IN4325

Query autocompletion and Interactive IR

Claudia Hauff (WIS, TU Delft)

The big picture

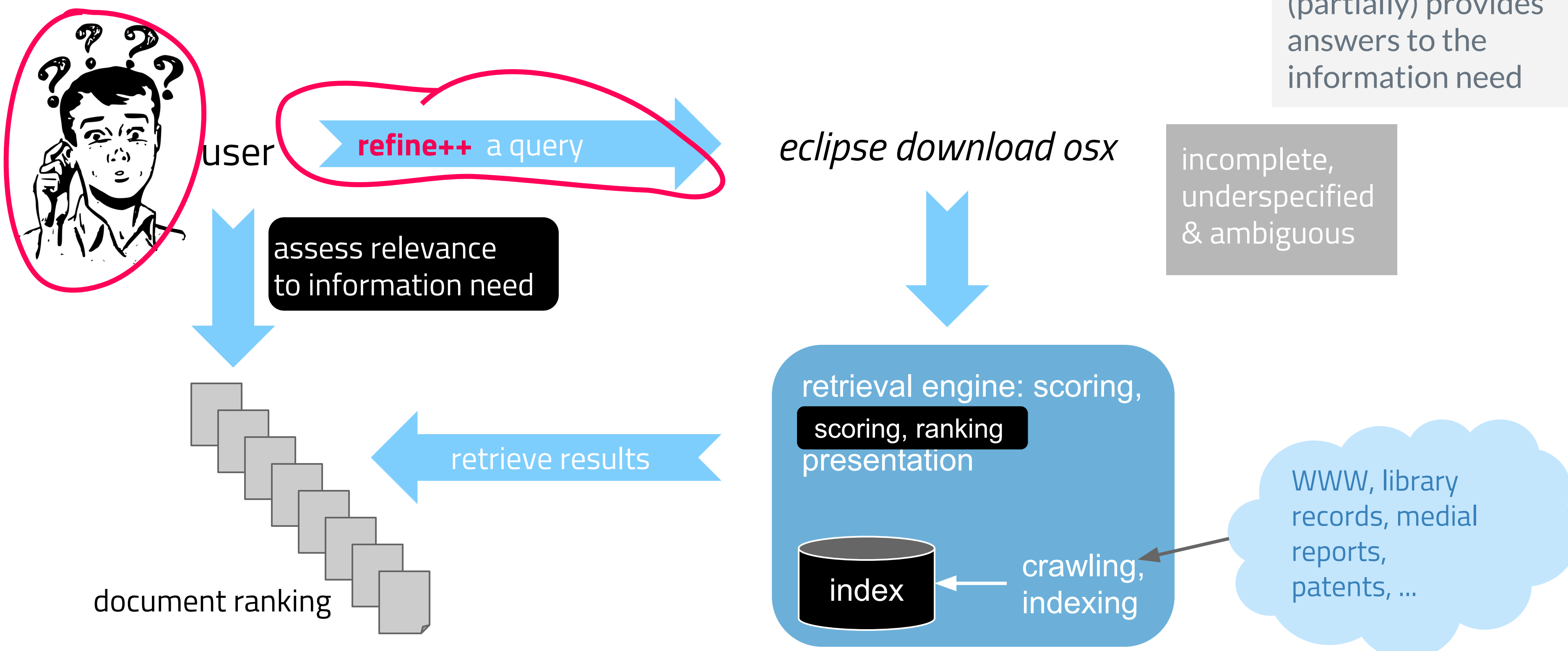
The essence of IR

Information need
Topic the user wants to know more about

Query
Translation of need into an input for the search engine

Relevance
A document is relevant if it (partially) provides answers to the information need

Information need: *Looks like I need Eclipse for this job. Where can I download the latest beta version for macOS Sierra?*



Query refinement

Query expansion
Pseudo-relevance feedback in LMs
Spell checking

Query autocompletion

Interactive query expansion

Select the term(s) to augment your original query with.

Query suggestions

Select the complete query to replace your original query with.

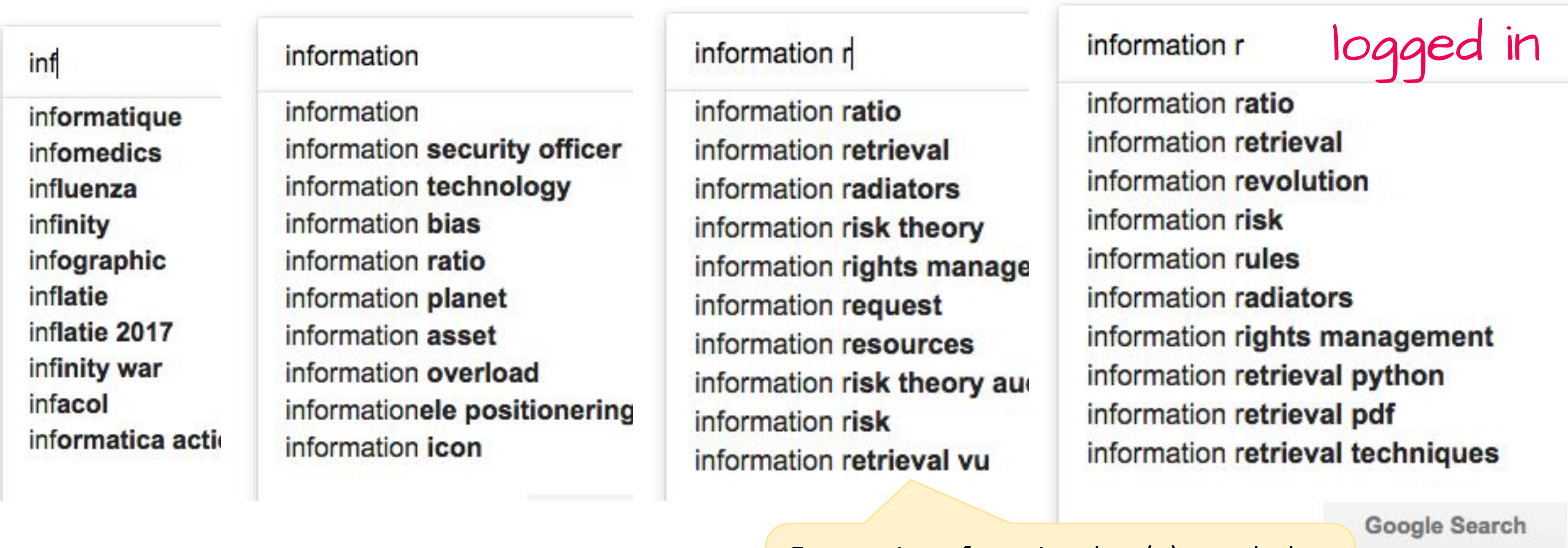
Query autocompletion

Select the complete query to replace your original query with whilst typing.

Related queries

Select the complete query to replace your original query with.

Overview



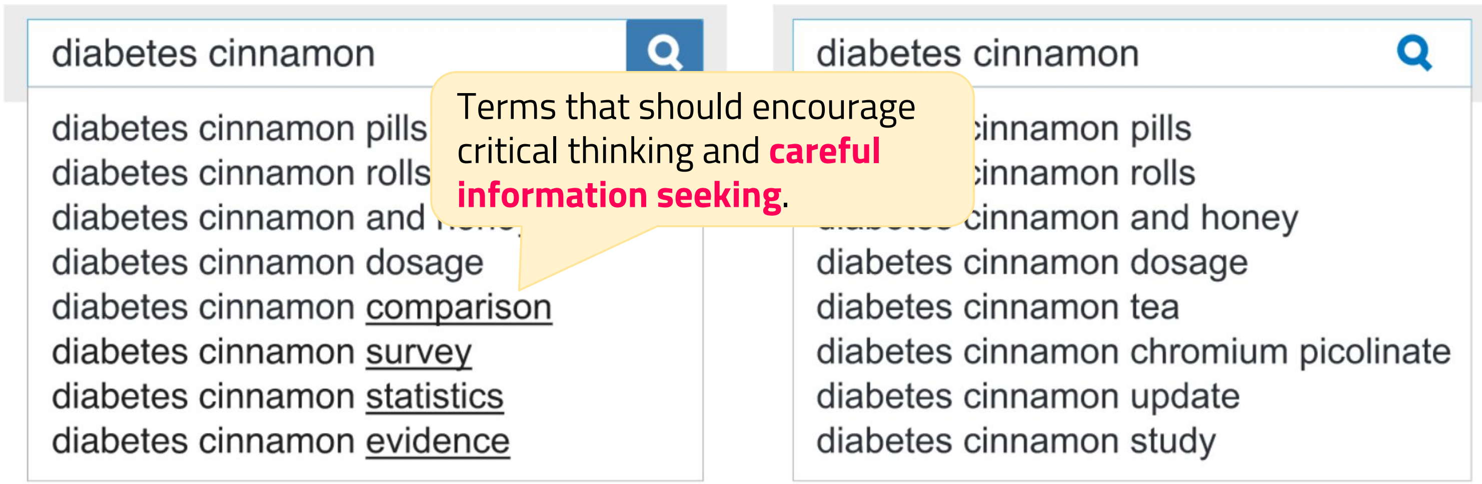
Suggestion of queries that (1) match the user's information needs and (2) yield a high-quality result ranking.

