

# Applying the User-over-Ranking Hypothesis to Query Formulation

Matthias Hagen   Benno Stein

Bauhaus-Universität Weimar  
matthias.hagen@uni-weimar.de

ICTIR 2011  
Bertinoro, Italy  
September 14, 2011

What is the User-over-Ranking hypothesis?

Queries returning as many results as the user can consider increase retrieval performance.

Queries returning as many results as the user can consider increase retrieval performance.



Fine print: If ranking works: great!  
Use case is not some query like ebay.  
But more involved information needs,  
automatic systems, etc.

# Assumption 1: More keywords = more specific

The screenshot shows a Google search interface. At the top, there are navigation links for Web, Images, Videos, Maps, News, Shopping, Mail, and more. The Google logo is on the left. The search bar contains the text "information retrieval" and shows "About 17,200,000 results (0.14 seconds)". A red box highlights the number "17,200,000". To the right of the search bar is a blue search button with a magnifying glass icon and a link to "Advanced search".

On the left side, there is a vertical menu with icons and labels: Everything, Images, Videos, News, Shopping, Blogs, Books, and Discussions.

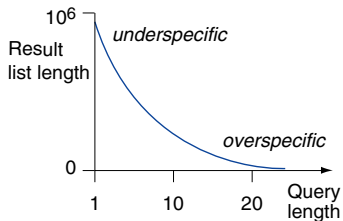
The search results are as follows:

- ▶ [Information retrieval - Wikipedia, the free encyclopedia](#)   
[en.wikipedia.org/wiki/Information\\_retrieval](https://en.wikipedia.org/wiki/Information_retrieval) - Cached  
**Information retrieval (IR)** is the area of study concerned with searching for documents, for information within documents, and for metadata about documents, ...  
[Relevance \(information retrieval\) - Information retrieval applications](#)
- ▶ [Introduction to Information Retrieval](#)   
[www-csli.stanford.edu/~hinrich/Information-retrieval-book.html](http://www-csli.stanford.edu/~hinrich/Information-retrieval-book.html) - Cached  
25+ items – Introduction to **Information Retrieval**. This is the companion ...
  - Front matter (incl. table of notations – pdf
  - 02 – The term vocabulary & postings lists
  - 03 – Dictionaries and tolerant retrieval[Slides - Irbook](#) - [Exercises](#) - [Text classification and Naive Bayes](#)

# Assumption 1: More keywords = more specific

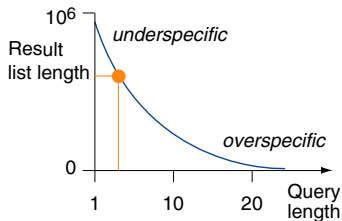
The screenshot shows a Google search interface. At the top, navigation links for Web, Images, Videos, Maps, News, Shopping, Mail, and more are visible. The Google logo is on the left. The search bar contains the query "information retrieval" "query formulation" and shows "About 87,100 results (0.40 seconds)". A red box highlights the number "87,100". To the right of the search bar is an "Advanced search" button. On the left side, there is a vertical menu with icons and labels for "Everything", "Images", "Videos", "News", "Shopping", and "More". The main search results area displays a section for "Scholarly articles for 'information retrieval' 'query formulation'" with a green atom icon. It lists three articles: "Modern information retrieval" by Baeza-Yates (8429 citations), "Extended Boolean information retrieval" by Salton (704 citations), and "Information filtering and information retrieval: two sides ..." by Belkin (1151 citations). Below this, a PDF result is shown for "Query Formulation for Information Retrieval by Intelligence Analysts" from www.cs.virginia.edu/~techrep/CS-2005-12.pdf, with a "Quick View" link. The snippet for this PDF reads: "File Format: PDF/Adobe Acrobat - Quick View by X Jin - Cited by 1 - Related articles proaches. We also suggest a method for comparing performance with other systems and suggest criteria that is. **Query Formulation** for **Information Retrieval** by ..."

# Assumption 1: More keywords = more specific



## Specificity of Queries

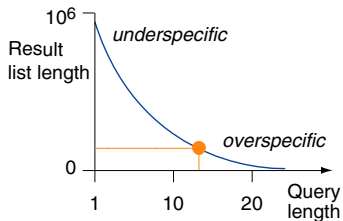
# Assumption 2: User can arbitrarily specify information need



## Specificity of Queries

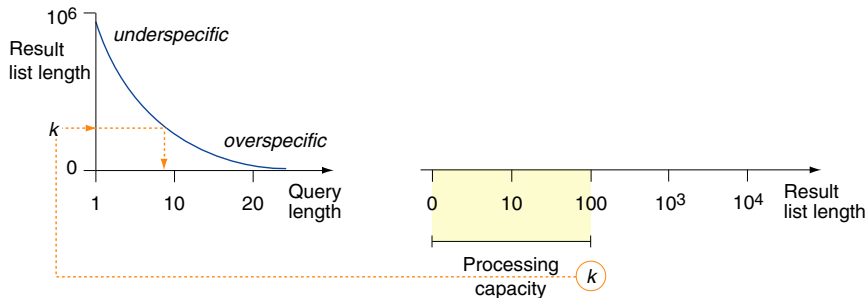


# Assumption 2: User can arbitrarily specify information need



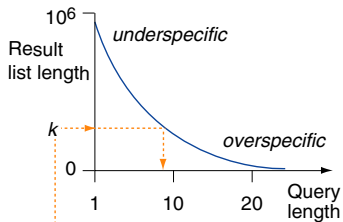
## Specificity of Queries

# Assumption 3: User can consider about $k$ results.

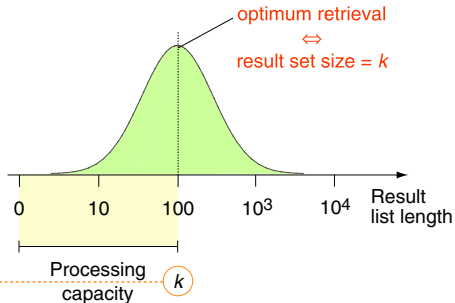


## Specificity of Queries

# Hypothesis: Specificity matches $k =$ Optimum retrieval



Specificity of Queries



Probability for Retrieval Success

What is this hypothesis good for?

## Scenario

- Given a set  $W$  of keywords
- Find a good query  $Q \subseteq W$

## Scenario

- Given a set  $W$  of keywords
- Find a good query  $Q \subseteq W$

## Previous approach

[Lee et al., CIKM 2009]

Learnt ranking function identifies the  $m$  best keywords from  $W$ .

