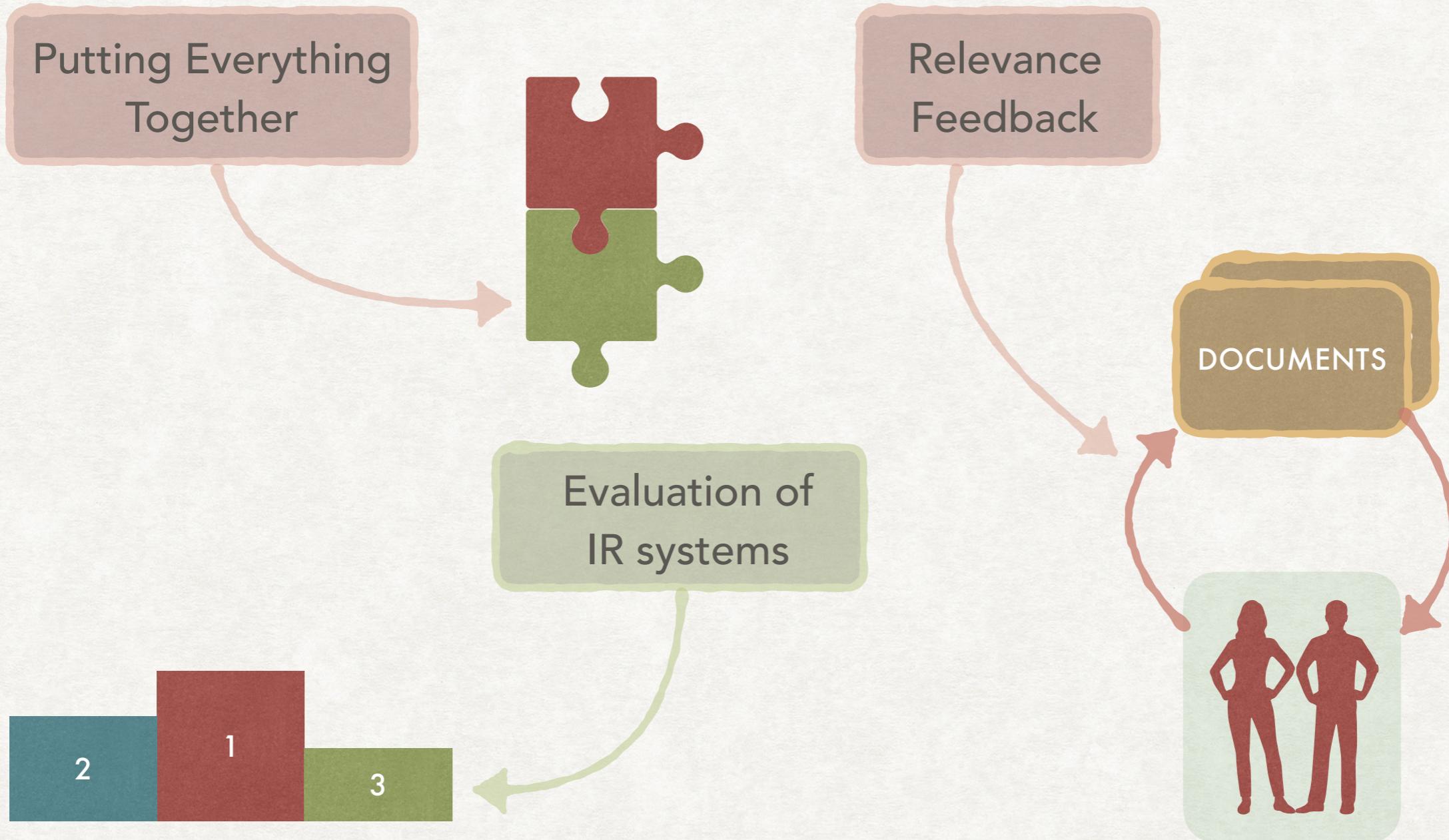


INFORMATION RETRIEVAL

Luca Manzoni
lmanzoni@units.it

LECTURE OUTLINE

* SIDE EFFECTS MAY INCLUDE SIDE EFFECTS



INTEGRATING EVERYTHING

TIERED INDEXES

GENERALISATION OF CHAMPION LISTS



We search for K documents in the rank 1 index,
if we have less than K we continue in the rank 2 index, and so on

QUERY TERM PROXIMITY

TOWARDS A “SOFT CONJUNCTIVE” SEMANTICS

- If we have a query $q = t_1 \ t_2 \ \dots, t_k$ we might want to give a higher score to documents in which the three terms appears close to each other.
- This is not a phrase query, but if the terms appears in close proximity the documents might be an indication that the document is more relevant.
- Let ω the length of the window (in term of number of words) in which t_1, t_2, \dots, t_k all appear.

QUERY TERM PROXIMITY

TOWARDS A “SOFT CONJUNCTIVE” SEMANTICS

Query: CAT XYLOPHONE

$$\omega = 5$$

Document 1: THE CAT JUMPED ON THE XYLOPHONE

$$\omega = \text{a lot more than } 5$$

Document 2: CAT: NOUN, A FELINE [...] XYLOPHONE: NOUN, AN [...]

How can we use ω in our scoring function?

- Hand-coding a scoring function using ω
- As an additional linear term whose weight we can learn from training samples

BOOLEAN RETRIEVAL

HOW TO PERFORM IT IN THE VECTOR SPACE MODEL

- We can use the vector space representation to perform Boolean retrieval:
- A document d is inside the set of documents denoted by t iff $\vec{v}(d)_t > 0$ (i.e., if the entry t of the vector of d is positive).
- The reverse is not true: the Boolean model does not keep trace of frequencies.
- The two models are different in a more fundamental way: in the Boolean model the queries are written to *select documents*, in the vector space model queries are a form of *evidence accumulation*.

WILDCARD QUERIES

CAN WE IMPLEMENT IT IN THE VECTOR SPACE MODEL?

- In most cases wildcard queries need an additional (and separate) index.
- We can return, from that index, the set of terms that satisfy the wildcards present in the query.
- Suppose that we have CAT* as a query. We obtain the terms “CAT”, “CATASTROPHE”, and “CATERPILLAR”.
- How can we score a document?
- We simply consider the three terms as “normal” query terms: if a document contains all three of them then it will probably be more relevant.