Goals:

1. Reduce query entry time
2. Prepare results in advance of query submission
3. Help users formulate a more precise query

Requires the search system to infer the user's *intent*.

Just released: query priming study



(1) QAC with query priming

(2) Conventional QAC

Findings:

1. With priming, users issue more queries
2. With priming, users (re)-visit the SERP more often
3. The priming effect varies relative to users' educational backgrounds (benefits highly educated users)

https://dl.acm.org/citation.cfm?id=3176377

Query-log based
Query autocompletion

Task

Given the current prefix (=query string the user has typed in so far), rank all possible candidates* (=complete queries).

Display the top ranked candidates to the user.

*assume for now that we have that list available

Two strong baselines

Assumptions:

1. Access to a query log and document clicks
2. Access to a corpus
3. Access to a user's past queries

Most popular ranker

Query candidates are ranked according to their past popularity

Clicked documents ranker

Cosine similarity between a user's profile (previously clicked docs by that user) and the candidate query profile (previously clicked docs across all users for that query) http://eprints.gla.ac.uk/105813/1/105813.pdf

Task

Mean reciprocal rank

Prefix length (#chars)

Approaches			MRR				
Ranking	Query-log Evidence	Personalized	2	4	6	8	10
Sentence occurrence ranker (SO)	No	No	0.005▼	0.0456▼	0.0696▼	0.1003▼	0.1546▼
Most Popular ranker (MP)	Yes	No	0.0964	0.2146	0.2851	0.3248	0.3641
Time Ranker (TR)	Yes	No	0.0324▼	0.1236▼	0.1995▼	0.2707▼	0.3281▼
Most Popular Time ranker (MT)	Yes	No	0.0961	0.2249▲	0.3112▲	0.3684▲	0.4153▲
Terms occurrence ranker (TO)	Yes	No	0.0021▼	0.0326▼	0.0773▼	0.1163▼	0.1617▼
Near Words Ranker (NW)	Yes	No	0.0611▼	0.1576▼	0.2347▼	0.2972▼	0.3611
String Similarity Ranker (SS)	No	Yes	0.0137▼	0.0711▼	0.1628▼	0.1149▼	0.2069▼
WordNet Similarity Ranker (WR)	Yes	Yes	0.089▼	0.0302▼	0.0711▼	0.0908▼	0.1055▼
N-Gram Similarity Ranker (NR)	Yes	Yes	0.0837	0.2927▲	**0.3693▲**	**0.4207▲**	**0.4602▲**
Kernel Similarity Ranker (KR)	Yes	Yes	0.907	0.2876▲	0.3356▲	0.3923▲	0.4121▲
Clicked Documents Ranker (CR)	Yes	Yes	**0.1442▲**	**0.2952▲**	0.3462▲	0.3938▲	0.4183▲

Table 1: Query auto-completion performance over the queries issued during the month of April '13 in our dataset, using the 11 presented ranking approaches. Statistically significant improvements/reductions in performance over the Most Popular ranker (MP) (p<0.05 paired t-test) are denoted ▲ and ▼, respectively.

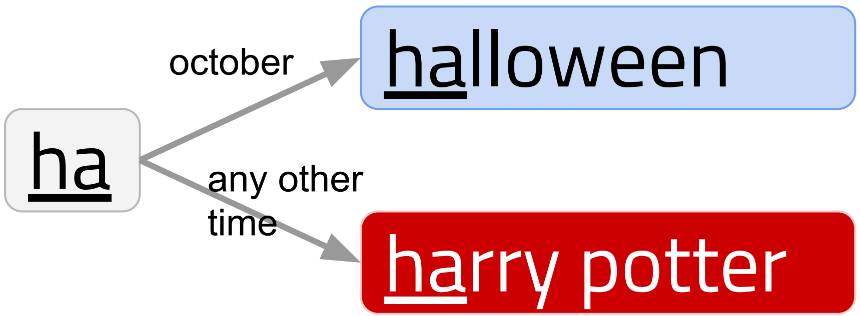
1,417,880 unique queries

November 2010 - March/April 2013

37,806 unique users

Medical search engine with 1.5M articles

Time-sensitive query autocompletion



Approach: apply time-series modeling and rank candidates according to their forecasted frequencies

https://dl.acm.org/citation.cfm?id=2806599

Rare prefixes

Query logs are a good source for *frequent* query prefixes.

The pool of candidate queries is usually drawn from a **pre-built prefix trie** (exact matching).

What happens if that does not yield any query candidates?

Idea: mine popular **query candidate suffixes** (popular n-grams) and generate **synthetic suggestion candidates** (prefix+suffix)that have never been observed in the log

https://dl.acm.org/citation.cfm?id=2806599

Rare prefixes

Query logs are a good s
prefixes.

The pool of candidate q
pre-built prefix trie (ex

What happens if that d
candidates?

Idea: mine popular **que**
n-grams) and generate
candidates (prefix+suff
observed in the log

what to cook with chicken and broccoli and
what to cook with chicken and broccoli *and bacon*
what to cook with chicken and broccoli *and noodles*
what to cook with chicken and broccoli *and brown sugar*
what to cook with chicken and broccoli *and garlic*
what to cook with chicken and broccoli *and orange juice*
what to cook with chicken and broccoli *and beans*
what to cook with chicken and broccoli *and onions*
what to cook with chicken and broccoli *and ham soup*

cheapest flights from seattle to
cheapest flights from seattle *to dc*
cheapest flights from seattle *to washington dc*
cheapest flights from seattle *to bermuda*
cheapest flights from seattle *to bahamas*
cheapest flights from seattle *to aruba*
cheapest flights from seattle *to punta cana*
cheapest flights from seattle *to airport*
cheapest flights from seattle *to miami*

https://dl.acm.org/citation.cfm?id=2806599

Rare prefixes: candidates generation

1. For each query in the query log, generate all possible n-grams from the end of the query

   amsterdam schiphol airport → airport, schiphol airport, amsterdam schiphol airport

2. Aggregate the n-grams across all queries and keep the most popular ones (precomputed)

AOL query log

Top suffixes	Top 2-word suffixes	Top 3-word suffixes
com	for sale	federal credit union
org	yahoo com	new york city
net	myspace com	in new york
gov	google com	or no deal
pictures	new york	disney channel com
lyrics	real estate	my space com
edu	of america	in new jersey
sale	high school	homes for sale
games	new jersey	department of corrections
florida	space com	chamber of commerce
for sale	aol com	bath and beyond
us	s com	in las vegas

3. For a given query prefix, extract the end-term

   + most popular ranker candidates

4. Match all suffixes that start with the end-term and create synthetic suggestion candidates

Rare prefixes: ranking features

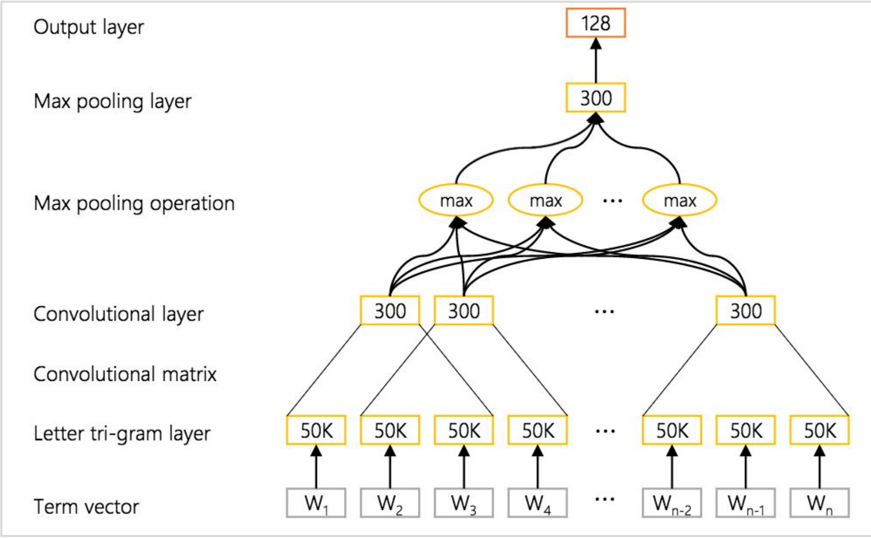
Supervised ranking model: features are computed for every query prefix and suggestion candidate (synthetic or previously observed); training data: [prefix,suggestion,judgment]