Based on:

- Known relevant documents
- Unrestricted index access
- Manually tuned  $m$  for each set  $W$

# Consider for instance ... Known-Item Finding

## Scenario

- User accessed a document
- But did not store it

### Making the Most of a Web Search Session

Benno Stein and Matthias Hagen  
Hamburg-Universität Wiener  
©first name-last name@uni-wiener.de

**Abstract.** We tackle problems related to Web query formulation given the set of keywords from a search session. If we had a function providing Web query, and if we consider a family of generating Web queries covering all keywords. A query is promising if it fairly well divides attention on the number of returned hits. We assume a real-world setting where the user is not given direct access to search engine's back-end, so optimization is possible only through an interface. The goal is to optimize the overall number of returned Web queries.

For both problems we develop search strategies based on maximum probability. The achieved performance gain is substantial compared to the maximum likelihood without an occurrence probability. The expected ranking are up to 50% in the number of returned queries, hits, answers, and results.

**Keywords:** Web Search Session, Query Formulation, Query Cost Optimization, Maximum Query Query Cost

#### 1. INTRODUCTION

The interaction between Web search users and search engines follows a classic scheme. The user comes up with a set of (in her opinion) appropriate keywords for a given information need. She submits a query containing some of these keywords and gets back a ranked result list. If the user does not find a result for her information need among the first results, she will hardly know all the items but submit different queries based on her keywords until she is satisfied or decides to give up. This process forms a search session—the set of consecutive Web-queries a user submits to a search engine in order to satisfy a given information need.

Experience shows that in many cases a user's first try is successful reasonably well, i.e., the first query brings up an appropriate result. Such one-query sessions are not the focus of this paper; we consider larger sessions where the user did not succeed with her first query. Search engines provide different means to support unsuccessed users, e.g., query expansion for queries containing hits of low ranking positions for queries returning no hits due to topic. In this paper we present two other approaches having a more combinatorial flavor, while being easily combinable with existing technology:

- 1) The maximum query for a given set of keywords, i.e., a query containing as many of the keywords as possible, while returning a reasonable number of results.
- 2) The query cover for a given set of keywords, i.e., a family of queries each returning a reasonable number of results that contains as many keywords as possible.

The idea is to use the keywords submitted in a search session up to the current query, and suggest a maximum query or a query cover as the user's next query. The requirement

to contain as many of the keywords as possible affects the following rationale. Taken together, the keywords of a search session describe the user's information need. Some of the keywords might not be appropriate (e.g., typos) and should be omitted, but the more keywords are contained in a maximum query or a query cover the better is the description of the user's information need.

The rationale for suggesting a reasonable number of hits per query also describes a user's consideration. Queries with empty result pages are useless and the same often applies to queries returning only a handful of hits. This gives a lower bound on the number of desired results. But there is also an upper bound since the number of results a user will consider for a single query is usually constrained by a processing capacity limit, determined by the user's reading time etc. If the user issues a query with millions of hits, she can only check a fraction of the results—typically the top-ranked ones. Relevant entries below are missed. We argue that the best queries are the ones that are sufficiently specific to not return millions of hits—but also not just one or two. For such queries the user can check the complete result list and will not miss any potential match for her information need due to search engine ranking issues but the content dilution.

Hence, from the user's perspective, maximum and covering queries contain the best possible description of the information need and offer the chance to check all the results. However, finding a maximum query or a query cover "by hand" is not fun straightforward. Several queries have to be submitted to identify appropriate keyword combinations. Hardly any user will take the time for such a lengthy procedure. Therefore, we give algorithms for both tasks. To be applicable as user side the algorithms are of general nature, i.e., they only use standard search engine interfaces. The Web search engine is handled as a black box, so that like an oracle that answers queries. There is no need to know the underlying retrieval model or implementation details.

Since Web searches are not for free but entail costs—at the very least some non-negligible amount of time is consumed, and memory charges come into play for large configurations of queries—we analyze the corresponding economic optimization problem for finding maximum and covering queries. We ask:

Which answer minimizes

the average number of submitted queries?

A. Related Work

A lot of research has been done on approaches for better results on better queries. An example of a very promising

# Consider for instance ... Known-Item Finding

## Scenario

- User accessed a document
- But did not store it

How can she find it again?

### Making the Most of a Web Search Session

Benno Stein and Matthias Hagen  
Hamburg-Universität WiSe04  
©first name-last name@uni-wi.se.edu

**Abstract.** We tackle problems related to Web query formulation given the set of keywords from a search session. It is that a function providing Web query, and (2) to construct a family of generating Web queries covering all keywords. A query is provided if it fulfills predefined constraints on the number of returned hits. We assume a real-world setting where the user is not given direct access to search engine's index, i.e., optimization is possible only through an interface. The goal is to optimize the overall number of submitted Web queries.

For both problems we develop search strategies based on incremental probabilities. The achieved performance gain is substantial compared to the straightforward baseline without an optimization procedure. The experimental settings are up to 500, the number of submitted queries, hits, answers, and results.

**Keywords:** Web search session, Query formulation, Query Cost Optimization, Maximum Query Query Cost

#### 1. INTRODUCTION

The interaction between Web search users and search engines follows a classic scheme. The user comes up with a set of (in her opinion) appropriate keywords for a given information need. She submits a query containing some of these keywords and gets back a ranked result list. If the user does not find a result for her information need among the first results, she will hardly know all the items but submit different queries based on her keywords until she is satisfied or decides to give up. This process forms a search session, the set of consecutive Web-queries a user submits to a search engine in order to satisfy a given information need.

Experience shows that in many cases a user's first try is successful reasonably well, i.e., the first query brings up an appropriate result. Such one-query sessions are not the focus of this paper; we consider larger sessions where the user did not succeed with her first query. Search engines provide different means to support unsuccessed users, e.g., query expansion for keywords missing hits or spelling correction for queries returning no hits due to typos. In this paper we present two other approaches having a more combinatorial flavor, while being easily combinable with existing technology:

- 1) The maximum query for a given set of keywords, i.e., a query containing as many of the keywords as possible, while remaining a reasonable number of results.
- 2) The query cover for a given set of keywords, i.e., a family of queries each containing a reasonable number of results that contain as many keywords as possible.

