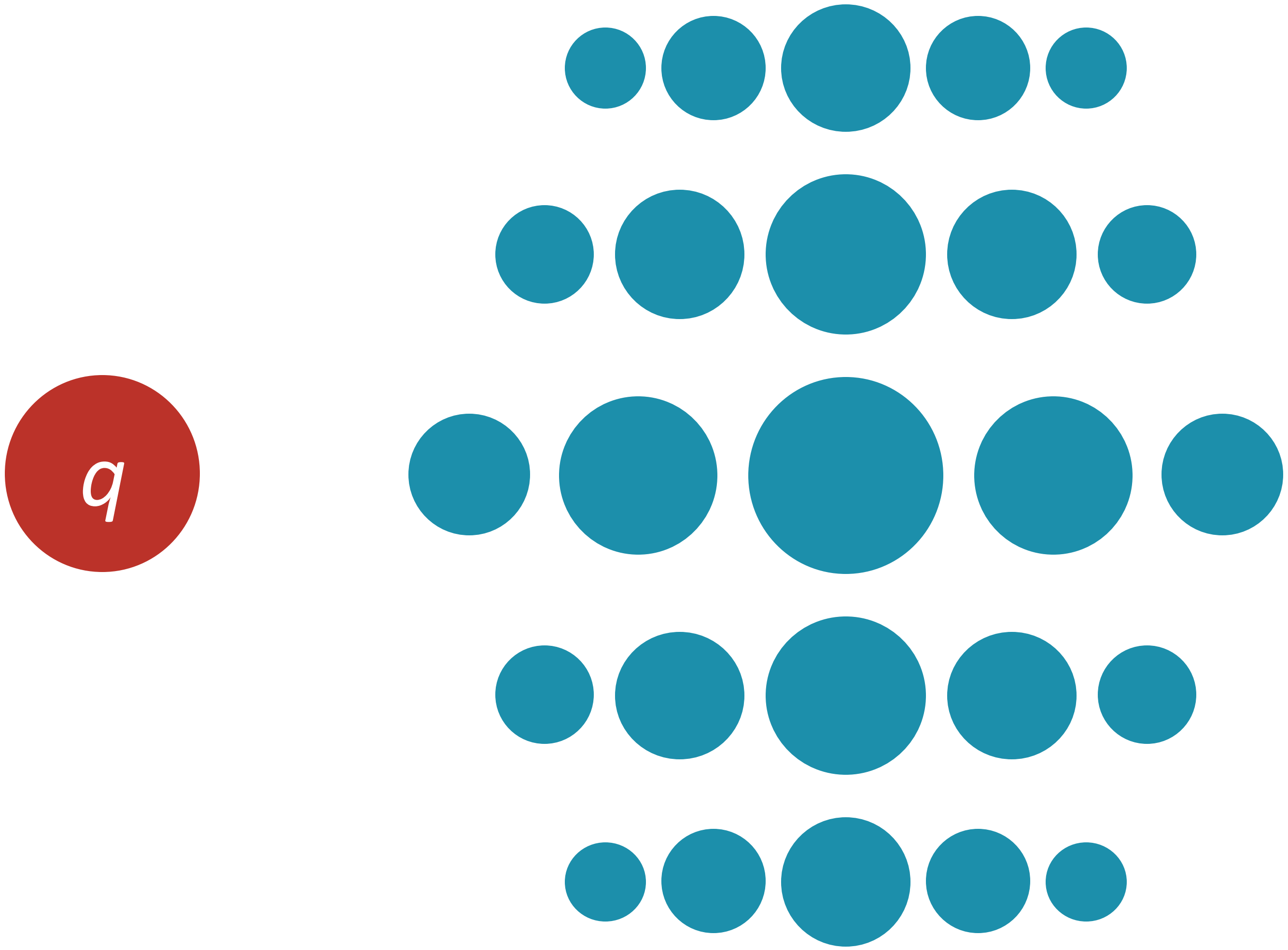
UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Information Retrieval