Main features for **LambdaMART**:

High-performing learning to rank approach

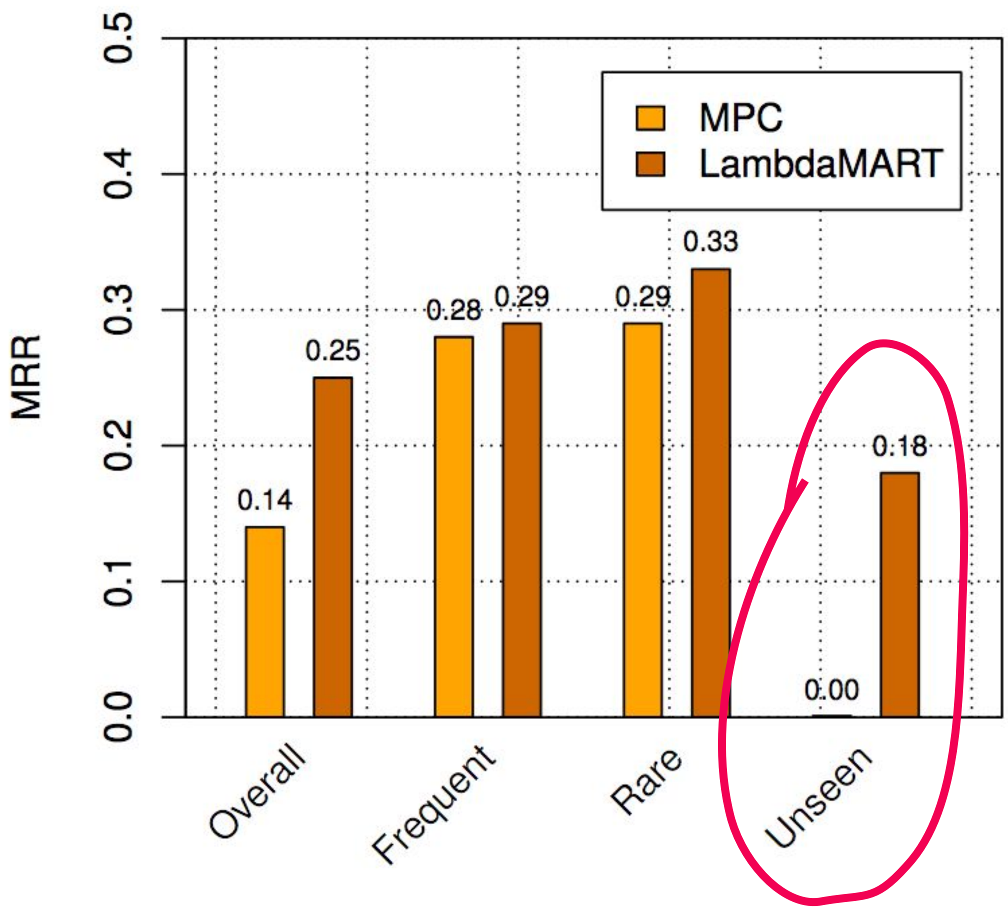
- Query log frequencies of N-grams appearing in a candidate suggestion
- **Convolutional latent semantic model** (training on prefix/suffix pairs generated from sampled queries)

$$clsmsim(\bar{p}, \bar{s}) = cosine(y_1, y_2) = \frac{y_1^\top y_2}{||y_1|| ||y_2||}$$

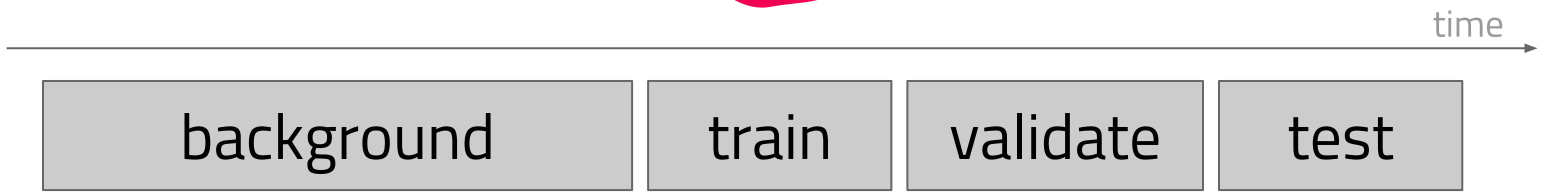


https://dl.acm.org/citation.cfm?id=2806599

Rare prefixes: results



Baseline: most popular Completion (MPC)



https://dl.acm.org/citation.cfm?id=2806599

Rare prefixes: results

Bing trade secrets

Models	AOL MRR	AOL % Improv.	Bing % Improv.
Full-query based candidates only			
MostPopularCompletion	0.1446	-	-
LambdaMART Model (n-gram features = no, CLSM feature = no)	0.1445	-0.1	-1.7*
LambdaMART Model (n-gram features = yes, CLSM feature = no)	0.1427	-1.4*	-1.2*
LambdaMART Model (n-gram features = no, CLSM feature = yes)	0.1445	-0.1	-1.2*
LambdaMART Model (n-gram features = yes, CLSM feature = yes)	0.1432	-1.0*	-1.5*
Full-query based candidates + Suffix based candidates (Top 10K suffixes)			
MostPopularCompletion	0.1446	-	-
LambdaMART Model (n-gram features = no, CLSM feature = no)	0.2116	+46.3*	+32.8*
LambdaMART Model (n-gram features = yes, CLSM feature = no)	0.2326	+60.8*	+42.6*
LambdaMART Model (n-gram features = no, CLSM feature = yes)	0.2249	+55.5*	+40.1*
LambdaMART Model (n-gram features = yes, CLSM feature = yes)	0.2339	**+61.7***	+43.8*

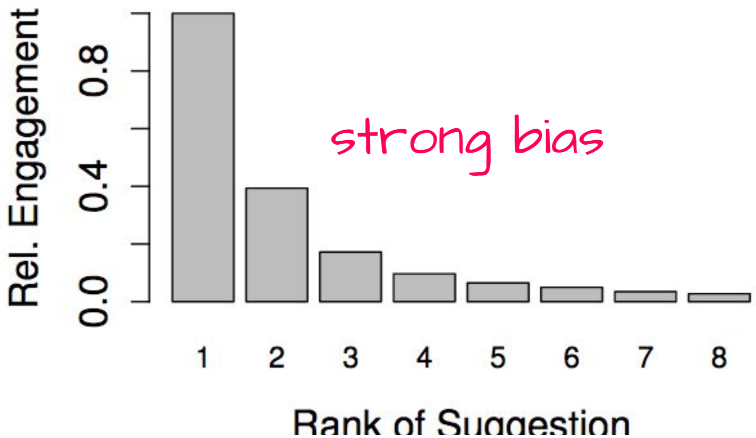
An **example** that shows how hard we (the IR community) have to work to yield significant gains from deep learning approaches. Gains are possible, but not guaranteed.

https://dl.acm.org/citation.cfm?id=2600428.2609508

User engagement with query autocompletion

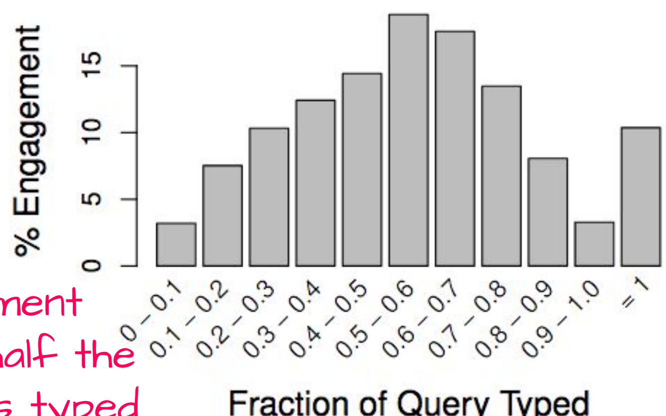
1.6M queries from Bing's search log

Most engagement at word boundaries and after 3 chars of typing a word

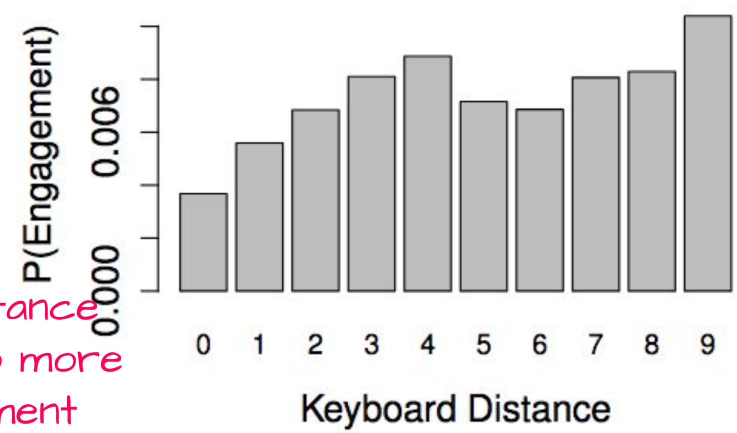


strong bias

Most engagement after half the query is typed



Larger key distance leads to more engagement



Cross-lingual IR: field of IR concerned with the retrieval of documents in a language different from the query language