The idea is to use the keywords submitted in a search session up to the current query, and suggest a maximum query or a query cover as the user's next query. The requirement

to contain as many of the keywords as possible reflects the following rationale. Taken together, the keywords of a search session describe the user's information need. Some of the keywords might not be appropriate (e.g. typos) and should be omitted, but the more keywords are contained in a maximum query or a query cover the better is the description of the user's information need.

The rationale for suggesting a reasonable number of hits per query also describes obvious considerations. Queries with empty result pages are useless and the same often applies to queries returning only a handful of hits. This gives a lower bound on the number of desired results. But there is also an upper bound since the number of results a user will consider for a single query is usually constrained by a processing capacity limit, determined by the user's reading time etc. If the user issues a query with millions of hits, she can only check a fraction of the results—typically the top-ranked ones. Relevant entries below are missed. We argue that the best queries are the ones that are sufficiently specific to not return millions of hits—but also not just one or two. For each query the user can check the complete result list and will not miss any potential match for her information need due to search engine ranking issues but the caused frustration.

Hence, from the user's perspective, maximum and covering queries contain the best possible description of the information need and offer the chance to check all the results. However, finding a maximum query or a query cover "by hand" is not fun straightforward. Several queries have to be submitted to identify appropriate keyword combinations. Hardly any user will take the time for such a lengthy procedure. Therefore, we give algorithms for both tasks. To be applicable as user-side algorithms we of course restrict, i.e., they only use standard search engine interfaces. The Web search engine is handled as a black box, acting like an oracle that answers queries. There is no need to know the underlying retrieval model or implementation details.

Since Web searches are not for free but entail costs—in the very least some non-negligible amount of time is consumed, and memory charges come into play for large configurations of queries—we analyze the corresponding economic optimization problem for finding maximum and covering queries. We ask:

Which answers minimize the average number of submitted queries?

#### A. Related Work

A lot of research has been done on approaches for better results on better queries. An example of a very promising



# Consider for instance ... Known-Item Finding

## Scenario

- User accessed a document
- But did not store it

How can she find it again?

### Making the Most of a Web Search Session

Benno Stein and Matthias Hagen  
Hamburg-Universität Wiener  
©1997 nameo-clat nameo-frank-wiener.de

**Abstract.** We tackle problems related to Web query formulation given the set of keywords from a search session. In a first step a function providing Web queries will be considered. A family of generating Web queries covering all keywords. A query is provided if it fulfills additional constraints on the number of returned hits. We assume a real-world setting where the user is not given direct access to search engine's index, i.e., operations is possible only through an interface. The goal is to optimize the overall number of submitted Web queries.

**For both problems we develop search strategies based on maximum probability. The achieved performance gain is substantial compared to the maximum likelihood without an occurrence probability (the expected ranking rate up to 50% in the number of submitted queries, hits, answers, and results).**

**Keywords:** Web search session, Query formulation, Query Cost Optimization, Maximum Query Query Cost

#### 1. INTRODUCTION

The interaction between Web search users and search engines follows a classic scheme. The user comes up with a set of (in her opinion) appropriate keywords for a given information need. She submits a query containing some of these keywords and gets back a ranked result list. If the user does not find a result for her information need among the first results, she will hardly know all the same but submit different queries based on her keywords until she is satisfied or decides to give up. This process forms a search session: the set of consecutive Web-queries a user submits to a search engine in order to satisfy a given information need.

Experience shows that in many cases a user's first try is successful reasonably well, i.e., the first query brings up an acceptable result. Such one-query sessions are not the focus of this paper; we consider larger sessions where the user did not succeed with her first query. Search engines provide different means to support unsuccessful users, e.g., query expansion for queries returning lists of hits, spelling correction for queries returning no hits due to typos. In this paper we present two other approaches having a more combinatorial flavor, while being easily combinable with existing technology:

- 1) The maximum query for a given set of keywords, i.e., a query containing as many of the keywords as possible, while returning a reasonable number of results.
- 2) The query cover for a given set of keywords, i.e., a family of queries each containing a reasonable number of results that contains as many keywords as possible.

The idea is to use the keywords submitted in a search session up to the current query, and suggest a maximum query or a query cover as the user's next query. The requirement

to contain as many of the keywords as possible reflects the following rationale. Taken together, the keywords of a search session describe the user's information need. Some of the keywords might not be appropriate in a typical and should be omitted, but the more keywords are contained in a maximum query or a query cover the better is the description of the user's information need.

The rationale for suggesting a reasonable number of hits per query also deserves closer consideration. Queries with empty result pages are useless and the same often applies to queries returning only a handful of hits. This gives a lower bound on the number of desired results. But there is also an upper bound since the number of results a user will consider for a single query is usually constrained by a processing capacity limit, determined by the user's reading time etc. If the user faces a query with millions of hits, she can only check a fraction of the results—typically the top-ranked ones. Relevant entries below are missed. We argue that the best queries are the ones that are sufficiently specific to not return millions of hits—but also not just one or two. For each query the user can check the complete result list and will not miss any potential match for her information need due to search engine ranking issues but the casual observer.

Hence, from the user's perspective, maximum and covering queries contain the best possible description of the information need and offer the chance to check all the results. However, finding a maximum query or a query cover "by hand" is not fun straightforward. Several queries have to be submitted to identify appropriate keyword combinations. Hardly any user will take the time for such a lengthy procedure. Therefore, we give algorithms for both tasks. To be applicable to our site the algorithms are of constant nature, i.e., they only use standard search engine interfaces. The Web search engine is handled as a black box, being like an oracle that answers queries. There is no need to know the underlying internal model or implementation details.

Since Web searches are not for free but entail costs—at the very least some non-negligible amount of time is consumed, and monetary charges come into play for large combinations of queries—we analyze the corresponding economic optimization problem for finding maximum and covering queries. We ask:

Which query maximizes  
the average number of submitted queries?

A. Related Work

A lot of research has been done on approaches for better results on better queries. An example of a very promising

## Solution

- Remember some keywords

information retrieval query formulation  
web search search session user support  
search engine cost optimization

- Query a search engine

But what query to formulate with the keywords?

# Single keywords?

information retrieval

~~http://information~~

~~web/search~~

~~search/session~~

~~web/support~~

~~search/engine~~

~~cs.stanford.edu/optimalization~~

Web Images Videos Maps News Shopping Mail more -

Google

"information retrieval"

About 17,200,000 results (0.14 seconds) [Advanced search](#)

Everything

Images

Videos

News

Shopping

Blogs

Books

Discussions

▶ [Information retrieval - Wikipedia, the free encyclopedia](#)

[en.wikipedia.org/wiki/Information\\_retrieval](http://en.wikipedia.org/wiki/Information_retrieval) - Cached

**Information retrieval (IR)** is the area of study concerned with searching for documents, for information within documents, and for metadata about documents, ...

