THE UNIVERSITY OF QUEENSLAND
AUSTRALIA
CREATE CHANGE

Information Retrieval 101

ie
lab

Why this video?

- Introduce key concepts in Information Retrieval
    - You are still required to follow upon these concepts individually
- Provide a common framework for discussion for your project
    - Common terminology, basic understanding of problem and techniques.

Information Retrieval (IR)

Information Retrieval is the field concerned with the structure, analysis, organisation, storage, searching and retrieval of information

Gerald Salton, IR pioneer

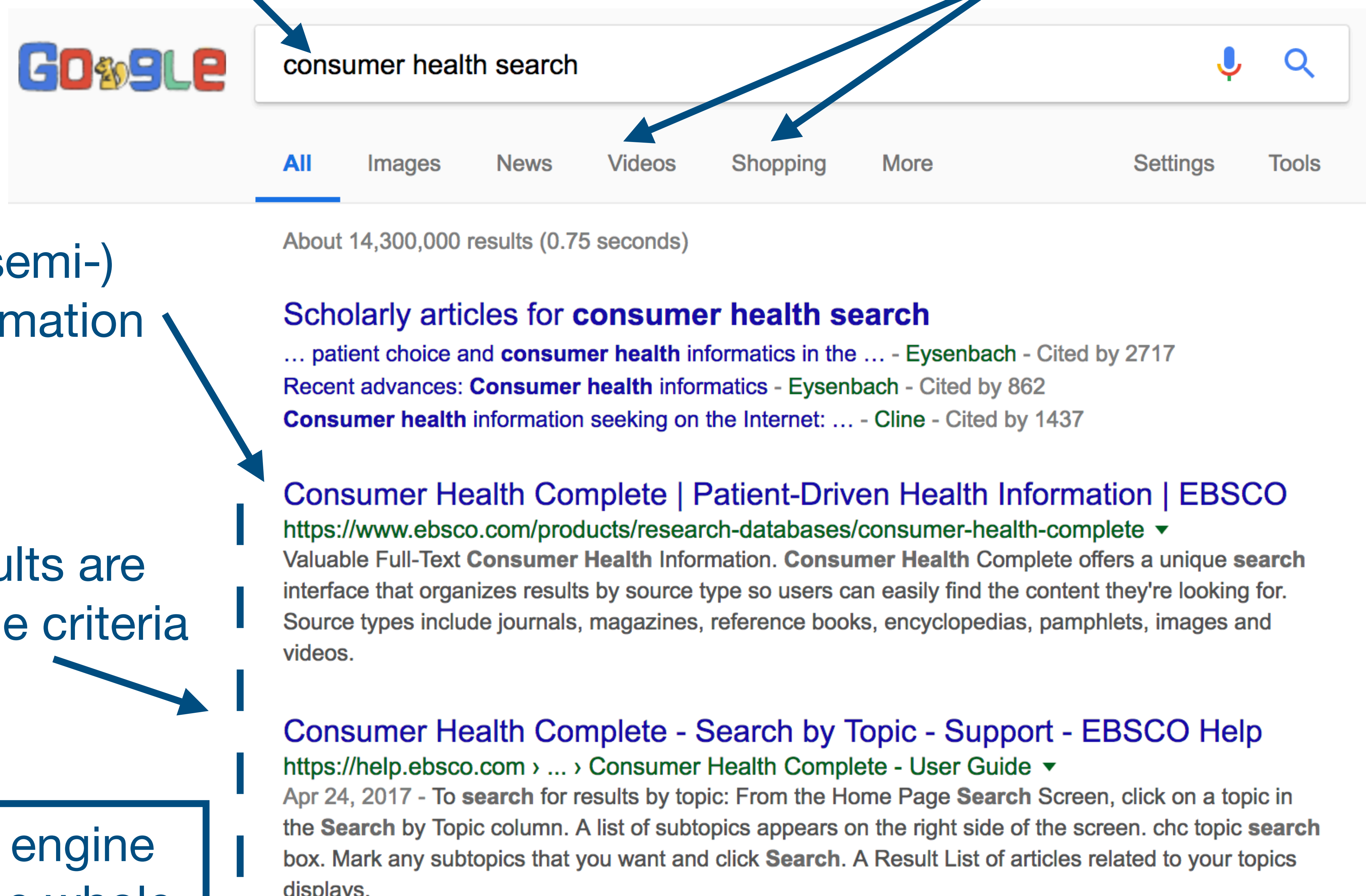
The perfect search engine understands exactly what you mean and gives you back exactly what you want

Larry Page, Google co-founder

A typical view of IR

Query: it expresses the user information need

Verticals: different search engines



Document: (semi-) instructed information

Ranking: results are ordered by some criteria

SERP: search engine result page - the whole page of results

Documents

- A search engine retrieves ***documents***
- The use of the term *Document* to describe the **unit of information** to retrieve is due to the initial IR research being on <u>text</u> documents
- However, document:
    - **Text** or **not-text** (e.g. image, song)
    - **Unstructured** or **semi-structured** (e.g. webpage), but generally not like a DB record
- Examples of documents: web pages, emails, books, news stories, Word files, LinkedIn profiles
- Usually we say documents are organised into a **collection** (i.e. the set of documents that are processed by the search engine)

Information Needs and Queries

- To search, common users pose a query to the search engine
    - *Exceptions exist: can you think of some?*
- Keyword queries are often **poor**, **approximate** descriptions of actual **information needs**

Relevance

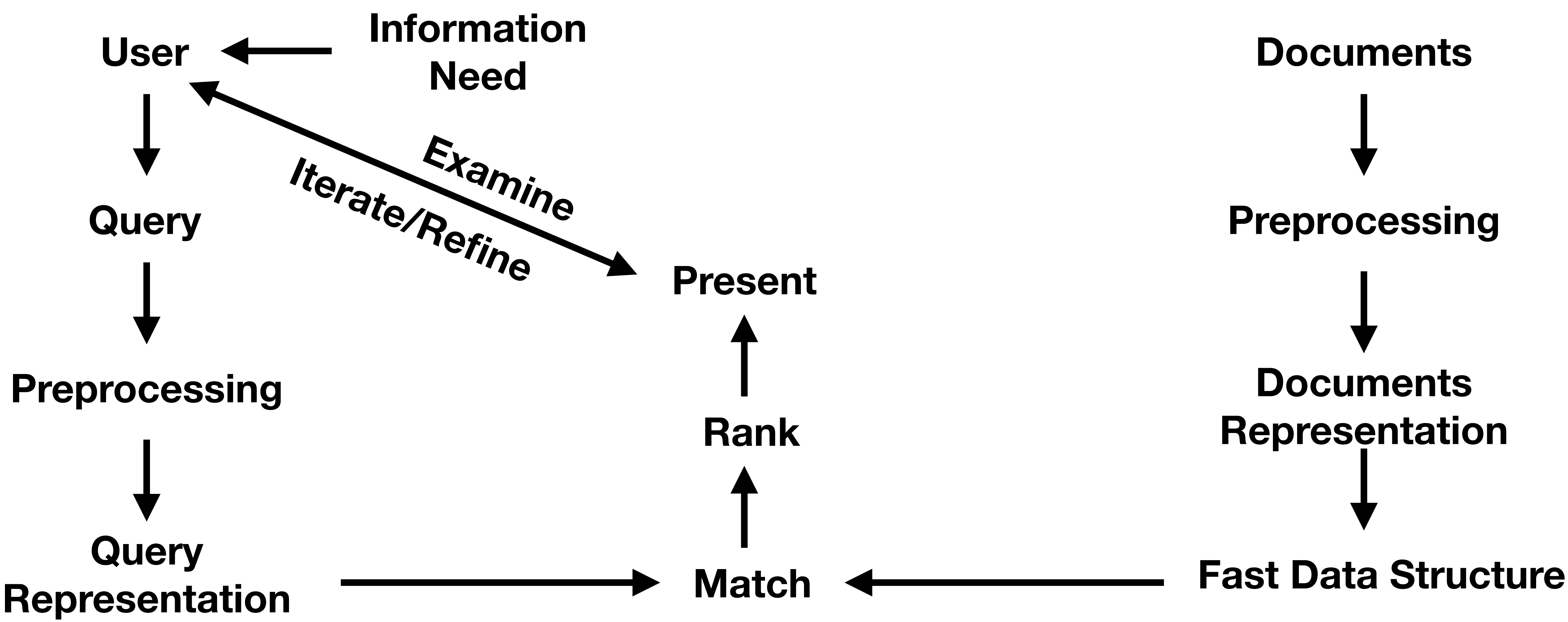
- Relevance is a relationship between a query and a document
- Different types of relevance:
    - **Topical relevance** (or aboutness): document is on same topic of query
    - **User relevance**: not just topical, but also: understandable, novel, trustworthy, etc
- **Algorithmical relevance** is the relevance that systems infer:
    - The objective is to get algorithmical relevance as close to user relevance
    - Most systems though only attempt to estimate topical relevance

Mizzaro. "Relevance: The whole history." Journal of the American society for information science 48.9 (1997): 810-832.