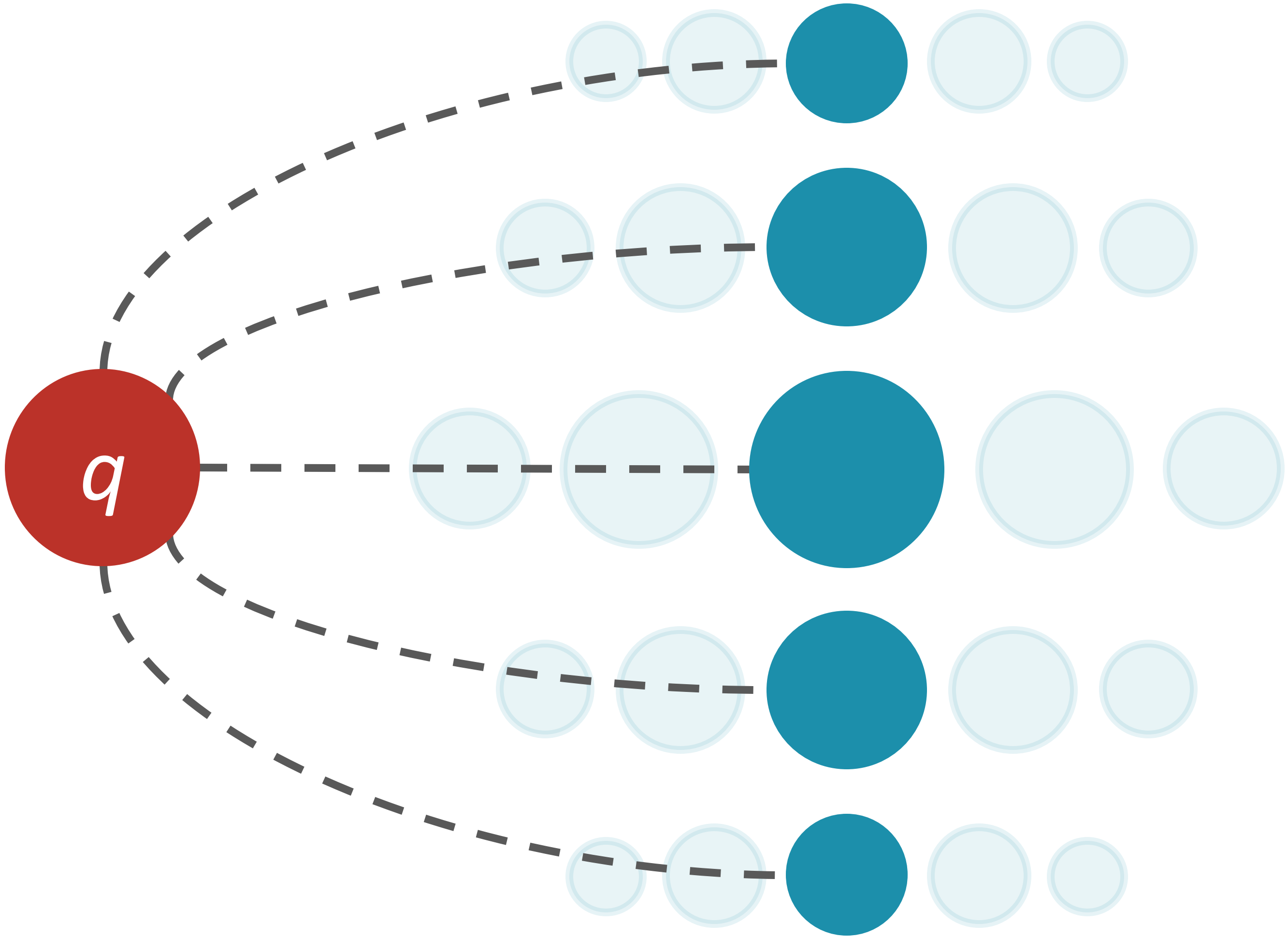
Feedback Models

Rodrygo L. T. Santos
rodrygo@dcc.ufmg.br

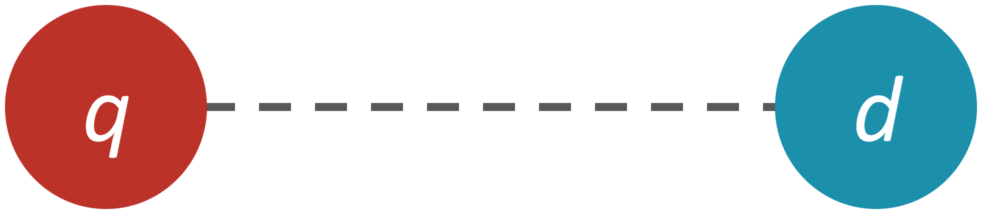
The ranking problem



The ranking problem

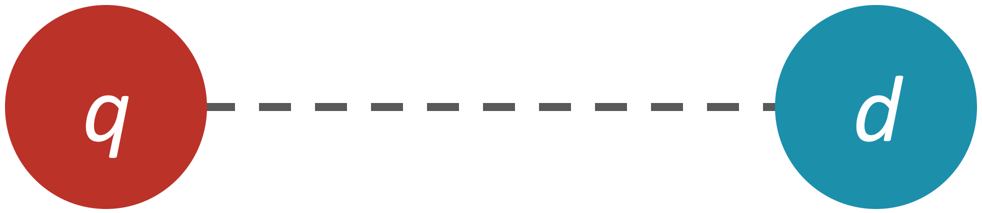


The ranking problem



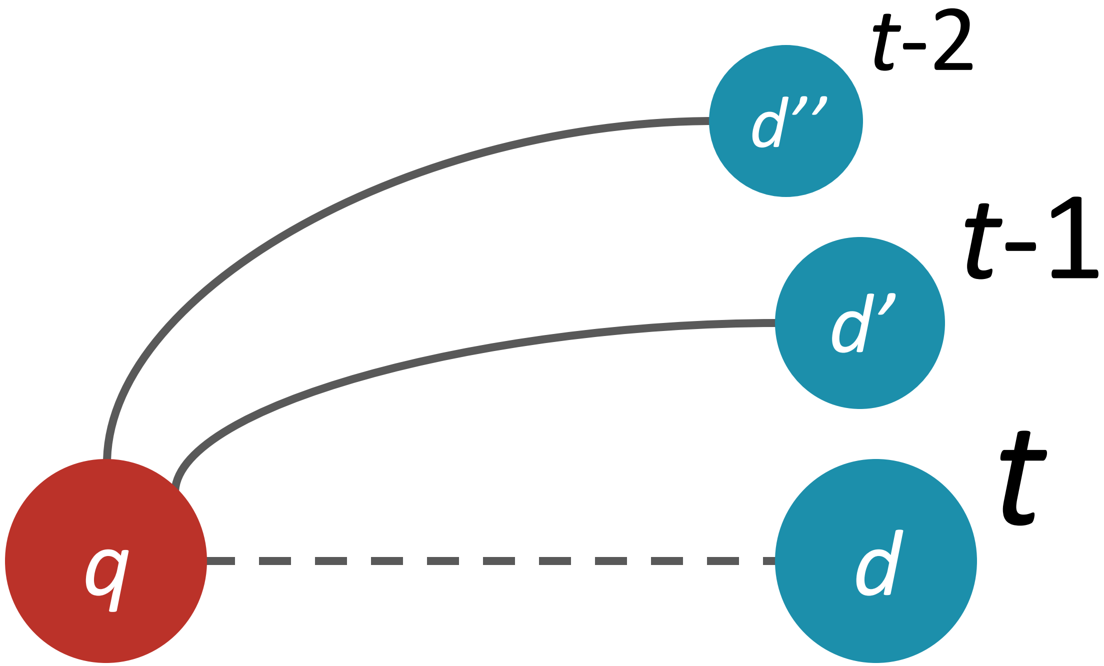
$$f(q, d)$$

The ranking problem