[Relevance \(information retrieval\) - Information retrieval applications](#)

[Introduction to Information Retrieval](#)

[www-csli.stanford.edu/~hinrich/Information-retrieval-book.html](http://www-csli.stanford.edu/~hinrich/Information-retrieval-book.html) - Cached

25+ items – Introduction to **Information Retrieval**. This is the companion ...

- Front matter (incl. table of notations – pdf
- 02 – The term vocabulary & postings lists
- 03 – Dictionaries and tolerant retrieval

[Slides - Irbook](#) - [Exercises](#) - [Text classification and Naive Bayes](#)

# Single keywords?

~~information/retrieval~~

query formulation

~~web/search~~

~~search/evaluation~~

~~web/support~~

~~search/engine~~

~~cost/optimization~~

Web Images Videos Maps News Shopping Mail more ▾

Google

"query formulation"

About 188,000 results (0.09 seconds)



Advanced search

Everything

Images

Videos

News

Shopping

More

▶ [PDF] [QUERY FORMULATION IN WEB INFORMATION SEARCH](#)

[citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.76.8657&rep...](#)

File Format: PDF/Adobe Acrobat - [Quick View](#)

by A Aula - [Cited by 41](#) - [Related articles](#)

**Query formulation** is an essential part of successful information retrieval. ... Based on the previous studies concerning **query formulation** and the results from the ...

[PDF] [DataGuides: Enabling Query Formulation and Optimization in ...](#)

[citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.55.8594&rep...](#)

File Format: PDF/Adobe Acrobat - [Quick View](#)

by R Goldman - [Cited by 1324](#) - [Related articles](#)

browsing and **query formulation**, and as a means of guiding the query ...

# Single keywords?

*information/retrieval*      *quality/algorithm*  
web search      *search/session*      *web/support*  
*search/engine*      *cost/optimization*

Web Images Videos Maps News Shopping Mail more ▾

Google

"web search"

About 96,800,000 results (0.09 seconds)

Advanced search

Everything

Images

Videos

News

Shopping

More

WebSearch

[www.websearch.com/](http://www.websearch.com/) - Cached

Web Local Rebates more ▾. Games Screensavers Maps. **WebSearch**. Search. Home About Privacy Policy Terms of Use Help. © 2011 **WebSearch**.com.

[Terms of Use](#) - [About Us](#) - [Help](#)

Dogpile Web Search

[www.dogpile.com/](http://www.dogpile.com/) - Cached

Dogpile.com makes searching the Web easy, because it has all the best search engines piled into one. Go Fetch!

[White Pages](#) - [Images](#) - [Preferences](#) - [Make Dogpile Your Home Page](#)

*information/technology*

*quality/algorithm*

web search

*search/session*

*web/support*

*search/engine*

*cost/optimization*

Web Images Videos Maps News Shopping Mail more ▾

Google

"web search"

About 96,800,000 results (0.09 seconds)

Advanced search

Everything

Images

Videos

News

Shopping

More

WebSearch

[www.websearch.com/](http://www.websearch.com/) - Cached

Web Local Rebates more ▾. Games Screensavers Maps. **WebSearch**. Search. Home About Privacy Policy Terms of Use Help. © 2011 **WebSearch**.com.

[Terms of Use](#) - [About Us](#) - [Help](#)

Dogpile Web Search

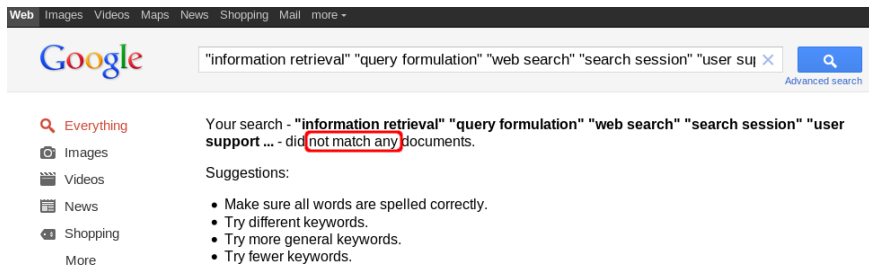
[www.dogpile.com/](http://www.dogpile.com/) - Cached

Dogpile.com makes searching the Web easy, because it has all the best search engines piled into one. Go Fetch!

[White Pages](#) - [Images](#) - [Preferences](#) - [Make Dogpile Your Home Page](#)

# All keywords at once?

information retrieval      query formulation  
web search      search session      user support  
search engine      cost optimization



The screenshot shows a Google search interface. At the top, there are navigation links: Web, Images, Videos, Maps, News, Shopping, Mail, and more. The Google logo is on the left, and the search bar contains the query: "information retrieval" "query formulation" "web search" "search session" "user support" "search engine" "cost optimization". A blue search button with a magnifying glass icon is on the right, with the text "Advanced search" below it. Below the search bar, there are navigation options: Everything (selected), Images, Videos, News, Shopping, and More. The search results area shows the message: "Your search - 'information retrieval' 'query formulation' 'web search' 'search session' 'user support ...' - did not match any documents." The phrase "not match any" is highlighted with a red box. Below this, there are suggestions: "Suggestions:" followed by a bulleted list: "• Make sure all words are spelled correctly.", "• Try different keywords.", "• Try more general keywords.", and "• Try fewer keywords."

information retrieval      query formulation  
web search      search session      user support  
search engine      cost optimization

The screenshot shows a Google search interface. The search bar contains the query: "information retrieval" "query formulation" "web search" "search session" "user support" "search engine" "cost optimization". The search results show a message: "Your search - 'information retrieval' 'query formulation' 'web search' 'search session' 'user support ...' - did not match any documents." The phrase "not match any" is highlighted with a red box. Below the message, there are suggestions for improving the search, such as "Make sure all words are spelled correctly.", "Try different keywords.", "Try more general keywords.", and "Try fewer keywords." The left sidebar shows navigation options like "Everything", "Images", "Videos", "News", "Shopping", and "More".

Web Images Videos Maps News Shopping Mail more ▾

Google

"information retrieval" "query formulation" "web search" "search session" "user support" "search engine" "cost optimization" X

Advanced search

Everything

Images

Videos

News

Shopping

More

Your search - "information retrieval" "query formulation" "web search" "search session" "user support ..." - did not match any documents.