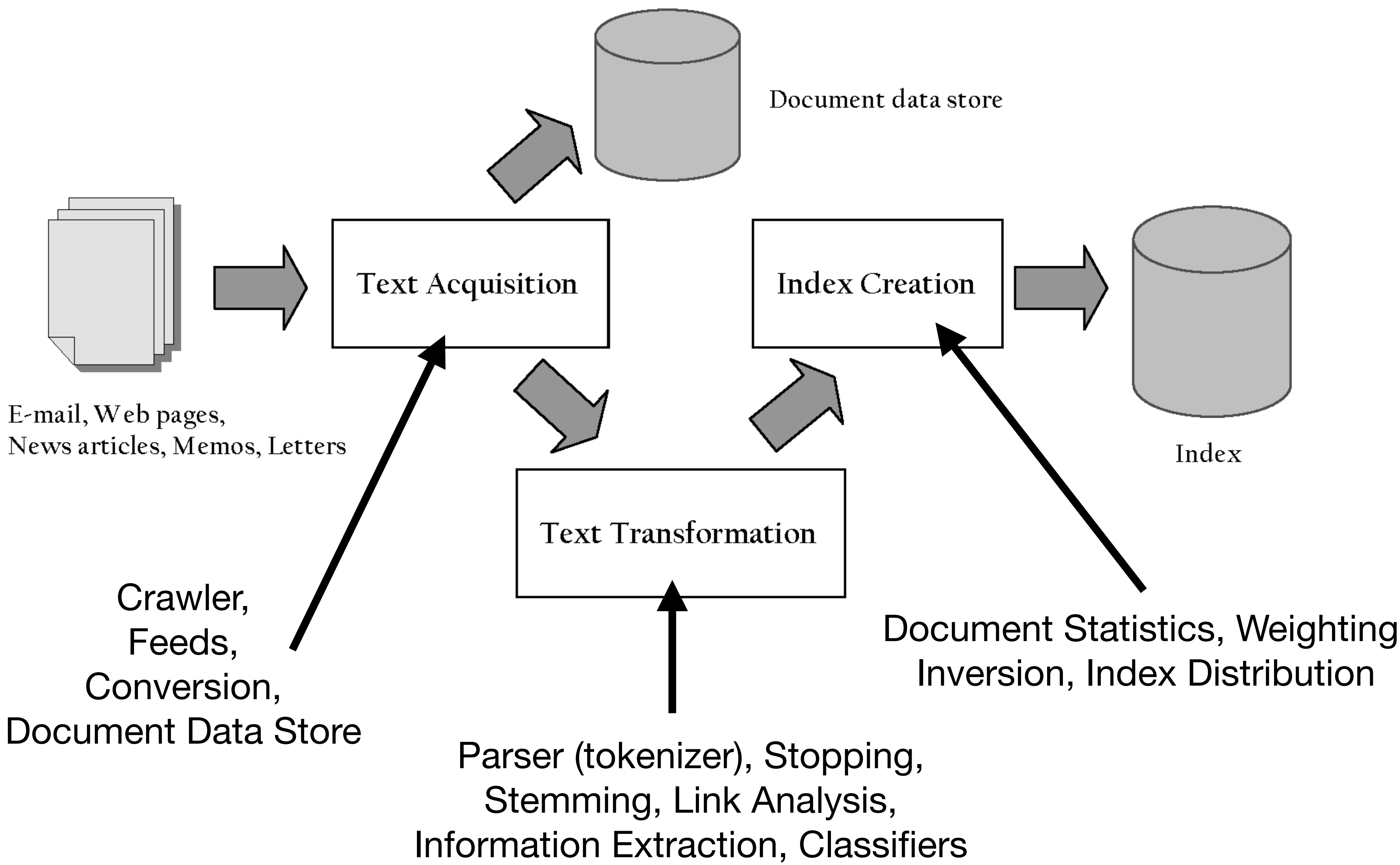
Why comparing text is not enough?

- Comparing the query text to the document text and determining what is a good match is the core issue of information retrieval
- Exact matching of words is not enough: Why?
  - Many different ways to write the same thing in a “natural language” like English — **vocabulary mismatch problem**
  - e.g., what text do we expect a relevant page for the query “most paid footballer” contain?
  - Some documents will be better matches than others; a direct answer (rather than a page) may be even preferred (Q&A search engine)
  - User relevance VS Topical Relevance/Aboutness