Cross-lingual query suggestions: suggest queries in a different language from the original query

Web search engines are not everything …

Large user base

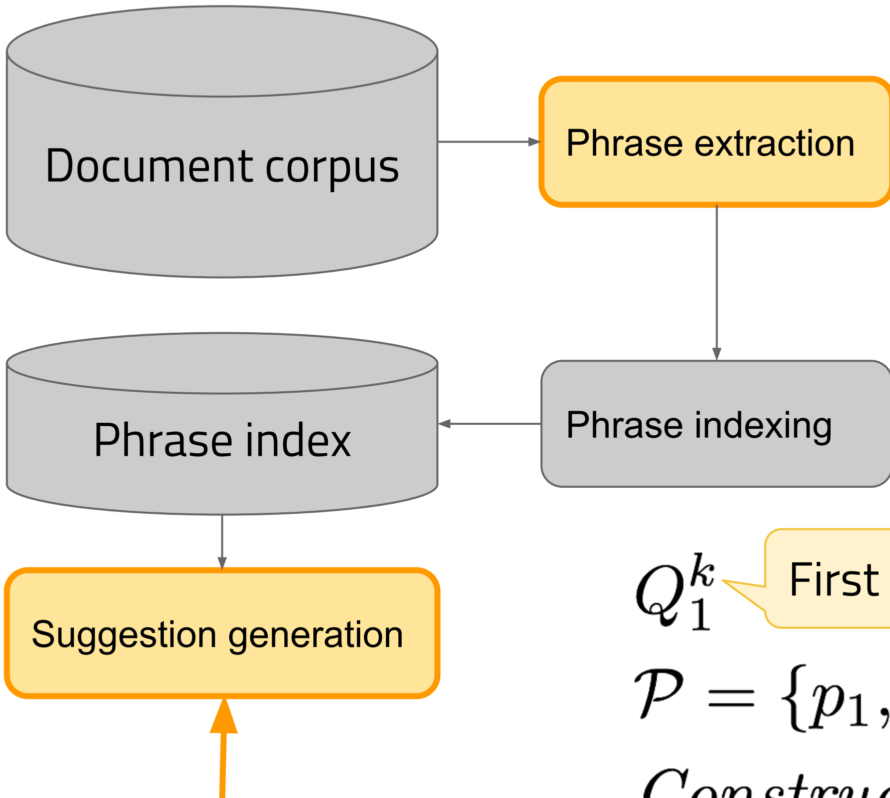
Assumptions:

1. **Access to a query log and document clicks**
2. Access to a corpus *always possible*
3. **Access to a user's past queries**

What about search in specialized domains or personal search systems (PIM)?

Corpus-based Query autocompletion

Corpus-based query suggestions



- N-grams: unigrams, bigrams, trigrams
- Ignore N-grams starting with a stopword

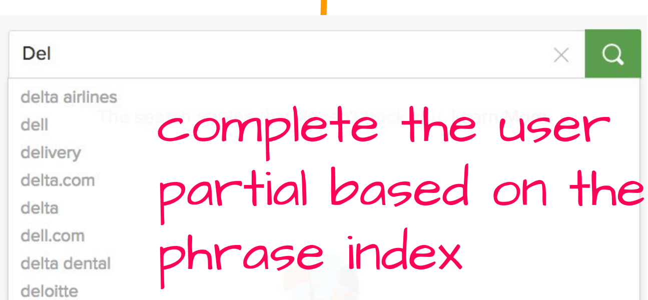
Q_1^k First k characters typed

$\mathcal{P} = \{p_1, .., p_n\}$ Set of all extracted phrases

$$Construct\ S \subset \mathcal{P},\ such\ that\ each$$

$$s \in S\ is\ a\ possible\ completion\ of\ Q_1^k$$

$$P(p_i|Q_1^k)$$



complete the user partial based on the phrase index

https://dl.acm.org/citation.cfm?id=2010023

Corpus-based query suggestions

$$P(p_i|Q_1^k)$$

Probability that the user will type p_i given her first k typed characters

https://dl.acm.org/citation.cfm?id=2010023

Corpus-based query suggestions

$$P(p_i|Q_1^k)$$

Probability that the user will type p_i given her first k typed characters

$$Q_1^k = \underline{Q_c} + \underline{Q_t}$$

Completed word(s) plus word the user is currently typing

https://dl.acm.org/citation.cfm?id=2010023

Corpus-based query suggestions

$$P(p_i|Q_1^k)$$

Probability that the user will type p_i given her first k typed characters

$$Q_1^k = \underline{Q_c} + \underline{Q_t}$$

Completed word(s) plus word the user is currently typing

$$P(p_i|Q_1^k) = \frac{P(p_i) \times P(Q_1^k|p_i)}{P(Q_1^k)}$$

according to Bayes' theorem

https://dl.acm.org/citation.cfm?id=2010023

Corpus-based query suggestions

$P(p_i|Q_1^k)$ Probability that the user will type p_i given her first k typed characters

$Q_1^k = \underline{Q_c} + \underline{Q_t}$ Completed word(s) plus word the user is currently typing

$$P(p_i|Q_1^k) = \frac{P(p_i) \times P(Q_1^k|p_i)}{P(Q_1^k)}$$ according to Bayes' theorem

$$= \frac{P(p_i) \times P(Q_t|p_i) \times P(Q_c|p_i)}{P(Q_1^k)}$$ Simplifying assumption: conditional independence

https://dl.acm.org/citation.cfm?id=2010023

Corpus-based query suggestions

$$P(p_i|Q_1^k)$$

Probability that the user will type p_i given her first k typed characters

$$Q_1^k = \underline{Q_c} + \underline{Q_t}$$

Completed word(s) plus word the user is currently typing

$$P(p_i|Q_1^k) = \frac{P(p_i) \times P(Q_1^k|p_i)}{P(Q_1^k)}$$

according to Bayes' theorem

$$= \frac{P(p_i) \times P(Q_t|p_i) \times P(Q_c|p_i)}{P(Q_1^k)}$$

$$= \frac{P(Q_t) \times P(p_i|Q_t) \times P(Q_c|p_i)}{P(Q_1^k)}$$

https://dl.acm.org/citation.cfm?id=2010023

Corpus-based query suggestions

$$P(p_i|Q_1^k)$$

Probability that the user will type p_i given her first k typed characters

$$Q_1^k = \underline{Q_c} + \underline{Q_t}$$

Completed word(s) plus word the user is currently typing

$$P(p_i|Q_1^k) = \frac{P(p_i) \times P(Q_1^k|p_i)}{P(Q_1^k)}$$

according to Bayes' theorem

$$= \frac{P(p_i) \times P(Q_t|p_i) \times P(Q_c|p_i)}{P(Q_1^k)}$$

$$P(p_i)P(Q_t|p_i)$$

$$= P(p_i, Q_t)$$

$$= P(Q_t)P(p_i|Q_t)$$

$$= \frac{P(Q_t) \times P(p_i|Q_t) \times P(Q_c|p_i)}{P(Q_1^k)}$$

https://dl.acm.org/citation.cfm?id=2010023

Corpus-based query suggestions

$$P(p_i|Q_1^k)$$

Probability that the user will type p_i given her first k typed characters

$$Q_1^k = \underline{Q_c} + \underline{Q_t}$$

Completed word(s) plus word the user is currently typing

$$P(p_i|Q_1^k) = \frac{P(p_i) \times P(Q_1^k|p_i)}{P(Q_1^k)}$$

according to Bayes' theorem

$$= \frac{P(p_i) \times P(Q_t|p_i) \times P(Q_c|p_i)}{P(Q_1^k)}$$

$$= \frac{\cancel{P(Q_t)} \times P(p_i|Q_t) \times P(Q_c|p_i)}{\cancel{P(Q_1^k)}}$$

Remains static for all p_i

$$\stackrel{rank}{=} P(p_i|Q_t) \times P(Q_c|p_i)$$

https://dl.acm.org/citation.cfm?id=2010023

Corpus-based query suggestions

$$P(p_i|Q_1^k) \stackrel{rank}{=} P(p_i|Q_t) \times P(Q_c|p_i)$$

Phrase that contains the completed word c_i

Phrase selection probability

Phrase-query correlation
`bill gate*` vs. `india gate*`
Context is needed!

$$P(p_{ij}|Q_t) = P(c_i|Q_t) \times P(p_{ij}|c_i)$$

Term completion probability; c_i is a possible word completion

Term to phrase probability

$$P(Q_c|p_i) = \frac{P(Q_c, p_i)}{P(p_i)}$$

Estimated based on corpus statistics; to avoid data sparseness, we simplify to the bag of words approach, i.e. search queries `linux install firefox` `install firefox linux` `firefox install linux` are treated in the same way.