Suggestions:

- Make sure all words are spelled correctly.
- Try different keywords.
- Try more general keywords.
- Try fewer keywords.



Remember the hypothesis . . . not too many results!

# Solution: As many keywords as possible!

information retrieval      query formulation  
web search      search session      ~~user/session~~  
search engine      cost optimization

The screenshot shows a Google search interface. The search bar contains the query: "information retrieval" "query formulation" "web search" "search session" "search engine" "cost optimization". The search results show 3 results in 0.19 seconds. The first result is a PDF titled "Search Strategies for Keyword Queries" from uni-weimar.de. The second result is a PDF titled "Making the Most of a Web Search Session" by B. Stein, also from uni-weimar.de. The search results are displayed in a list format with icons for PDF files and search engines.

# “As many keywords as possible” -Query

## Characteristics

- Captures most of the remembered keywords
- Best possible description of the known-item
- Not too many results → user can check complete list

# “As many keywords as possible” -Query

## Characteristics

- Captures most of the remembered keywords
- Best possible description of the known-item
- Not too many results → user can check complete list

## Problem

- Relevant documents not known
- No web index at user site
- Query size not known

# “As many keywords as possible” -Query

## Characteristics

- Captures most of the remembered keywords
- Best possible description of the known-item
- Not too many results → user can check complete list

## Problem

- Relevant documents not known
- No web index at user site → Lee et al. not applicable
- Query size not known

We propose an approach for this scenario . . .

## PROMISING QUERY

- Given:
  - ① A set  $W$  of keywords
  - ② An upper bound  $k$  on the result list length
- Find a largest query  $Q \subseteq W$  yielding at most  $k$  results

## Optimization Problem!

Minimize the number of submitted web queries to find  $Q$ .

## PROMISING QUERY

- Given:
  - ① A set  $W$  of keywords
  - ② An upper bound  $k$  on the result list length
- Find a largest query  $Q \subseteq W$  yielding at most  $k$  results

## Optimization Problem!

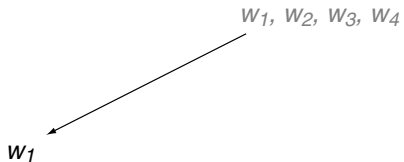
Minimize the number of submitted web queries to find  $Q$ .