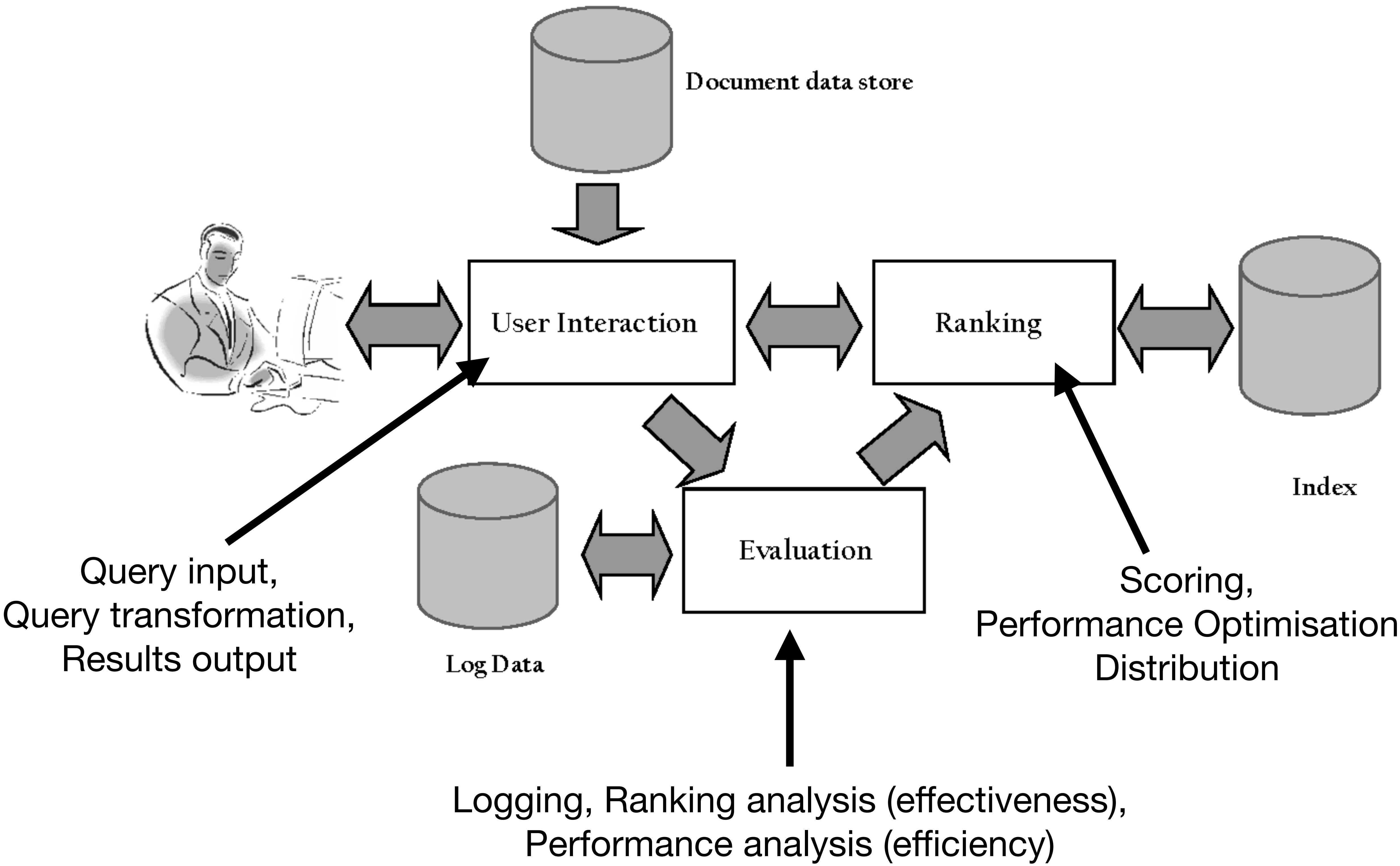
Search - the big picture



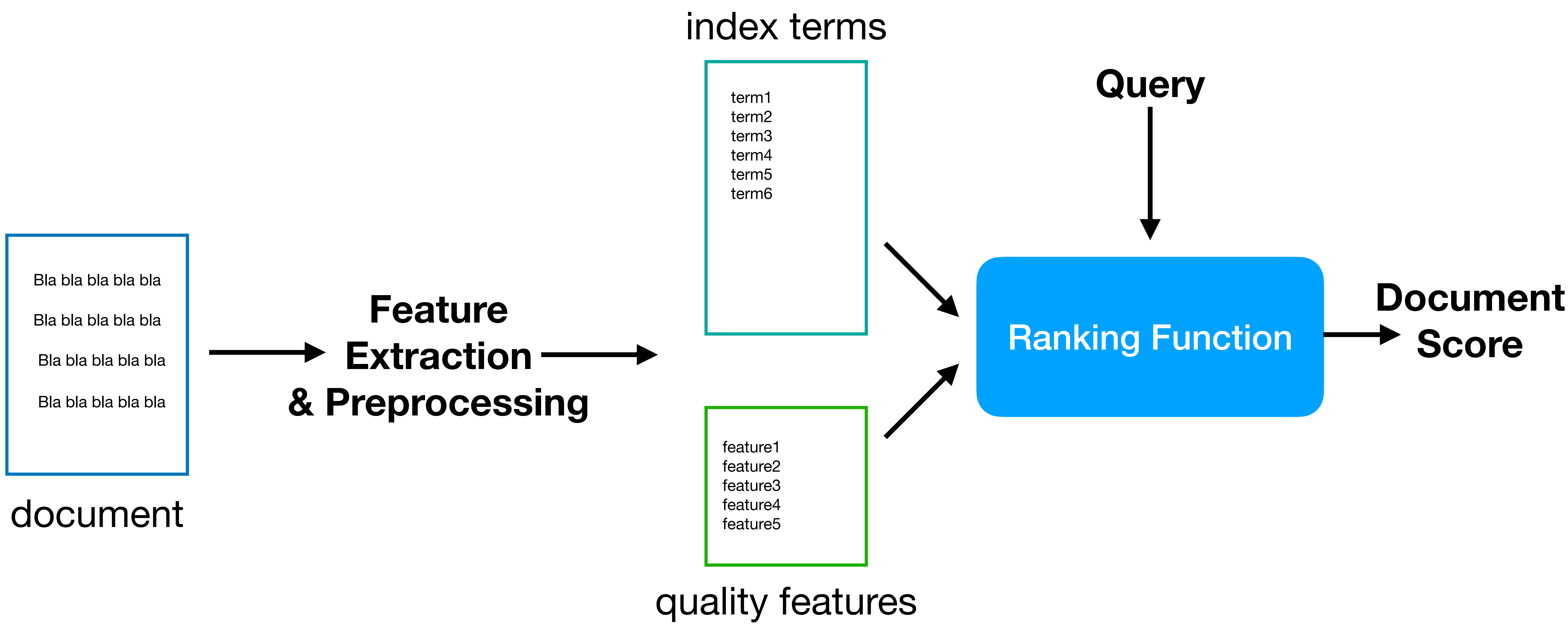
The indexing process



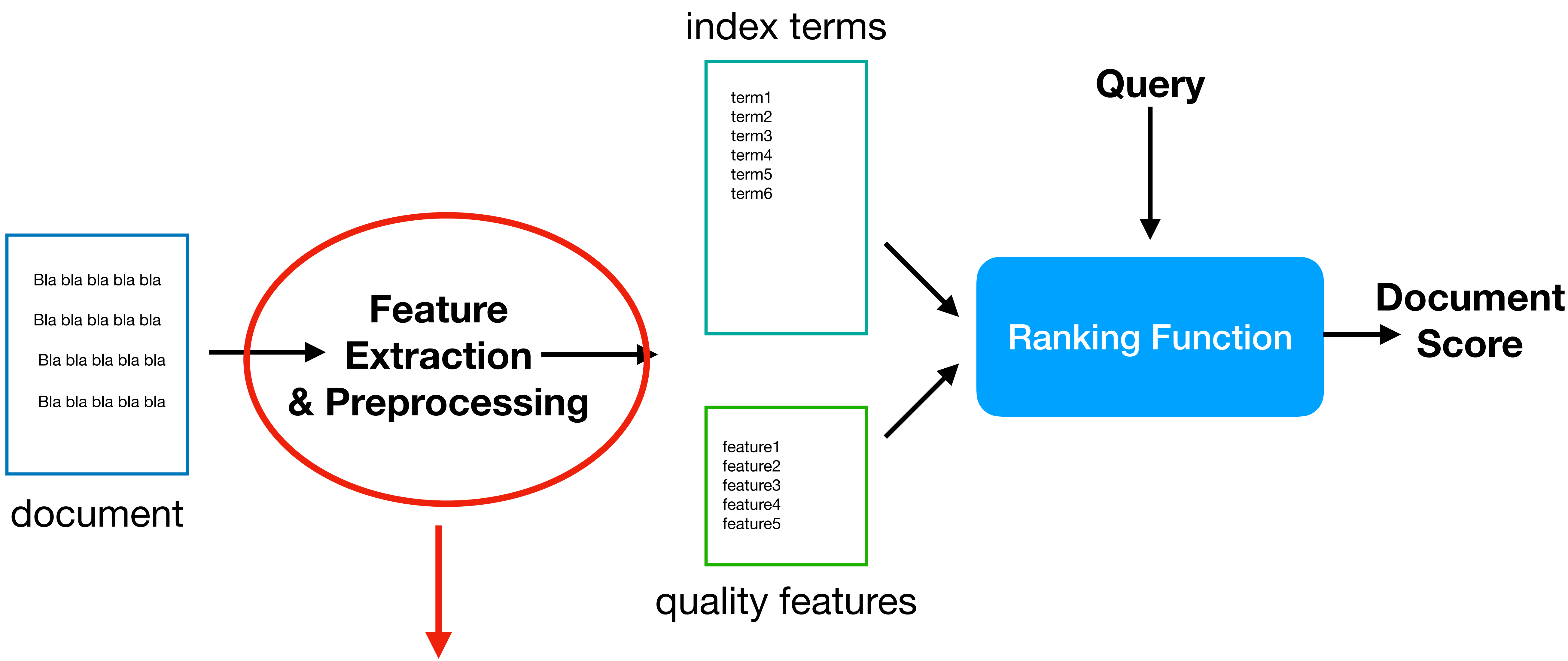
The querying process



An abstract model of search and ranking

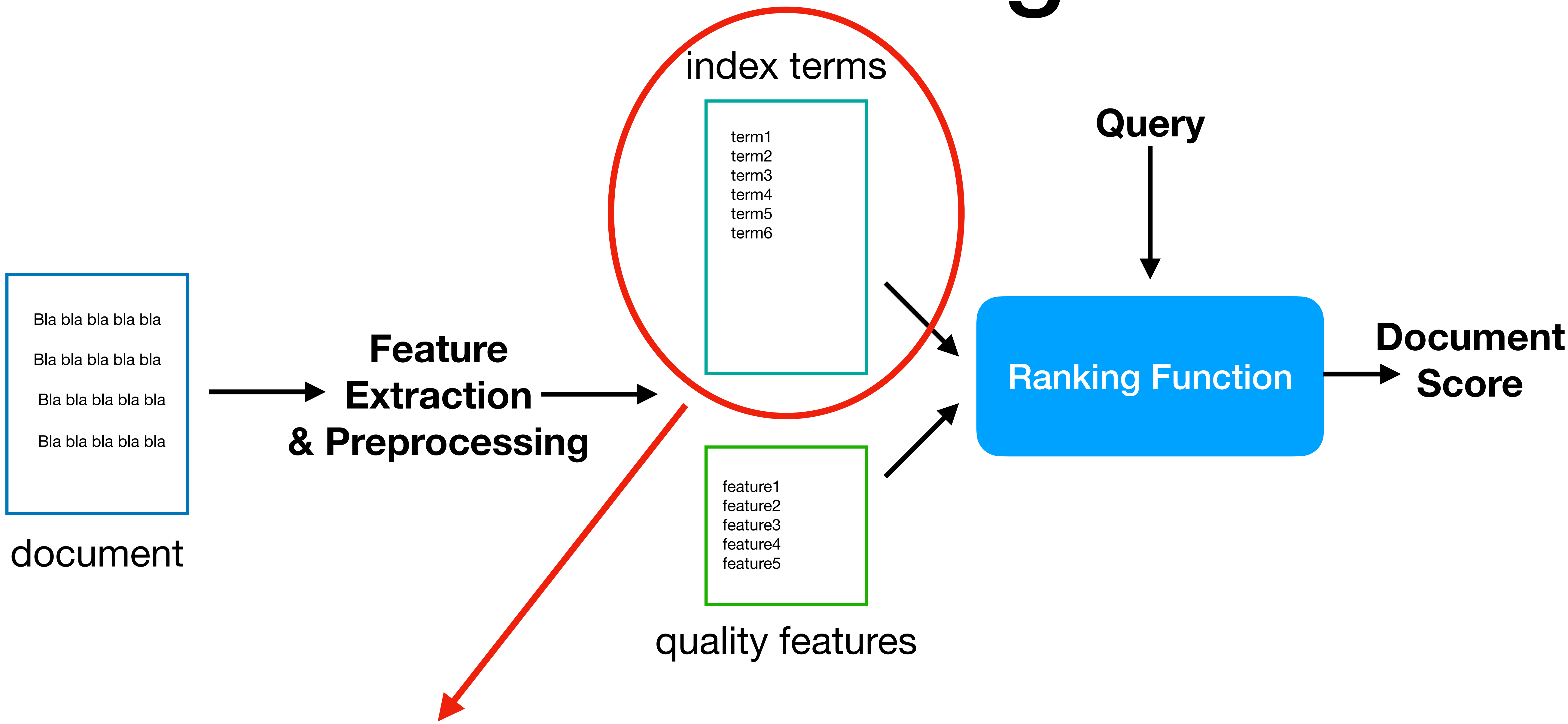


An abstract model of search and ranking



- i.e. text preprocessing for the text component (tokenisation, stemming, etc)
- But also other features, e.g. anchors, last time page was updated, how many pages link to this, etc.

