

# Evaluation Metrics

Jaime Arguello

INLS 509: Information Retrieval

[jarguell@email.unc.edu](mailto:jarguell@email.unc.edu)

# Batch Evaluation

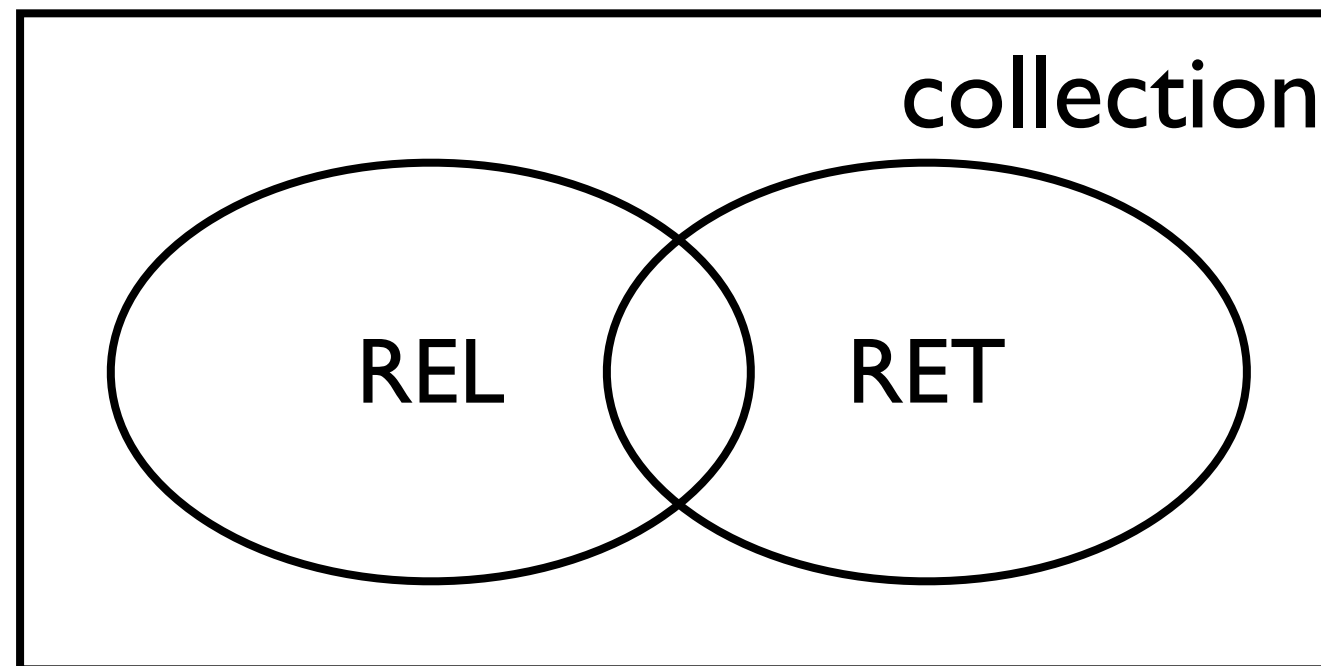
## evaluation metrics

- At this point, we have a set of queries, with identified relevant and non-relevant documents
- The goal of an **evaluation metric** is to measure the quality of a particular ranking of known relevant/non-relevant documents

# Set Retrieval

precision and recall

- **Precision (P)**: the proportion of retrieved documents that are relevant

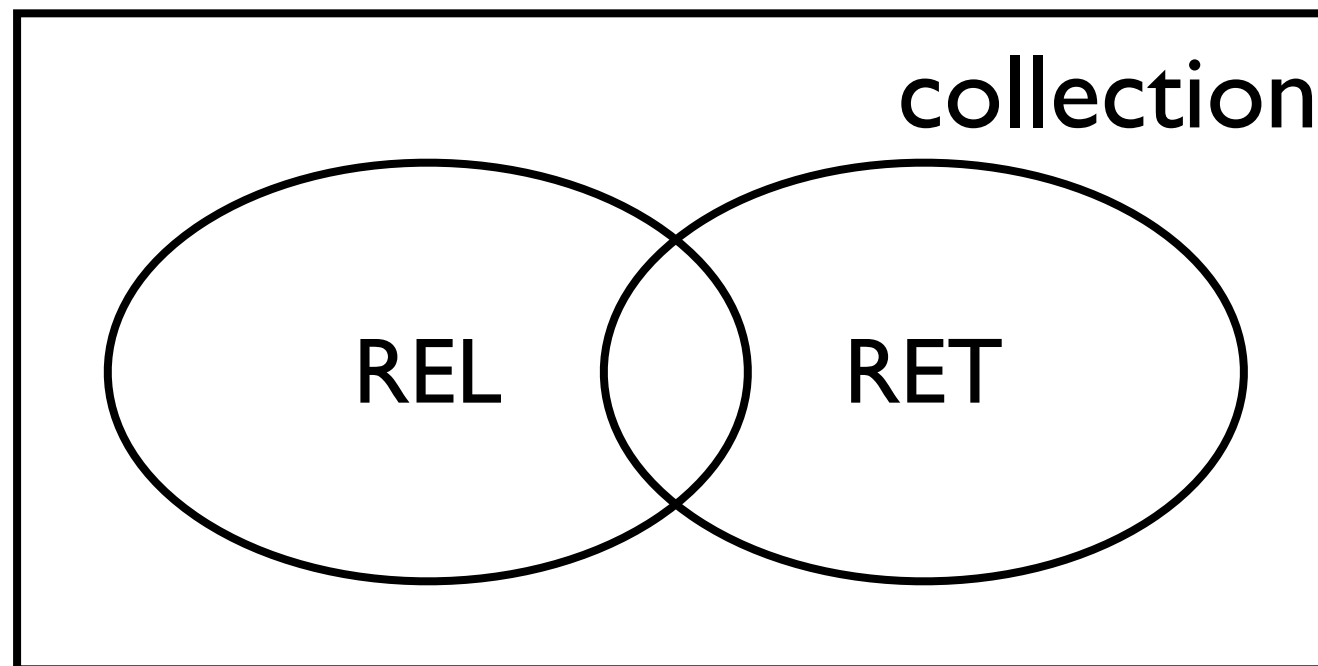


$$\mathcal{P} = \frac{|RET \cap REL|}{|RET|}$$

# Set Retrieval

## precision and recall

- **Recall (R)**: the proportion of relevant documents that are retrieved



$$\mathcal{R} = \frac{|RET \cap REL|}{|REL|}$$

# Set Retrieval

## precision and recall

- A system can make two types of errors:
  - a false positive error: the system retrieves a document that is not relevant (should not have been retrieved)
  - a false negative error: the system fails to retrieve a document that is relevant (should have been retrieved)
- How do these types of errors affect precision and recall?
- Precision is affected by the number of false positive errors
- Recall is affected by the number of false negative errors

