. However, AI comes with its own limitations.

Explainable Artificial Intelligence - DARPA
https://www.darpa.mil/program/**explainable-artificial-intelligence** ▾
The Need for **Explainable AI** Dramatic success in machine learning has led to a torrent of Artificial Intelligence (AI) applications. Continued advances promise to produce autonomous systems that will perceive, learn, decide, and act on their own.

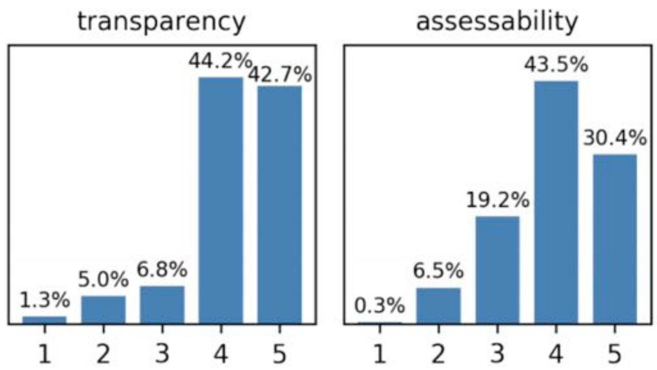
  - Query-centric, with keywords highlighted
  - Explain why a webpage is retrieved

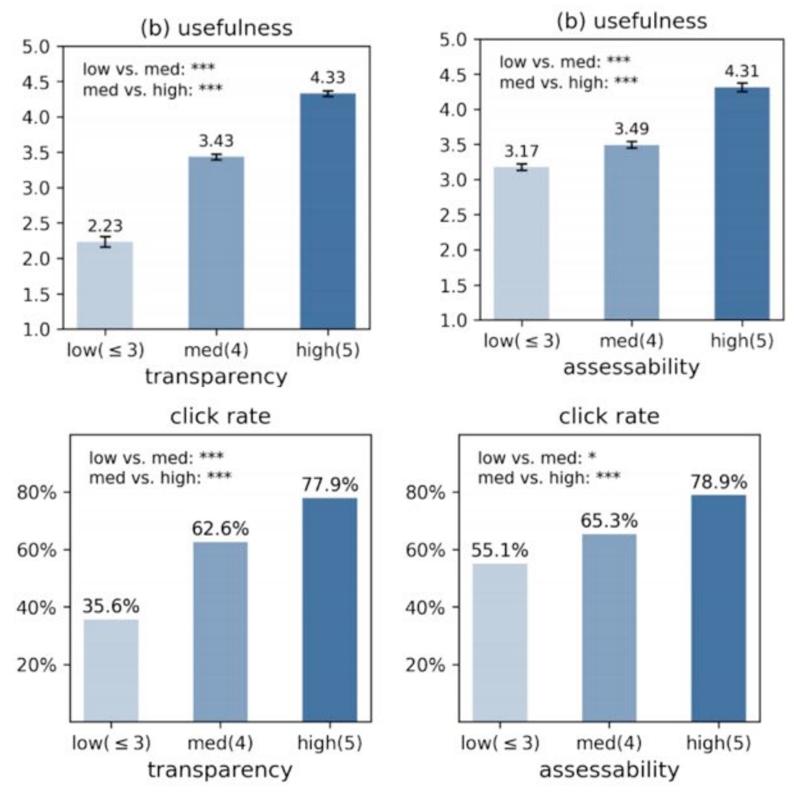
Motivation: Why do we need explainable search?

- Investigating the interpretability of search result summaries in a user study (Mi an Jiang, 2019)

Table 1: Search result summary judgment questions.

Transparency	By looking at the snippet, I can understand why the search engine returned this result for my keywords "$q".
Assessability	By looking at the snippet, I can tell if the result is useful or not without opening the link.
Usefulness	By looking at the snippet, I expect the result webpage to include useful information for the search task.





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

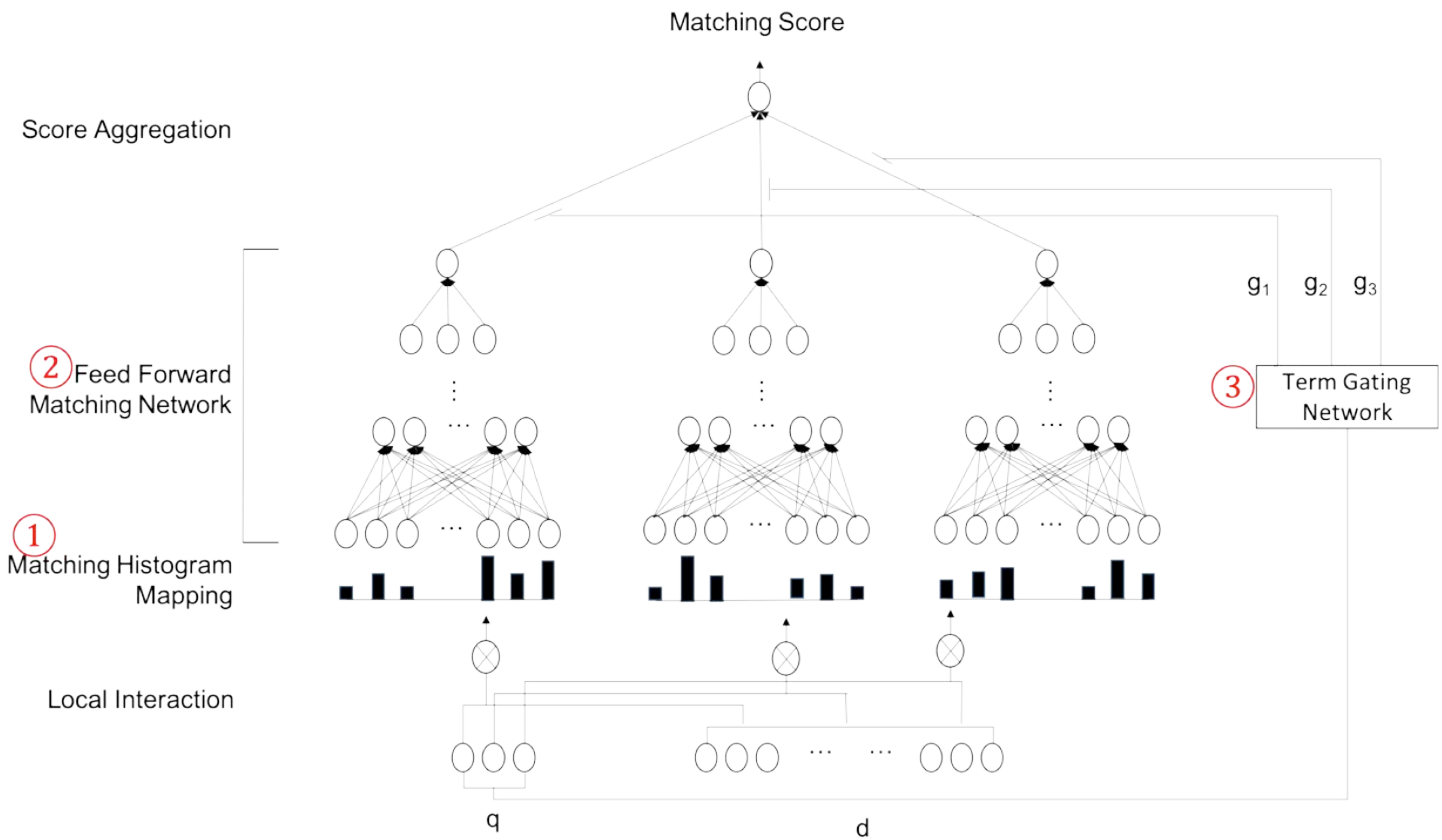
- Existing retrieval models are quite explainable, for example:
  - TF-IDF model:
    - $f_{tfidf}(q,d) = \sum_{w \in q \cap d} tf(w) \cdot idf(w)$
  - Based on exact match between query and document terms
  - Modeling the importance of query term with inverse document frequency : $idf(w) = \log \frac{N}{n_w}$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