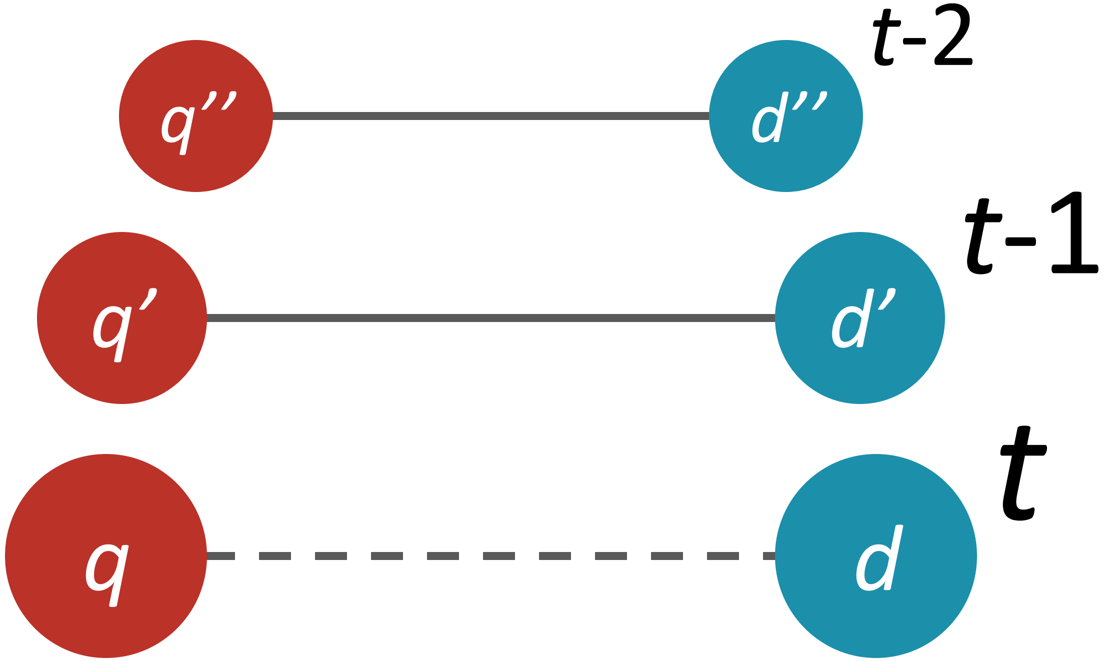
$$f(q, d, c)$$

Exploiting interactions



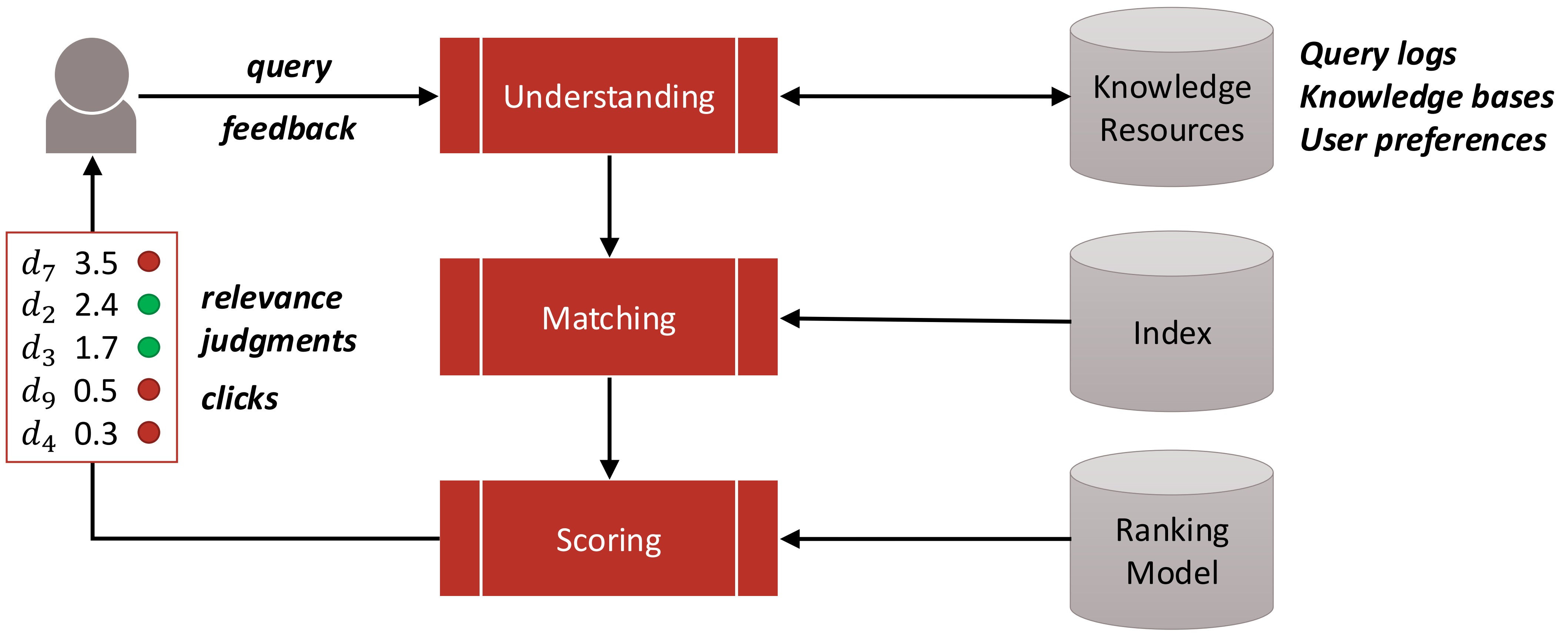
$$f(q, d, c)$$

Exploiting interactions



$$f(q, d, c)$$

Eliciting feedback



Eliciting feedback

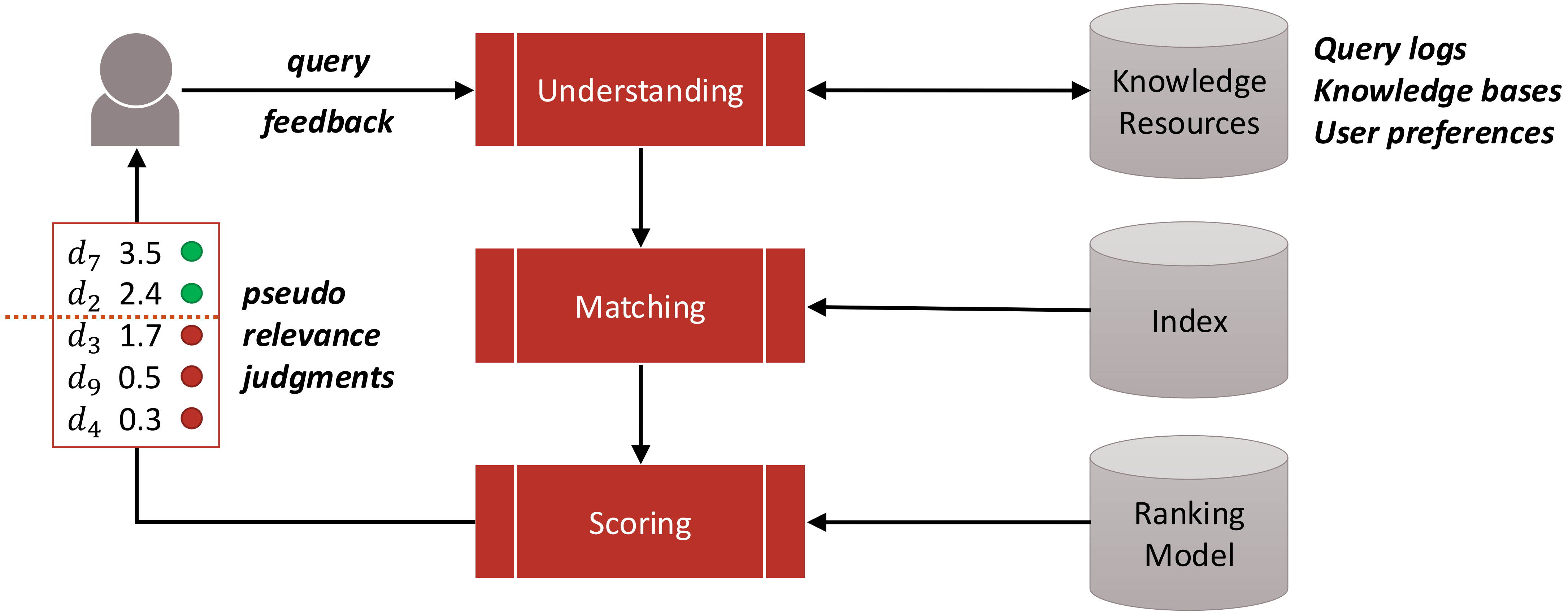
Explicit feedback

- Explicit relevance judgments
- Reliable, but costly

Implicit feedback

- Positive-only ”judgments” (e.g., clicks, dwell time)
- Noisy and biased, but cheap and abundant

Simulating feedback



Simulating feedback

Pseudo-relevance feedback

◦ Top-*k* results are assumed to be relevant

◦ Very sensitive to ranking quality, but automatic

Exploiting feedback

Machine-learned ranking

- User feedback can be treated as supervision for learning effective ranking models
- See classes on ***Learning to Rank***

Exploiting feedback

Query expansion

- Feedback documents can help enhance the user’s query by providing related expansion terms

Example: [information retrieval]

- Relevant or pseudo-relevant documents may provide related terms like ”search engine”, ”ranking”