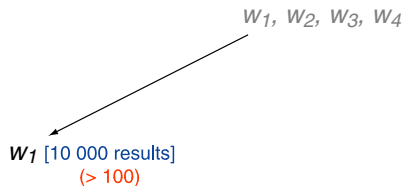
# Baseline: Depth-First Search

$W_1, W_2, W_3, W_4$

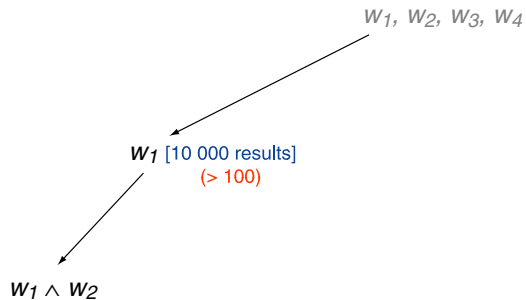
# Baseline: Depth-First Search



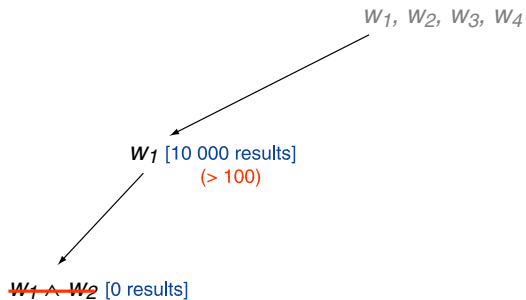
# Baseline: Depth-First Search



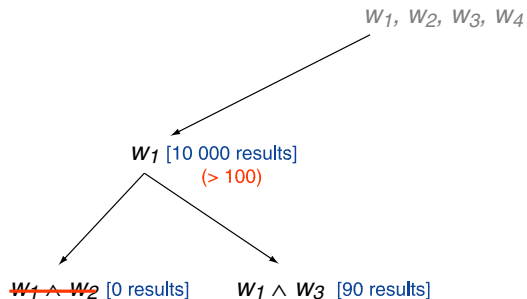
# Baseline: Depth-First Search



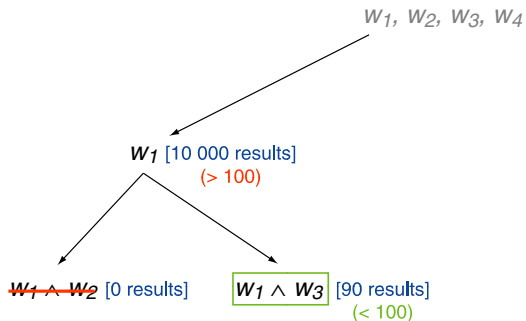
# Baseline: Depth-First Search



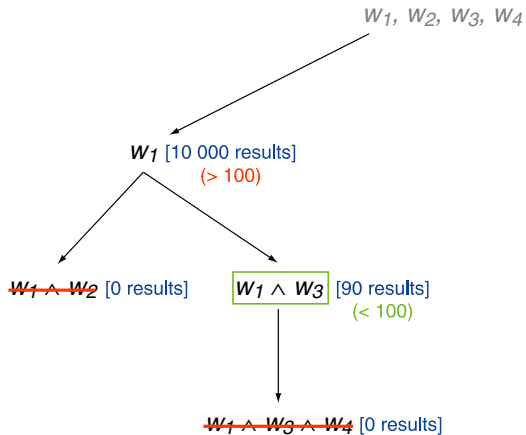
# Baseline: Depth-First Search



# Baseline: Depth-First Search

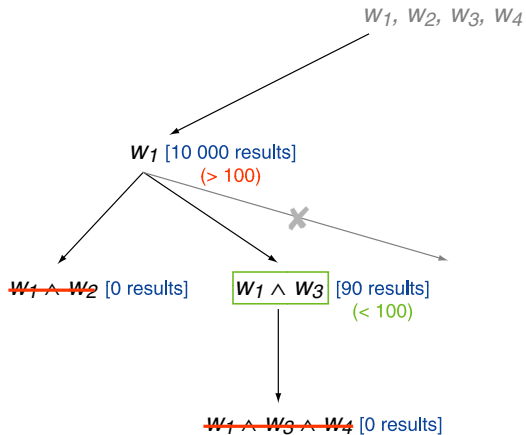


# Baseline: Depth-First Search

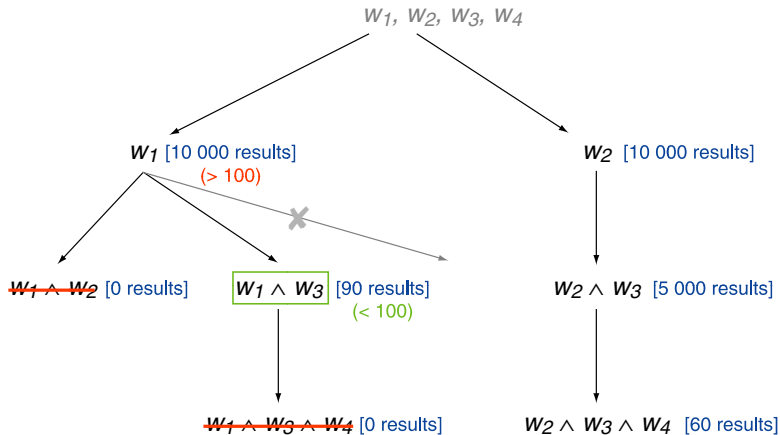




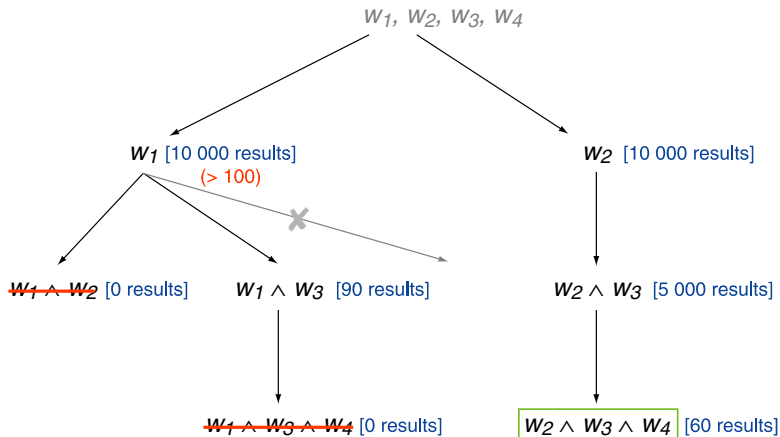
# Baseline: Depth-First Search



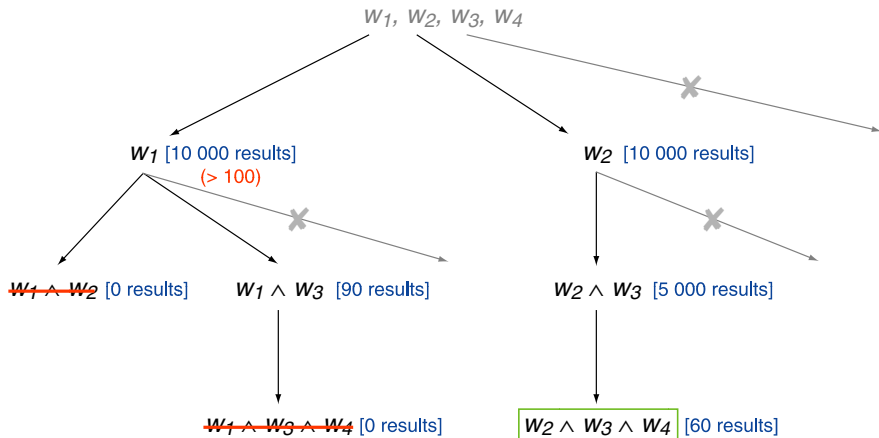
# Baseline: Depth-First Search



# Baseline: Depth-First Search



# Baseline: Depth-First Search



## Major Drawback

All intermediate queries submitted. → Bad run time!

## Major Drawback

All intermediate queries submitted. → Bad run time!

## Idea

Estimate the result list length before query submission.

# Co-occurrence based Estimations

Estimate: "information retrieval" "query formulation" + "web search"

# Co-occurrence based Estimations

Estimate: "information retrieval" "query formulation" + "web search"

Known: "information retrieval" "query formulation" 87 100 results



# Co-occurrence based Estimations

Estimate: "information retrieval" "query formulation" + "web search"

Known: "information retrieval" "query formulation" 87 100 results  
"information retrieval" + "web search" 16 % remain

# Co-occurrence based Estimations

Estimate: "information retrieval" "query formulation" + "web search"

Known:	"information retrieval" "query formulation"	87 100 results
	"information retrieval" + "web search"	16 % remain
	"query formulation" + "web search"	22 % remain

# Co-occurrence based Estimations

Estimate: "information retrieval" "query formulation" + "web search"

Known:	"information retrieval" "query formulation"	87 100 results
	"information retrieval" + "web search"	16 % remain
	"query formulation" + "web search"	22 % remain

Our estimation scheme:  $\text{avg}(16\% , 22\%) = 19\%$   
 $87\,100 \cdot 0.19 = 16\,500$  results

# Co-occurrence based Estimations

Estimate: "information retrieval" "query formulation" + "web search"

Known:	"information retrieval" "query formulation"	87 100 results
	"information retrieval" + "web search"	16 % remain
	"query formulation" + "web search"	22 % remain

Our estimation scheme:  $\text{avg}(16\% , 22\%) = 19\%$   
 $87\,100 \cdot 0.19 = 16\,500$  results

Control:  35 700 results

# Co-occurrence based Estimations

Estimate: "information retrieval" "query formulation" + "web search"

Known:	"information retrieval" "query formulation"	87 100 results
	"information retrieval" + "web search"	16 % remain
	"query formulation" + "web search"	22 % remain

Our estimation scheme:  $\text{avg}(16\% , 22\%) = 19\%$   
 $87\,100 \cdot 0.19 = 16\,500$  results

Control:  35 700 results

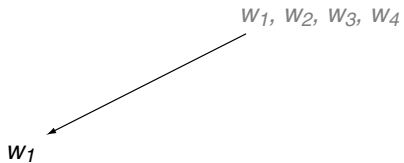
## Observation

Our scheme usually underestimates the real result list length.