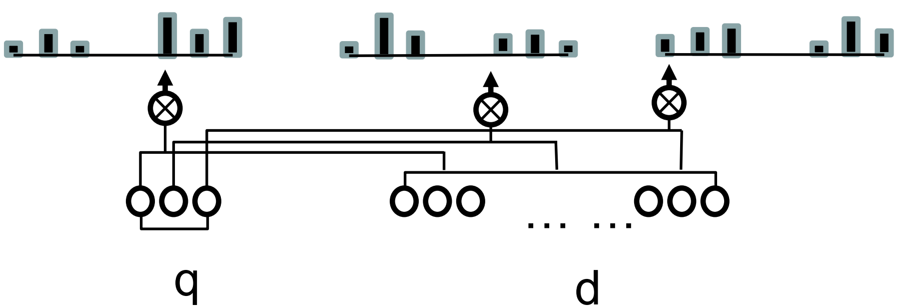
Model Architecture



(Guo 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



$$z_i^{(0)} = h\left(w_i^{(q)} \otimes d\right), i = 1, \dots, M$$

cosine similarity

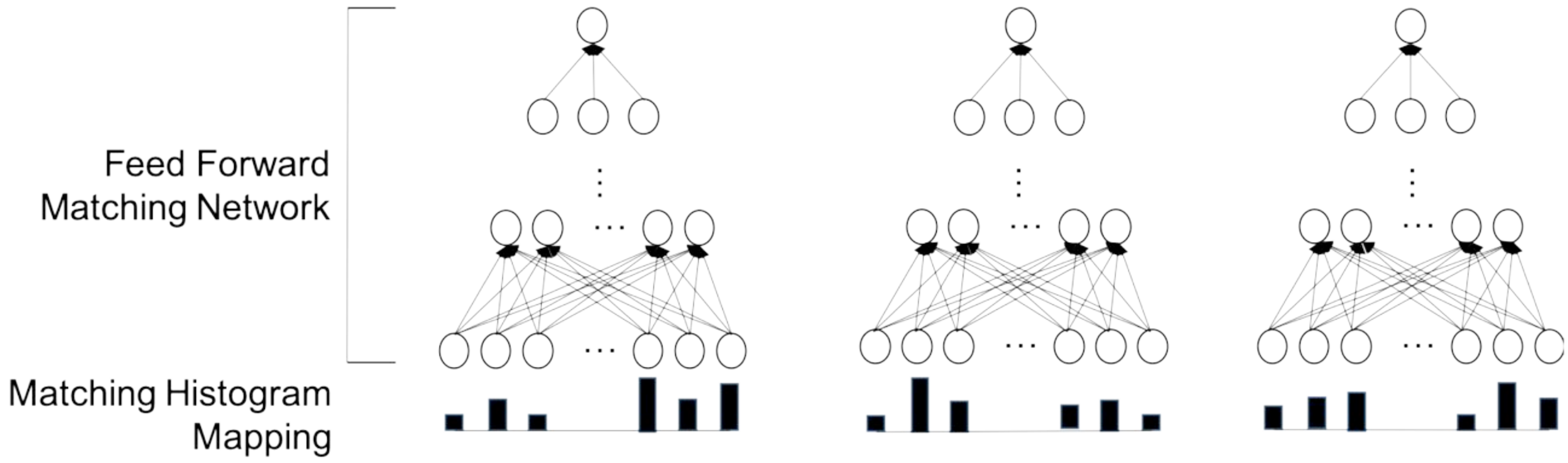
  - Different mappings *h()*:
    - Count-based histogram: frequency
    - Normalized histogram: relative frequency
    - LogCount-based histogram: logarithm

Model Architecture

- 2. Feed forward Matching Network
  - Extract hierarchical matching patterns from different levels of interaction signals

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

$$z_i^{(l)} = tanh\left(W^{(l)} z_i^{(l-1)} + b^{(l)}\right), \qquad i = 1, \ldots, M, l = 1, \ldots, L$$

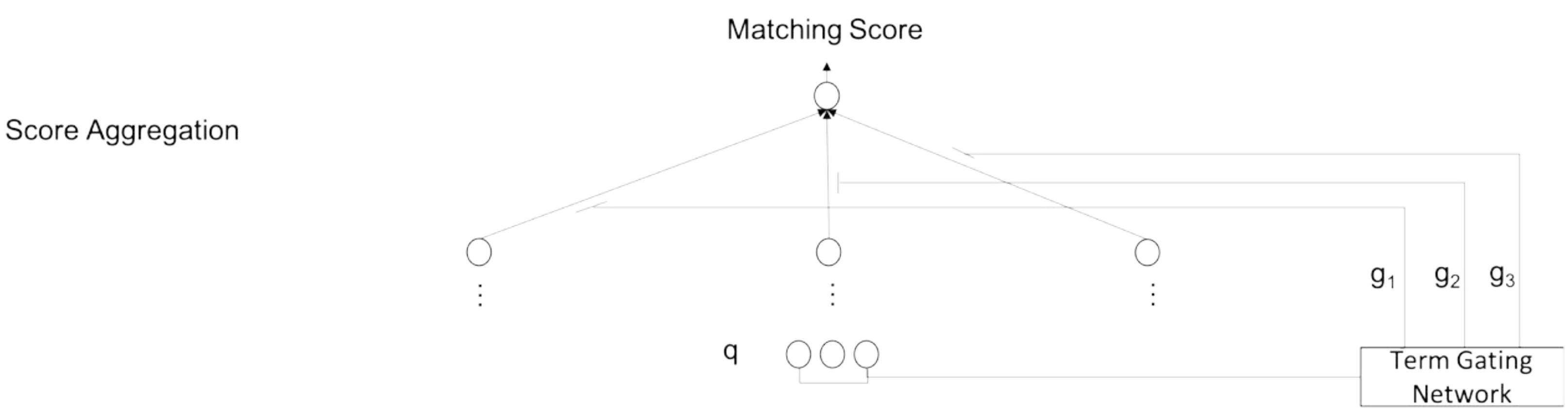


Model Architecture

- 3. Term Gating Network
  - Modeling term importance by control how much relevance score on each query term contribute to the final relevance score

$$s = \sum_{i=1}^{M} g_i \, z_i^{(L)} \qquad g_i = \frac{\exp(w_g x_i^{(q)})}{\sum_{j=1}^{M} \exp(w_g x_j^{(q)})}, \qquad i = 1, \ldots, M$$

  - 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

	Robust04	**ClueWeb09-Cat-B**
Vocabulary	0.6M	38M
Document Count	0.5M	34M
Collection Length	252M	26B
Query Count	250	150

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	$\text{DRMM}_{CH\times TV}$	0.253	0.407	0.357	0.247	0.404	0.341
	$\text{DRMM}_{NH\times TV}$	0.160−	0.293−	0.258−	0.132−	0.217−	0.186−
	$\text{DRMM}_{LCH\times TV}$	0.268+	0.423	0.381	0.265+	0.423+	0.360+
	$\text{DRMM}_{CH\times IDF}$	0.259	0.412	0.362	0.255	0.410+	0.344
	$\text{DRMM}_{NH\times IDF}$	0.187−	0.312−	0.282−	0.145−	0.243−	0.199−
	$\text{DRMM}_{LCH\times IDF}$	**0.279**+	**0.431**+	**0.382**+	**0.275**+	**0.437**+	**0.371**+