PHRASE QUERIES

PHRASES IN A “BAG OF WORDS” MODEL

- In the vector space model our documents are “bags of words”, without any ordering, while in phrase queries the ordering is important.
- The two models are, in some sense, incompatible: a bag of words model cannot be directly used for phrase queries.
- They can still be combined in some meaningful way:
 - Perform the phrase query and rank only the documents returned by the query.
 - If less than K documents are present then “reduce” the share query and start again.

EVALUATION OF IR SYSTEMS

STANDARD TEST COLLECTIONS

STANDARD BENCHMARKS

CRANFIELD COLLECTION

ONE OF THE OLDEST, NOW TOO SMALL.
1398 ABSTRACTS OF AERODYNAMICS
JOURNAL ARTICLES AND 225 QUERIES.

TREC

(TEXT RETRIEVAL CONFERENCE)

NOT A SINGLE COLLECTION. THERE IS A
RANGE OF TEXT COLLECTIONS ON
DIFFERENT TOPICS.
SEE : [HTTPS://TREC.NIST.GOV](https://trec.nist.gov)

REUTERS

REUTERS-21578 (21578 DOCUMENTS) AND
REUTERS-RCV1 (806791 DOCUMENTS)
COLLECT A LARGE NUMBER OF NEWSWIRE
ARTICLES

Also see: http://ir.dcs.gla.ac.uk/resources/test_collections/

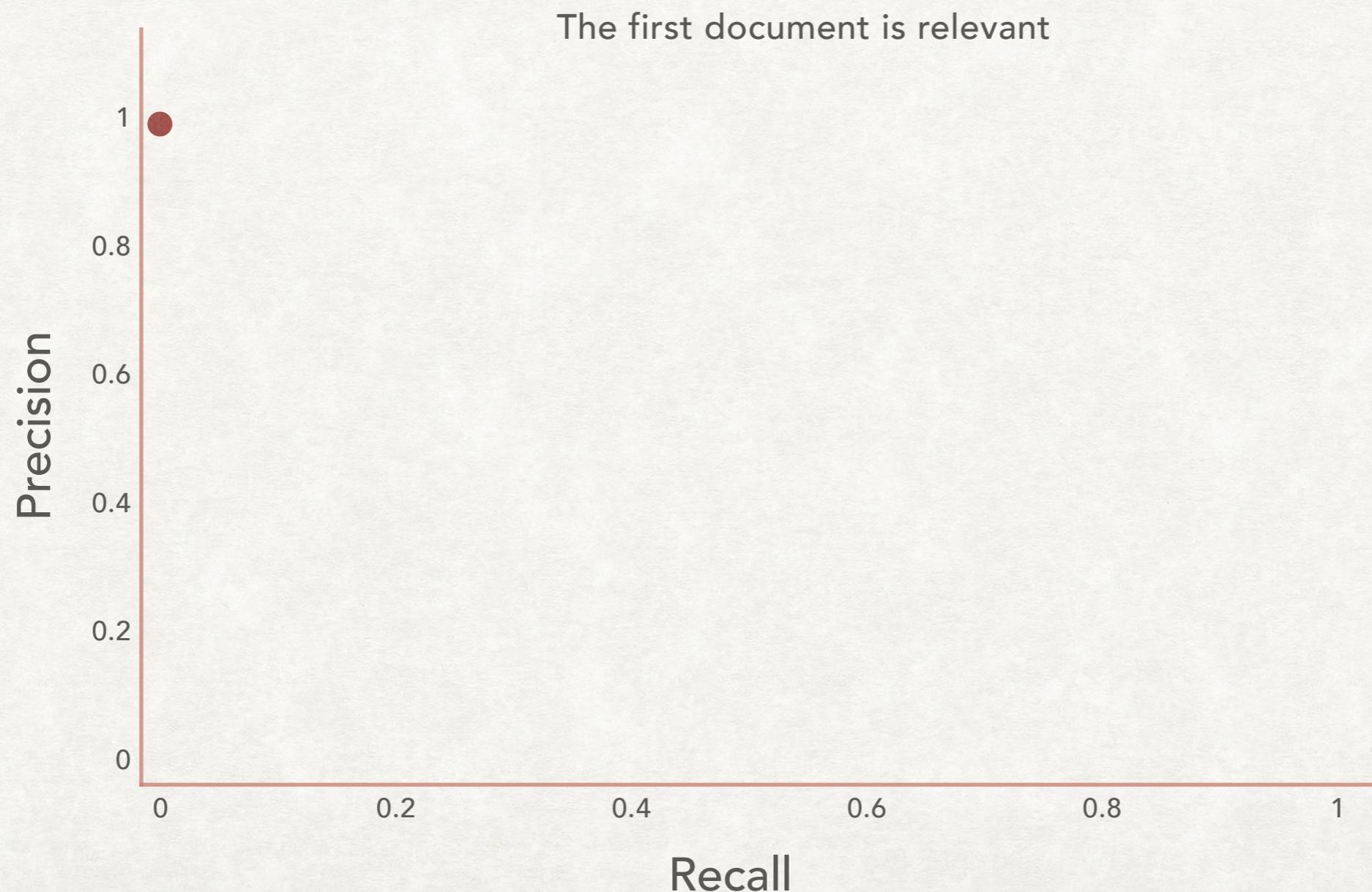
RANKED RETRIEVAL

HOW TO COMPUTE PRECISION AND RECALL?