# Set Retrieval

## F-measure (also known as F1)

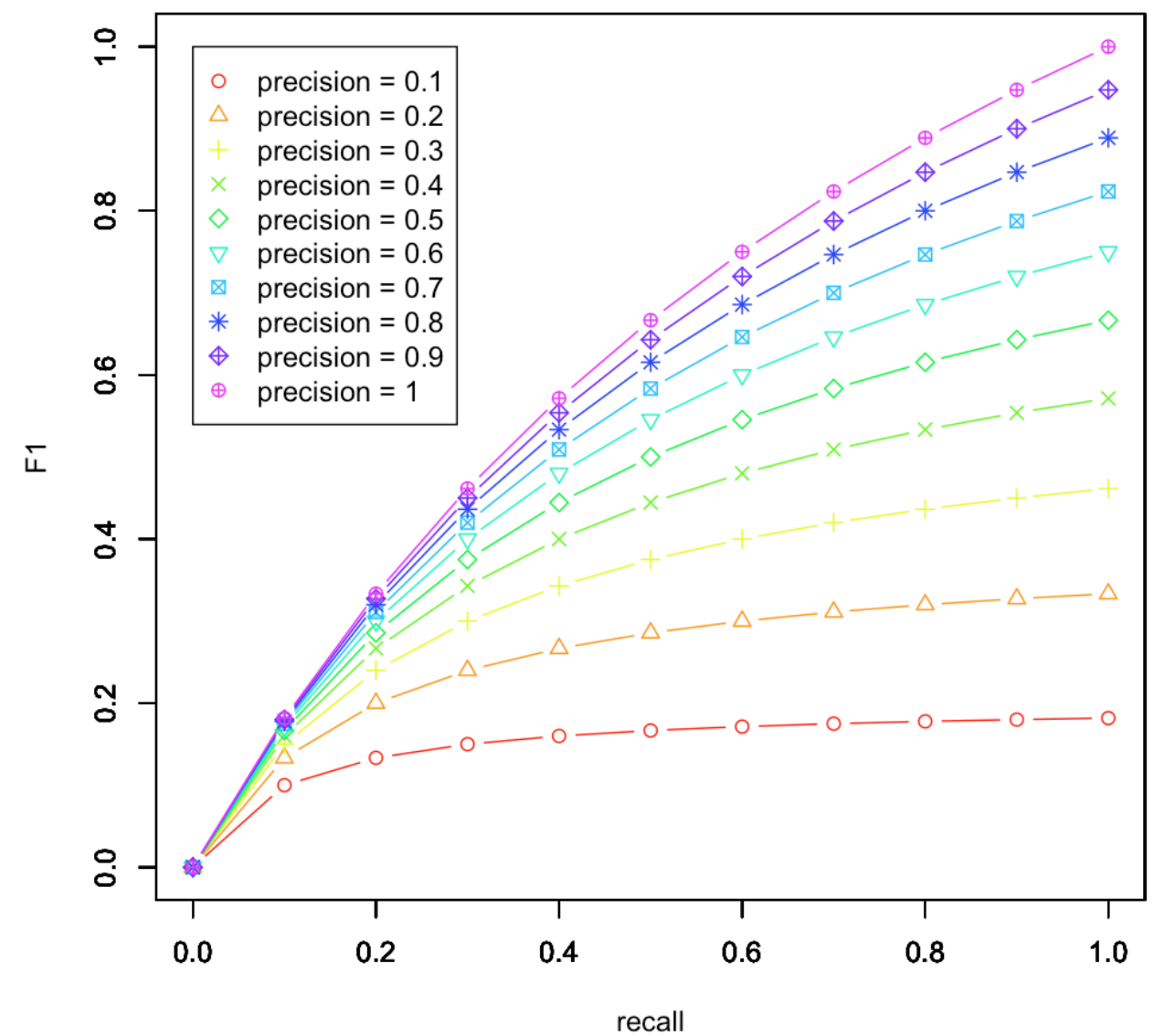
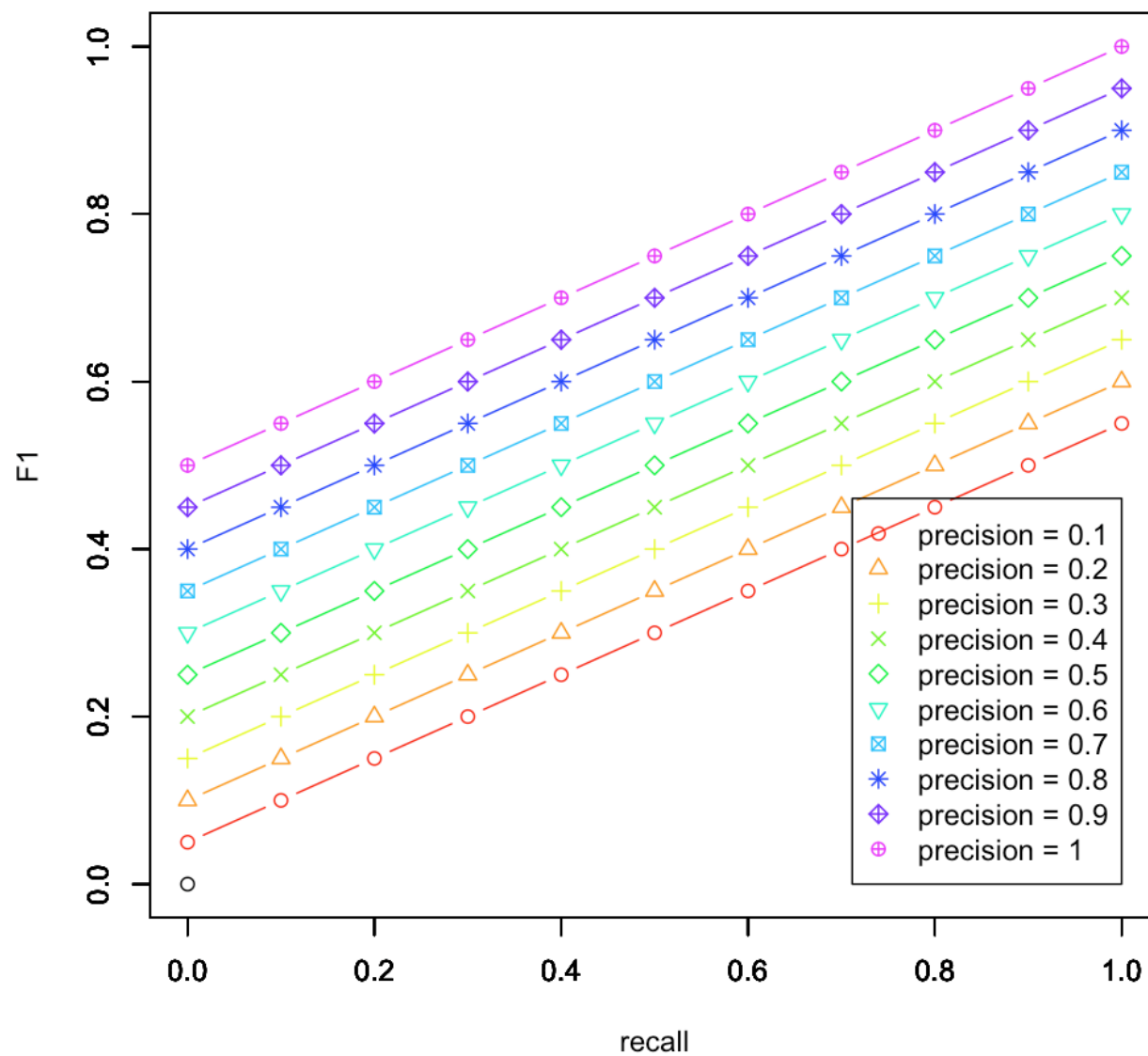
- A system that retrieves a single relevant document would get **1.0** precision and near **0.0** recall
- A system that retrieves the entire collection would get **1.0** recall and near **0.0** precision
- **Solution:** use the harmonic mean rather than the arithmetic mean
- **F-measure:**

$$\mathcal{F} = \frac{1}{\frac{1}{2} \left( \frac{1}{\mathcal{P}} + \frac{1}{\mathcal{R}} \right)} = \frac{2 \times \mathcal{P} \times \mathcal{R}}{\mathcal{P} + \mathcal{R}}$$

# Set Retrieval

F-measure (also known as F1)

- The harmonic mean punishes small values



(slide courtesy of Ben Carterette)

# Ranked Retrieval

## precision and recall

- In most situations, the system outputs a ranked list of documents rather than an unordered set
- User-behavior assumption:
  - ▶ The user examines the output ranking from top-to-bottom until he/she is satisfied or gives up
- Precision/Recall @ rank  $K$



# Ranked Retrieval

precision and recall: exercise

- Assume 20 **relevant** documents

K	P@K	R@K
1	$(1/1) = 1.0$	$(1/20) = 0.05$
2	$(1/2) = 0.5$	$(1/20) = 0.05$
3	$(2/3) = 0.67$	$(2/20) = 0.10$
4	$(3/4) = 0.75$	$(3/20) = 0.15$
5	$(4/5) = 0.80$	$(4/20) = 0.20$
6	$(5/6) = 0.83$	$(5/20) = 0.25$
7	$(6/7) = 0.86$	$(6/20) = 0.30$
8	$(6/8) = 0.75$	$(6/20) = 0.30$
9	$(7/9) = 0.78$	$(7/20) = 0.35$
10	$(7/10) = 0.70$	$(7/20) = 0.35$

K = 10



# Ranked Retrieval

what do these statements mean?

- As with most metrics, experimenters report average values (averaged across evaluation queries)
- System **A** obtains an average **P@10** of 0.50
- System **A** obtains an average **P@10** of 0.10
- System **A** obtains an average **P@1** of 0.50
- System **A** obtains an average **P@20** of 0.20

# Ranked Retrieval

## comparing systems

- **Good practice:** always ask yourself “Are users likely to notice?”
- System **A** obtains an average **P@I** of 0.10
- System **B** obtains an average **P@I** of 0.20
- This is a 100% improvement.
- Are user’s likely to notice?

# Ranked Retrieval

## comparing systems

- **Good practice:** always ask yourself “Are users likely to notice?”
- System **A** obtains an average **P@I** of 0.05
- System **B** obtains an average **P@I** of 0.10
- This is a 100% improvement.
- Are user’s likely to notice?

# Ranked Retrieval

## P/R@K

- Advantages:
  - ▶ easy to compute
  - ▶ easy to interpret
- Disadvantages:
  - ▶ the value of **K** has a huge impact on the metric
  - ▶ how do we pick **K**?

# Ranked Retrieval

motivation: average precision

- Ideally, we want the system to achieve high precision for varying values of **K**
- The metric **average precision** accounts for precision and recall without having to set **K**