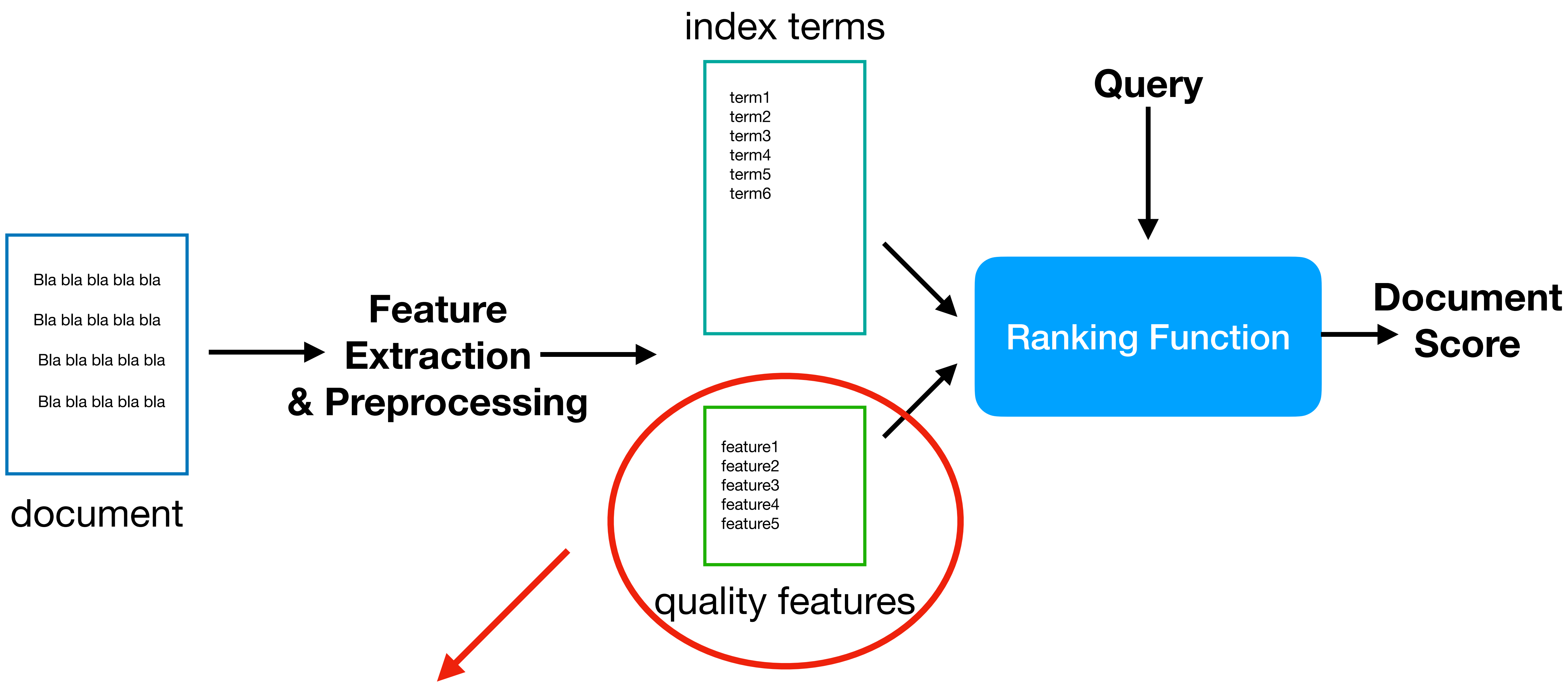
An abstract model of search and ranking



- The data structure where we store the terms - usually **inverted index** (other structure possible though, e.g. signature files)

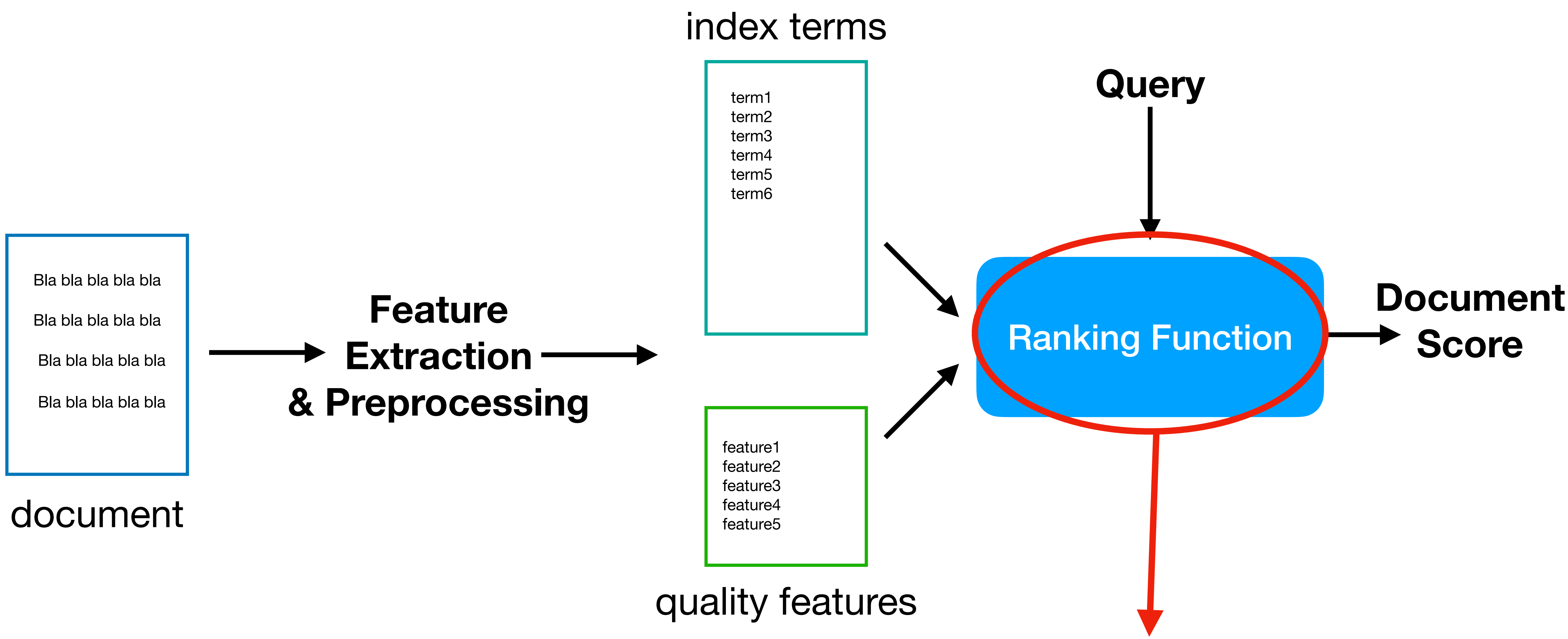
- Usually contains, along with term: frequency info, positional info

An abstract model of search and ranking



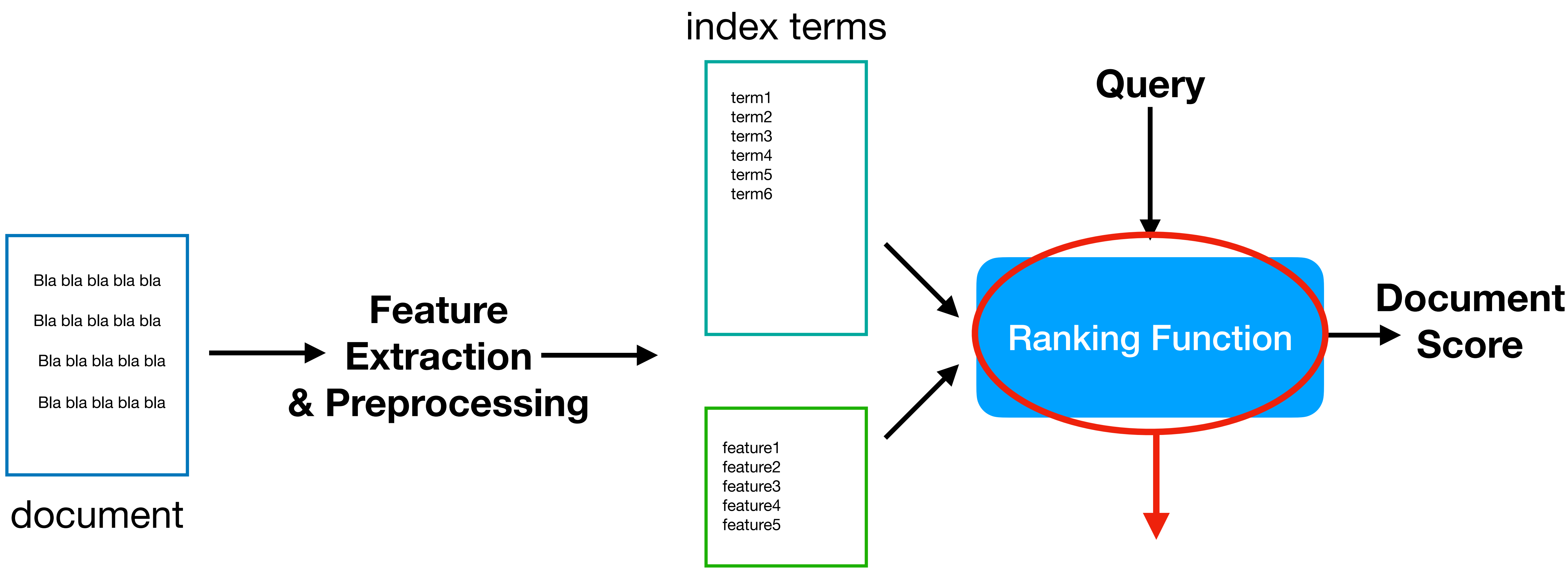
- Contains features like anchors, last time page was updated, etc
- A feature is an attribute of the document that is expressed numerically