- We usually evaluate the effectiveness of a IR system with precision and recall (other measures are also possible)...
- ...and this works well with *unranked* results.
- How can we extend it to *ranked* results, where position is important?
 - Precision-recall curve and interpolated precision
 - Eleven-point interpolated average precision
 - Mean average precision (MAP)
 - Precision at k and R -precision

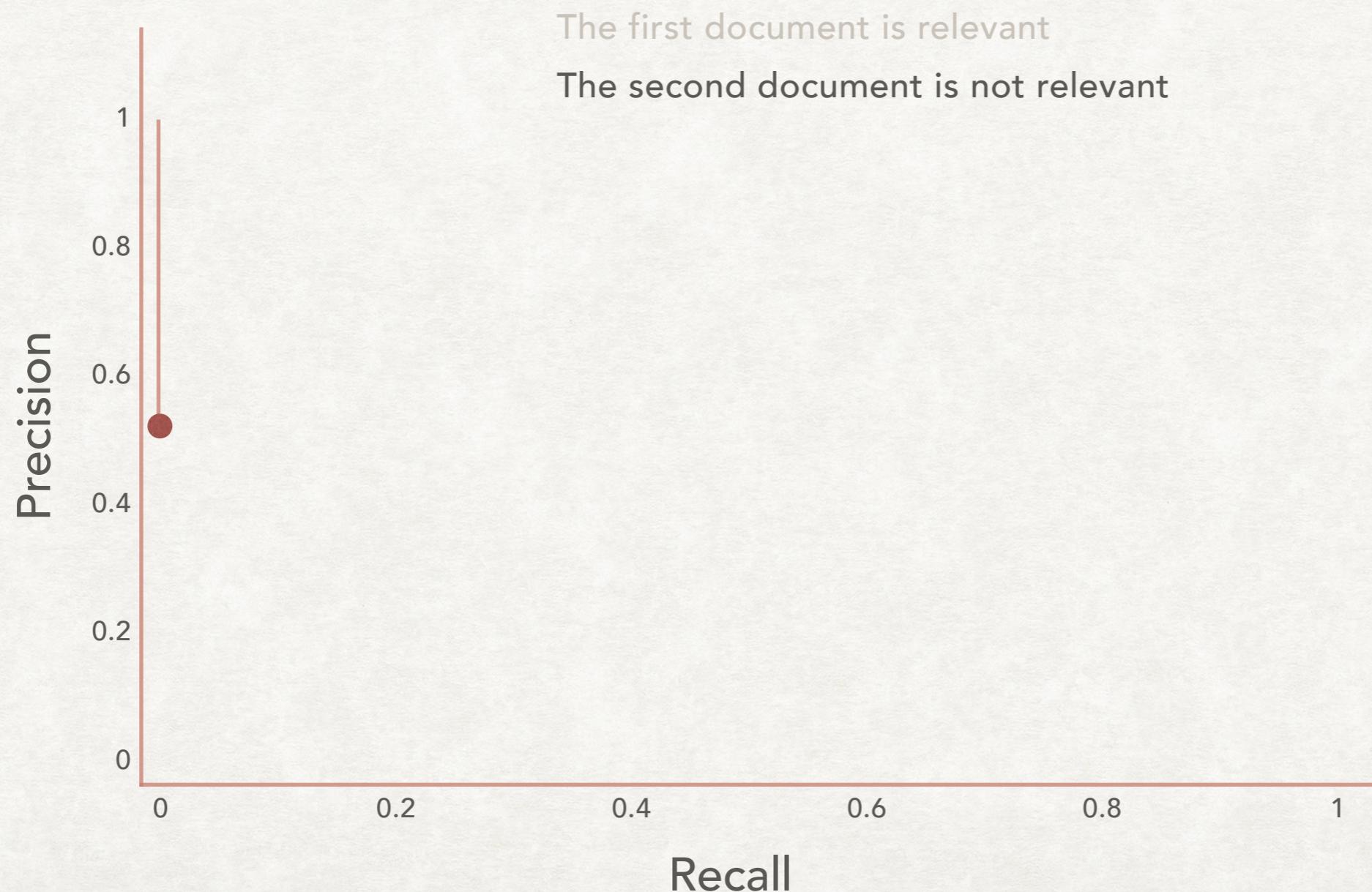
PRECISION-RECALL CURVE

We compute precision and recall for the first 1, 2, 3, 4, etc. retrieved documents:



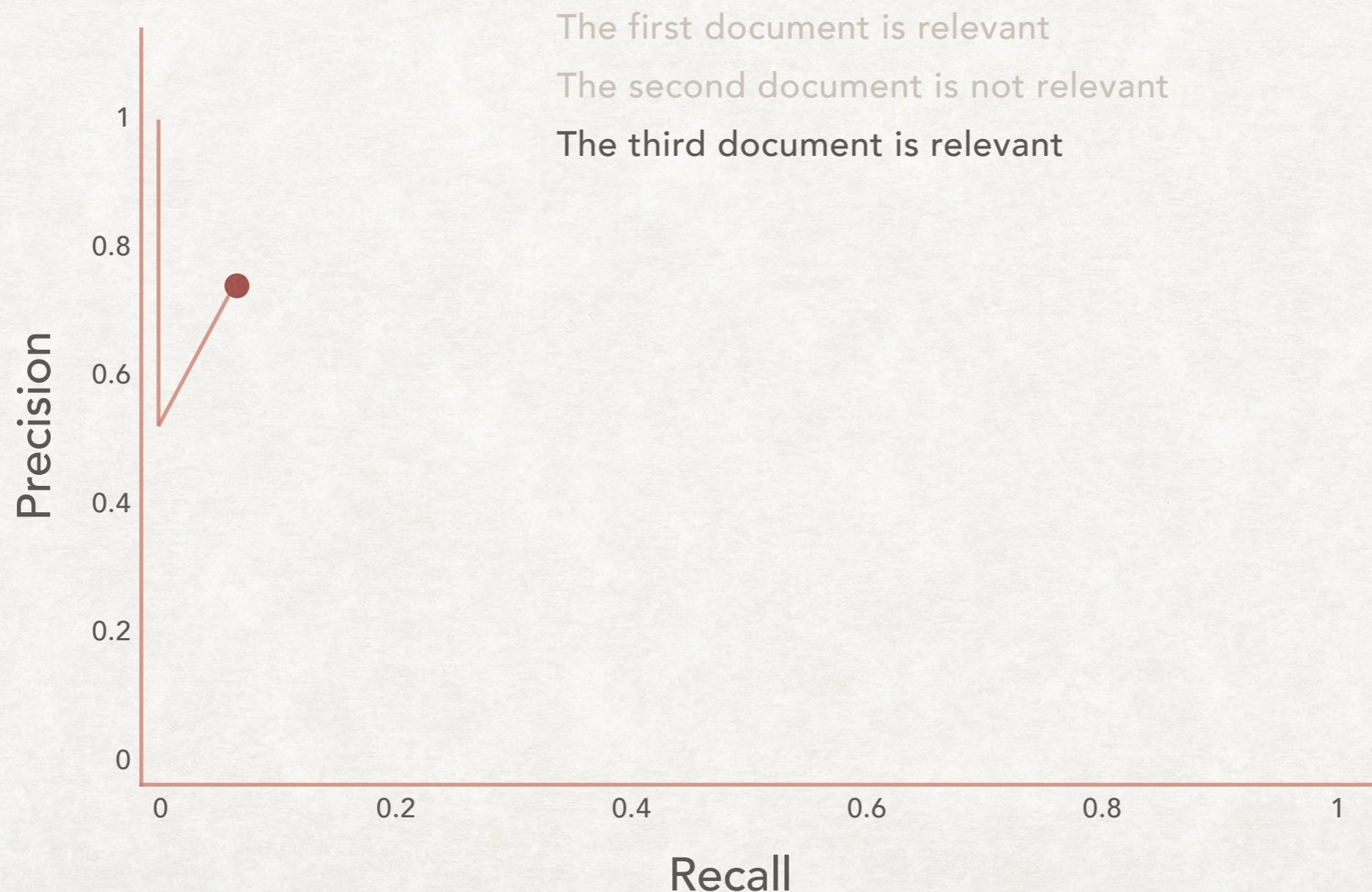
PRECISION-RECALL CURVE

We compute precision and recall for the first 1, 2, 3, 4, etc. retrieved documents:



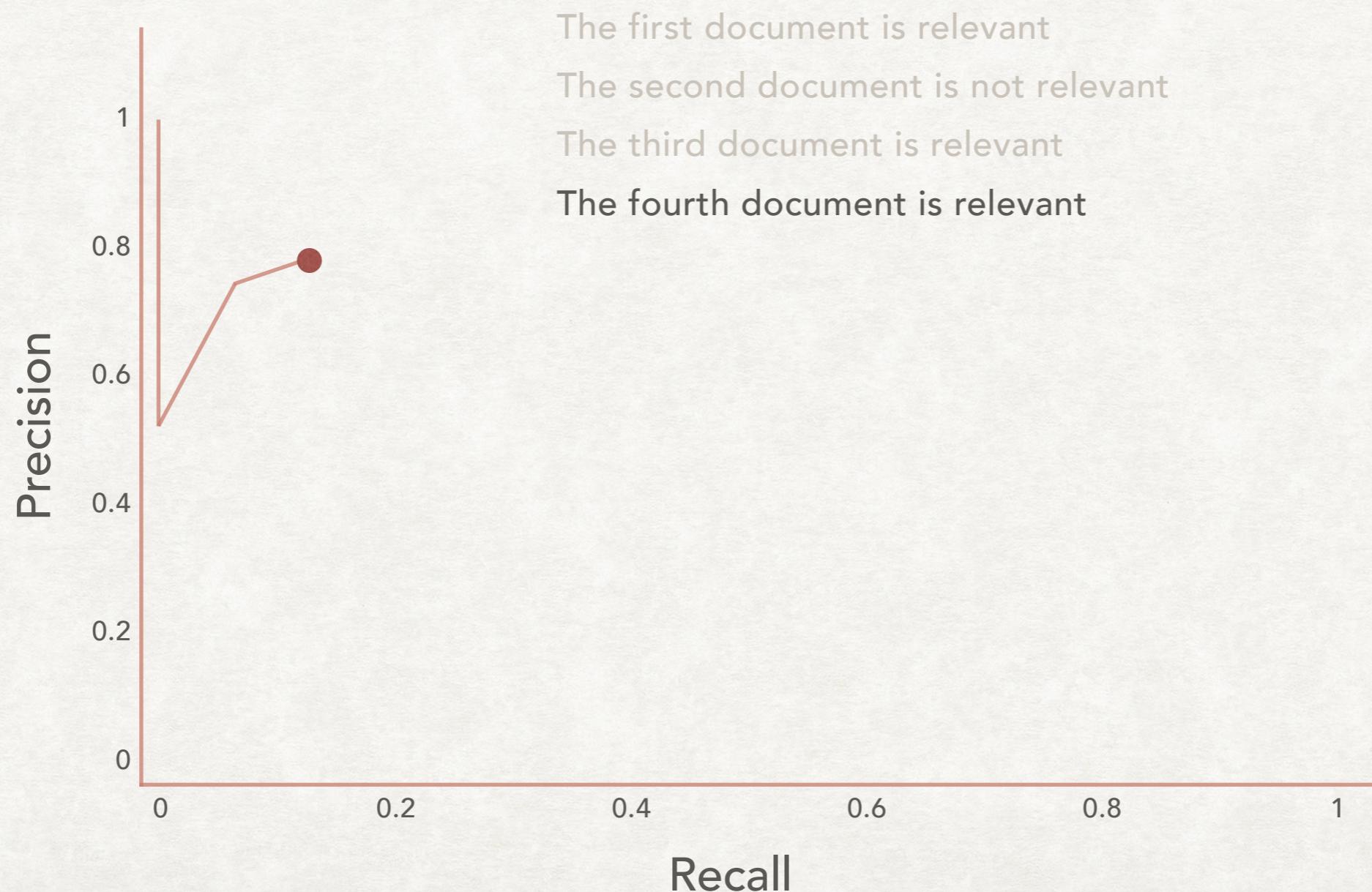
PRECISION-RECALL CURVE

We compute precision and recall for the first 1, 2, 3, 4, etc. retrieved documents:



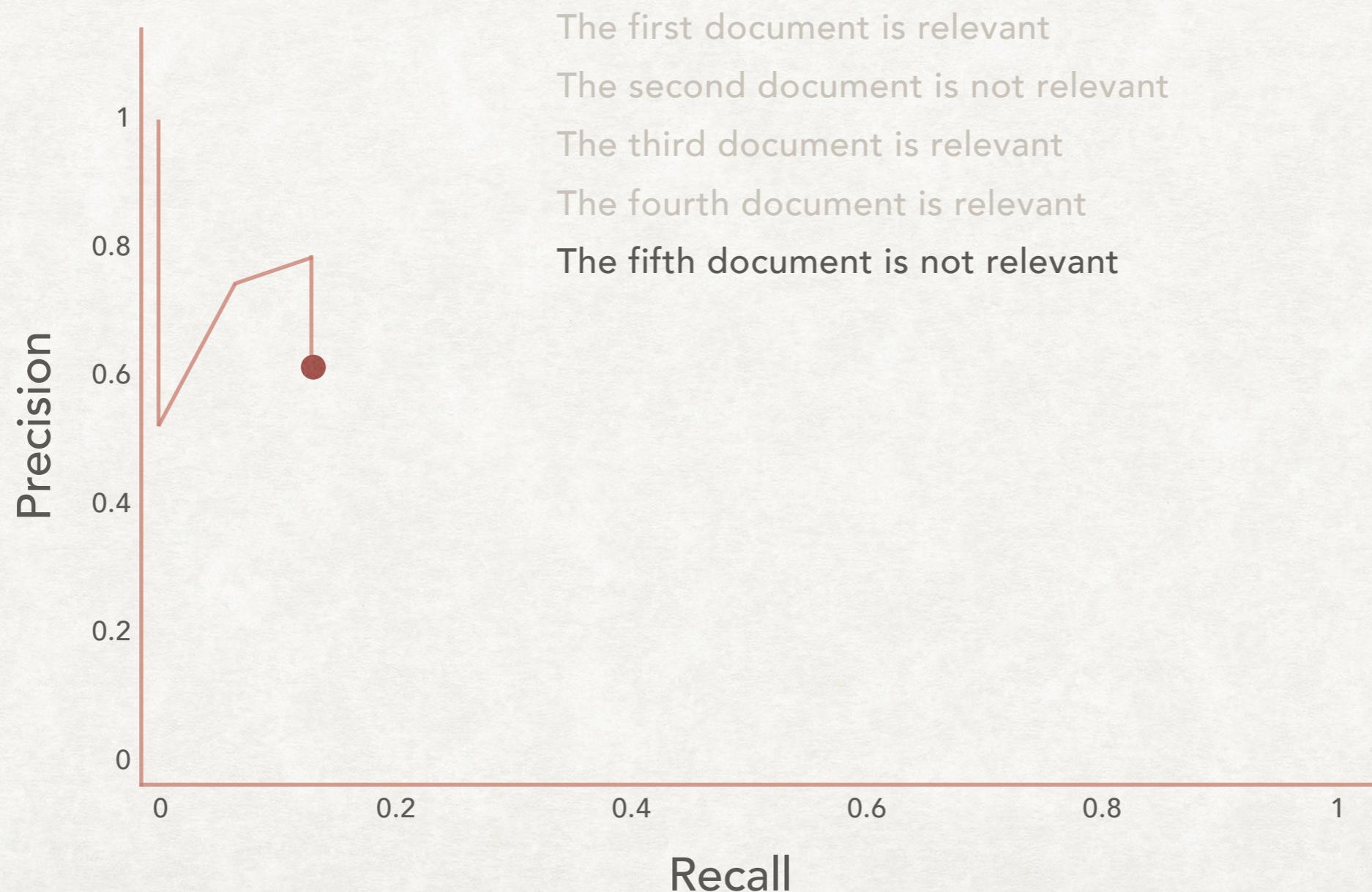
PRECISION-RECALL CURVE

We compute precision and recall for the first 1, 2, 3, 4, etc. retrieved documents:



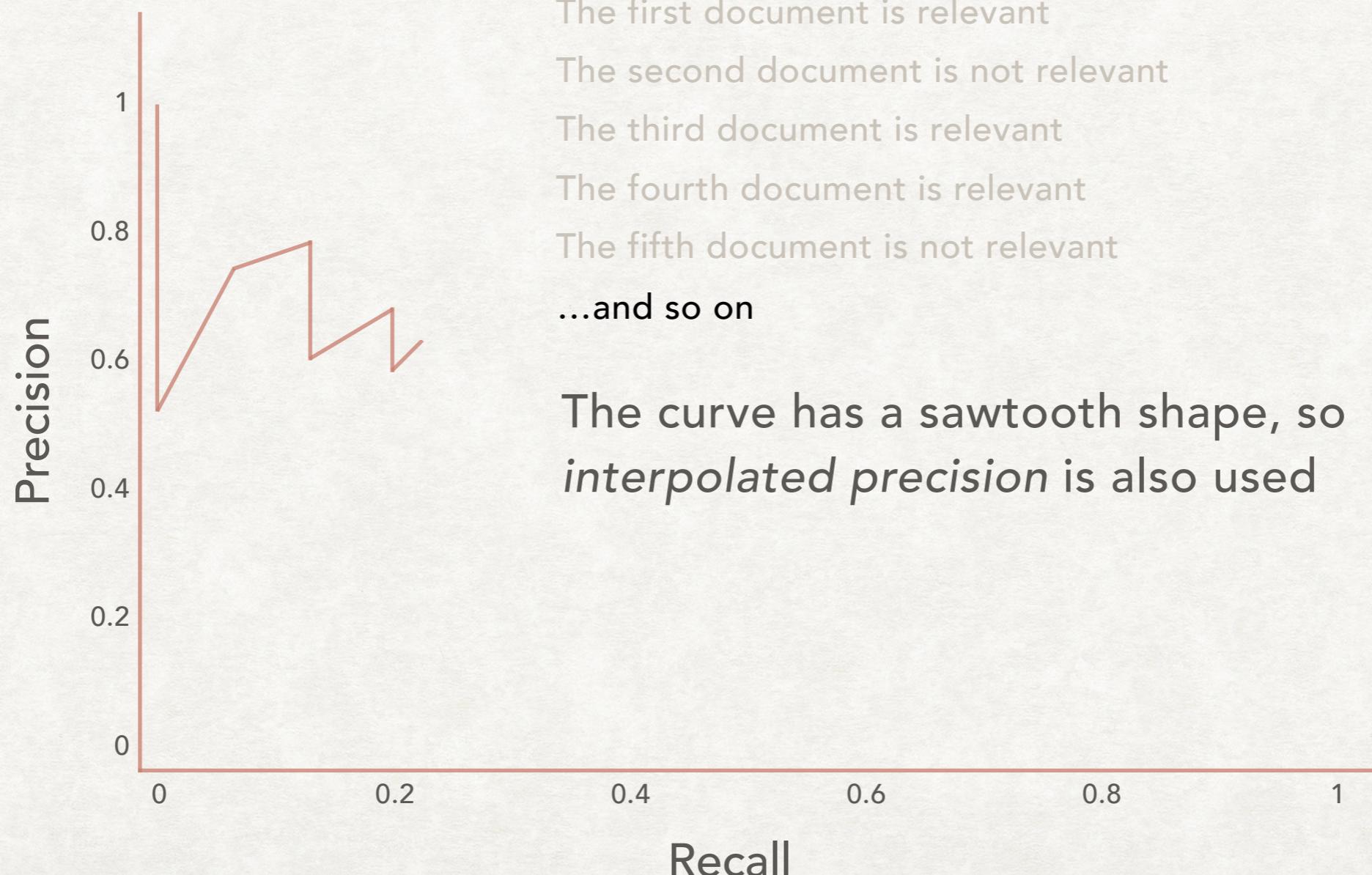
PRECISION-RECALL CURVE

We compute precision and recall for the first 1, 2, 3, 4, etc. retrieved documents:



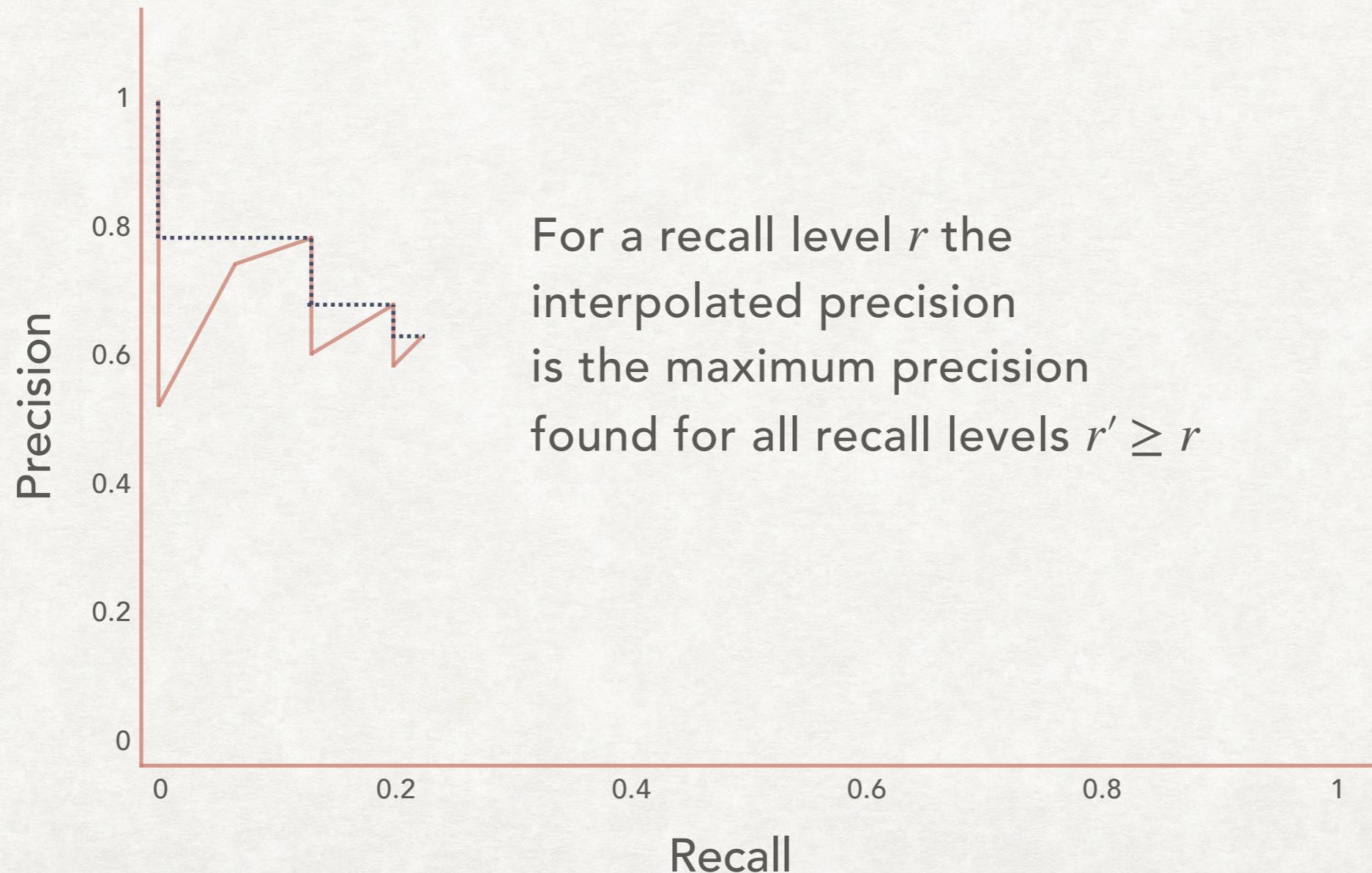
PRECISION-RECALL CURVE

We compute precision and recall for the first 1, 2, 3, 4, etc. retrieved documents:



PRECISION-RECALL CURVE

We compute precision and recall for the first 1, 2, 3, 4, etc. retrieved documents:



ELEVEN POINT INTERPOLATED PRECISION

PRECISION AT ELEVEN RECALL LEVELS

Recall	Precision
0,0	1,0
0,1	0,73
0,2	0,64
0,3	0,58
0,4	0,51
0,5	0,45
0,6	0,38
0,7	0,27
0,8	0,21
0,9	0,13
1,0	0,09

The recall levels are fixed and for each recall level the corresponding precision is recorded.