Feedback in vector space models

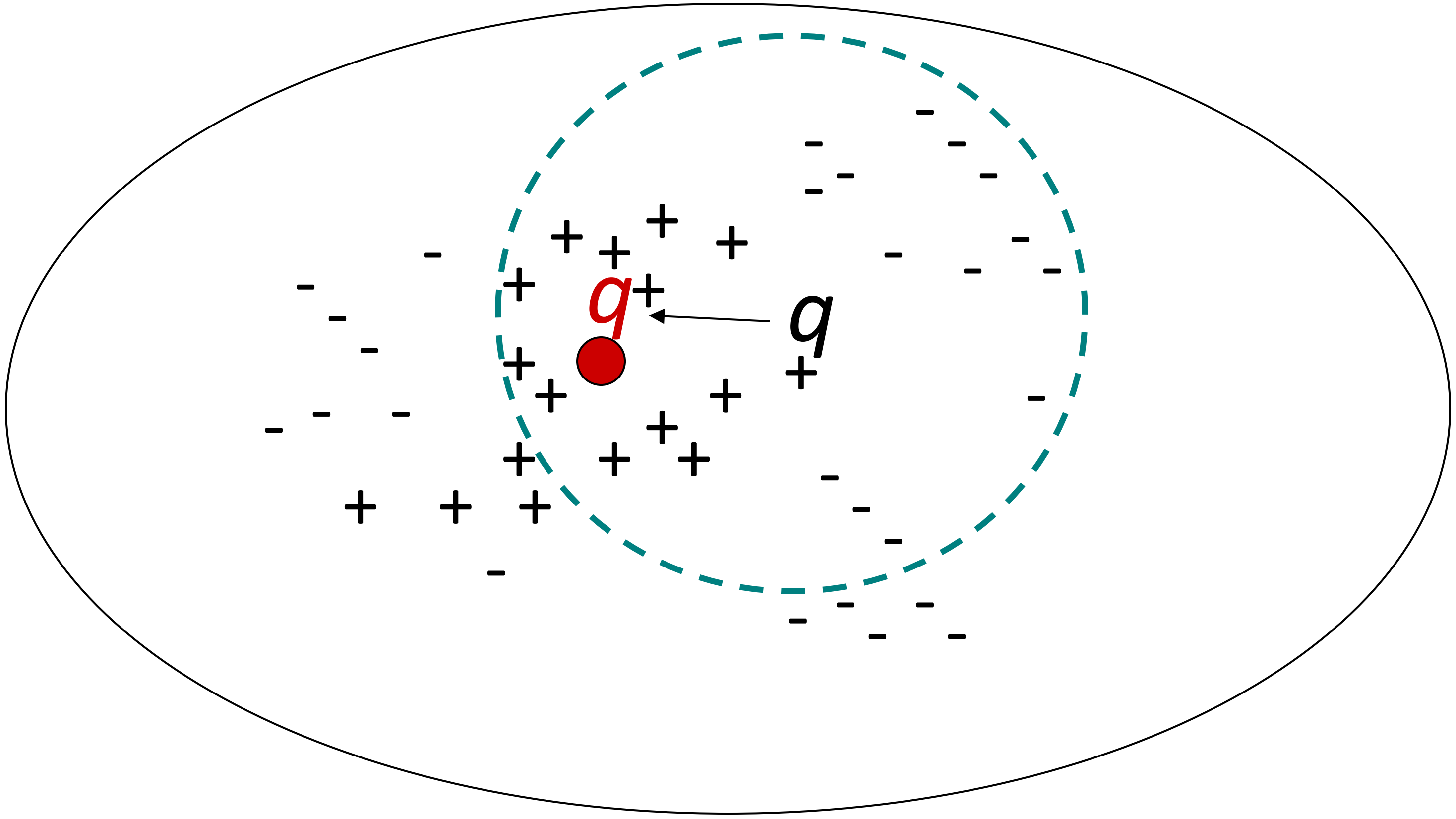
General idea: query modification

- Adding new (weighted) terms
- Adjusting weights of old terms

Rocchio (1971): most well-known approach

- Also effective and robust in practice

Rocchio method



Rocchio method

Standard operation in vector space

Parameters

Modified query

$$\vec{q}_m = \alpha\vec{q} + \frac{\beta}{|G|}\sum_{\vec{d}_i \in G} \vec{d}_i - \frac{\gamma}{|\bar{G}|}\sum_{\vec{d}_j \in \bar{G}} \vec{d}_j$$

Original query

Rel docs

Non-rel docs

Rocchio example

$$V = \{news, about, presidential, campaign, food, text\}$$

$$\vec{q} = \{1, 1, 1, 1, 0, 0\}$$

		{	news	about	pres.	campaign	food	text	}
$-$	d_1	{	1.5	0.1	0.0	0.0	0.0	0.0	}
$-$	d_2	{	1.5	0.1	0.0	2.0	2.0	0.0	}
$+$	d_3	{	1.5	0.0	3.0	2.0	0.0	0.0	}
$+$	d_4	{	1.5	0.0	4.0	2.0	0.0	0.0	}
$-$	d_5	{	1.5	0.0	0.0	6.0	2.0	0.0	}

Rocchio example

		{	news	about	pres.	campaign	food	text	}
−	d_1	{	1.5	0.1	0.0	0.0	0.0	0.0	}
−	d_2	{	1.5	0.1	0.0	2.0	2.0	0.0	}
+	d_3	{	1.5	0.0	3.0	2.0	0.0	0.0	}
+	d_4	{	1.5	0.0	4.0	2.0	0.0	0.0	}
−	d_5	{	1.5	0.0	0.0	6.0	2.0	0.0	}

		{	news	about	pres.	campaign	food	text	}
+	C_r	{	$\frac{1.5+1.5}{2}$	0.0	$\frac{3.0+4.0}{2}$	$\frac{2.0+2.0}{2}$	0.0	0.0	}
−	C_n	{	$\frac{1.5+1.5+1.5}{3}$	$\frac{0.1+0.1+0.0}{3}$	0.0	$\frac{0.0+2.0+6.0}{3}$	$\frac{0.0+2.0+2.0}{3}$	0.0	}