Assumption: phrases in the corpus that are more important have a higher chance of being used by the user for querying. Estimated based on corpus statistics.

Corpus-based query suggestions

Data sets

TREC: 200K news articles by the Financial Times published between 1991-1994

Ubuntu: 100K discussion threads crawled from ubuntuforums.org

Given a complete query, retain only the first keyword (Type-A) or the first keyword plus $k>2$ characters (Type-B)

Baseline

SimSearch: search the phrase index for all phrases containing the partial user query; rank them in order of decreasing corpus frequency

`Radioactive waste`
(TREC Topic 387)

`Radioactive`
(Type-A)

`Radioactive was`
(Type-B)

https://dl.acm.org/citation.cfm?id=2010023

What are good metrics?

Corpus-based query suggestions

Data sets

TREC: 200K news articles by the Financial Times published between 1991-1994

Baseline

SimSearch: search the phrase index for all phrases containing the partial user query; rank them in order of decreasing corpus frequency

presented approach

Query = mount			Query = falkland		
SimSearch	**CompSearch**	**Prob**	**SimSearch**	**CompSearch**	**Prob**
mount	mount	mount	falklands	falklands	falklands
mounted	mounted	unable to mount	falkland	falkland	falklands war
mounting	mounting	mount point type	falkland islands	falklanders	falkland islands
mounts	mounts	sudo mount	falklands war		falklands conflict
sudo mount	mountpoint	able to mount	falklands conflict		1982 falklands
unable to mount	mountcifs	mountpoint	1982 falklands		1982 falklands conflict
system mount	mountable	try to mount	falkland islands government		falkland islands government
file system mount	mounter	mount the drive	1982 falklands conflict		falklands war in 1982
mount point type	mountunmount	mount the partition	falkland arms		1982 falklands war
system mount point type	mountpoints	file system mount	falklanders		invasion of the falklands

https://dl.acm.org/citation.cfm?id=2010023

What are other options besides the generic document corpus frequencies?

Corpus-based query suggestions

presented approach

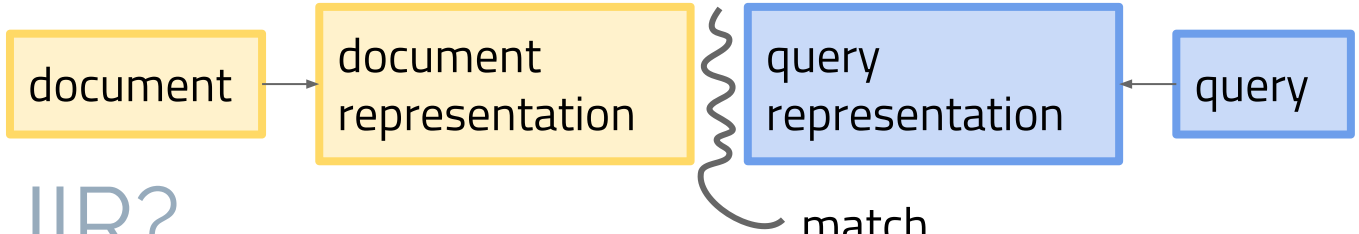
Ubuntu			
	SimSearch	CompSearch	Probabilistic
Type-A	1.00	1.00	1.00
Type-B	0.75	1.00^s	1.00^s
Overall	0.875	1.00^s	1.00^s
TREC			
	SimSearch	CompSearch	Probabilistic
Type-A	1.00	1.00	1.00
Type-B	0.15	0.95^S	1.00^S
Overall	0.575	0.975^S	1.00^S

Success rate: at least one meaningful suggestion for the partial query

Table 4: Success Rate of different query suggestion methods for the two datasets. Superscripts s and S indicate statistically significant improvements over SimSearch with $p < 0.05$ and $p < 0.01$, respectively (one-tailed t-test).

Interactive Information Retrieval

"classic" IR model



What is IIR?

*"The area of interactive information retrieval covers research related to **studying** and **assisting** these diverse end users of information access and retrieval systems."*
(Ian Ruthven)

*"In interactive information retrieval, **users are typically studied** along with their interactions with systems and information."*
(Diane Kelly)

*"... the interactive approach to IR has led to a **focus on the user-oriented activities** of query formulation and reformulation, and inspection and judgement of retrieved items ..."* (Nick Belkin)

From past to present

Many (many!) models have been proposed over the years. This is only a small selection.

Conceptual, observational and empirical work

Bates' berrypicking

Kuhlthau's ISP

Fuhr's IPRP

approximately equivalent

Search Economic Theory

Mathematical models of information seeking and search

- Observe users
- Propose a model that *describes* the observations well and has intuitive appeal

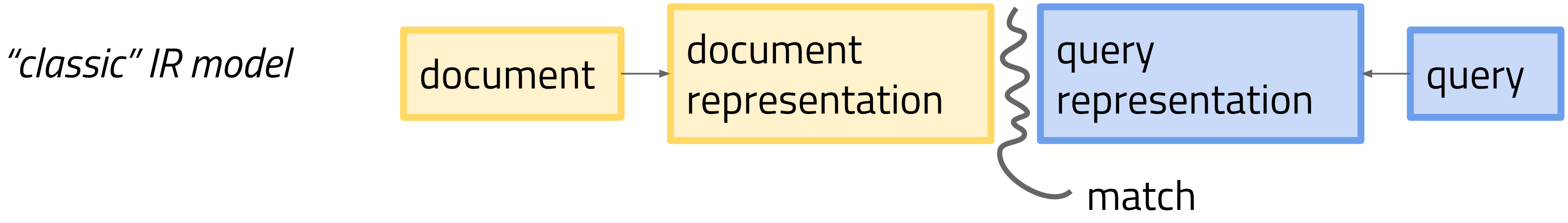
- Narrow down the 'search space' of **testable hypotheses**
- Pick the most promising hypotheses
- Design & execute user studies to (in)validate the hypotheses

Most often in IR when we talk about models we mean retrieval models.