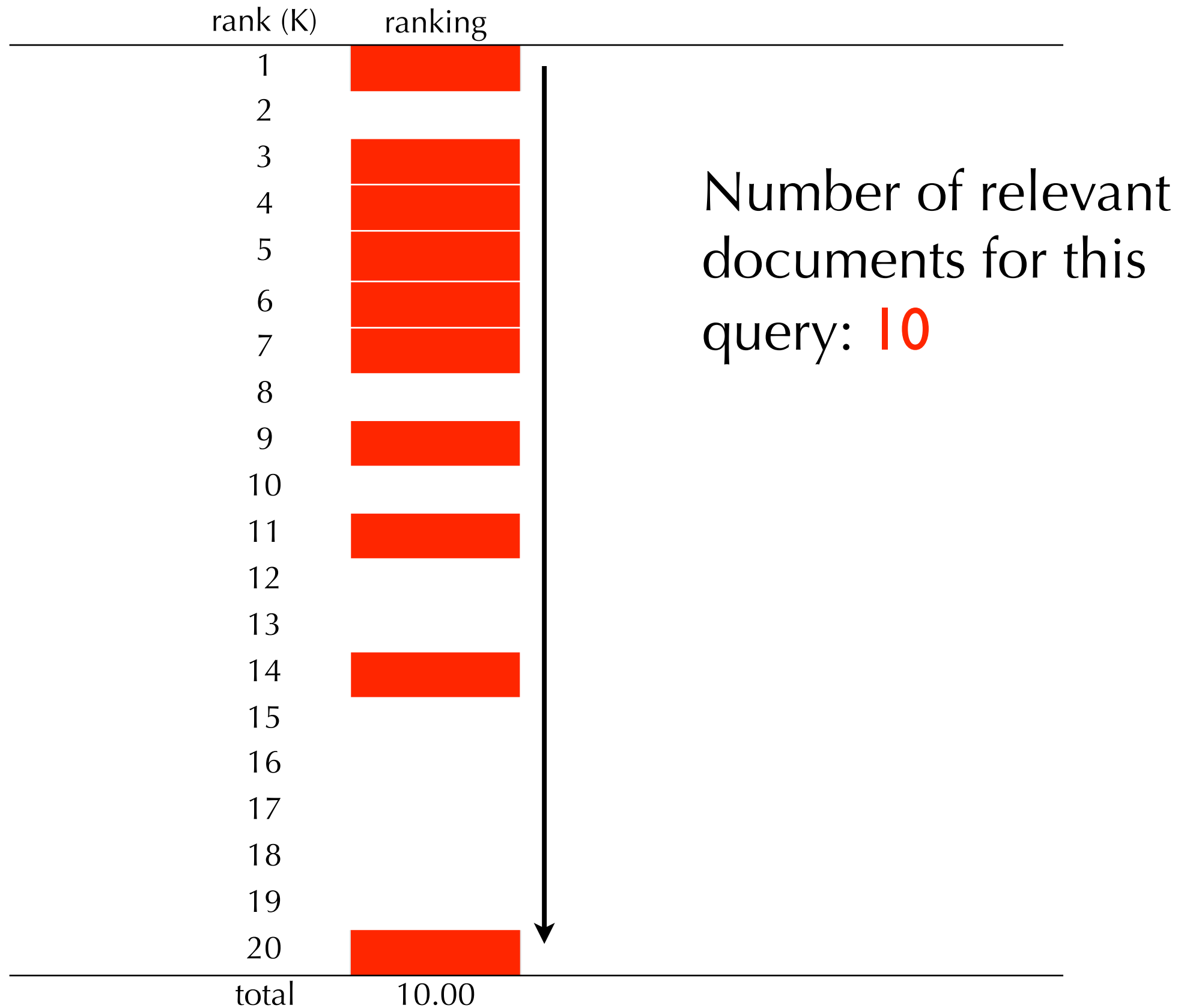
# Ranked Retrieval

## average precision

1. Go down the ranking one-rank-at-a-time
2. If the document at rank  $K$  is relevant, measure  $P@K$ 
  - ▶ proportion of top- $K$  documents that are relevant
3. Finally, take the average of  $P@K$  values
  - ▶ the number of  $P@K$  values will equal the number of relevant documents

# Ranked Retrieval

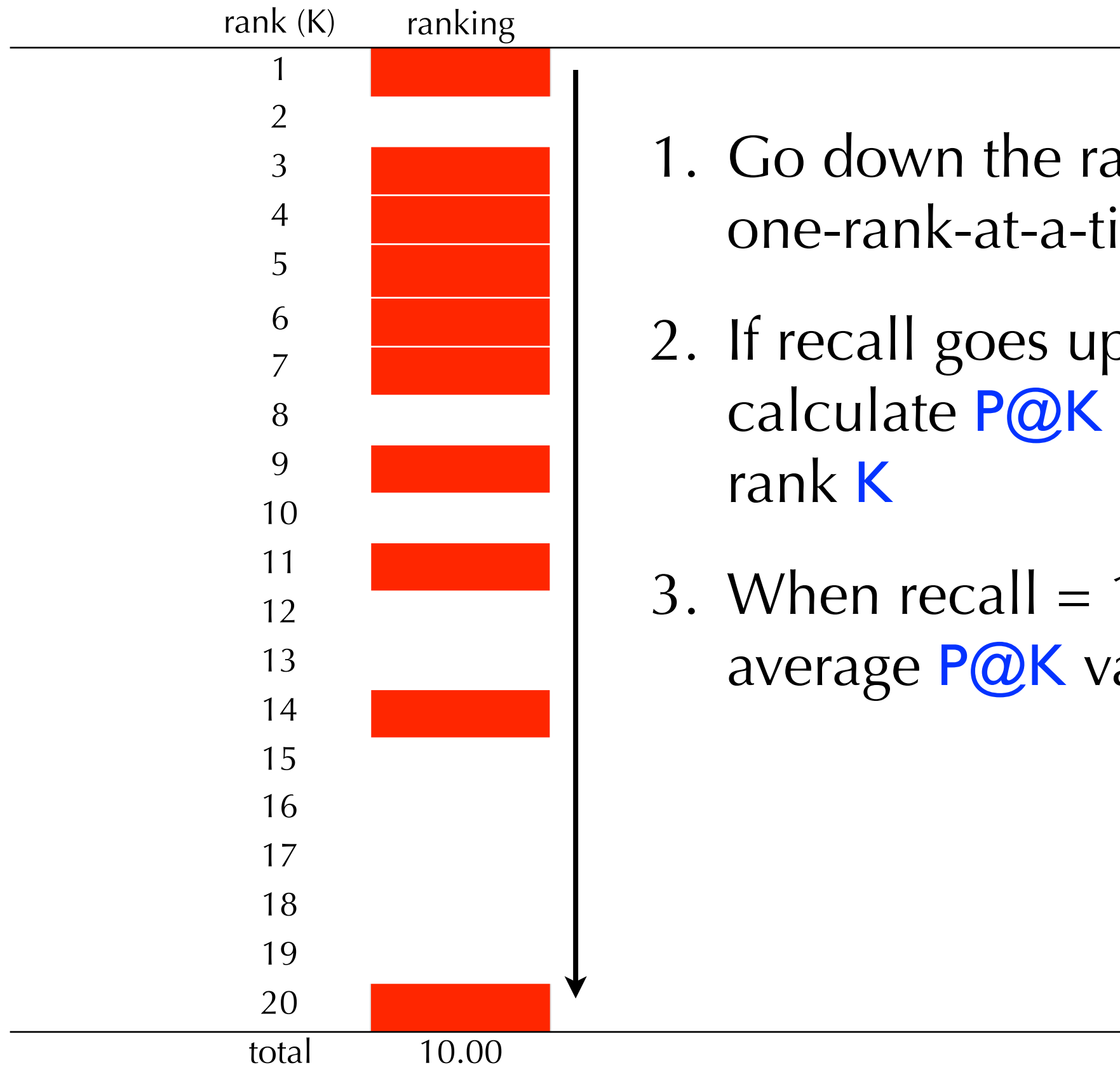
## average-precision





# Ranked Retrieval

## average-precision



1. Go down the ranking one-rank-at-a-time
2. If recall goes up, calculate  $P@K$  at that rank  $K$
3. When recall = 1.0, average  $P@K$  values

# Ranked Retrieval

## average-precision

rank (K)	ranking	R@K	P@K
1	█	0.10	1.00
2		0.10	0.50
3	█	0.20	0.67
4	█	0.30	0.75
5	█	0.40	0.80
6	█	0.50	0.83
7	█	0.60	0.86
8		0.60	0.75
9	█	0.70	0.78
10		0.70	0.70
11	█	0.80	0.73
12		0.80	0.67
13		0.80	0.62
14	█	0.90	0.64
15		0.90	0.60
16		0.90	0.56
17		0.90	0.53
18		0.90	0.50
19		0.90	0.47
20	█	1.00	0.50
total	10.00	average-precision	0.76

# Ranked Retrieval

## average-precision

rank (K)	ranking	R@K	P@K
1		0.10	1.00
2		0.20	1.00
3		0.30	1.00
4		0.40	1.00
5		0.50	1.00
6		0.60	1.00
7		0.70	1.00
8		0.80	1.00
9		0.90	1.00
10		1.00	1.00
11		1.00	0.91
12		1.00	0.83
13		1.00	0.77
14		1.00	0.71
15		1.00	0.67
16		1.00	0.63
17		1.00	0.59
18		1.00	0.56
19		1.00	0.53
20		1.00	0.50
total	10.00	average-precision	1.00

# Ranked Retrieval

## average-precision

rank (K)	ranking	R@K	P@K
1		0.00	0.00
2		0.00	0.00
3		0.00	0.00
4		0.00	0.00
5		0.00	0.00
6		0.00	0.00
7		0.00	0.00
8		0.00	0.00
9		0.00	0.00
10		0.00	0.00
11		0.10	0.09
12		0.20	0.17
13		0.30	0.23
14		0.40	0.29
15		0.50	0.33
16		0.60	0.38
17		0.70	0.41
18		0.80	0.44
19		0.90	0.47
20		1.00	0.50
total	10.00	average-precision	0.33

# Ranked Retrieval

## average-precision

rank (K)	ranking	R@K	P@K
1	█	0.10	1.00
2		0.10	0.50
3	█	0.20	0.67
4	█	0.30	0.75
5	█	0.40	0.80
6	█	0.50	0.83
7	█	0.60	0.86
8		0.60	0.75
9	█	0.70	0.78
10		0.70	0.70
11	█	0.80	0.73
12		0.80	0.67
13		0.80	0.62
14	█	0.90	0.64
15		0.90	0.60
16		0.90	0.56
17		0.90	0.53
18		0.90	0.50
19		0.90	0.47
20	█	1.00	0.50
total	10.00	average-precision	0.76

# Ranked Retrieval

## average-precision

swapped  
ranks 2 and 3

rank (K)	ranking	R@K	P@K
1		0.10	1.00
2		0.20	1.00
3		0.20	0.67
4		0.30	0.75
5		0.40	0.80
6		0.50	0.83
7		0.60	0.86
8		0.60	0.75
9		0.70	0.78
10		0.70	0.70
11		0.80	0.73
12		0.80	0.67
13		0.80	0.62
14		0.90	0.64
15		0.90	0.60
16		0.90	0.56
17		0.90	0.53
18		0.90	0.50
19		0.90	0.47
20		1.00	0.50
total	10.00	average-precision	0.79