RUTGERS 清華大學 Tsinghua University UMASS AMHERST

Retrieval Performance on ClueWeb-09-Cat-B

Model Type	Model Name	Topic Titles			Topic Descriptions		
		MAP	nDCG@20	P@20	MAP	nDCG@20	P@20
Traditional Retrieval Baselines	QL	0.100	0.224	0.328	0.075	0.183	0.234
	BM25	0.101	0.225	0.326	0.080	0.196	0.255+
Representation-Focused Matching Baselines	DSSM$_T$	0.054−	0.132−	0.185−	0.046−	0.119−	0.143−
	DSSM$_D$	0.039−	0.099−	0.131−	0.034−	0.078−	0.103−
	CDSSM$_T$	0.064−	0.253−	0.214−	0.055−	0.139−	0.171−
	CDSSM$_D$	0.054−	0.134−	0.177−	0.049−	0.125−	0.160−
	ARC-I	0.024−	0.073−	0.089−	0.017−	0.036−	0.051−
Interaction-Focused Matching Baselines	ARC-II	0.033−	0.087−	0.123−	0.024−	0.056−	0.075−
	MP$_{IND}$	0.056−	0.139−	0.208−	0.043−	0.118−	0.158−
	MP$_{COS}$	0.066−	0.158−	0.222−	0.057−	0.140−	0.171−
	MP$_{DOT}$	0.044−	0.109−	0.158−	0.033−	0.073−	0.102−
Our Approach	DRMM$_{CH \times TV}$	0.103	0.245	0.347	0.072	0.188	0.253
	DRMM$_{NH \times TV}$	0.065−	0.151−	0.213−	0.031−	0.075−	0.100−
	DRMM$_{LCH \times TV}$	0.111+	0.250+	0.361+	0.083	0.213	0.275
	DRMM$_{CH \times IDF}$	0.104	0.252+	0.354+	0.077	0.204	0.267
	DRMM$_{NH \times IDF}$	0.066−	0.151−	0.216−	0.038−	0.087−	0.113−
	DRMM$_{LCH \times IDF}$	**0.113+**	**0.258+**	**0.365+**	**0.087+**	**0.235+**	**0.310+**

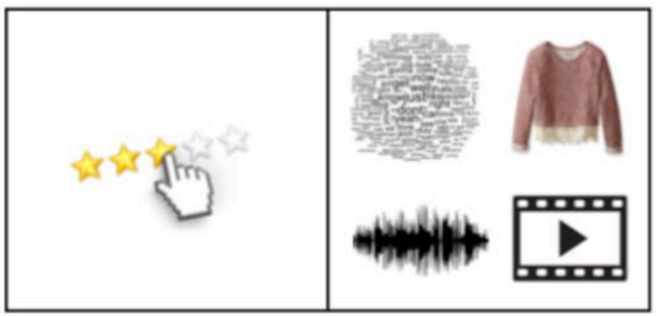
Retrieval Performance

Model Type	Model Name	Topic Titles			Topic Descriptions		
		MAP	nDCG@20	P@20	MAP	nDCG@20	P@20
Our Approach	$DRMM_{CH \times TV}$	0.253	0.407	0.357	0.247	0.404	0.341
	$DRMM_{NH \times TV}$	0.160	0.293	0.258	0.132	0.217	0.186
	$DRMM_{LCH \times TV}$	0.268	0.423	0.381	0.265	0.423	0.360
	$DRMM_{CH \times IDF}$	0.259	0.412	0.362	0.255	0.410	0.344
	$DRMM_{NH \times IDF}$	0.187	0.312	0.282	0.145	0.243	0.199
	$DRMM_{LCH \times IDF}$	**0.279**	**0.431**	**0.382**	**0.275**	**0.437**	**0.371**

- 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

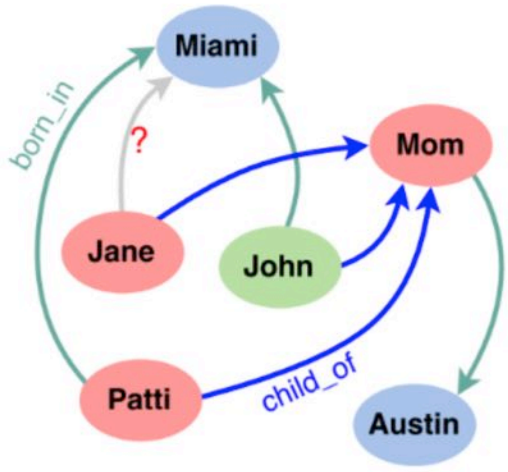
Leverage structured knowledge

- 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

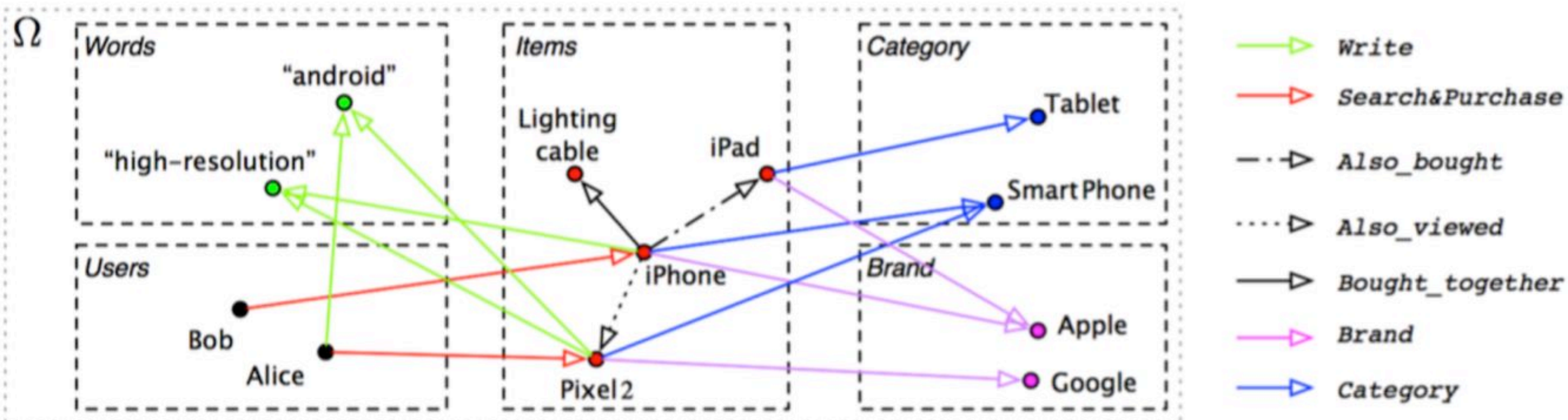
$$\boldsymbol{h} + \boldsymbol{\ell} \approx \boldsymbol{t} \qquad d(\boldsymbol{h} + \boldsymbol{\ell}, \boldsymbol{t})$$

Minimize the hinge-loss to learn entity and relation embeddings

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

User-Product Knowledge Graph

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



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

$$= \sum_{(u,v,i)\in D_{(u,v,i)}} \log \sigma\big((\boldsymbol{u} + f(q))\cdot \boldsymbol{i}\big) + k\cdot \mathbb{E}_{i'\sim P_i}[\log \sigma\big(-(\boldsymbol{u}+f(q))\cdot \boldsymbol{i'}\big)]$$

$$+ \sum_{(x,r,y)\in S_{(x,r,y)}} \log \sigma\big((\boldsymbol{x}+\boldsymbol{r})\cdot \boldsymbol{y}\big) + k\cdot \mathbb{E}_{y'\sim P_r}[\log \sigma\big(-(\boldsymbol{x}+\boldsymbol{r})\cdot \boldsymbol{y'}\big)]$$