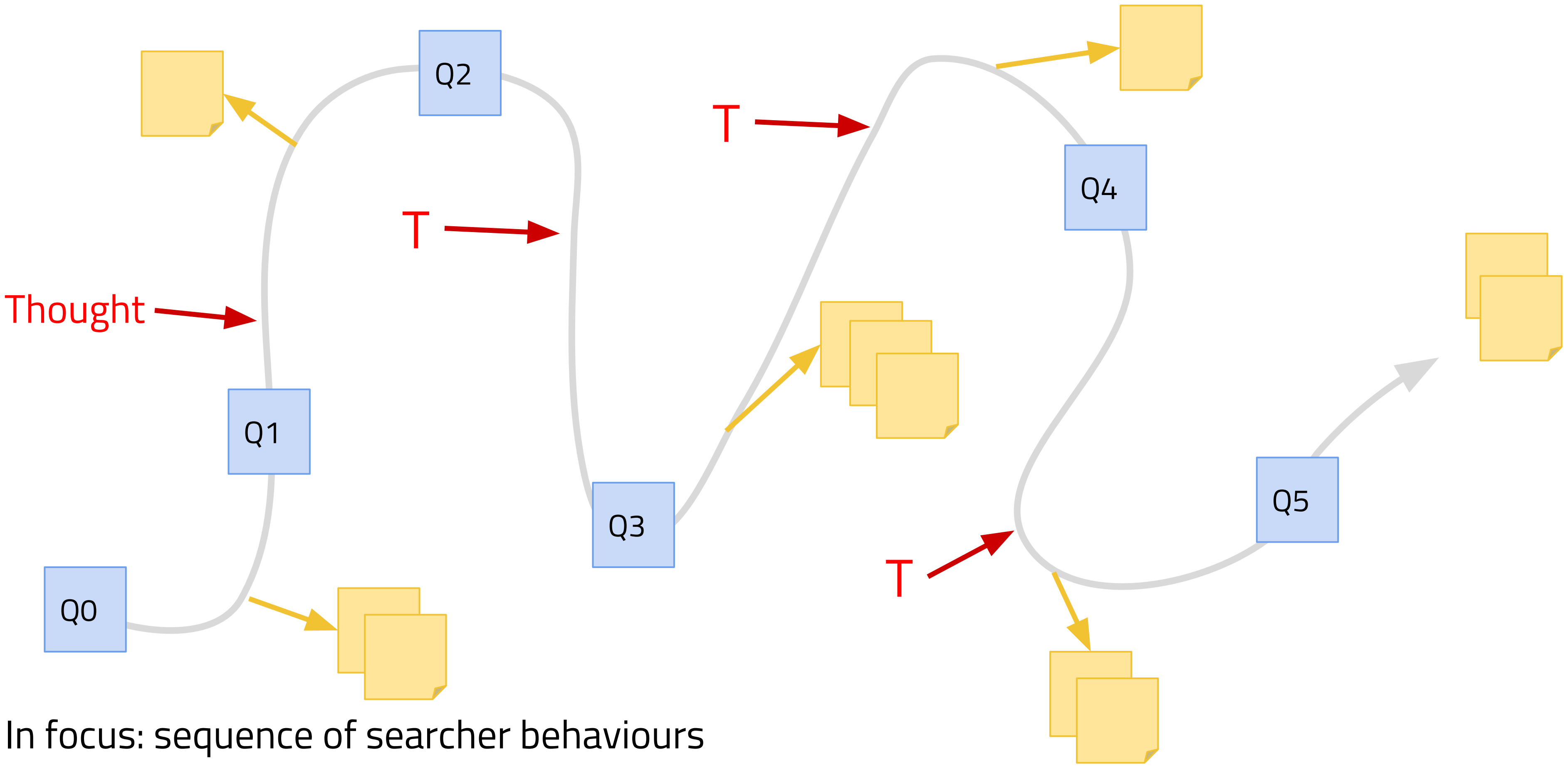
Not now though!

Now: models for interactive information seeking and retrieval

Two early models of IIR



Bates' berrypicking model (1989)



In focus: sequence of searcher behaviours
Based on intuitions, informal observations

https://www.emeraldinsight.com/doi/pdfplus/10.1108/eb024320

Bates' berrypicking model (1989)

- Information needs **evolve over time**, they are not static throughout the search

- Users frequently start their search with just one sub-topic of a broader topic

- Each found piece of information can result in new ideas and search directions

- A query is not satisfied by a final retrieved set of documents, but by a **series of selections** of bits of information at each stage of the evolving search

bit-at-a-time retrieval ***= berrypicking***

Kuhlthau's Information Search Process model (1988)

Model designed based on **observations** of **high school students**' application of library skills (i.e. qualitative research)

Motivation: "*Findings are needed that define the **experience** of people in an information search from their **own perspective**.*"

Systematic development of theory

Goal: **grounded theory** of the library search process

flickr@juanillooo

Kuhlthau's Information Search Process model (1988)

Exploratory study based on:

- Observations in the natural setting (school library)
- Interviews (45 minutes)
- Journals (diaries)
- Search logs
- Time lines
- Flow charts
- Assessed writing probes

Participants: 26 college-bound high school seniors

Assignment: write a paper

Describe how you felt when the teacher announced the research assignment.

Describe how and why you chose your topic.

How did you know when your search was completed?

What did you find most difficult about your search?

Kuhlthau’s Information Search Process model (1988)

Six stages

	Task initiation	Topic selection	Prefocus exploration	Focus formulation	Information collection	Search closure
Feelings (affective)	uncertainty	optimism	confusion, frustration, doubt	clarity	sense of direction, confidence	relief
Thoughts (cognitive)		ambiguity →	→	→	specificity	
				increased interest →	→	
Actions (physical)	seeking relevant information →	→	→	→	→	seeking pertinent information

Kuhlthau's Information Search Process model (1988)

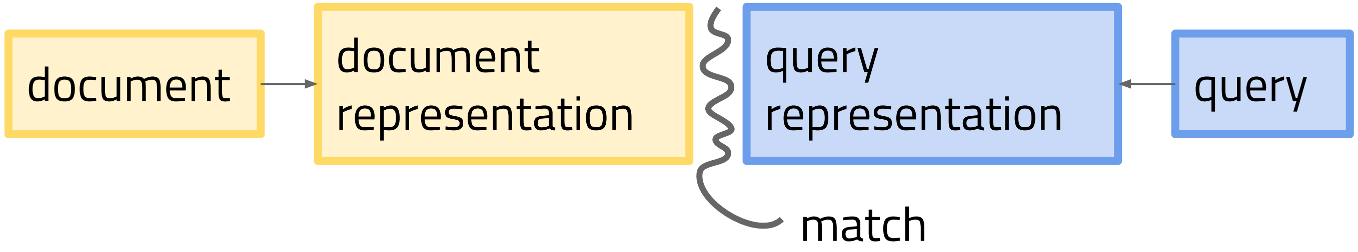
Six stages

	Task initiation	Topic selection	Prefocus exploration	Focus formulation	Information collection	Search closure
Feelings (affective)	uncertainty	optimism	confusion, frustration, doubt	clarity	sense of direction, confidence	relief
Thoughts (cognitive)		iguity →	→	→	specificity	
				increased interest →	→	
Actions (physical)	seeking relevant information →	→	→	→	→	seeking pertinent information

Hidden from any IR system we know today ...

One of today's prevalent IIR modeling approaches

"classic" IR model



Predictive models are needed

- Observational studies and descriptive models allow us to think but not to reason about interactive IR design decisions

e.g. is it better to show 20 query autocompletion items or just 3?

- Interactive IR experiments have shown that system effectiveness and **user performance** do not necessarily correlate

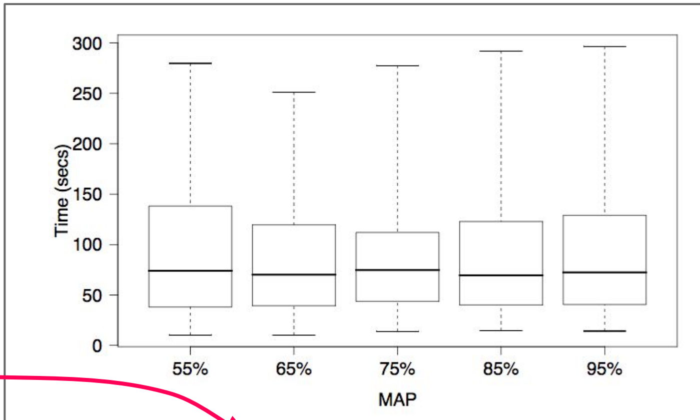


Figure 3: Time taken to find the first relevant document versus the mean average precision of the system used.

space of all possible UI changes

UIs predicted to be useful by a model

Economic models of interaction

(Azzopardi et al, 2011-today)

Focus on understanding/predicting the behaviour of economic agents within an environment.

Economics is a field ripe with predictive models of costs and benefits; can we make use of them?

User interactions re-interpreted:

- Users take **actions** to advance towards their **goals**
- Each action has a **cost** (time, effort, cognitive load, etc.)
- An action may or may not lead to a **benefit** (saving time, finding new information, etc.)

flickr@arabani

Economic models of interaction

(Azzopardi et al, 2011-today)

Representation of reality in an abstracted form; requires assumptions.

Having formulated a **mathematical model**, we can examine what actions:

- accrue the **most benefits** for a given cost
- incur the **least cost** for a given benefit level
- a rational user should take (given a task, interface, context, constraints) to achieve **optimal** results

flickr@arabani

Economic models of interaction

(Azzopardi et al, 2011-today)

Assumptions:

- Economic agents are **rational** and attempt to maximize their benefits
- Economic agents can **adapt** their strategies towards the optimal course of interaction

Tutorial: http://zuccon.net/publications/azzopardi-zuccon-2017-tutorial-economics.pdf

Let's look at two IR examples!