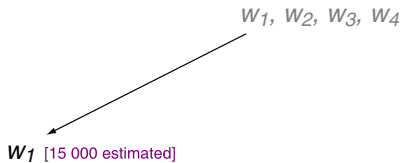
# “Informed” Baseline

$W_1, W_2, W_3, W_4$

# “Informed” Baseline

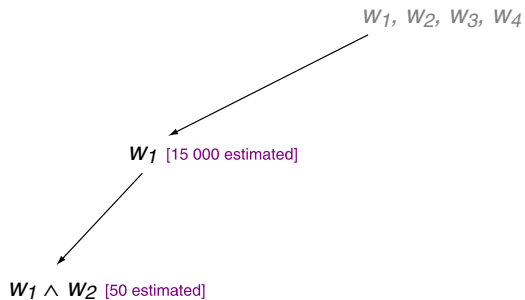


# “Informed” Baseline

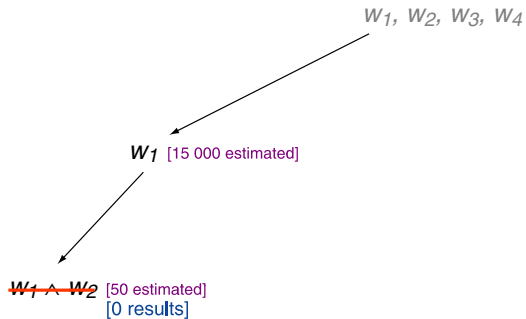




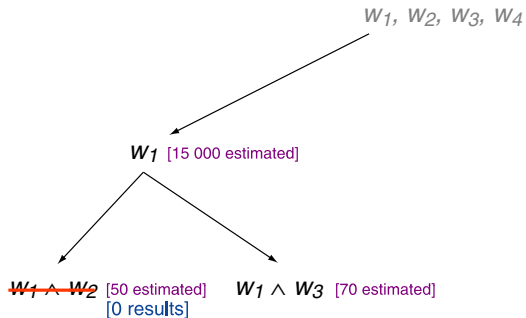
# “Informed” Baseline



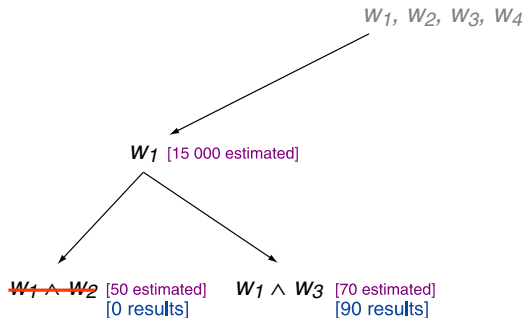
# “Informed” Baseline



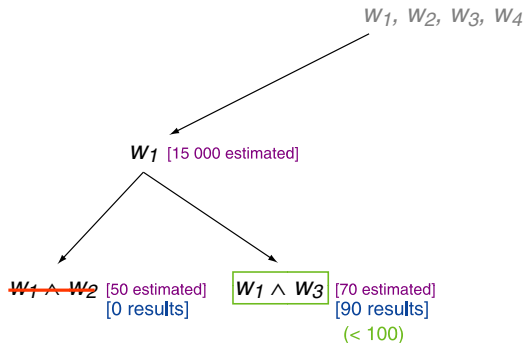
# “Informed” Baseline



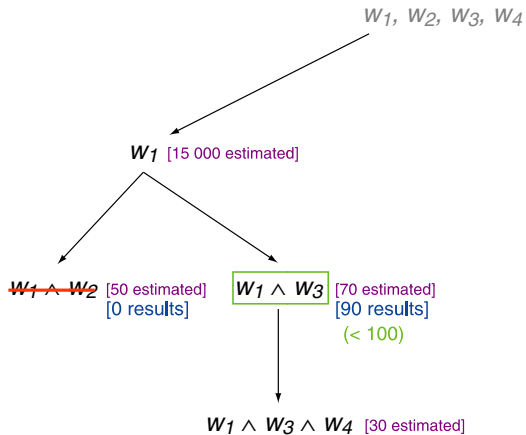
# “Informed” Baseline



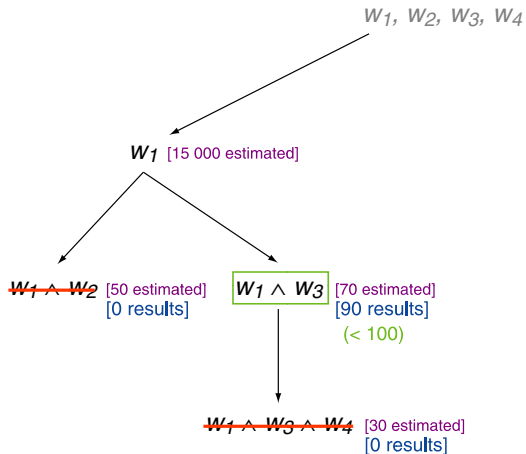
# “Informed” Baseline



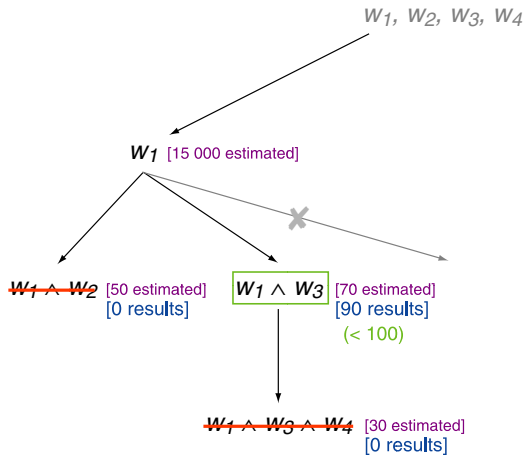
# “Informed” Baseline



# “Informed” Baseline

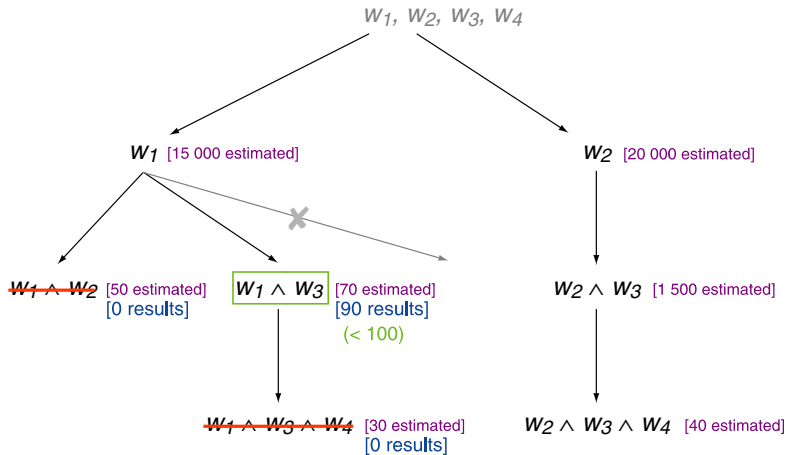


# “Informed” Baseline

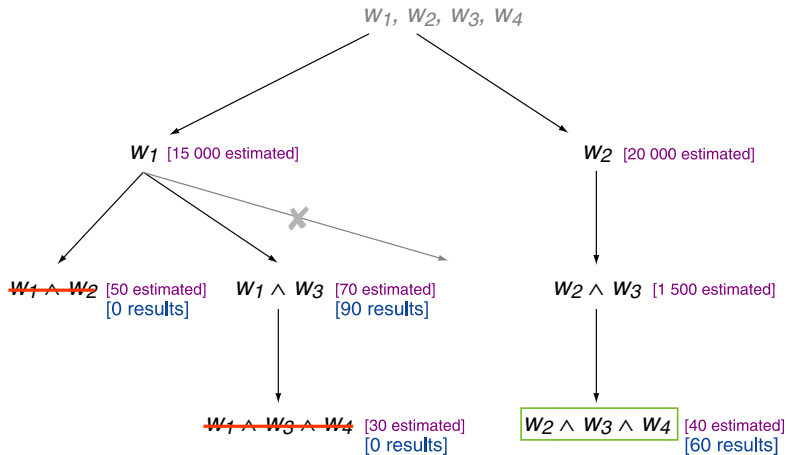




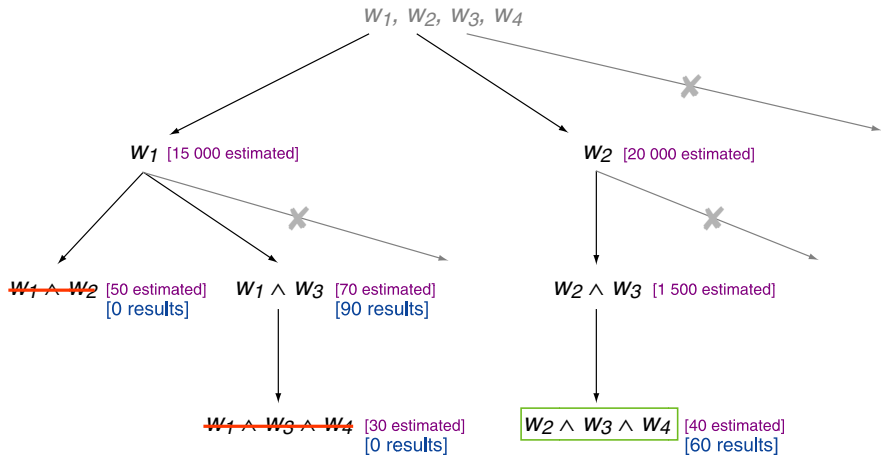
# “Informed” Baseline



# “Informed” Baseline



# “Informed” Baseline



Informed baseline + heuristic reordering of the keywords at each step

## Corpus

- 775 papers on Computer Science (the known-items)
- 15 keywords extracted from each

## System

- Bing API as search engine
- Set  $k = 100$
- Measure number of submitted Web queries

## Corpus

- 775 papers on Computer Science (the known-items)
- 15 keywords extracted from each

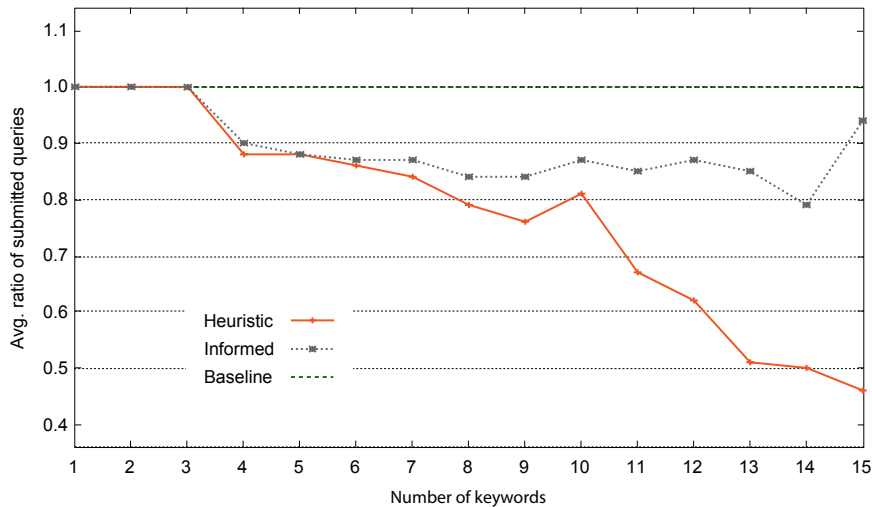
## System

- Bing API as search engine