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

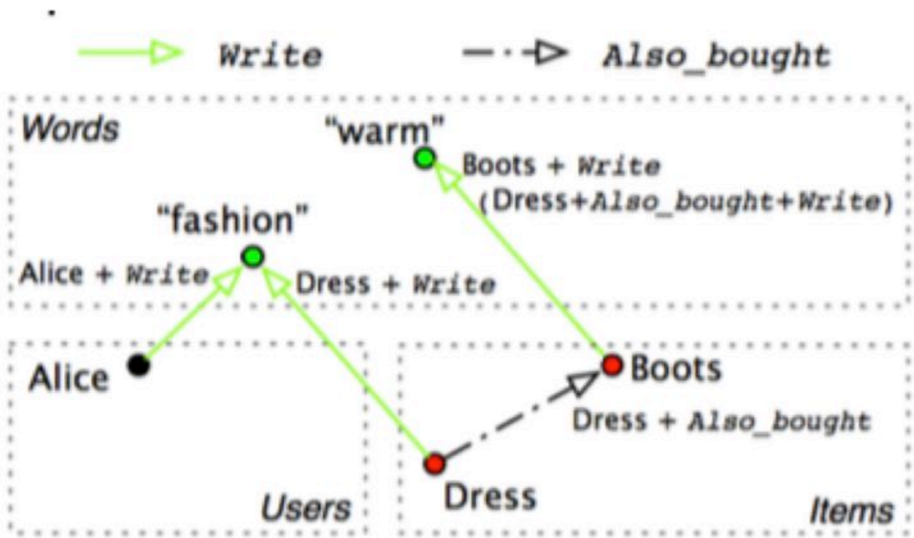
$$f(q) = \tanh(W \cdot \frac{\sum_{w_q \in q} \boldsymbol{w_q}}{|q|} + b)$$

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

	Electronics	*Kindle Store*	*CDs & Vinyl*	*Cell Phones & Accessories*
Corpus				
Number of reviews	1,689,188	982,618	1,097,591	194,439
Number of users	192,403	68,223	75,258	27,879
Number of items	63,001	61,934	64,443	10,429
Number of brands	3,525	1	1,414	955
Number of categories	983	2,523	770	206
Relationships				
Write per user	777.23±1748.6	1174.23±3682.39	1846.88±7667.51	500.01±979.78
Write per item	2373.62±5860.33	1293.47±1916.72	2156.83±4024.15	1336.64±2698.30
Also_bought per item	36.70±38.56	82.56±29.92	57.28±39.22	56.53±35.82
Also_viewed per item	4.36±9.44	0.16±1.66	0.27±1.86	1.24±4.29
Bought_together per item	0.59±0.72	0.00±0.04	0.68±0.80	0.81±0.77
Brand per item	0.47±0.50	0.00±0.00	0.21±0.41	0.52±0.50
Category per item	4.39±0.95	9.85±2.61	7.25±3.13	3.49±1.08
Train/Test				
Number of reviews	1,275,432/413,756	720,006/262,612	804,090/293,501	150,048/44,391
Number of queries	904/85	3313/1290	534/160	134/31
Number of user-query pairs	1,204,928/5,505	1,490,349/232,668	1,287,214/45,490	114,177/665
Relevant items per pair	1.12±0.48/1.01±0.09	1.87±3.30/1.48±1.94	2.57±6.59/1.30±1.19	1.52±1.13/1.00±0.05

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 vj
  - 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)

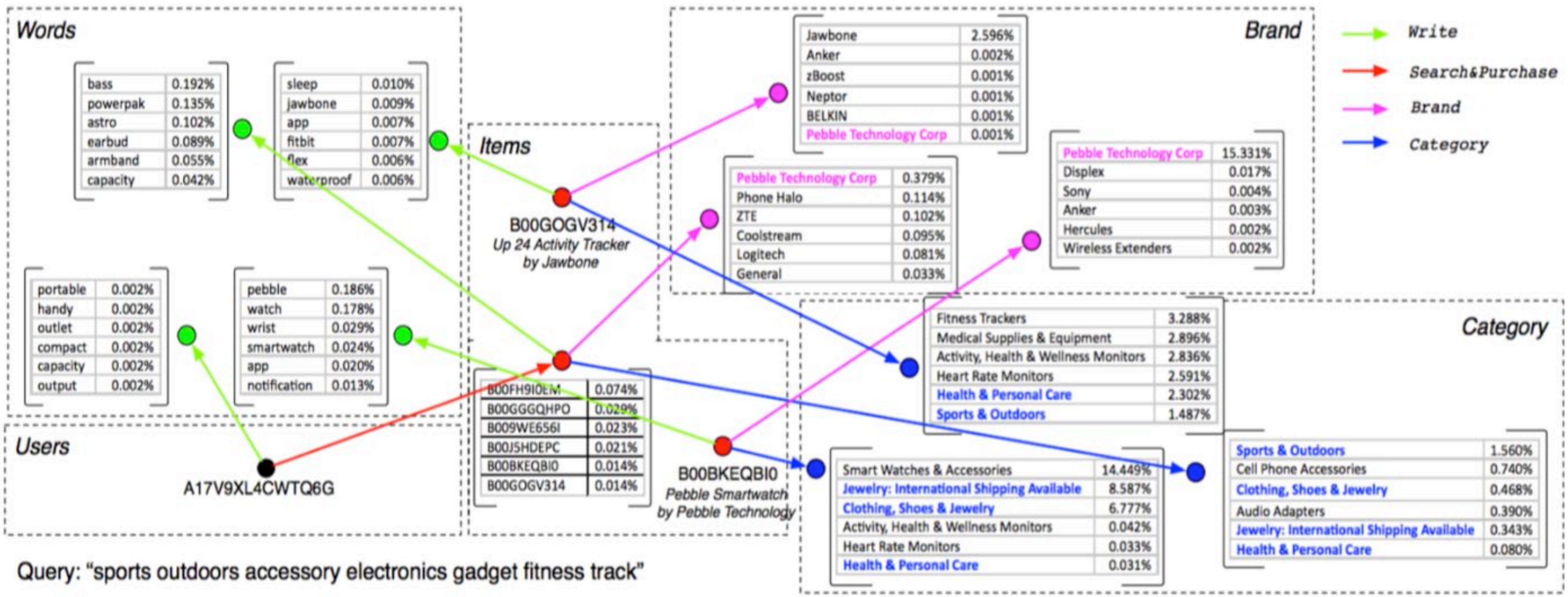
Search Performance

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

	Electronics			*Kindle Store*		
Model	MAP	MRR	NDCG	MAP	MRR	NDCG
QL	0.289	0.289	0.316	0.011	0.012	0.013
BM25	0.283	0.280	0.304	0.021	0.013	0.014
LambdaMART	0.180	0.181	0.237	0.028	0.029	0.018
LSE	0.233	0.234	0.239	0.006	0.007	0.007
HEM	0.308^{*+}	0.309^{*+}	0.329^{*+}	0.029	0.035^{*}	0.033^{*}
DREM_{NoMeta}	0.291	0.291	0.319	0.036^{*}	0.044^{*}	0.042^{*}
DREM_{AB}	0.283	0.283	0.312	0.043^{*+}	0.052^{*+}	0.050^{*+}
DREM_{AV}	0.318^{*+}	0.319^{*+}	0.349^{*+}	0.035^{*}	0.043^{*}	0.041^{*}
DREM_{BT}	0.320^{*+}	0.321^{*+}	0.346^{*+}	0.037^{*}	0.045^{*}	0.042^{*}
DREM_{Bnd}	0.314^{*+}	0.315^{*+}	0.340^{*+}	0.037^{*}	0.044^{*}	0.043^{*}
DREM_{Cat}	0.299^{+}	0.300^{+}	0.360^{*+}	0.048^{*+}	0.056^{*+}	0.056^{*+}
DREM_{All}	$\mathbf{0.366}^{*+\dagger}$	$\mathbf{0.367}^{*+\dagger}$	$\mathbf{0.408}^{*+\dagger}$	$\mathbf{0.057}^{*+\dagger}$	$\mathbf{0.067}^{*+\dagger}$	$\mathbf{0.067}^{*+\dagger}$