# Ranked Retrieval

## average-precision

rank (K)	ranking	R@K	P@K
1	█	0.10	1.00
2		0.10	0.50
3	█	0.20	0.67
4	█	0.30	0.75
5	█	0.40	0.80
6	█	0.50	0.83
7	█	0.60	0.86
8		0.60	0.75
9	█	0.70	0.78
10		0.70	0.70
11	█	0.80	0.73
12		0.80	0.67
13		0.80	0.62
14	█	0.90	0.64
15		0.90	0.60
16		0.90	0.56
17		0.90	0.53
18		0.90	0.50
19		0.90	0.47
20	█	1.00	0.50
total	10.00	average-precision	0.76

# Ranked Retrieval

## average-precision

rank (K)	ranking	R@K	P@K
1	■	0.10	1.00
2		0.10	0.50
3	■	0.20	0.67
4	■	0.30	0.75
5	■	0.40	0.80
6	■	0.50	0.83
7	■	0.60	0.86
8	■	0.70	0.88
9		0.70	0.78
10		0.70	0.70
11	■	0.80	0.73
12		0.80	0.67
13		0.80	0.62
14	■	0.90	0.64
15		0.90	0.60
16		0.90	0.56
17		0.90	0.53
18		0.90	0.50
19		0.90	0.47
20	■	1.00	0.50
total	10.00	average-precision	0.77

swapped ranks  
8 and 9



# Ranked Retrieval

## average precision

- Advantages:
  - ▶ no need to choose  $K$
  - ▶ accounts for both precision and recall
  - ▶ mistakes at the top are more influential
  - ▶ mistakes at the bottom are still accounted for
- Disadvantages
  - ▶ not quite as easy to interpret as  $P/R@K$

# Ranked Retrieval

MAP: mean average precision

- So far, we've talked about average precision for a single query
- **Mean Average Precision (MAP):** average precision averaged across a set of queries
  - ▶ yes, confusing. but, better than calling it "average average precision"!
  - ▶ one of the most common metrics in IR evaluation

# Ranked Retrieval

## precision-recall curves

- In some situations, we want to understand the trade-off between precision and recall
- A precision-recall (PR) curve expresses precision as a function of recall

# Ranked Retrieval

precision-recall curves: general idea

- Different tasks require different levels of recall
- Sometimes, the user wants a few relevant documents
- Other times, the user wants most of them
- Suppose a user wants some level of recall **R**
- The goal for the system is to minimize the number of false negatives the user must look at in order to achieve a level of recall **R**

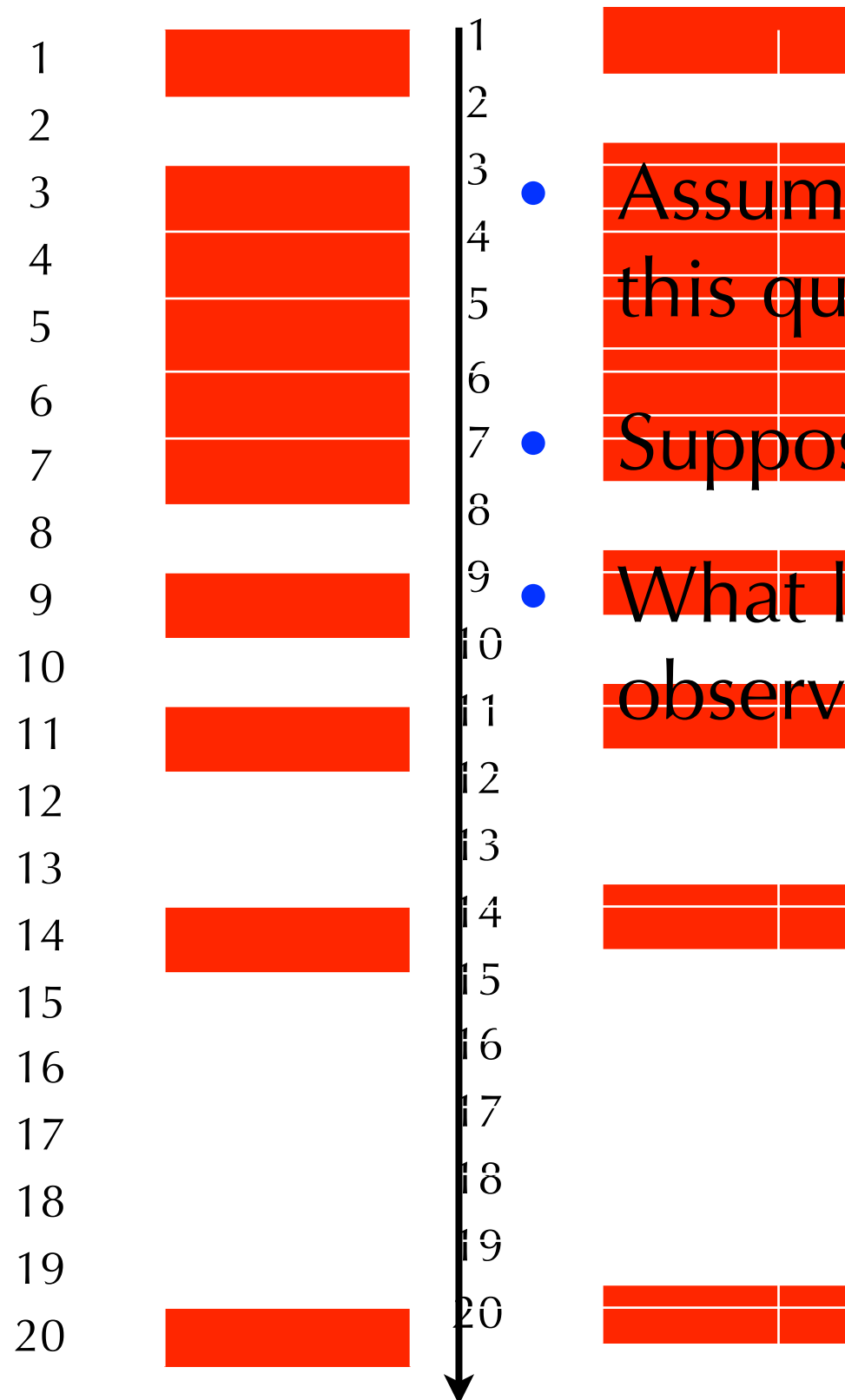
# Ranked Retrieval

precision-recall curves: general idea

- **False negative error:** not retrieving a relevant document
  - ▶ false negative errors affects recall
- **False positive errors:** retrieving a non-relevant document
  - ▶ false positives errors affects precision
- If a user wants to avoid a certain level of false-negatives, what is the level of false-positives he/she must filter through?

# Ranked Retrieval

## precision-recall curves



- Assume 10 relevant documents for this query
- Suppose the user wants  $R = (1/10)$
- What level of precision will the user observe?

# Ranked Retrieval

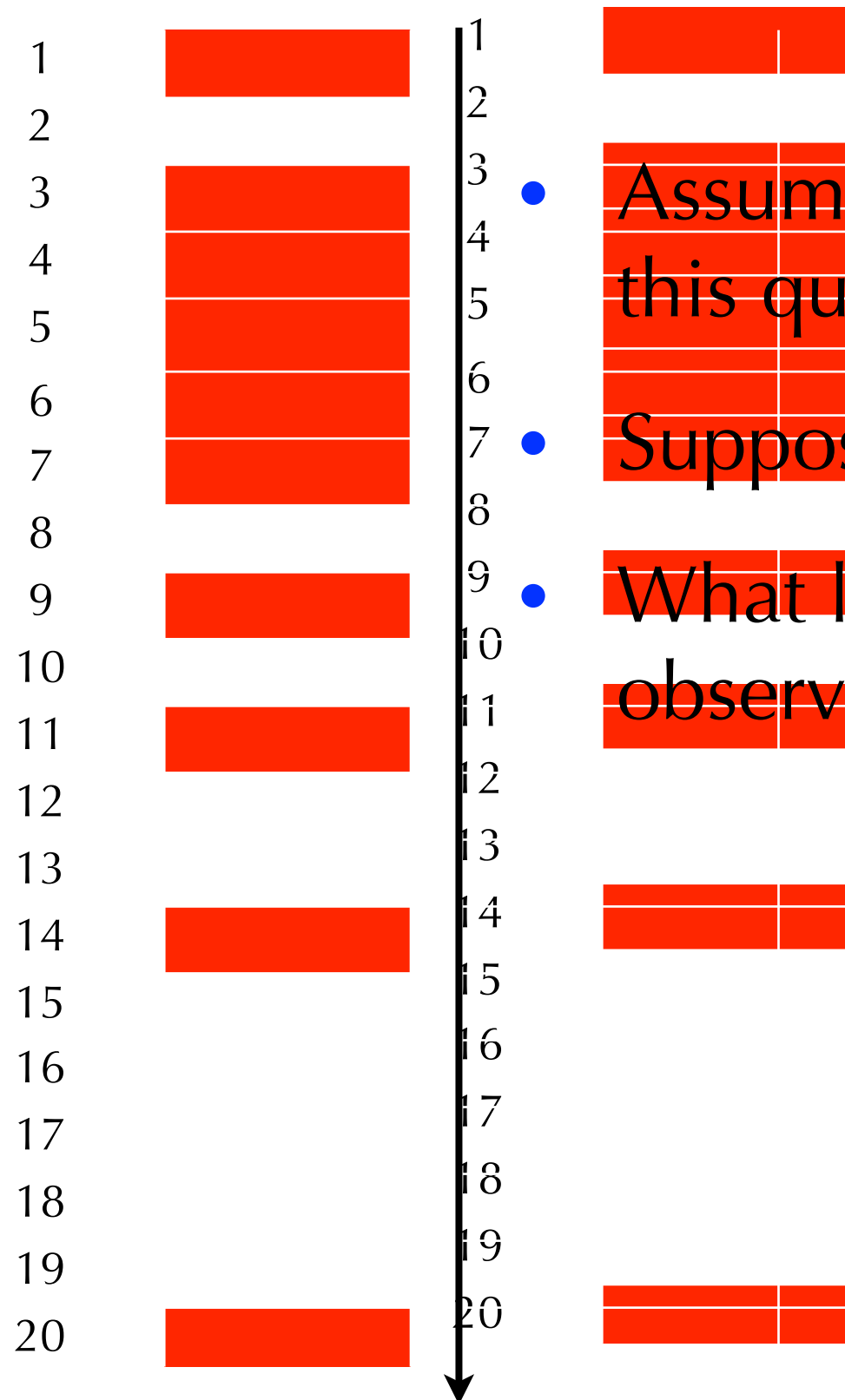
## precision-recall curves



- Assume 10 relevant documents for this query
- Suppose the user wants  $R = (2/10)$
- What level of precision will the user observe?