Rocchio example

		{	news	about	pres.	campaign	food	text	}
$+$	C_r	{	$\frac{1.5+1.5}{2}$	0.0	$\frac{3.0+4.0}{2}$	$\frac{2.0+2.0}{2}$	0.0	0.0	}
$-$	C_n	{	$\frac{1.5+1.5+1.5}{3}$	$\frac{0.1+0.1+0.0}{3}$	0.0	$\frac{0.0+2.0+6.0}{3}$	$\frac{0.0+2.0+2.0}{3}$	0.0	}

$$\vec{q} = \{1, 1, 1, 1, 0, 0\}$$

$$\vec{q}_m = \alpha \cdot \vec{q} + \beta \cdot C_r - \gamma \cdot C_n$$

$$= \{\alpha + 1.5\beta - 1.5\gamma, \alpha - 0.067\gamma, \alpha + 3.5\beta, \alpha + 2\beta - 2.67\gamma, -1.33\gamma, 0\}$$

Rocchio in practice

Non-relevant documents lack coherence

- Keep low weight for negative examples (γ)

Training set is small and noisy and may be biased

- Keep relatively high weight on the original query (α)

Feedback in language models

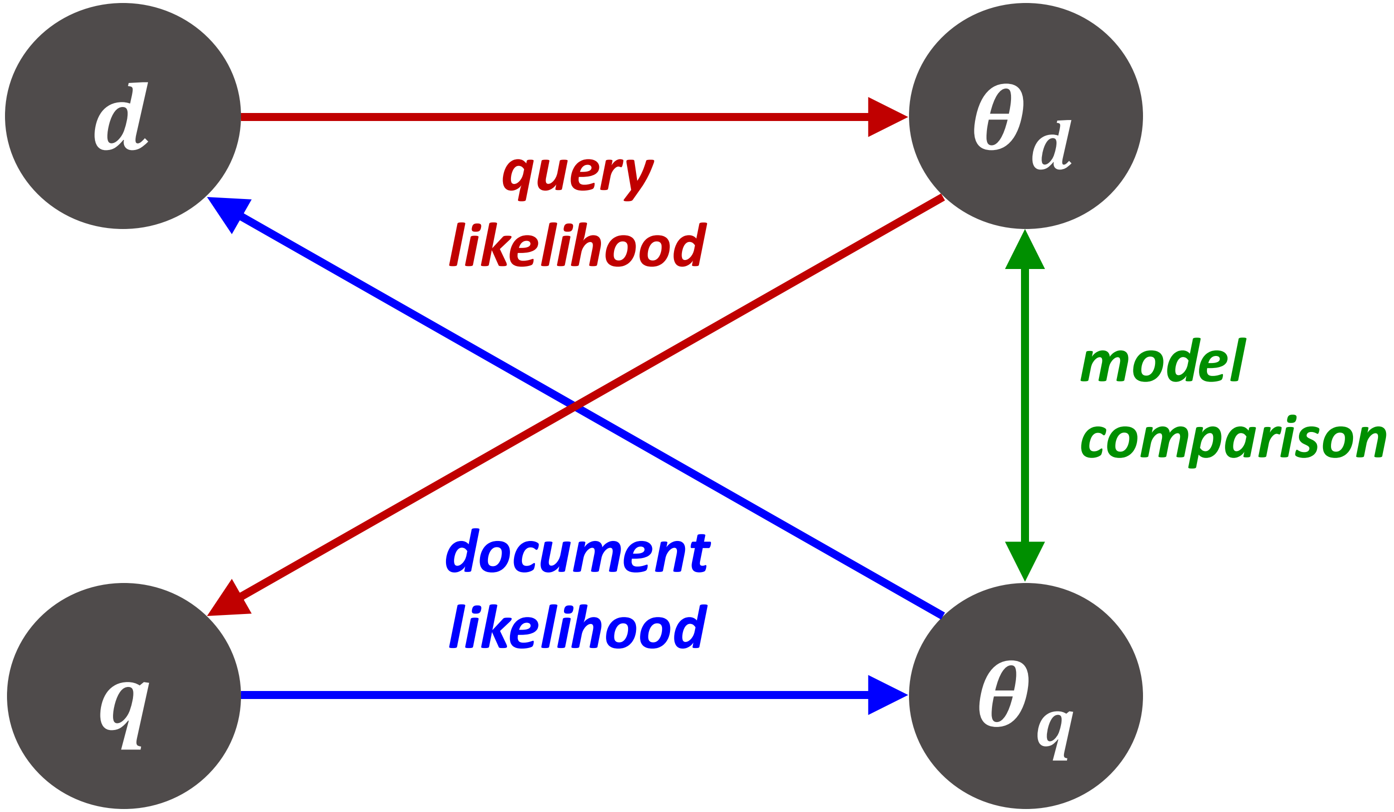
Query likelihood model

- $$\begin{aligned} f(q,d) &= P(q|\theta_d) \\ &\propto \log P(q|\theta_d) \\ &= \sum_{t\in q} \mathrm{tf}_{t,q} \log P(t|\theta_d) \end{aligned}$$