- Set  $k = 100$
- Measure number of submitted Web queries

# Experimental Results

Number of keywords		5	10	15
Promising query	not possible	614	328	86
	found	161	447	689
Avg. queries submitted	heuristic	10.39	<b>24.93</b>	<b>53.78</b>
	informed	<b>10.36</b>	27.01	108.78
	baseline	11.81	30.94	116.22

# Experimental Results





Almost the end: The take-away messages!

## Results

- User-over-Ranking
  - longer queries → fewer results
  - optimum retrieval performance  
→ user capacity
- Heuristic for promising queries
- Use cases:
  - Known-item finding
  - Empty results lists
  - Query sessions

## Future Work

- Co-occurrence source
- User study

# What we have (not) done

## Results

- User-over-Ranking
  - longer queries → fewer results
  - optimum retrieval performance  
→ user capacity
- Heuristic for promising queries
- Use cases:
  - Known-item finding
  - Empty results lists
  - Query sessions

## Future Work

- Co-occurrence source
- User study

# What we have (not) done

## Results

- User-over-Ranking
  - longer queries → fewer results
  - optimum retrieval performance  
→ user capacity
- Heuristic for promising queries
- Use cases:
  - Known-item finding
  - Empty results lists
  - Query sessions

## Future Work

- Co-occurrence source
- User study

**Thank you**  