	CDs & Vinyl			*Cell Phones & Accessories*		
Model	MAP	MRR	NDCG	MAP	MRR	NDCG
QL	0.009	0.011	0.010	0.081	0.081	0.092
BM25	0.027	0.018	0.016	0.083	0.081	0.115
LambdaMART	0.054^{*+}	0.057^{*+}	0.051^{*+}	0.121	0.121	0.148
LSE	0.018	0.022	0.020	0.098	0.098	0.084
HEM	0.034	0.040	0.040	0.124^{*+}	0.124^{*+}	0.153^{*+}
DREM_{NoMeta}	0.034	0.041	0.040	0.107	0.107	0.127
DREM_{AB}	0.046^{+}	0.054^{+}	0.054^{+}	0.098	0.098	0.120
DREM_{AV}	0.034	0.041	0.040	0.095	0.096	0.096
DREM_{BT}	0.037^{+}	0.044^{+}	0.042^{+}	0.089	0.089	0.096
DREM_{Bnd}	0.035	0.041	0.040	0.134^{*+}	0.134^{*+}	0.152^{+}
DREM_{Cat}	0.059^{*+}	0.068^{*+}	0.070^{*+}	0.193^{*+}	0.193^{*+}	0.229^{*+}
DREM_{All}	$\mathbf{0.074}^{*+\dagger}$	$\mathbf{0.084}^{*+\dagger}$	$\mathbf{0.086}^{*+\dagger}$	$\mathbf{0.249}^{*+\dagger}$	$\mathbf{0.249}^{*+\dagger}$	$\mathbf{0.282}^{*+\dagger}$

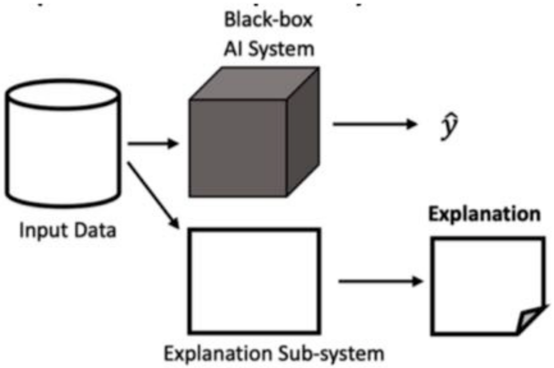
Generating Explanations



- $u+\vec{SP}+\vec{B}\rightarrow$ **Pebble Technology** $\leftarrow i_p+\vec{B}$ (5.81%): "Based on your profile and query, you may like to see something by *Pebble Technology*, and *Pebble Smartwatch by Pebble Technology* is a top product of this brand."
- $u+\vec{SP}+\vec{C}\rightarrow$ **Clothing, Shoes, Jewelry** $\leftarrow i_p+\vec{C}$ (3.17%): "Based on your profile and query, you may like to see something in *Clothing, Shoes, Jewelry*, and *Pebble Smartwatch by Pebble Technology* is a top product in this category."
- $u+\vec{SP}+\vec{C}\rightarrow$ **Health&Personal Care** $\leftarrow i_j+\vec{C}$ (0.184%): "Based on your profile and query, you may like to see something in *Health&Personal Care*, and *Up 24 Activity Tracker by Jawbone* is a top product in this category."
- $u+\vec{SP}+\vec{C}\rightarrow$ **Sports&Outdoors** $\leftarrow i_j+\vec{C}$ (2.32%): "Based on your profile and query, you may like to see something 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)



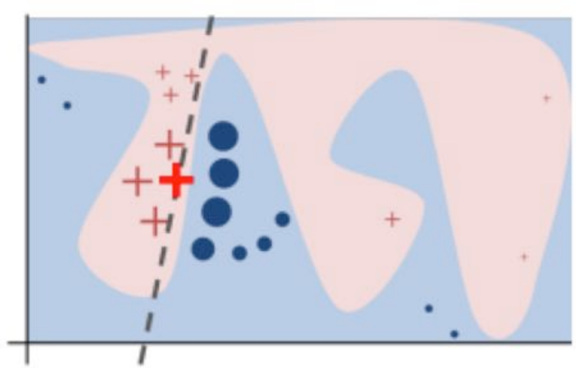
- EXS: Explainable Search Using Local Model Agnostic Interpretability (J.Singh and A.Anand 2019)
- Primary goal: aid users in answering the following questions:
  - Why is this document relevant to the query?
  - Why is this document ranked higher than the other?
  - What is the intent of the query according to the ranker?
- Basic Idea: Adapt LIME to search task

LIME (Ribeiro et al. 2016)

- **L**ocal **I**nterpretable **M**odel-agnostic **E**xplanations
- For a trained model f and an instance x, the explanation by LIME is obtained by:

$$\xi(x) = argmin_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

  - 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. $\pi_x(z) = \exp(-\frac{D(x,z)^2}{\sigma^2})$)
  - $L(f, g, \pi_x)$: a measure of how unfaithful g is in approximating f in the locality defined by π_x (e.g. $L(f, g, \pi_x) = \sum_{z,z' \in Z} \pi_x(z)\big(f(z) - g(z')\big)^2$)



(Ribeiro et al. 2016)

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 $P(X = relevant\,|q, d', R)$
  - Top-k Binary: $P(X = relevant\,|q, d', R) = 1$ if $R(q, d') > R(q, d'_k)$
  - Score based: $P(X = relevant\,|q, d', R) = 1 - \frac{R(q,d_1) - R(q,d')}{R(q,d_1)}$
  - Rank based: $P(X = relevant\,|q, d', R) = 1 - \frac{rank(d')}{k}$

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 $k = rank(d_B)$ 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

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

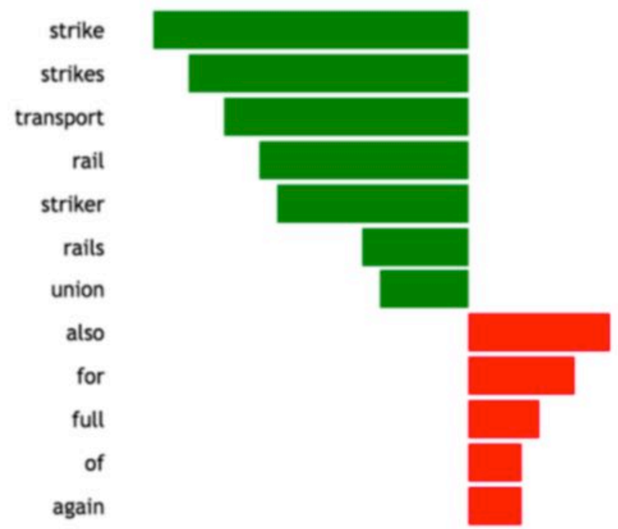


Figure 2: Intent explanation for the query 'Rail Strikes' when using DRMM to rank documents from a news collection.