An abstract model of search and ranking



- Defines how the document is scored against the query
- Usually depends from features: index terms, but also quality features
- Considers also query features

An abstract model of search and ranking



$$R(Q, D) = \sum_i g_i(Q) f_i(D)$$

- *f(.)* and *g(.)* are feature functions: *f* for documents, *g* for queries
- Usually millions of features for documents; but very few for queries
- Computationally then, we only consider those features for which *g(Q)* is not zero

Evaluation

- What is evaluation?
    - Measure the **effectiveness**, **efficiency** and **cost** of a system
- Search **Effectiveness**: how **good** a system is in retrieving relevant documents
- Search **Efficiency**: how **fast** a system is in retrieving documents
- Often there is **trade-off** between effectiveness and efficiency
- **Cost**: how much does it cost to run the system ($$, Kw/h, etc)
    - Usually cost is determined by the desired level of effectiveness and efficiency

Evaluation

- Why do we want to evaluate a system?
    - Say whether the system is **any good**
    - **Compare** two systems, so as to choose the “best” (or best fit)
    - Understand where the system **succeeds** and where it fails (**diagnostic**)

Test your evaluation intuition

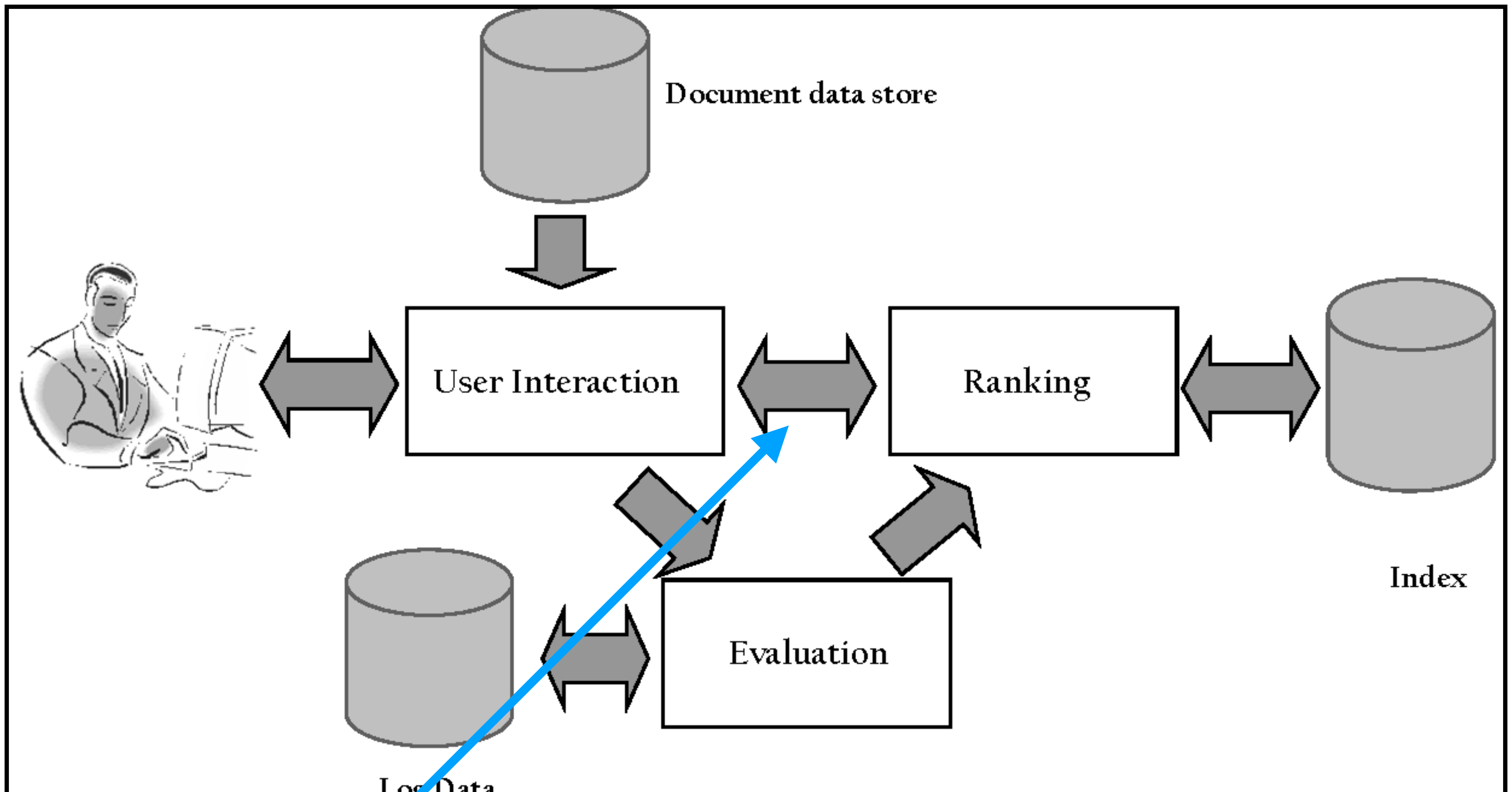
Which SERP is better?

SERP one	SERP two
Bad	Good
Bad	Bad
Bad	Bad
Good	Bad
Good	Bad
Good	Bad

Efficiency Measures

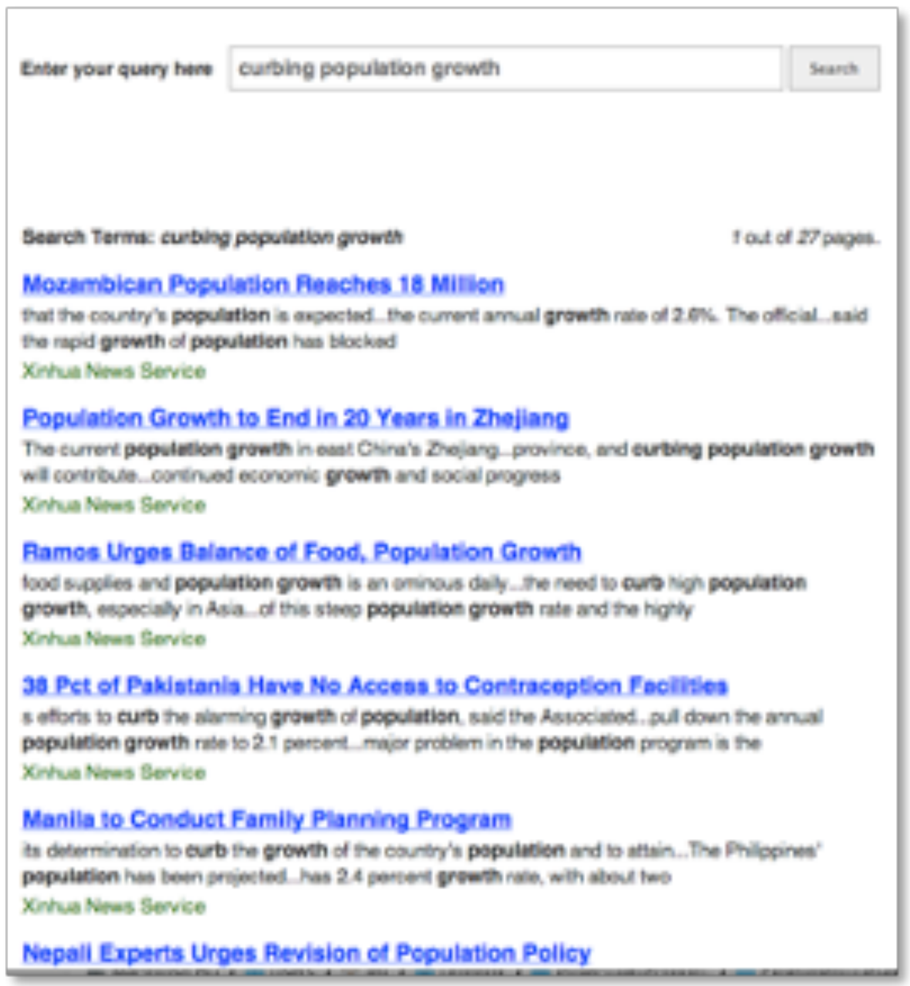
Query Latency

Search Process



Query latency: Amount of time a user must wait to receive a response to their query