Difficulty

- Query as fixed sample (documents modeled instead)

Extended approaches



Relevance models

Language model representing information need

- Query and feedback documents are samples

$P(d|\theta_G)$: probability of generating the text in a document d given a relevance model θ_G

- Kind of document likelihood model (ext. of $P\left(d\middle|\theta_q\right)$)
- Less effective than query likelihood – hard to compare documents as samples with different lengths

Divergence-based ranking

Estimate relevance model from query and feedback

- Rank documents by similarity to relevance model

Kullback-Leibler divergence (KL-divergence)

- $f(q, d) = -D_{\mathrm{KL}}(\theta_G || \theta_d)$

Divergence-based ranking

$$f(q,d) = -D_{\mathrm{KL}}(\theta_G || \theta_d)$$

$$= -\sum_t P(t|\theta_G) \log \frac{P(t|\theta_G)}{P(t|\theta_d)}$$

$$= \sum_t P(t|\theta_G) \log P(t|\theta_d) - \underbrace{\sum_t P(t|\theta_G) \log P(t|\theta_G)}_{\textit{document independent}}$$

Divergence-based ranking

$$f(q,d) \propto \sum_t P(t|\theta_G) \log P(t|\theta_d)$$

- Without feedback, under MLE: $P(t|\theta_G) \propto \mathrm{tf}_{t,q}$
- Relevance model degenerates to query likelihood

Estimating relevance models

Probability of pulling a word t out of the "bucket" representing the relevance model depends on the query terms we have just pulled out

$$P(t|\theta_G) \approx P(t|q)$$

$$= \frac{P(t,q)}{P(q)}$$

Estimating relevance models

$$
\begin{aligned}
P(t, q) &= \sum_{d \in G} p(d) P(t, q|d) \\
&= \sum_{d \in G} p(d) P(t|q, d) P(q|d) \\
&\approx \sum_{d \in G} p(d) P(t|d) \prod_{t_i \in q} P(t_i|d)
\end{aligned}
$$

Estimating relevance models

$$P(t, q) \approx \sum_{d \in G} P(d) P(t|d) \prod_{t_i \in q} P(t_i|d)$$

Assuming uniform $P(d)$

- $P(t, q)$ is an average of query likelihood scores across feedback documents, weighted by $P(t|d)$

Example from top 10 docs

president lincoln	*abraham lincoln*	*fishing*	*tropical fish*
lincoln	lincoln	fish	fish
president	america	farm	tropic
room	president	salmon	japan
bedroom	faith	new	aquarium
house	guest	wild	water
white	abraham	water	species
america	new	caught	aquatic
guest	room	catch	fair
serve	christian	tag	china
bed	history	time	coral
washington	public	eat	source
old	bedroom	raise	tank
office	war	city	reef
war	politics	people	animal
long	old	fishermen	tarpon
abraham	national	boat	fishery

Example from top 50 docs

president lincoln	*abraham lincoln*	*fishing*	*tropical fish*
lincoln	lincoln	fish	fish
president	president	water	tropic
america	america	catch	water
new	abraham	reef	storm
national	war	fishermen	species
great	man	river	boat
white	civil	new	sea
war	new	year	river
washington	history	time	country
clinton	two	bass	tuna
house	room	boat	world
history	booth	world	million
time	time	farm	state
center	politics	angle	time
kennedy	public	fly	japan
room	guest	trout	mile

Summary

Acquiring feedback

- Explicit, implicit, simulated (pseudo) feedback

Exploiting feedback via query expansion

- Rocchio for VSM
- Feedback language models for LM

Challenges

Long queries are inefficient for typical search engines

- Only reweight certain prominent terms

Users are often reluctant to provide explicit feedback

- Effective pseudo-relevance feedback is challenging
- Implicit feedback is abundant, yet often biased

Feedback is also useful as a learning signal

References

Text Data Management: A Practical Introduction to Information Retrieval and Text Mining, Ch. 6
Zhai and Massung, 2016

Introduction to Information Retrieval, Ch. 9
Manning et al., 2008

Search Engines: Information Retrieval in Practice, Ch. 7
Croft et al., 2009

References

Relevance feedback in information retrieval
Rocchio, 1971

Relevance based language models
Lavrenko and Croft, SIGIR 2001

A survey of automatic query expansion in information retrieval
Carpineto and Romano, ACM Comp. Surveys 2012

UFMG UNIVERSIDADE FEDERAL DE MINAS GERAIS

Coming next...

Diversification Models

Rodrygo L. T. Santos
rodrygo@dcc.ufmg.br