# Ranked Retrieval

## precision-recall curves













- Assume 10 relevant documents for this query
- Suppose the user wants  $R = (10/10)$
- What level of precision will the user observe?



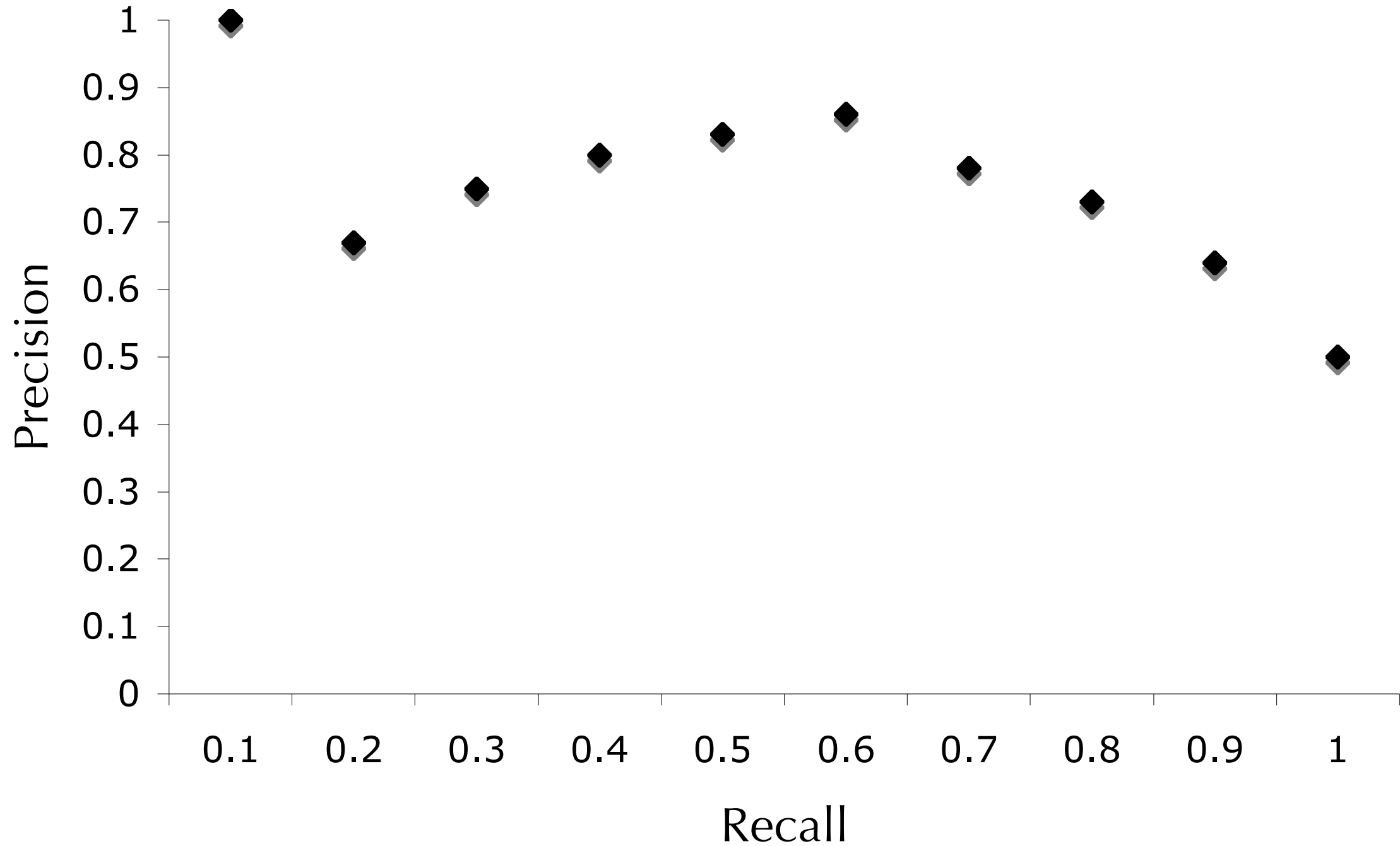
# Ranked Retrieval

## precision-recall curves

rank (K)	ranking	R@K	P@K
1		0.10	1.00
2		0.10	0.50
3		0.20	0.67
4		0.30	0.75
5		0.40	0.80
6		0.50	0.83
7		0.60	0.86
8		0.60	0.75
9		0.70	0.78
10		0.70	0.70
11		0.80	0.73
12		0.80	0.67
13		0.80	0.62
14		0.90	0.64
15		0.90	0.60
16		0.90	0.56
17		0.90	0.53
18		0.90	0.50
19		0.90	0.47
20		1.00	0.50