Our approximation of a SERP



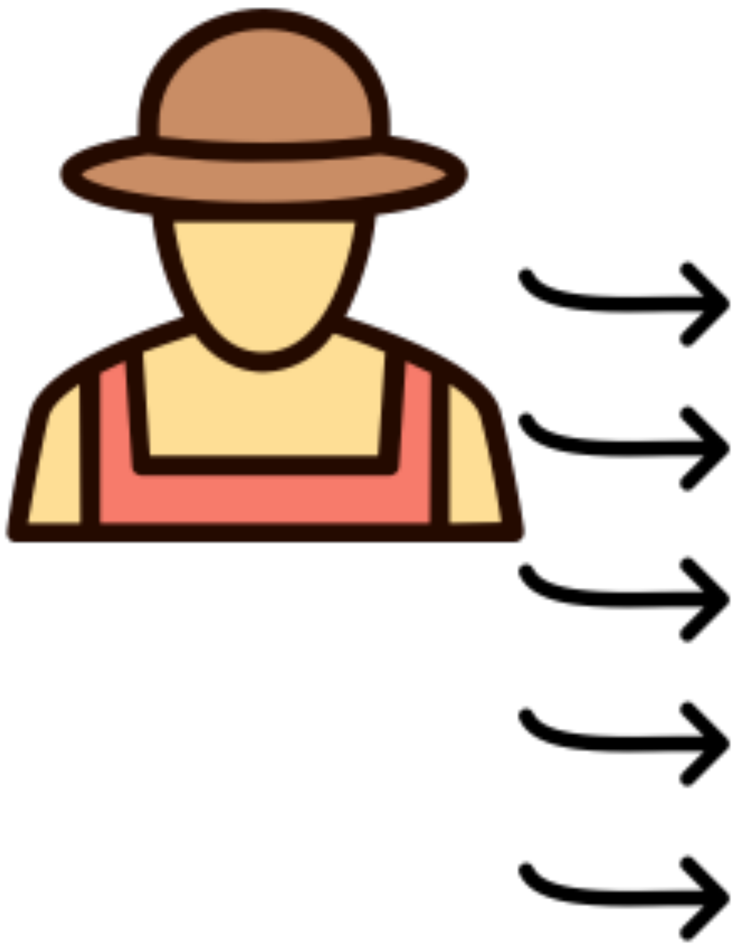
sort of approximates

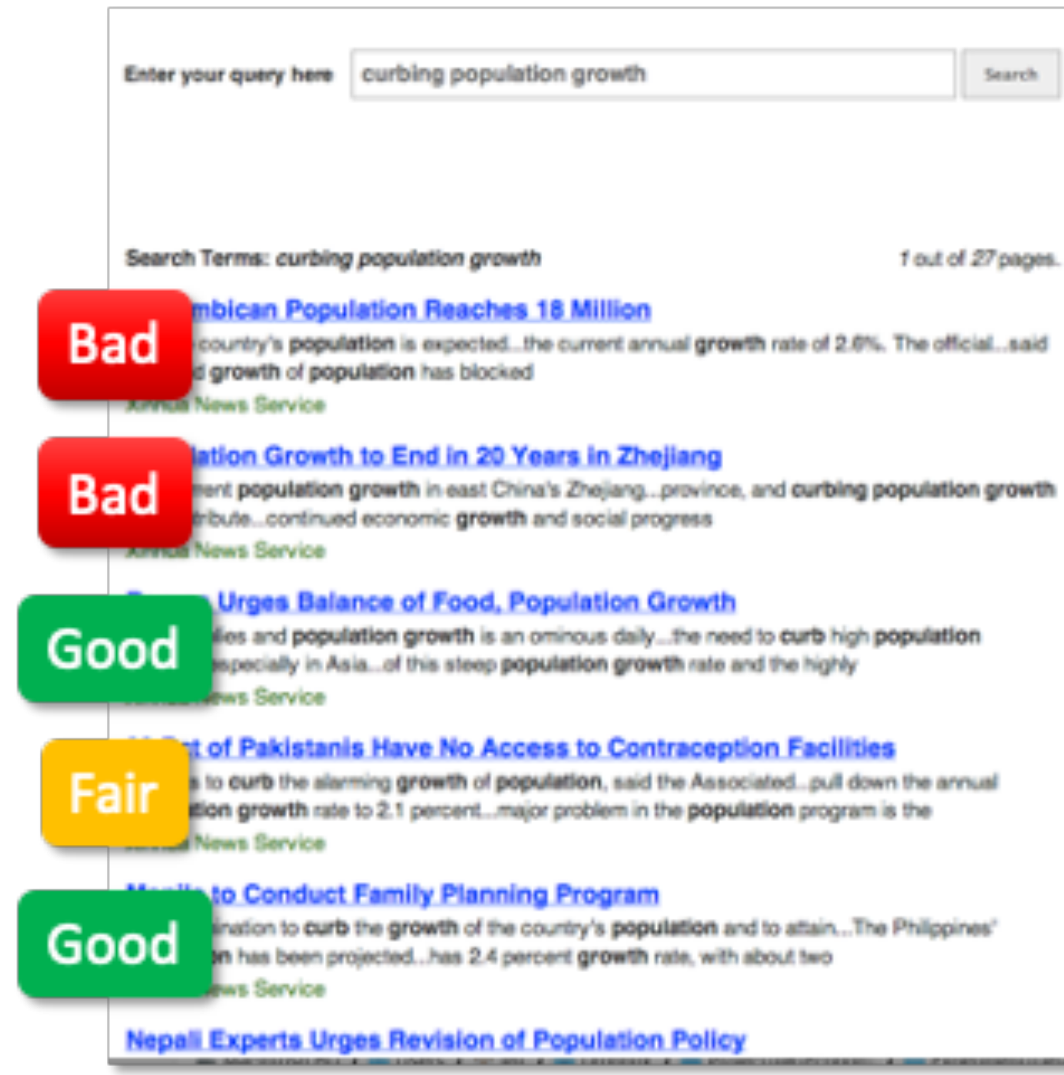
Ranked list, top down

Homogenous elements

Web SERP

How do we model how to assess a ranking





Weight × Gain

Framework for evaluation: Cranfield/TREC

- In practice, how do we go about using these measures?
- The Cranfield/TREC experiments:
- Formalise **a way to experimentally evaluate IR systems**
- Predicates the development of **test collections** to measure IR effectiveness
    - A set of **queries**: sufficiently large & representative
    - A set of **documents**: large & representative
    - A set of relevance **assessments** for query-doc pairs: need for completeness/exhaustivity?

TREC Topic Example

<top>
<num> Number: 794

<title> pet therapy

<desc> Description:
How are pets or animals used in therapy for humans and what are the benefits?

<narr> Narrative:
Relevant documents must include details of how pet- or animal-assisted therapy is or has been used. Relevant details include information about pet therapy programs, descriptions of the circumstances in which pet therapy is used, the benefits of this type of therapy, the degree of success of this therapy, and any laws or regulations governing it.

</top>

Relevance Assessments

- Obtaining relevance assessments is an expensive, time-consuming process
    - who does it?
    - what are the instructions?
    - what is the level of agreement?
- TREC judgments
    - depend on task being evaluated
    - Early collections had binary assessments; recent ones are graded
    - agreement good because of “narrative”

A qrel file

101 0 AP880212-0047 1
101 0 AP880219-0139 0
101 0 AP880219-0166 0
101 0 AP880222-0172 0
101 0 AP880223-0104 0
101 0 AP880229-0146 0
101 0 AP880314-0113 0
101 0 AP880314-0121 0
101 0 AP880314-0145 0
101 0 AP880320-0041 0
101 0 AP880321-0117 0
101 0 AP880323-0210 0
101 0 AP880323-0211 0
101 0 AP880324-0256 0
101 0 AP880326-0149 0
101 0 AP880329-0195 0
101 0 AP880329-0201 0
101 0 AP880330-0014 1
101 0 AP880330-0182 0
101 0 AP880404-0207 0
101 0 AP880414-0171 0
……………………

A TREC result file

101	Q0	WSJ870226-0091	1	0.7194	Brkly3
101	Q0	WSJ861216-0134	2	0.7078	Brkly3
101	Q0	AP890130-0077	3	0.7005	Brkly3
101	Q0	WSJ880523-0063	4	0.6999	Brkly3
101	Q0	WSJ881007-0136	5	0.6932	Brkly3
101	Q0	AP881030-0049	6	0.6912	Brkly3
101	Q0	AP880714-0012	7	0.6844	Brkly3
101	Q0	AP890426-0036	8	0.6844	Brkly3
101	Q0	AP881024-0011	9	0.6800	Brkly3
101	Q0	AP880608-0123	10	0.6766	Brkly3
101	Q0	WSJ870408-0045	11	0.6745	Brkly3
101	Q0	AP880314-0145	12	0.6743	Brkly3
101	Q0	AP890717-0130	13	0.6683	Brkly3
101	Q0	WSJ870715-0122	14	0.6663	Brkly3
101	Q0	AP891215-0115	15	0.6651	Brkly3
101	Q0	WSJ880712-0128	16	0.6614	Brkly3
101	Q0	AP890718-0020	17	0.6609	Brkly3
101	Q0	AP880611-0055	18	0.6601	Brkly3
101	Q0	DOE1-76-0712	19	0.6598	Brkly3
101	Q0	AP880610-0262	20	0.6585	Brkly3

……………………………..

Precision, recall

- Precision: The ability to retrieve documents that are relevant.

$$precision = \frac{Number\ \ of\ \ relevant\ documents\ \ retrieved}{Total\ number\ of\ \ documents\ \ retrieved}$$

- Recall: The ability to retrieve all of the relevant documents in the corpus.

$$recall = \frac{Number\ of\ \ relevant\ documents\ retrieved}{Total\ number\ of\ \ relevant\ documents}$$

Many other Evaluation Measures

- Please see Evaluation in IR playlist: https://www.youtube.com/watch?v=dEYM2euPtUY&list=PLCg0q-84NyLGReuSwhf7SaiDYoXPDH3Ek
    - Skip videos related to C/W/L
    - Pay particular attention to P@n, MAP, nDCG

How do we retrieve&rank docs given a q?