Building economic models

iterate

1. Describe the problem context (who/what/how)
2. Specify the cost and benefit functions (keep it simple and then refine)
3. Solve the model (analytically, computationally, or graphically)
4. Use the model to generate hypotheses about behaviours (how do different variables influence interaction and behaviour)
5. Compare the predictions with observations in the literature and/or experimental data (model as a guide and evidence that [in]validates our models, leading to refinement)

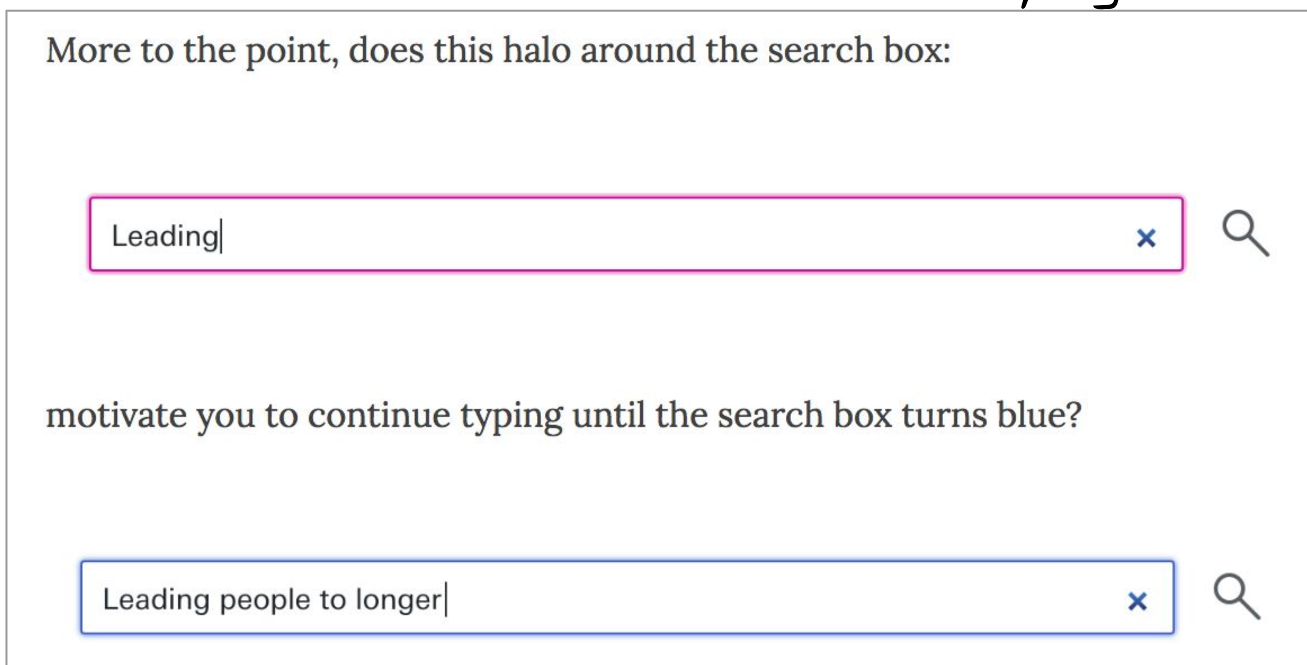
Economic model of querying

Goal: a model that describes the relationship between the length of the query and the costs/benefits of the query given its length

How about trying this?

Longer queries tend to lead to better results; users do not use long queries.

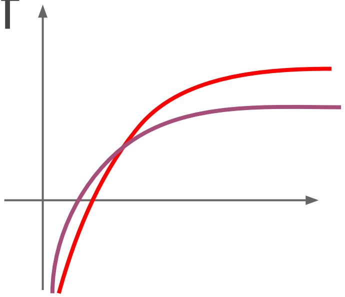
Can we incentivize them?



Economic model of querying

Goal: a model that describes the relationship between the length of the query $\mathbf{W}$ (in words) and the costs/benefits of the query given its length.

Modeling assumption: cost/benefit are a function of query length alone.



$$\underbrace{b(\mathbf{W})}_{\text{benefit function}} = \mathbf{k} \times \log_a(\mathbf{W} + \mathbf{1})$$

benefit function

Diminishing returns ($\mathbf{a}$ determines steepness) as the length increases with $\mathbf{k}$ as scaling factor (e.g. SE quality).

$$c(\mathbf{W}) = \mathbf{W} \times \mathbf{c_w}$$

cost function
(i.e. the effort in querying)

Effort to enter one word.

Economic model of querying

Given the cost and benefit functions, we can compute the **profit (net benefit)** that the user receives for a query of length $\mathbf{W}$:

$$\pi = b(\mathbf{W}) - c(\mathbf{W}) = \mathbf{k} \times log_a(\mathbf{W} + \mathbf{1}) - \mathbf{W} \times \mathbf{c_w}$$

Which query length maximizes the user's net benefit? Differentiate with respect to $\mathbf{W}$ and solve:

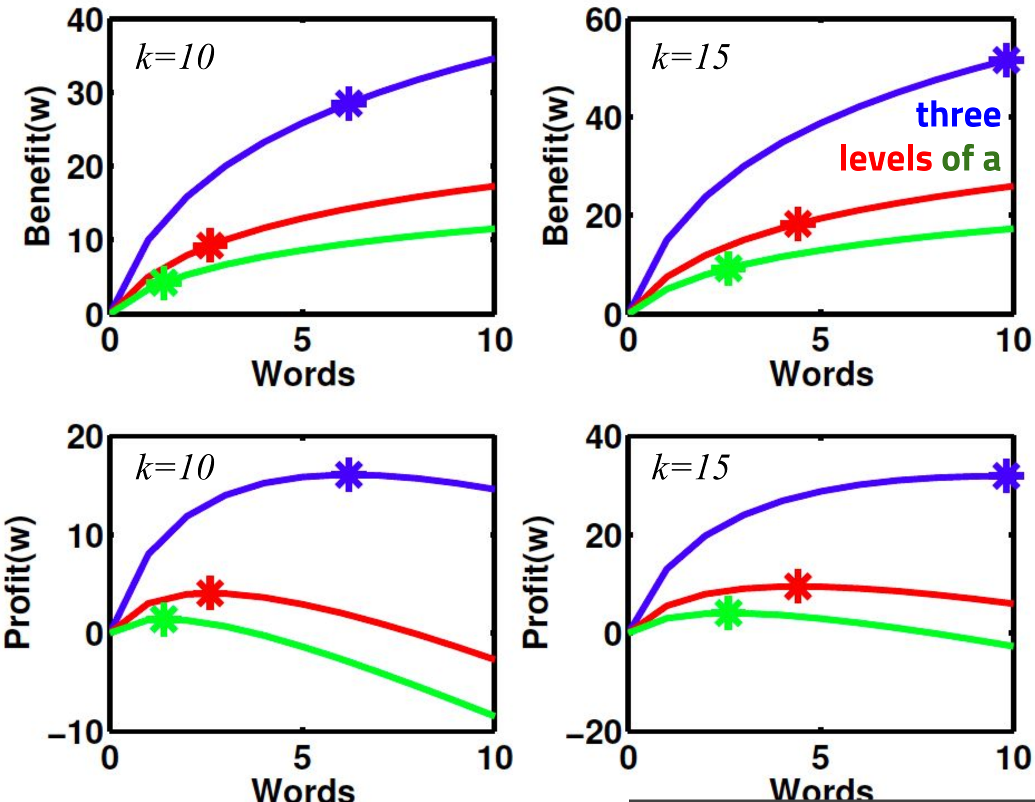
$$\frac{\partial \pi}{\partial \mathbf{W}} = \frac{\mathbf{k}}{\log \mathbf{a}} \times \frac{\mathbf{1}}{\mathbf{W} + \mathbf{1}} - \mathbf{c_w} = 0$$

$$\mathbf{W}^* = \frac{\mathbf{k}}{\mathbf{c_w} \times \log \mathbf{a}} - \mathbf{1}$$

Economic model of querying

$$\mathbf{W}^* = \frac{\mathbf{k}}{\mathbf{c_w} \times \log \mathbf{a}} - \mathbf{1}$$

What does the model say about:
query halo effect
query autocompletion
SE with AND between query terms



Hypotheses based on this model:

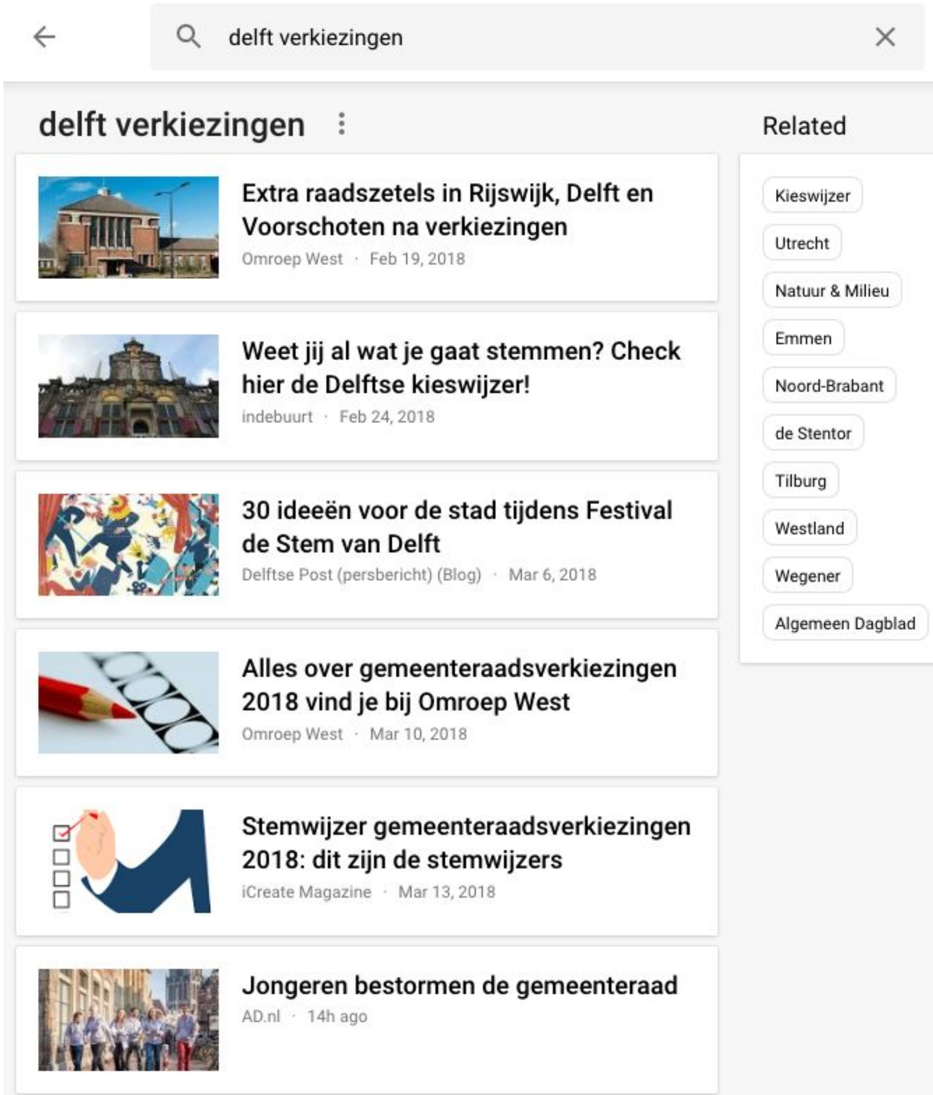
- As the system performance ($\mathbf{k}$) increases, the query length increases
- If additional terms provide less and less benefit ($\mathbf{a}$ increases), queries decrease in length
- With decreasing cost of entering a word ($\mathbf{c_w}$), users tend to pose longer queries

Economic model of assessing

Goal: a model that describes how users interact with a list of search results after having posed a query. Also known as “stopping behaviour”.

Empirical findings: users stop when having found ‘enough’ or after N relevant docs or …

Example: news retrieval



Economic model of assessing

Goal: a model that describes how users interact with a list of search results after having posed a query. Also known as “stopping behaviour”.

Modeling assumption: a user interacts with one list of results.

cost of assessing $\mathbf{A}$ items

cost of the query

cost of assessing 1 doc.

Cost function: $$c(\mathbf{A}) = \mathbf{c_q} + \mathbf{A} \times \mathbf{c_a}$$

Benefit function: $$b(\mathbf{A}) = \mathbf{k} \times \mathbf{A}^{\beta}$$

Determines how quickly the benefit from information diminishes
$\beta < 1$ *usually*

Economic model of assessing

Given the cost and benefit functions, we can compute the **profit (net benefit)** the user receives when assessing to a depth of $\mathbf{A}$:

$$\pi = b(\mathbf{W}) - c(\mathbf{W}) = \mathbf{k} \times \mathbf{A}^{\beta} - \mathbf{c_q} - \mathbf{A} \times \mathbf{c_a}$$

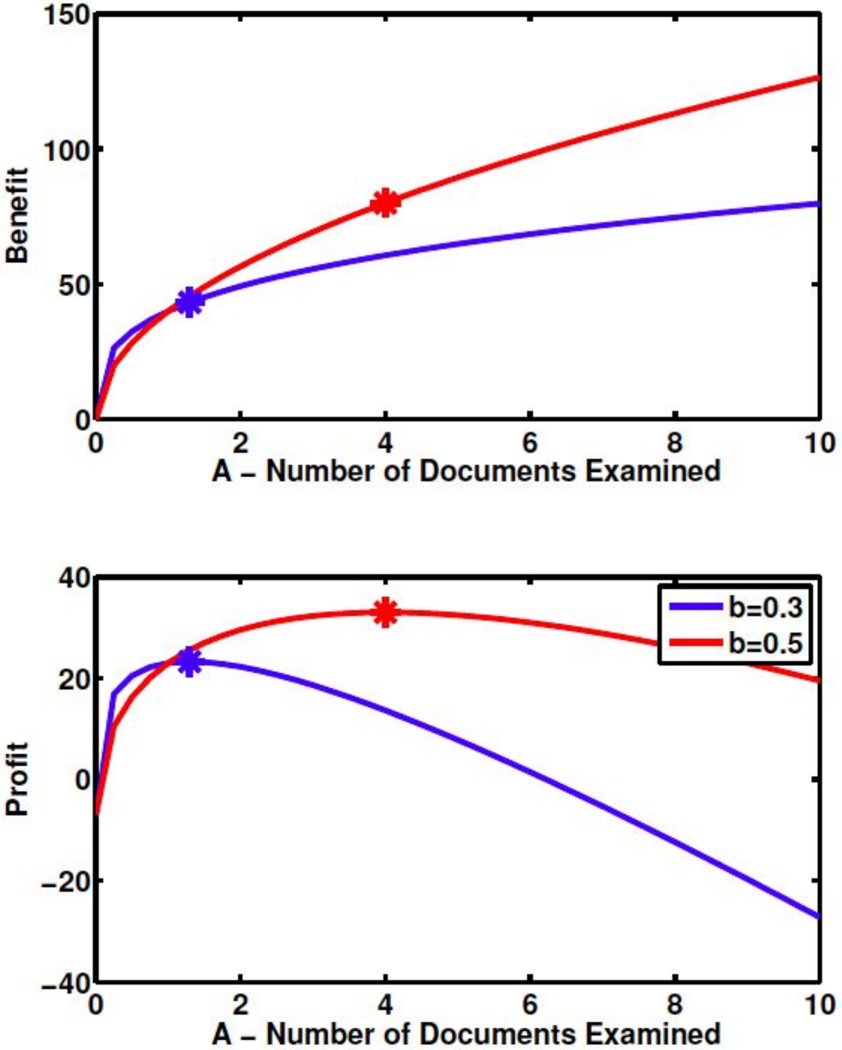
Differentiate with respect to $\mathbf{A}$ and solve:

$$\frac{\partial \pi}{\partial \mathbf{A}} = \mathbf{k} \times \beta \times \mathbf{A}^{\beta - 1} - \mathbf{c_a} = 0$$

$$\mathbf{A}^* = \left(\frac{\mathbf{c_a}}{\mathbf{k} \times \beta} \right)^{\frac{1}{\beta - 1}}$$

Economic model of assessing

$$\mathbf{A}^* = \left(\frac{\mathbf{c_a}}{\mathbf{k} \times \beta}\right)^{\frac{1}{\beta - 1}}$$



Model interpretation:

- If the performance of the query is poor, there is little incentive to examine search results.
- If the cost of assessing documents is very high, fewer documents are examined.
- The cost of a query does not impact user behaviour (as it is a fixed cost).

Economic model of searching

Goal: a model that describes the process of searching over a session - numerous queries can be issued, the user examines a number of items per query.

It gets more complicated quickly ...

$$c(\mathbf{Q}, \mathbf{V}, \mathbf{S}, \mathbf{A}) = \mathbf{c_q}.\mathbf{Q} + \mathbf{c_v}.\mathbf{V}.\mathbf{Q} + \mathbf{c_s}.\mathbf{S}.\mathbf{Q} + \mathbf{c_a}.\mathbf{A}.\mathbf{Q}$$

A user poses a number of queries

... examines a number of SERPs per query

... examines a number of snippets per query

Take-away message: models can be as simple/complex as desired.

Economic models of interaction

(Azzopardi et al., 2011-today)

Challenges:

- **Estimation of costs and benefits** and their respective units (temporal, fiscal, satisfaction, enjoyment, ...)
- Assumption that users seek to max. their benefit
- Is the model sufficiently realistic wrt. user and environment?
- **Design of experiments**

That's it for today!

Don't forget that milestone M4 (March 19) **is coming up Monday.**

Slack: in4325.slack.com

Email: in4325-ewi@tudelft.nl