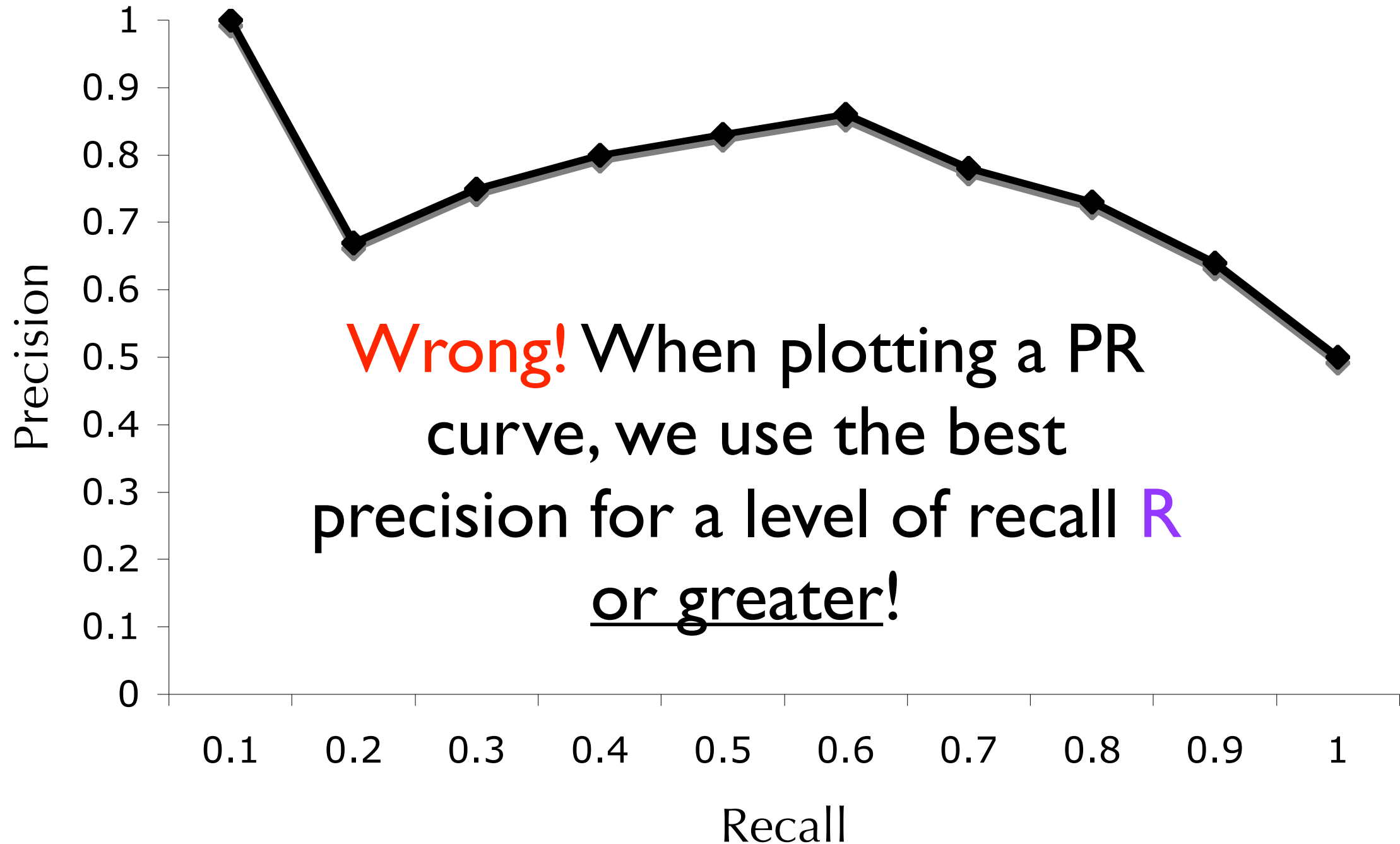
# Ranked Retrieval

## precision-recall curves



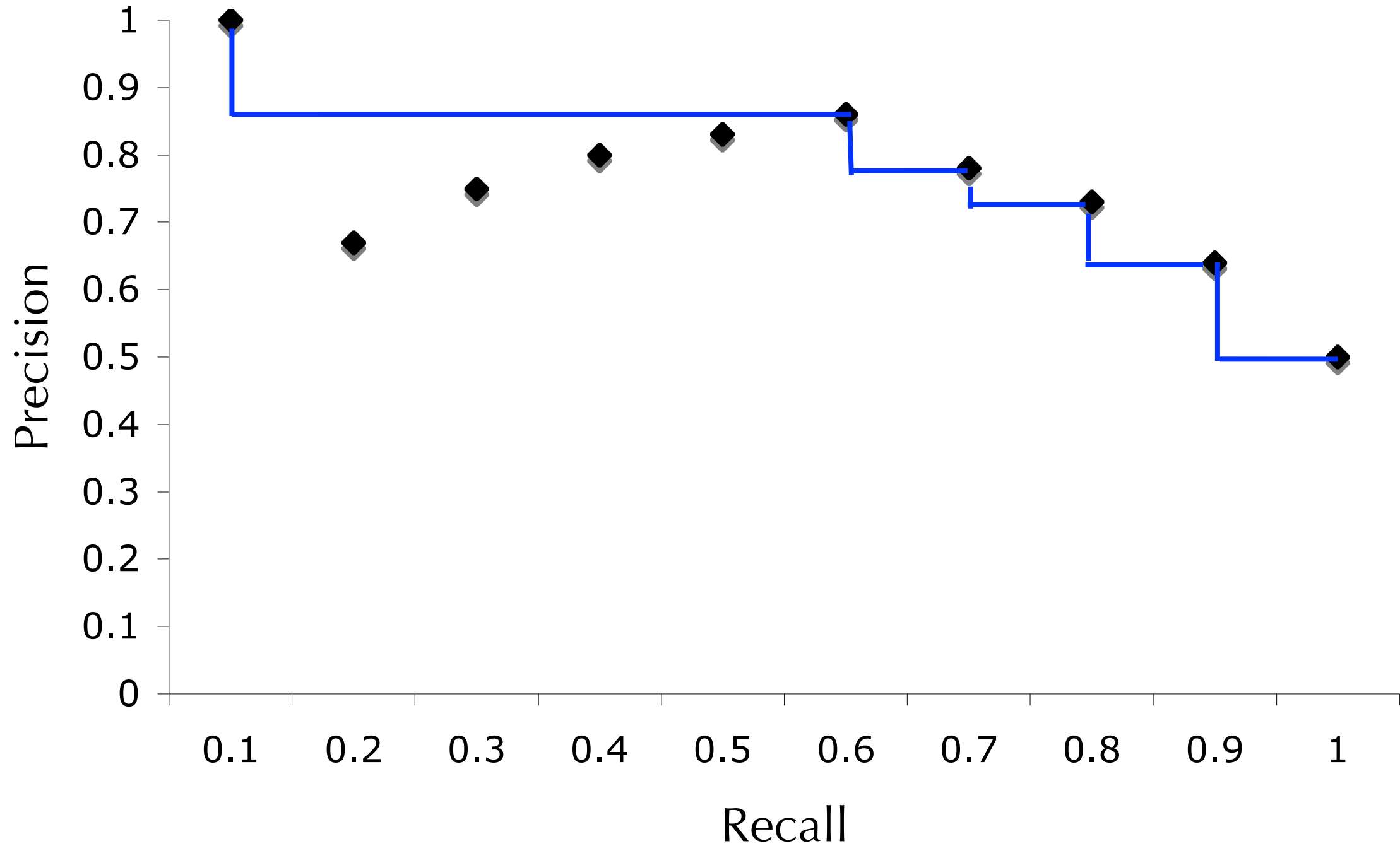
# Ranked Retrieval

## precision-recall curves



# Ranked Retrieval

## precision-recall curves



# Ranked Retrieval

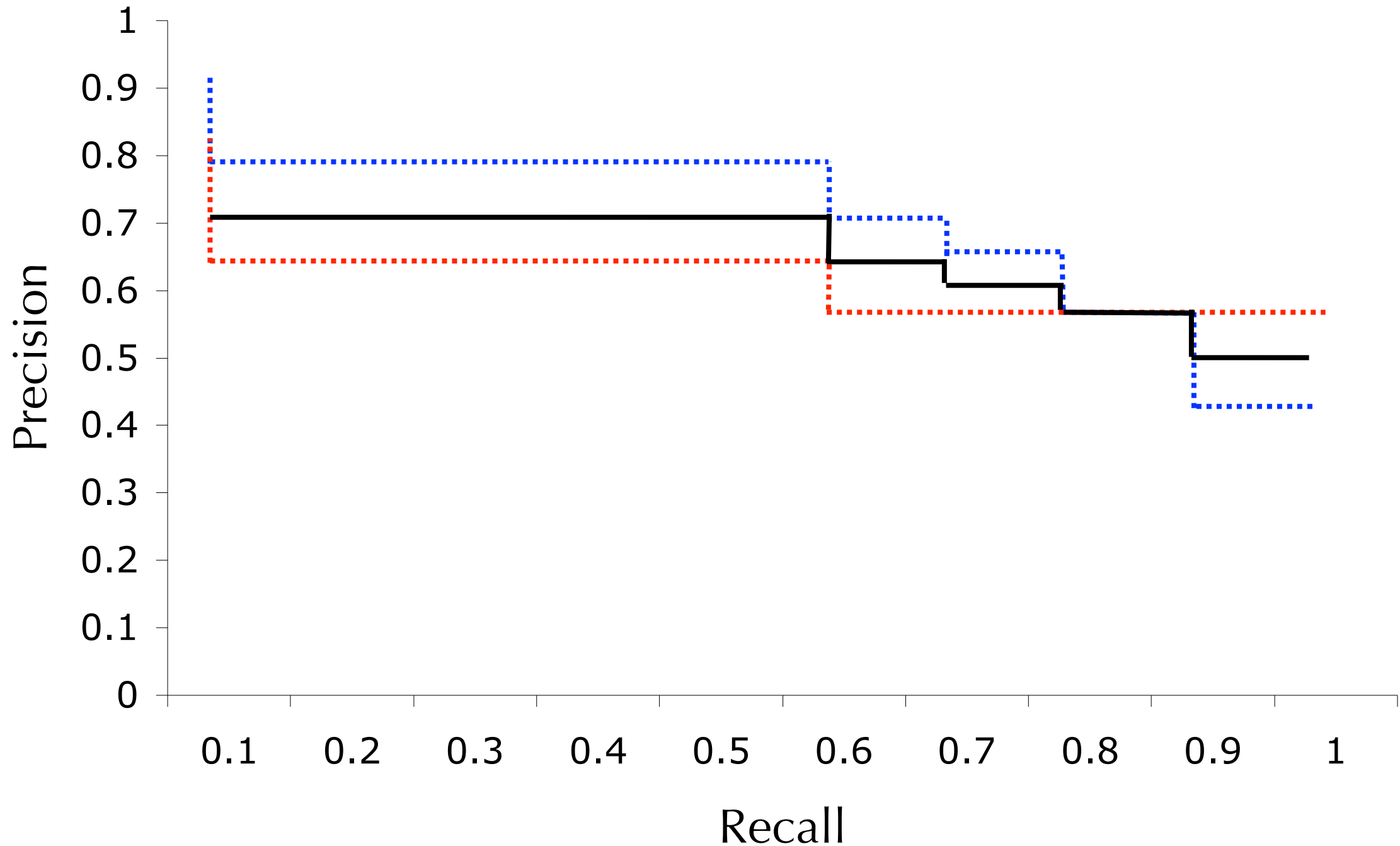
## precision-recall curves

- For a single query, a PR curve looks like a step-function
- For multiple queries, we can average these curves
  - ▶ Average the precision values for different values of recall (e.g., from 0.01 to 1.0 in increments of 0.01)
- This forms a smoother function