- Docs and queries are text
- For matching, we follow a (mathematical) model that formalises
    - How docs and queries are represented
    - How matching between doc and query occur
- These matching functions are often based on
    - Statistics of text in collection: term frequency, inverse document frequency
    - Statistical similarity of words

BM25

- Popular and effective ranking algorithm **based on the binary independence model** (derivation BIM -> BM25 in Croft book)
- **Expands the tf-idf idea** (under the most general form of no relevance information provided), with two main difference
    - Term frequency **saturation**
    - Document length **normalisation**
- Empirically has been shown to be a quite **reliable and robust** model, which works well out-of-the-box in most situations
- Default retrieval model in many open source search engines, e.g. Lucene, Elasticsearch

Components of BM25: RSJ weight

$$\sum_{i=1}^{|Q|} \boxed{log\left(\frac{(r_i+0.5)/(R-r_i+0.5)}{(n_i-r_i+0.5)/(N-n_i-R+r_i+0.5)}\right)} \cdot \frac{(k_1+1)f_i}{k_1B+f_i} \cdot \frac{(k_2+1)qf_i}{k_2+qf_i}$$

This is known as the Robertson-Sparck Jones weight

- It considers **knowledge** about number of relevant documents (*R*) and number of relevant documents that contain term *i* (r_i)
- N and n_i refer to the number of documents that have been judged to obtain R and r_i
- Thus, if **no relevance** information is provided, it becomes:

$$log\frac{N-n_i+0.5}{n_i+0.5} \longleftarrow$$ Approximation of the classical IDF

Components of BM25: saturation

$$\sum_{i=1}^{|Q|} log\left(\frac{(r_i+0.5)/(R-r_i+0.5)}{(n_i-r_i+0.5)/(N-n_i-R+r_i+0.5)}\right) \cdot \frac{(k_1+1)f_i}{k_1 B+f_i} \cdot \frac{(k_2+1)qf_i}{k_2+qf_i}$$

This is known as the **saturation component**

- The contribution of the occurrences of term to a document score cannot exceed a saturation point
- **k_1 parameter** controls the saturation, $k_1 > 0$
  - k_1 high -> f_i contributes significantly to score
  - k_1 low -> f_i additional contributions of further term occurrences tail off quickly
  - typically set to 1.2

Components of BM25: doc length

$$\sum_{i=1}^{|Q|} log\left(\frac{(r_i+0.5)/(R-r_i+0.5)}{(n_i-r_i+0.5)/(N-n_i-R+r_i+0.5)}\right) \cdot \frac{(k_1+1)f_i}{k_1 B+f_i} \cdot \frac{(k_2+1)qf_i}{k_2+qf_i}$$

$$B = (1-b) + b \cdot \frac{dl}{avdl}$$

This is known as the **document length normalisation**

- The author of a document decided to write shorter/longer document than the average document. Why?
  verbosity-> prefer shorter, *scope*->prefer longer

- This component provides a **soft normalisation** of doc length

- **b parameter** that controls normalisation, 0 <= b <=1
  - b=1: full normalisation; b=0: no normalisation
  - Typically set to 0.75

- B is used to normalise the term frequency f_i

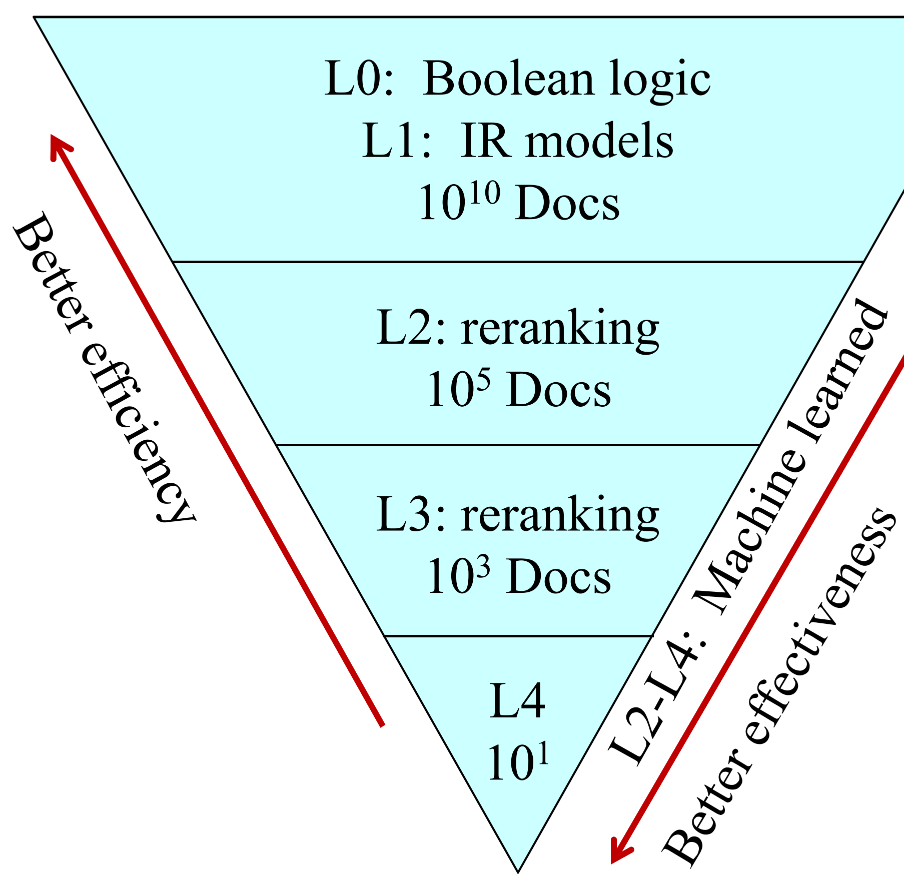
Components of BM25: query

$$\sum_{i=1}^{|Q|} log\left(\frac{(r_i+0.5)/(R-r_i+0.5)}{(n_i-r_i+0.5)/(N-n_i-R+r_i+0.5)}\right) \cdot \frac{(k_1+1)f_i}{k_1B+f_i} \cdot \frac{(k_2+1)qf_i}{k_2+qf_i}$$

This is known as the **within-query component**

- Useful for longer queries where a term may occur multiple times
- Similar saturation as the within-document component
- Has its own constant k_2 (other times referred as k_3)
- Experiments suggested this term is not important, i.e. simply treat multiple occurrences of a term in a query as different terms

Cascade of Rankers



- Cascade architecture of rankers
- Each layer ranks, prunes, and returns results
- Layered evaluation gives control over search costs
- Simpler models are applied to massive data
    - Efficient
- Sophisticated models are applied to little data
    - Effective

Sparse Representations

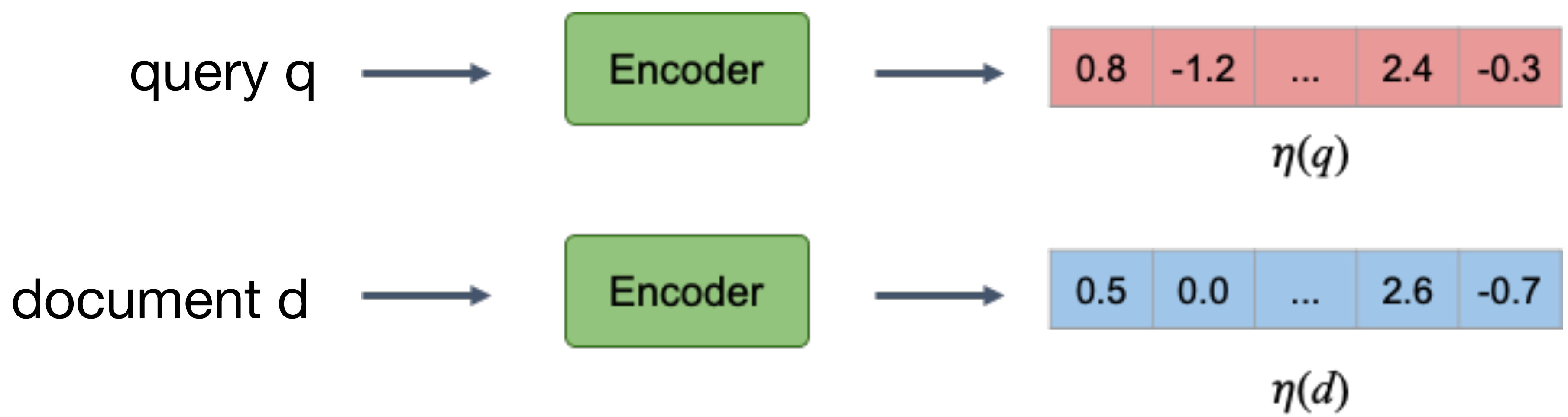
- The document and the query are represented in a sparse space, e.g. that of all terms in the vocabulary
  - BOW is sparse
  - VSM is sparse

$$\mathrm{BM25}(q,d) = \sum_{\boxed{t \in q \cap d}} \log \frac{N - \mathrm{df}(t) + 0.5}{\mathrm{df}(t) + 0.5} \cdot \frac{\mathrm{tf}(t,d) \cdot (k_1 + 1)}{\mathrm{tf}(t,d) + k_1 \cdot \left(1 - b + b \cdot \frac{l_d}{L}\right)}$$