MEAN AVERAGE PRECISION

A SINGLE FIGURE

We have a set of queries $Q = \{q_1, \dots, q_n\}$

For each q_j we know the set of documents $\{d_1, \dots, d_{m_j}\}$ that are relevant

Let R_{jk} the set of ranked documents retrieved for the j^{th} query that we get to obtain k relevant documents

Then the mean average precision $\text{MAP}(Q)$ is:

$$\frac{1}{n} \sum_{j=1}^n \left(\frac{1}{m_j} \sum_{k=1}^{m_j} \text{Precision}(R_{jk}) \right)$$

Average precision of the j^{th} query

PRECISION AT K AND R-PRECISION

OTHER SINGLE FIGURES

- Precision at k simply means that we record the precision of the first k retrieved documents. Like “precision at 10”.
- If there are less than k relevant documents then the value cannot be one. Its value is highly dependant on the number of relevant documents that exists.
- A solution to this is the R -precision. If there are R relevant documents for a query, the R -precision is the precision of the top R ranked documents returned by the query.
- R -precision can be averaged across queries.

RELEVANCE FEEDBACK

WHAT IS RELEVANCE FEEDBACK

RECEIVING FEEDBACK FROM THE USER

- The main idea is to involve the user in giving feedback on the initial set of results:
- The user issues a query.
- The system returns an initial set of results.
- The user decides which results are relevant and which are not.
- The system computes a new set of results based on the feedback received by the user.
- If necessary, repeat.

WHAT RELEVANCE FEEDBACK CAN SOLVE AND WHAT IT CANNOT SOLVE

- Relevance feedback can help the user in refining the query without having him/her reformulate it manually.
- It is a *local method*, where the initial query is modified, in contrast to *global methods* that change the wording of the query (like spelling correction).
- Relevance feedback can be ineffective when in the case of
 - Misspelling (but we have seen spelling correction techniques).
 - Searching documents in another language.
 - Vocabulary mismatch between the user and the collection.

THE ROCCHIO ALGORITHM

FEEDBACK FOR THE VECTOR SPACE MODEL

- It is possible to introduce relevance feedback in the vector space model
- We will see the Rocchio Algorithm (1971)
- It was introduced in the SMART (*System for the Mechanical Analysis and Retrieval of Text*) information retrieval system...
- ...which is also where the vector space model was firstly developed