# Ranked Retrieval

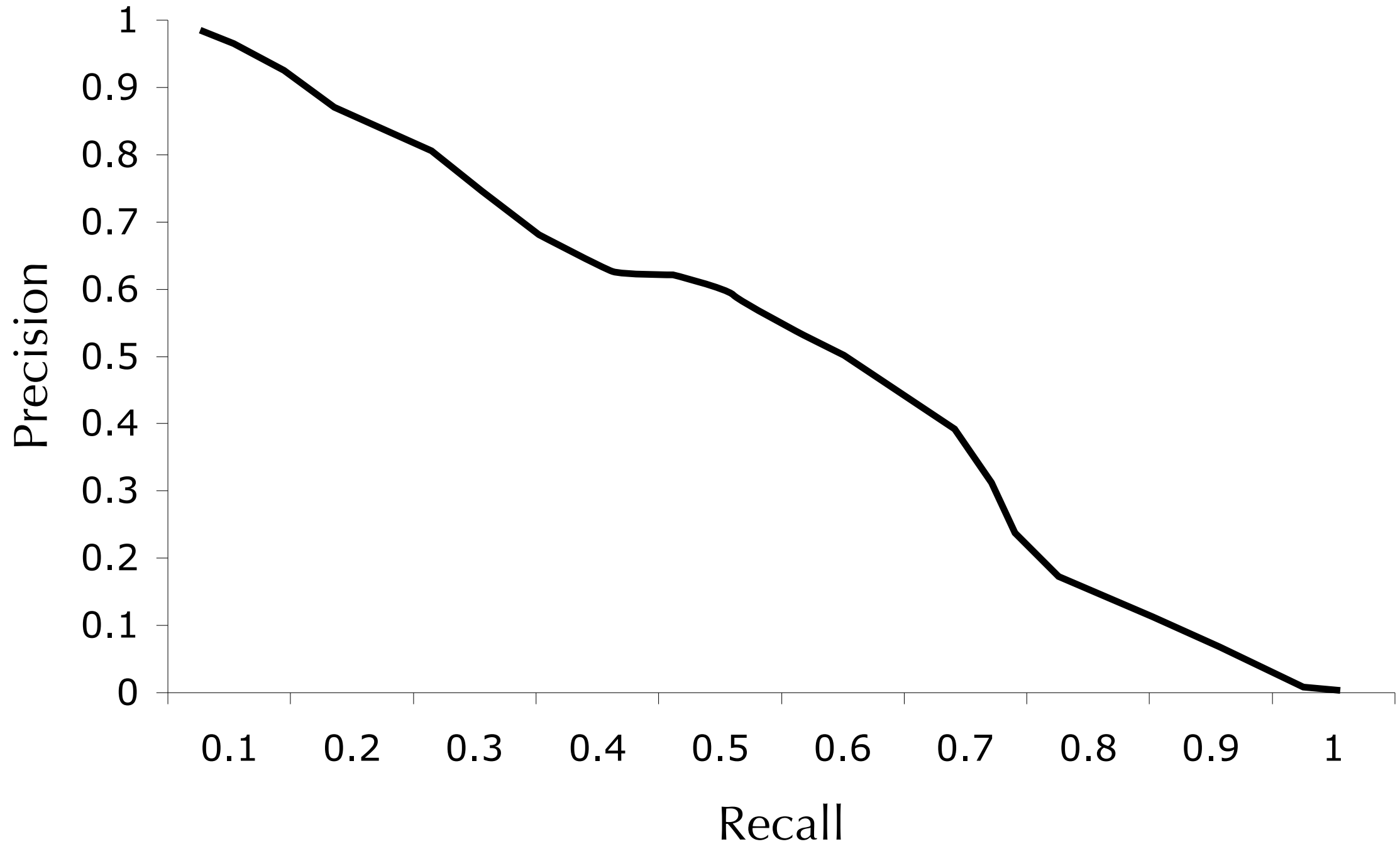
## precision-recall curves

- PR curves can be averaged across multiple queries



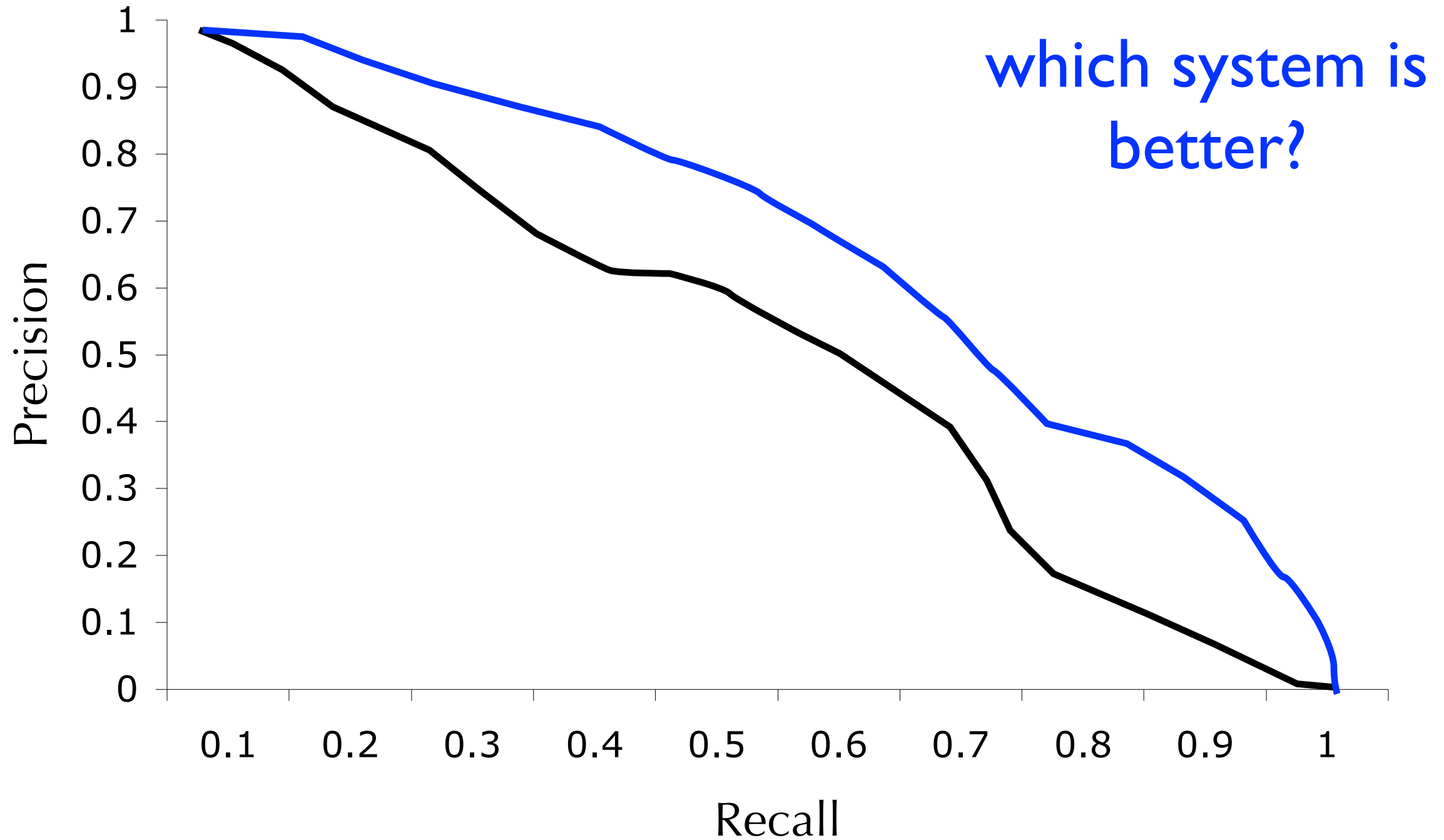
# Ranked Retrieval

precision-recall curves



# Ranked Retrieval

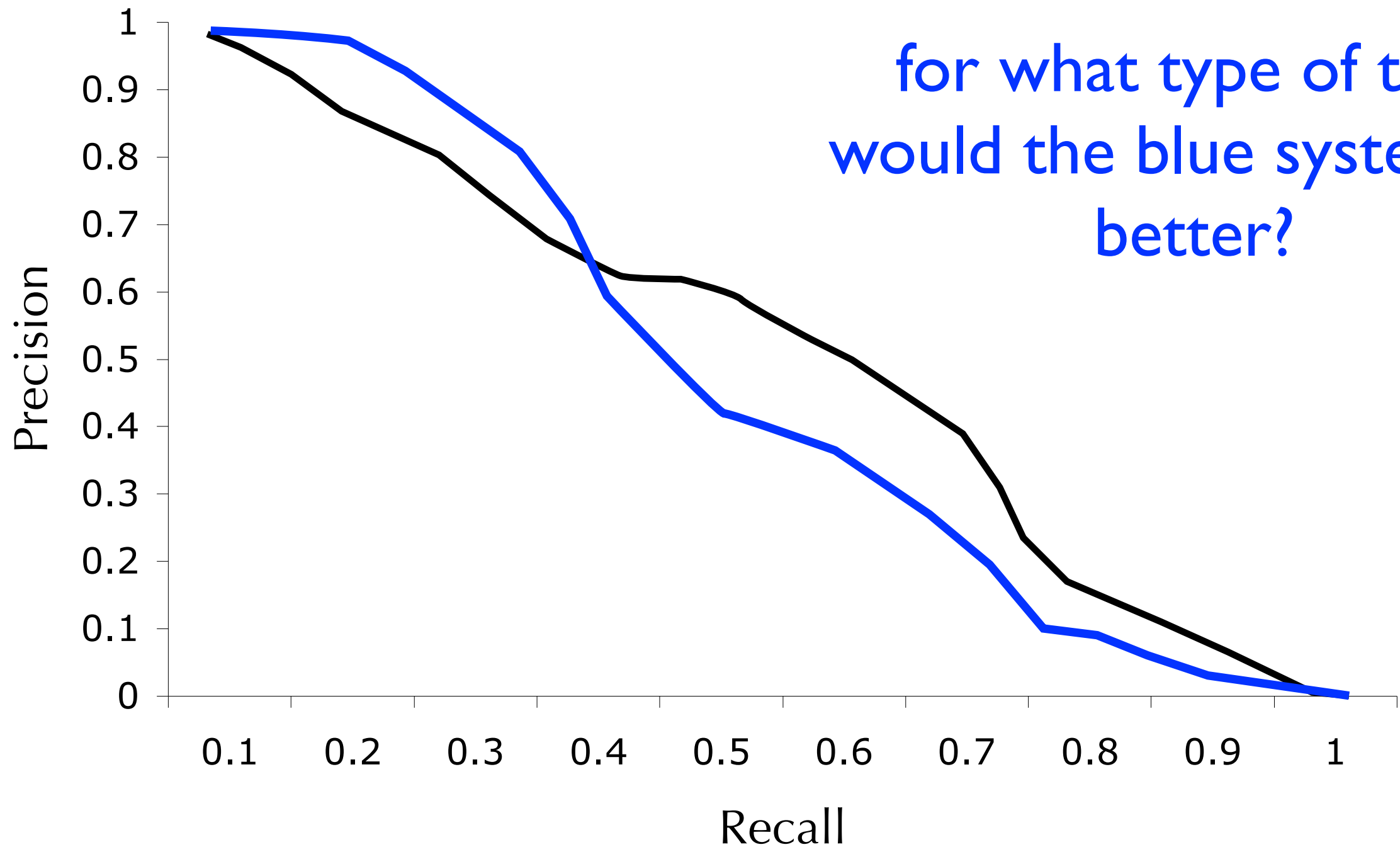
precision-recall curves





# Ranked Retrieval

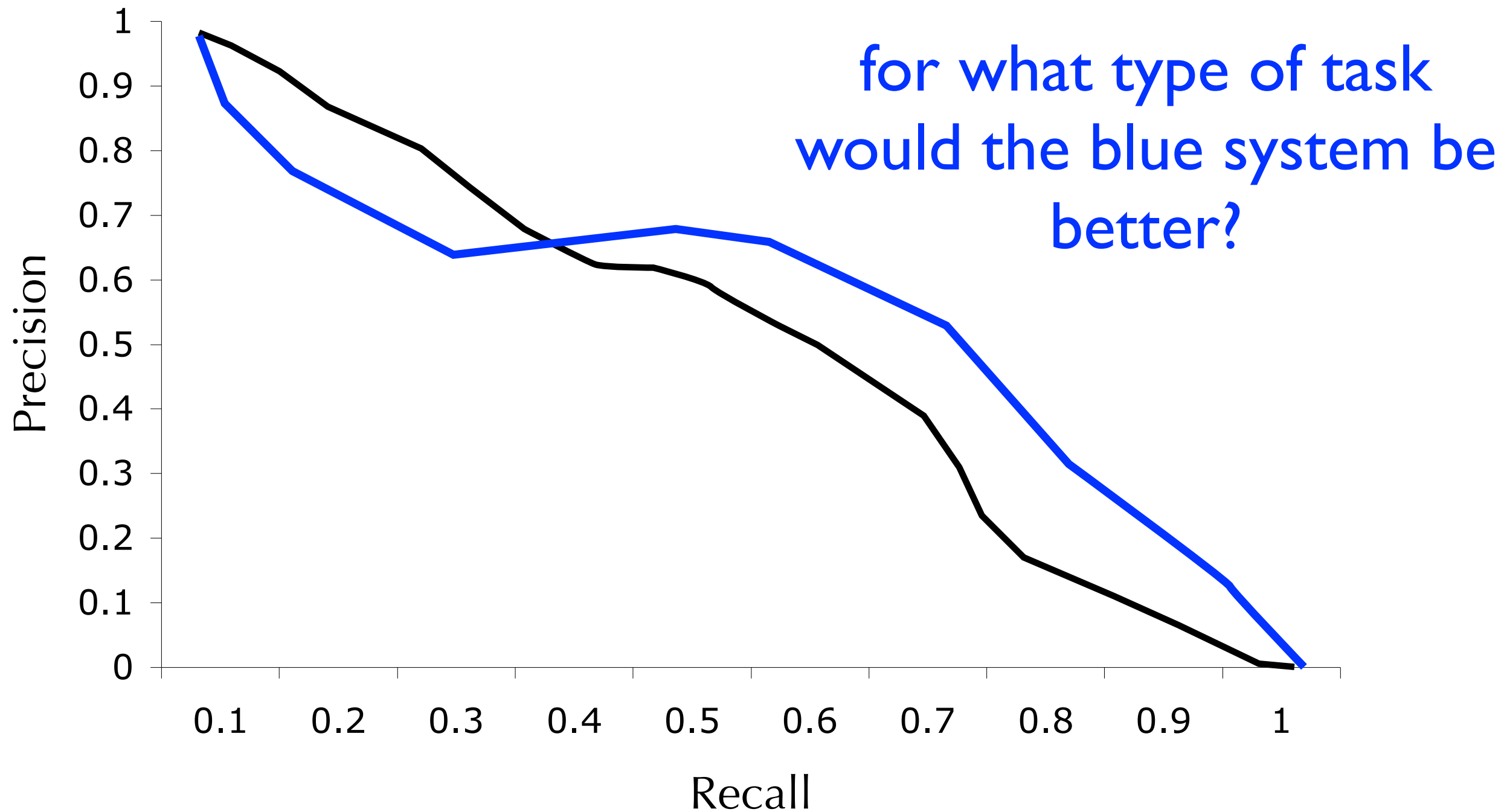
precision-recall curves



for what type of task  
would the blue system be  
better?

# Ranked Retrieval

precision-recall curves



# Ranked Retrieval

## discounted-cumulative gain

- In some retrieval tasks, we really want to focus on precision at the top of the ranking
- A classic example is web-search!
  - ▶ users rarely care about recall
  - ▶ users rarely navigate beyond the first page of results
  - ▶ users may not even look at results below the “fold”
- Are any of the metrics we’ve seen so far appropriate for web-search?

# Ranked Retrieval

## discounted-cumulative gain

### CTR Curve



Study Conducted by  Slingshot<sup>seo</sup>

# Ranked Retrieval

## discounted-cumulative gain

- We could potentially evaluate using  $P@K$  with several small values of  $K$