- Advantages:
  - Fast to retrieve candidates from inverted index because q is usually short
  - Fast to compute because q ∩ d is usually small
- Disadvantage: Terms need to match exactly

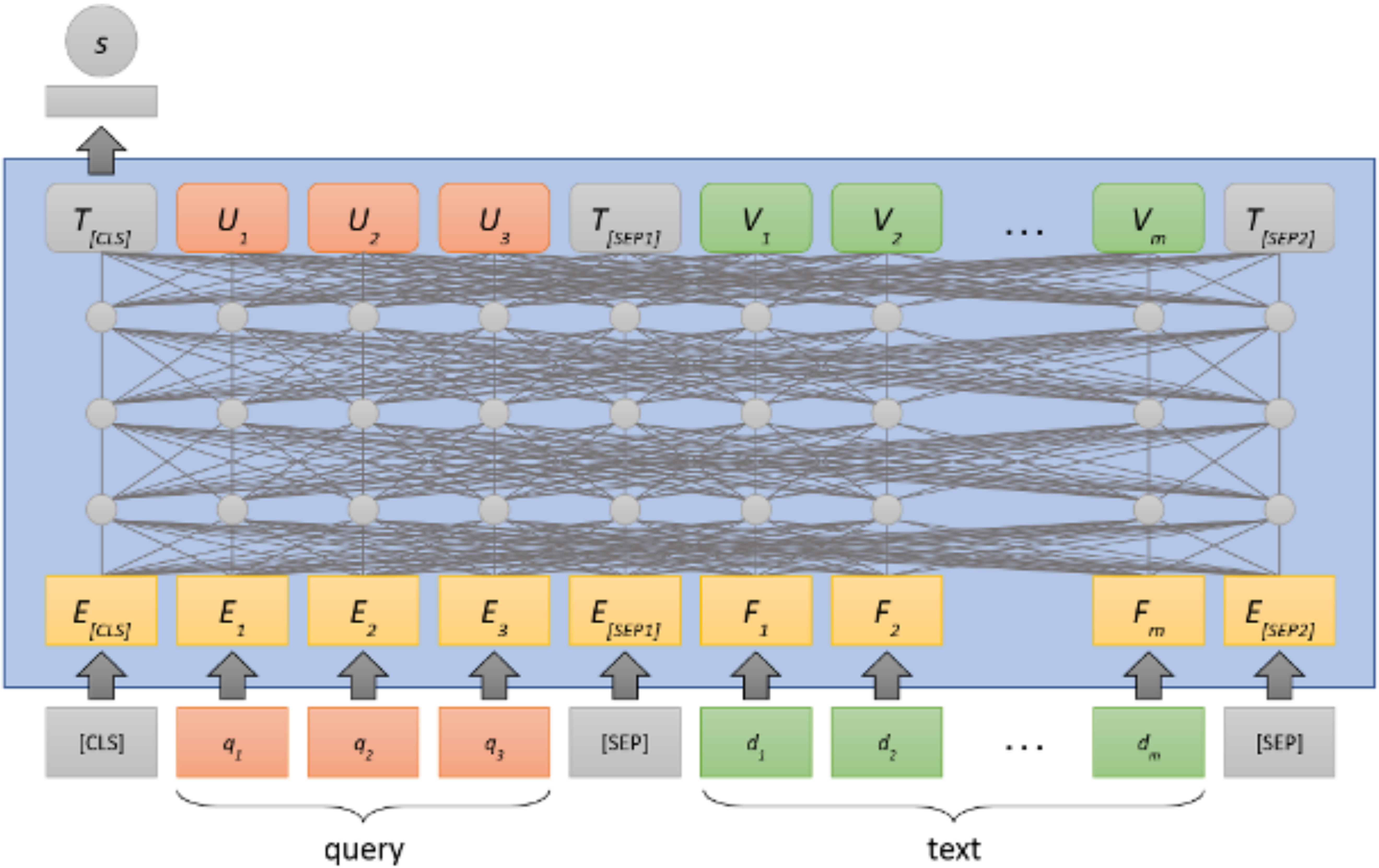
Dense Representations

- Encode document d or query q into a “low” dimensional vector
- Compute the similarity b/w the vector of q and that of d

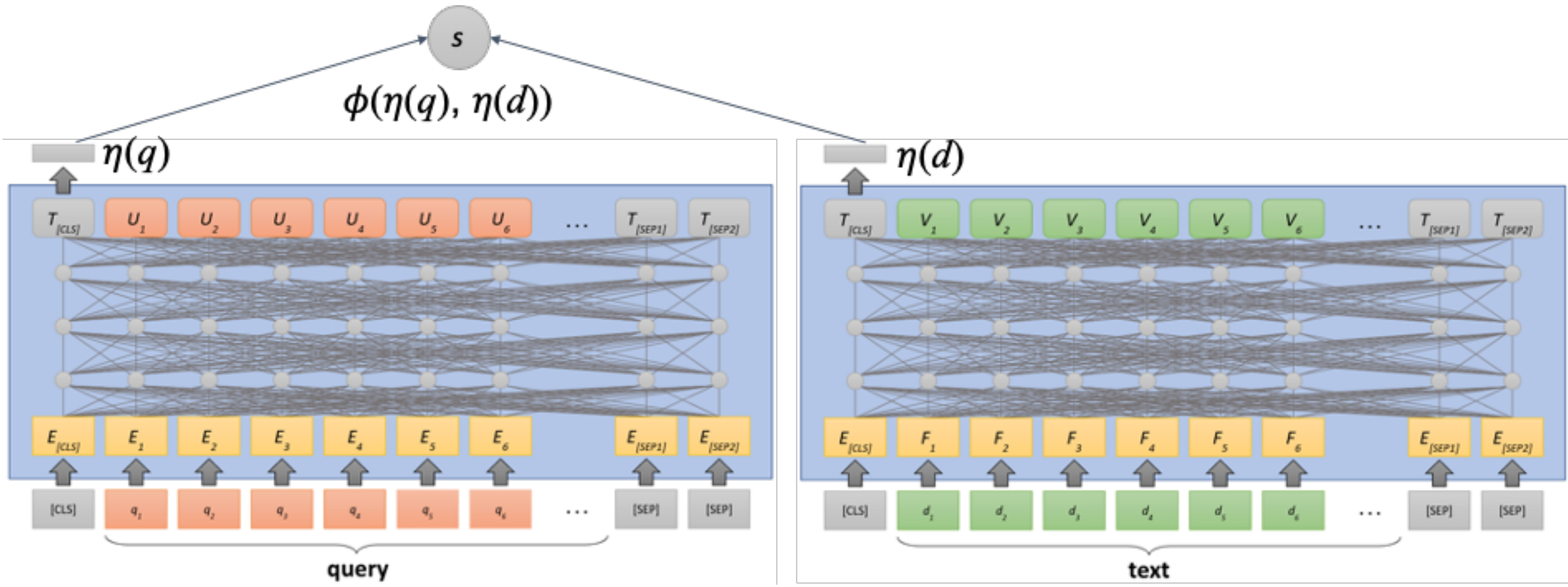


- ϕ is a similarity function (e.g., inner product or cosine similarity)
- $\phi(\eta(\text{q}), \eta(\text{d}))$ → ideally measures how relevant q and d are to each other

Cross-encoder



Bi-encoder



s
$\phi(\eta(q), \eta(d))$
$\eta(q)$
$T_{[CLS]}$
U_1
U_2
U_3
U_4
U_5
U_6
...
$T_{[SEP1]}$
$T_{[SEP2]}$
$E_{[CLS]}$
E_1
E_2
E_3
E_4
E_5
E_6
$E_{[SEP1]}$
$E_{[SEP2]}$
[CLS]
q_1
q_2
q_3
q_4
q_5
q_6
...
[SEP]
[SEP]
query
$\eta(d)$
$T_{[CLS]}$
V_1
V_2
V_3
V_4
V_5
V_6
...
$T_{[SEP1]}$
$T_{[SEP2]}$
$E_{[CLS]}$
F_1
F_2
F_3
F_4
F_5
F_6
$E_{[SEP1]}$
$E_{[SEP2]}$
[CLS]
d_1
d_2
d_3
d_4
d_5
d_6
...
[SEP]
[SEP]
text

Dense Retrievers: key characteristics

- **Representation**: choose how to represent a document/ query, e.g. CLS token embedding, all tokens embeddings, …
- **Similarity function**: how to compare dense vectors?
- **Fine-tuning**: change BERT weights through learning the similarity computation task
  - Which **loss function** to use for fine-tuning?
  - How to select positive/negative **samples**?

More about BERT, Sparse and Dense Retrievers

- Watch the playlists:
- https://www.youtube.com/watch?v=dUHWnpfdho0&list=PLCg0q-84NyLG2s-rJH7laqN2eUOqddQQs
- https://www.youtube.com/watch?v=4h6DlydWUdM&list=PLCg0q-84NyLFfkDJ_lJudX-TfMPJZR0Of