(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

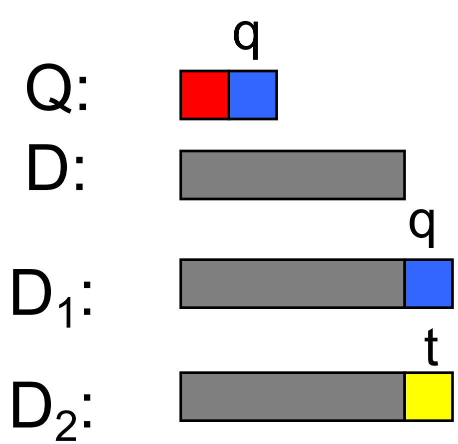
Seven Relevance Constraints

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

(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 $S(Q, D \cup \{q\}) > S(Q, D \cup \{t\})$



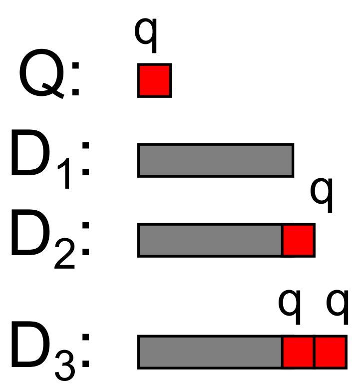
$$S(Q, D_1) > S(Q, D_2)$$

(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 $S(Q, D \cup \{q\}) - S(Q, D) > S(Q, D \cup \{q\} \cup \{q\}) - S(Q, D \cup \{q\})$



$$S(D_2,Q)-S(D_1,Q)>S(D_3,Q)-S(D_2,Q)$$

(Fang and Zhai, 2014)

Term Frequency Constraints

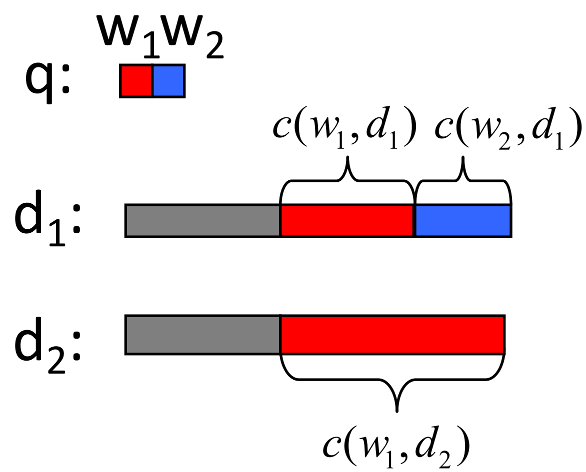
- TFC3
  - Intuition: favor a document with more distinct query terms
  - Let q be a query and w_1, w_2 be two query terms.

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

    if $c(w_1, d_2) = c(w_1, d_1) + c(w_2, d1)$

    and $c(w_2, d_2) = 0, c(w_1, d_1) \neq 0, c(w_2, d_1) \neq 0$

    then $S(q, d_1) > S(q, d_2)$



$S(d_1, q) > S(d_2, q)$

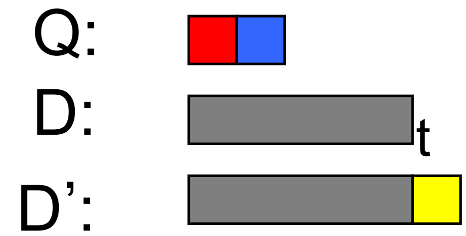
(Fang and Zhai, 2014)

Term Discrimination Constraint

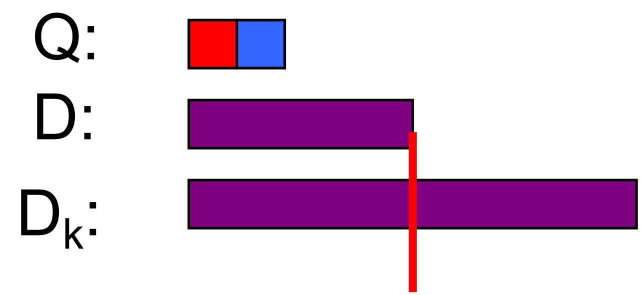
- TDC
  - Intuition: to penalize the words popular in the collection and assign higher weights to discriminative terms
  - Let $Q = \{q_1, q_2\}$ Assume $|D_1| = |D_2|$ and $c(q_1, D_1) + c(q_2, D_1) = c(q_1, D_2) + c(q_2, D_2)$. If $idf(q_1) \geq idf(q_2)$ and $c(q_1, D_1) \geq\ c(q_1, D_2)$, we have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

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 $D \cap Q \neq \phi$, and D_k is constructed by concatenating D with itself k times, then $S(Q, D_k) \geq S(Q, D)$



$$S(Q, D') < S(Q, D)$$



$$S(Q, D_k) \geq S(Q, D)$$

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!)

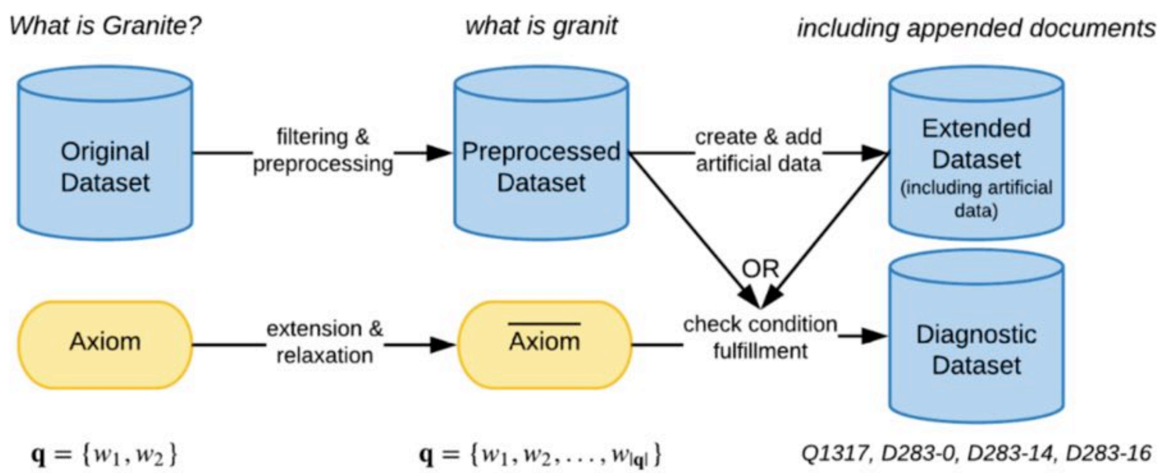


Fig. 1: Overview of the diagnostic dataset creation pipeline. In *italics*, we show an example for the $\overline{\texttt{TFC2}}$ axiom as extracted from question 1317 on passages from Wikipedia document 283 in the `WikiPassageQA` dataset, and refer to appended documents as an example of artificial data (for $\overline{\texttt{LNC2}}$).

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 2: Overview of models' retrieval effectiveness and fraction of fulfilled axiom instances. $^{1/2/3/4}$ denote statistically significant improvements (Wilcoxon signed rank test with $p < 0.05$) in retrieval effectiveness.

	Retrieval effectiveness			Performance per axiom				
	MAP	MRR	P@5	TFC1	TFC2	M-TDC	LNC2^{Test}	LNC2^{All}
[1] BM25	$0.52^{3,4}$	$0.60^{3,4}$	0.18^{3}	0.73	**0.98**	**1.00**	**0.80**	**0.80**
[2] QL	$\mathbf{0.54}^{1,3,4}$	$\mathbf{0.62}^{1,3,4}$	$\mathbf{0.19}^{3}$	**0.87**	0.63	**0.94**	0.68	0.68
[3] Duet	0.25	0.29	0.10	0.69	0.56	0.48	0.19	0.47
[4] MatchPyramid	0.44^{3}	0.51^{3}	0.18^{3}	0.79	0.58	0.63	0.00	0.19
[5] DRMM	$0.55^{1,2,3,4}$	$0.64^{1,2,3,4}$	$0.20^{1,2,3,4}$	0.84	**0.60**	**0.76**	0.05	0.12
[6] aNMM	$\mathbf{0.57}^{1,2,3,4}$	$\mathbf{0.66}^{1,2,3,4}$	$\mathbf{0.21}^{1,2,3,4}$	**0.85**	0.56	0.69	**0.38**	**0.47**

- 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 the labeled data for training neural ranking models
  - For each document d and constraint Δ_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 Δ_i on the pair d and $d^{(i)}$