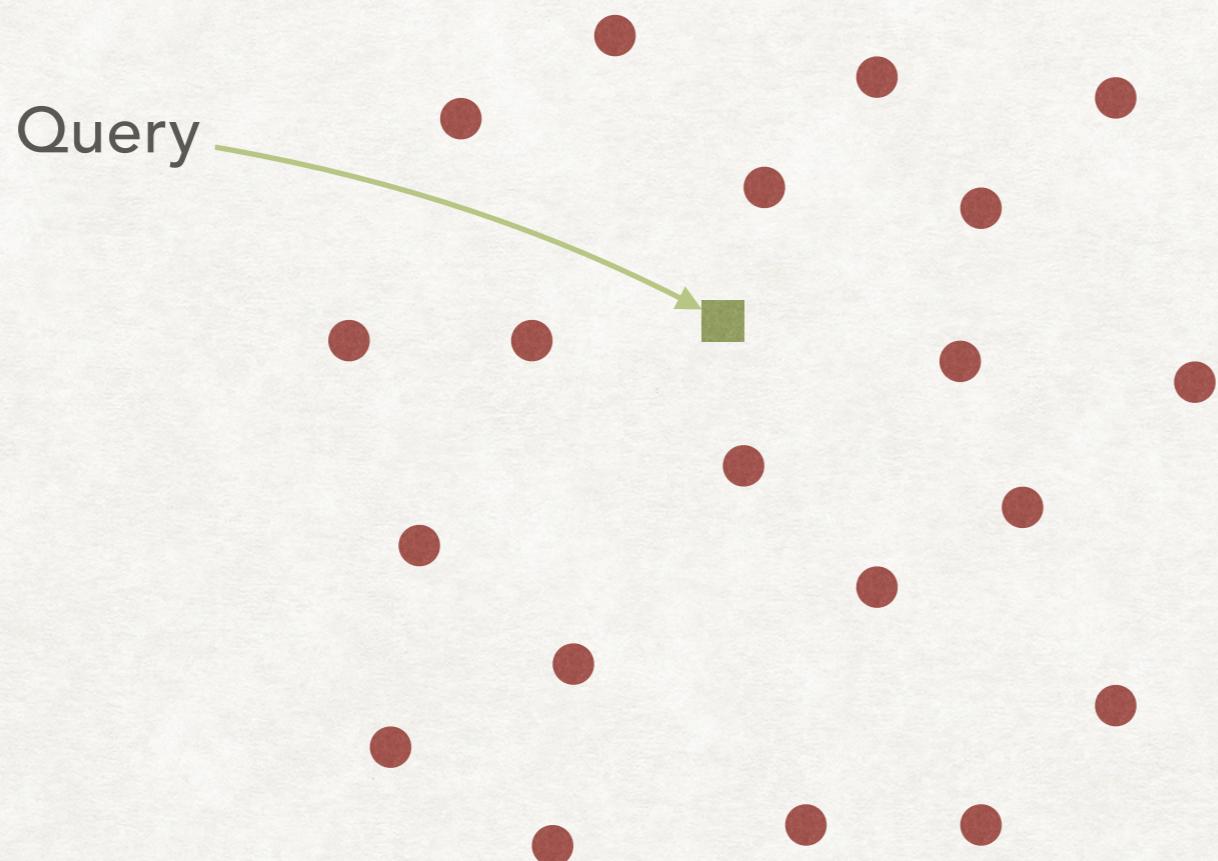
ROCCHIO ALGORITHM: MAIN IDEA

MOVING THE QUERY VECTOR



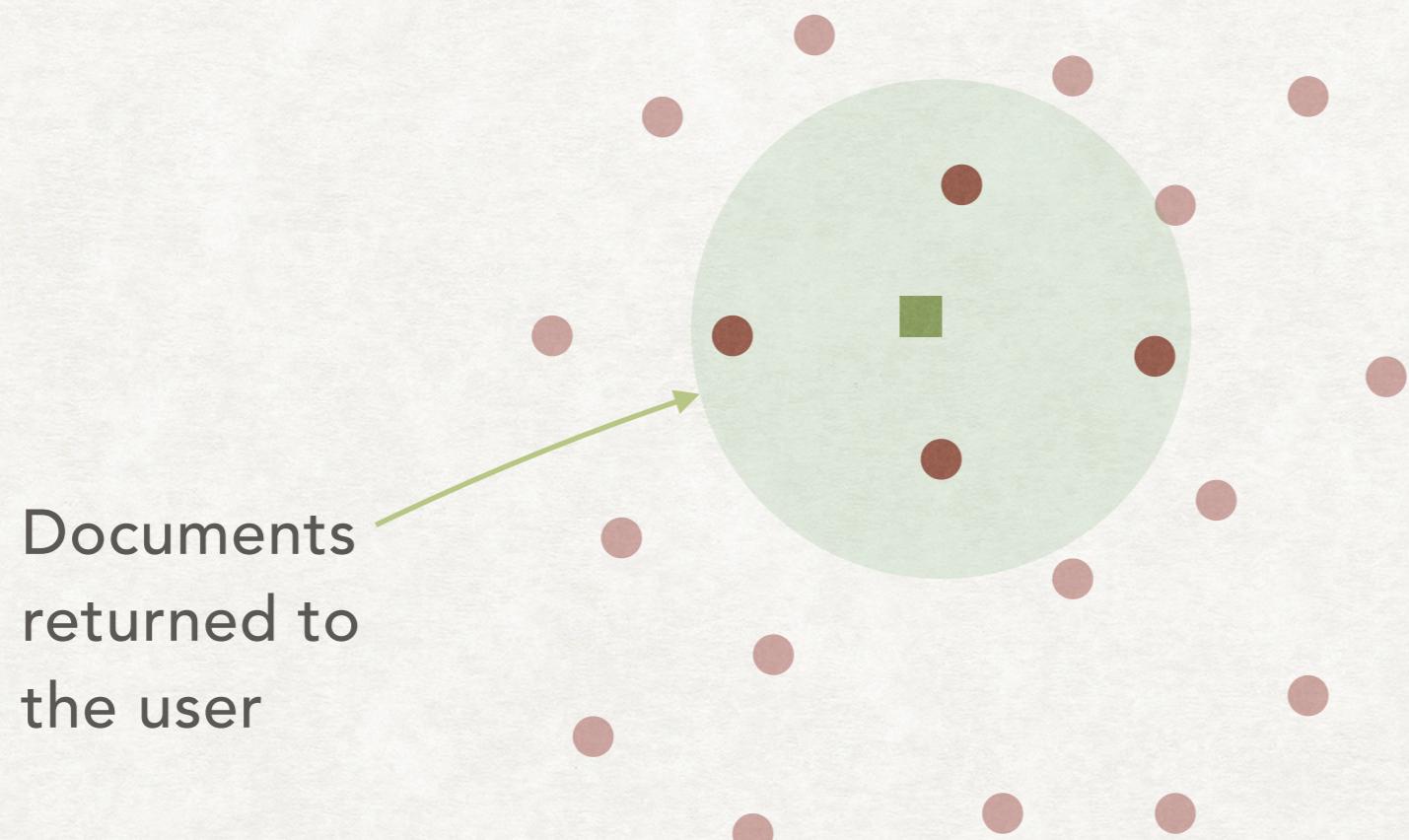
ROCCHIO ALGORITHM: MAIN IDEA

MOVING THE QUERY VECTOR



ROCCHIO ALGORITHM: MAIN IDEA

MOVING THE QUERY VECTOR



ROCCHIO ALGORITHM: MAIN IDEA

MOVING THE QUERY VECTOR



ROCCHIO ALGORITHM: MAIN IDEA

MOVING THE QUERY VECTOR



ROCCHIO ALGORITHM: THEORY

- The user gives us two sets of documents:
 - The relevant documents C_r
 - The non-relevant documents C_{nr}
- We want to maximise the similarity of the query with the set of relevant documents...
- ...while minimising it with respect to the set of non-relevant documents.

ROCCHIO ALGORITHM: THEORY

This can be formalised as defining the *optimal* query \vec{q}_{opt} as:

$$\vec{q}_{opt} = \arg \max_{\vec{q}} [\text{sim}(\vec{q}, C_r) - \text{sim}(\vec{q}, C_{nr})]$$

If we use cosine similarity, we can reformulate the definition as:

$$\vec{q}_{opt} = \frac{1}{|C_r|} \sum_{\vec{d} \in C_r} \vec{d} - \frac{1}{|C_{nr}|} \sum_{\vec{d} \in C_{nr}} \vec{d}$$

Centroid of relevant documents Centroid of non-relevant documents

ROCCHIO ALGORITHM

However, we usually do not have knowledge of the relevance of *all* documents in the system. Instead we have:

- a set D_r of *known relevant* documents
- a set D_{nr} of *known non-relevant* documents

We also have the original query \vec{q}_0 performed by the user.

We can perform a linear combination of:

- The centroid of D_r
- The centroid of D_{nr}
- The original query \vec{q}_0

ROCCHIO ALGORITHM

In the Rocchio algorithm the query is updated as follows:

$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d} \in C_r} \vec{d} - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d} \in C_{nr}} \vec{d}$$

Original query

Centroid of the known relevant documents

Centroid of the known non-relevant documents

The diagram illustrates the Rocchio formula with three main components highlighted by red circles and arrows: 1) The original query vector \vec{q}_0 is scaled by α and added to the sum. 2) The centroid of known relevant documents C_r is scaled by β and added to the sum. 3) The centroid of known non-relevant documents C_{nr} is scaled by γ and subtracted from the sum.

If one of the components of \vec{q}_m is less than 0, we set it to 0
(all documents have non-negative coordinates)

ROCCHIO ALGORITHM

SELECTING THE WEIGHTS

- We need to select reasonable weights α , β , and γ :
- Positive feedback is more valuable than negative feedback, so usually $\gamma < \beta$.
- Reasonable values might be $\alpha = 1$, $\beta = 0.75$, and $\gamma = 0.15$.
- It is also possible to also have only positive feedback with $\gamma = 0$.

PSEUDO-RELEVANCE FEEDBACK

NOW WITHOUT THE USER

- It is possible to perform a relevance feedback without the user...
- ...even before he/she receives the results of the first query.
- Perform the query \vec{q} as usual.
- Consider the first k retrieved documents in the ranking as relevant.
- Perform relevance feedback using this assumption.
- Might provide better results, but the retrieved documents might drift the query in an unwanted direction.