Axiom	Perturbation	Expected result
TFC1-A	Sample a query term from query q and insert it at a random position in d	$d^{(i)} >_q d$
TFC1-D	Sample a query term from query q and delete it in d	$d^{(i)} <_q d$
TFC3	Sample a query term not present in d, and insert it in d.	$d^{(i)} >_q d$
LNC	Sample k terms and insert them at random positions in d	$d^{(i)} <_q d$

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

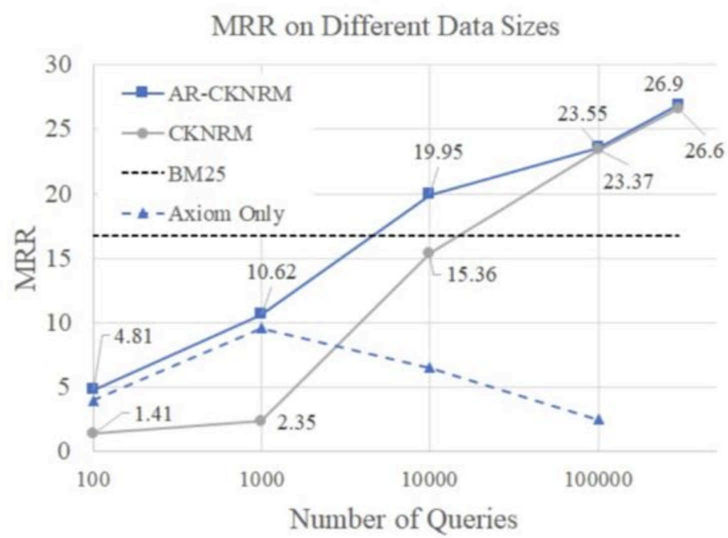


Figure 1: MRR results of training CKNRM and its axiomatic variant on datasets with 100, 1k, 10k, 100k, and all MS-MARCO queries on the dev set. Each point represents the ensemble of four independently trained models.

Ablation on 10k Queries

	MAP	MRR
CKNRM	15.13	15.36
+ TFC1-A	19.33	19.56
+ TFC1-D	18.16	18.38
+ TFC3	19.05	19.28
+ LNC	11.42	11.47
+ All Axioms	19.70	19.95

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

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

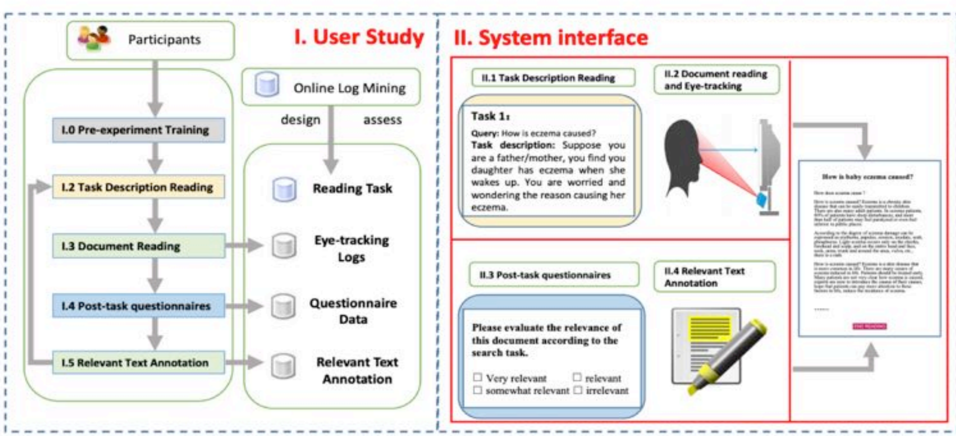
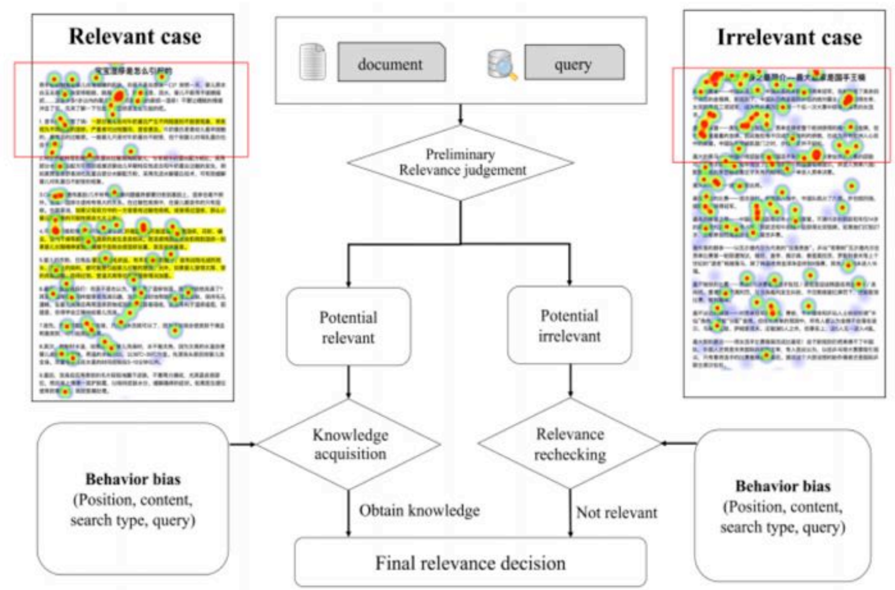


Figure 2: User study procedure. The system interface is translated from Chinese.



Heuristic Constraints from Reading Behavior

- Define six reading heuristic

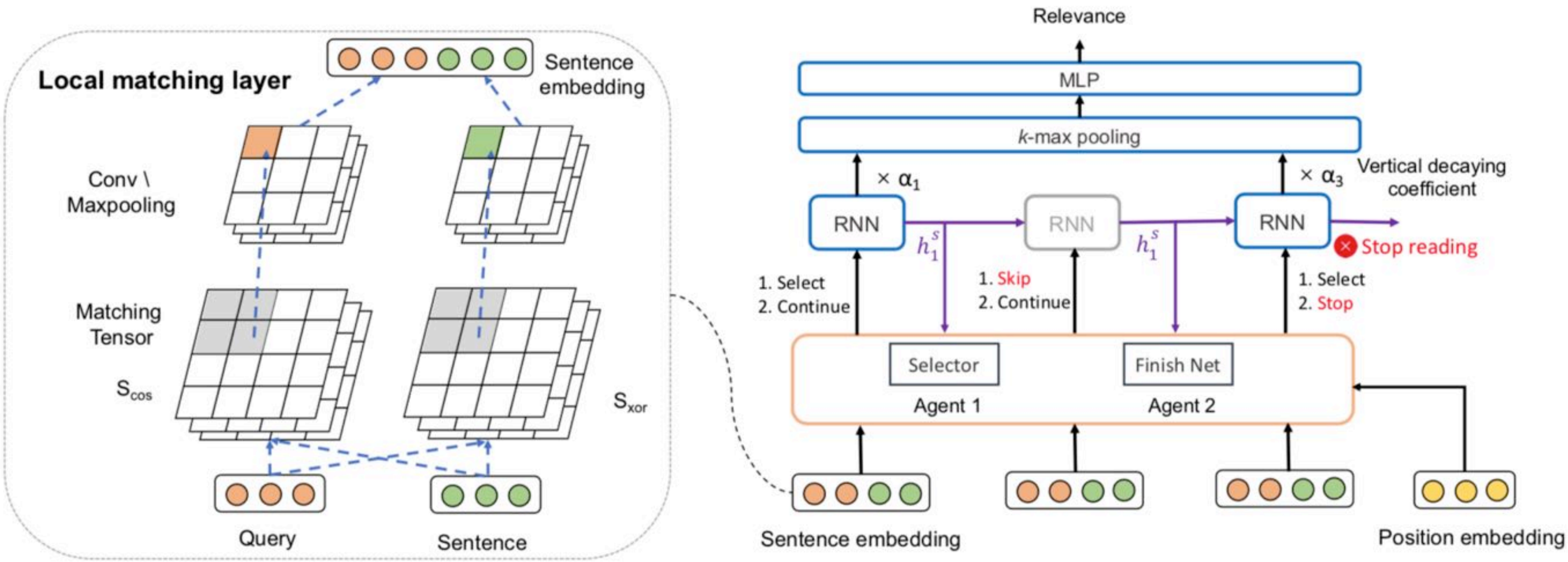
#	Heuristic	Description	Implication for retrieval models
a	**Sequential reading**	Reading direction is from top to bottom	The presented order of the content may affect relevance
b	**Vertical decaying attention**	Reading attention is decaying vertically	Retrieval model should assign more weights to the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

- Analyze whether existing neural IR models satisfy these heuristics

Models	a	b	c	d	e	f
ARC-I						
ARC-II			√			
DRMM			√			
Match Pyramid			√			
KNRM			√			
PACRR			√			
DeepRank			√		√	
HiNT	√		√	√		

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

	Test-SAME (PSCM)				Test-DIFF (UBM)			
	NDCG@1	NDCG@3	NDCG@5	NDCG@10	NDCG@1	NDCG@3	NDCG@5	NDCG@10
BM25	0.7048*	0.7202*	0.7414*	0.7967*	0.6127*	0.6509*	0.6819*	0.7429*
ARC-I	0.7583*	0.7647	0.7804	0.8286	0.6489*	0.6869	0.7142	0.7677
ARC-II	0.7239*	0.7347*	0.7519*	0.8061*	0.6303*	0.6667*	0.6948*	0.7523*
DRMM	0.6958*	0.7141*	0.7352*	0.7923*	0.6024*	0.6471*	0.6790*	0.7404*
MatchPyramid	0.6851*	0.7028*	0.7248*	0.7857*	0.5938	0.6386	0.6716*	0.7366*
KNRM	0.6997*	0.7121*	0.7336*	0.7917*	0.6048	0.6465*	0.6775*	0.7400*
PACRR	0.7072*	0.7219*	0.7411*	0.7981*	0.6172*	0.6557*	0.6860*	0.7465*
DeepRank	0.7058*	0.7227*	0.7452*	0.8059*	0.6099*	0.6566*	0.6891*	0.7540*
HiNT	0.7550*	0.7592*	0.7751*	0.8264	0.6564	0.6895	0.7072*	0.7603*
RIM	**0.7746**	**0.7705**	**0.7830**	**0.8304**	**0.6602**	**0.6918**	**0.7170**	**0.7689**

	NTCIR-13				NTCIR-14			
	NDCG@1	NDCG@3	NDCG@5	NDCG@10	NDCG@1	NDCG@3	NDCG@5	NDCG@10
BM25	0.6099	0.6194	0.6253	0.6391	0.4324	0.4432	0.4383	0.4706
ARC-I	0.5933	0.6153	0.6184	0.6222	0.4726	0.4690	0.4643	0.4814
ARC-II	0.6466	0.6649	0.6523	0.6426	0.4556	0.4369	0.4405	0.4700
DRMM	0.6866	0.6490	0.6487	0.6378	0.4345	0.4651	0.4657	0.4847
MatchPyramid	0.6866	0.6507	0.6458	0.6436	0.3586	0.3838	0.3998	0.4378
KNRM	0.6700	0.6564	0.6557	0.6591	0.4367	0.4252	0.446	0.4739
PACRR	0.6700	0.6661	0.6659	0.6620	0.4219	0.4483	0.4541	0.4689
DeepRank	0.6750	0.6606	0.6617	**0.6648**	0.4894	0.4588	0.4640	0.4793
HiNT	0.6566	0.6599	0.6548	0.6449	0.4746	0.4643	0.4617	0.4898
RIM	**0.7050**	**0.6797**	**0.6749**	0.6570	**0.4979**	**0.4887**	**0.4911**	**0.5021**

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

	Global vs Local	Intrinsic vs Post-hoc
Building Interpretable search models (Guo et al. 2016)	Global	Intrinsic
Using structured knowledge (Explainable Product Search with Knowledge Base Embedding)	Local	Intrinsic
Post-hoc explanation methods for search (J.Singh and A.Anand 2019)	Local	Post-hoc
Axiomatic analysis of search models (Fang et al. 2004) (Rennnings et al. 2019) (Rosset et al. 2019) (Li et al. 2019)	Global	Post-hoc

Reference

- (Mi an Jiang, 2019) Mi, Siyu and Jiang, Jiepu. “Understanding the Interpretability of Search Result Summaries.”, In SIGIR 2019.
- (B. Kim, et al. NIPS 2016) Kim, Been, Rajiv Khanna, and Oluwasanmi O. Koyejo. "Examples are not enough, learn to criticize! criticism for interpretability." *Advances in Neural Information Processing Systems*. 2016.
- (Doshi-Velez and Kim, 2017) Doshi-Velez, Finale, and Been Kim. "Towards a rigorous science of interpretable machine learning." *arXiv preprint arXiv:1702.08608* (2017).
- (X.Wang, NExT++ Workshop 2019) http://www.nextcenter.org/wp-content/uploads/2019/03/Wang-Xiang.pdf
- (Joachims 2012) Joachims, T. (2002), "Optimizing Search Engines using Clickthrough Data", Proceedings of the ACM Conference on Knowledge Discovery and Data Mining
- (Wu et al. 2010) Wu, Qiang, et al. "Adapting boosting for information retrieval measures." Information Retrieval 13.3 (2010): 254-270.
- (Guo et al. 2016) Guo, Jiafeng, et al. "A deep relevance matching model for ad-hoc retrieval." Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. ACM, 2016.

Reference

- (Van Gysel et al. 2016) Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2016. Learning latent vector spaces for product search. In CIKM 2016.
- (Ai et al. 2017) Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W Bruce Croft. 2017. Learning a Hierarchical Embedding Model for Personalized Product Search. In SIGIR 2017.
- (J.Singh and A.Anand 2019) Singh, Jaspreet, and Avishek Anand. "EXS: Explainable Search Using Local Model Agnostic Interpretability." Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. ACM, 2019.
- (Ribeiro et al. 2016) Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why Should I Trust You?: Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 1135–1144.
- (Fang et al. 2004) Fang, Hui, Tao Tao, and ChengXiang Zhai. "A formal study of information retrieval heuristics." Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2004.
- (Fang et al. 2011) Fang, Hui, Tao Tao, and ChengXiang Zhai. Diagnostic evaluation of information retrieval models. H. Fang, T. Tao and C. Zhai. TOIS, 2011.

Reference

- (Fang and Zhai, 2014) Axiomatic Analysis and Optimization of Information Retrieval Models. SIGIR 2014 Tutorial (https://www.eecis.udel.edu/~hfang/pubs/sigir14-axiomatic.pptx)
- (Rennnings et al. 2019) Rennings, Daniël, Felipe Moraes, and Claudia Hauff. "An Axiomatic Approach to Diagnosing Neural IR Models." European Conference on Information Retrieval. Springer, Cham, 2019.
- (Rosset et al. 2019) Rosset, Corby, et al. "An Axiomatic Approach to Regularizing Neural Ranking Models." arXiv preprint arXiv:1904.06808(2019).
- (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
- (Dai et al. 2018) Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In Proceedings of the eleventh ACM international conference on web search and data mining. ACM, 126–134.
- (Li et al. 2018) Li, Xiangsheng, et al. "Understanding Reading Attention Distribution during Relevance Judgement." Proceedings of the 27th ACM International Conference on Information and Knowledge Management. ACM, 2018.


THANKS
Q
A