- But, this has some limitations
- What are they?

# Ranked Retrieval

discounted-cumulative gain

- Which retrieval is better?



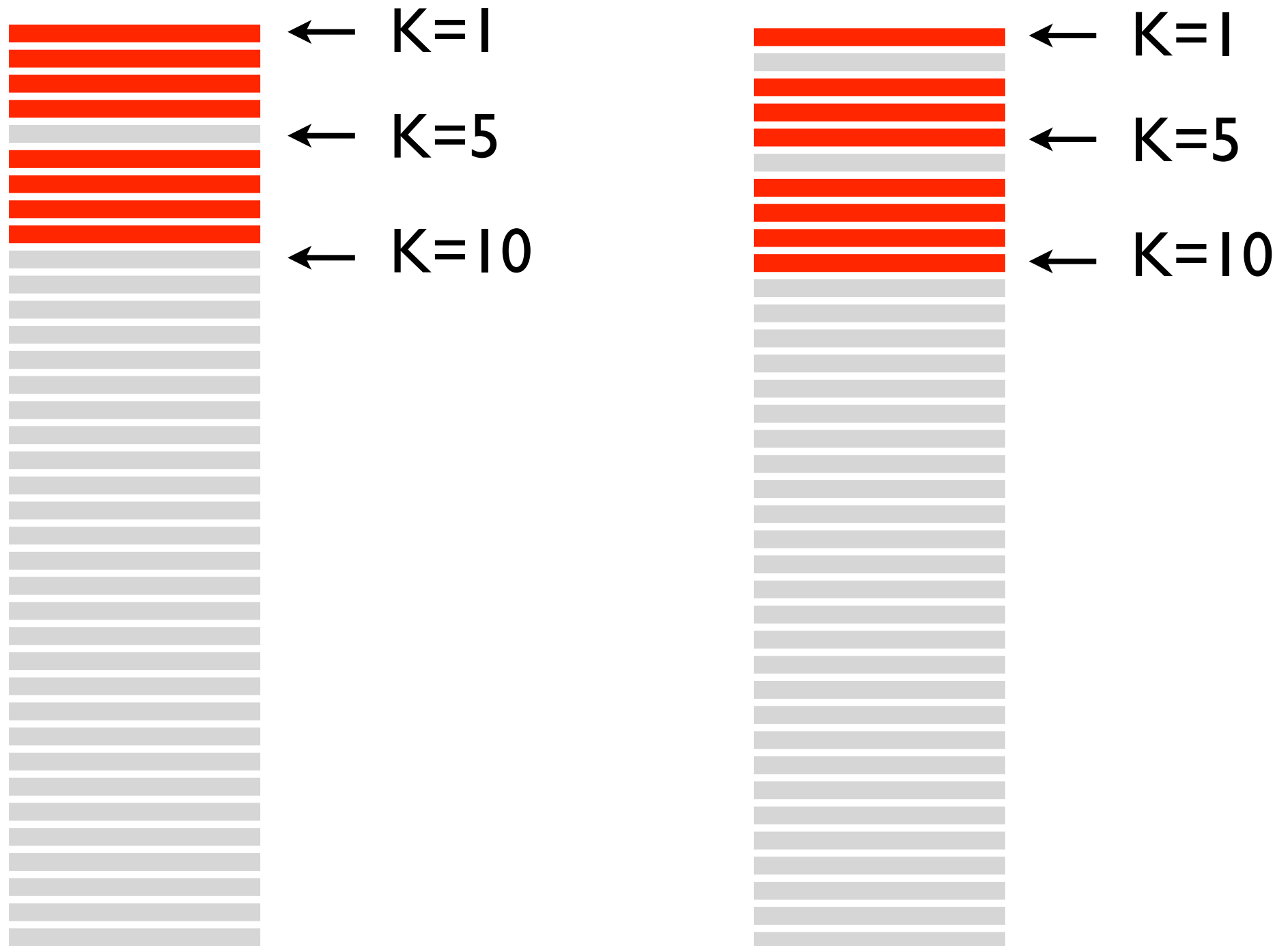
# Evaluation Metrics

Jaime Arguello  
INLS 509: Information Retrieval  
[jarguell@email.unc.edu](mailto:jarguell@email.unc.edu)

October 26, 2016

# Ranked Retrieval

- Evaluation based on  $P@K$  can be too coarse





# Ranked Retrieval

## discounted-cumulative gain

- DCG: discounted cumulative gain
- Assumptions:
  - ▶ Graded vs. binary relevance: perfect, excellent, good, fair, bad
  - ▶ The likelihood that a user will examine a search result decreases rapidly with rank (more rapidly than linearly)

# Ranked Retrieval

## discounted-cumulative gain

- Let  $REL_i$  be the relevance associated with the document at rank  $i$ 
  - ▶ perfect  $\rightarrow 4$
  - ▶ excellent  $\rightarrow 3$
  - ▶ good  $\rightarrow 2$
  - ▶ fair  $\rightarrow 1$
  - ▶ bad  $\rightarrow 0$

# Ranked Retrieval

## discounted-cumulative gain

- **DCG**: discounted cumulative gain

$$DCG@K = \sum_{i=1}^K \frac{REL_i}{\log_2(\max(i, 2))}$$

# Ranked Retrieval

## discounted-cumulative gain

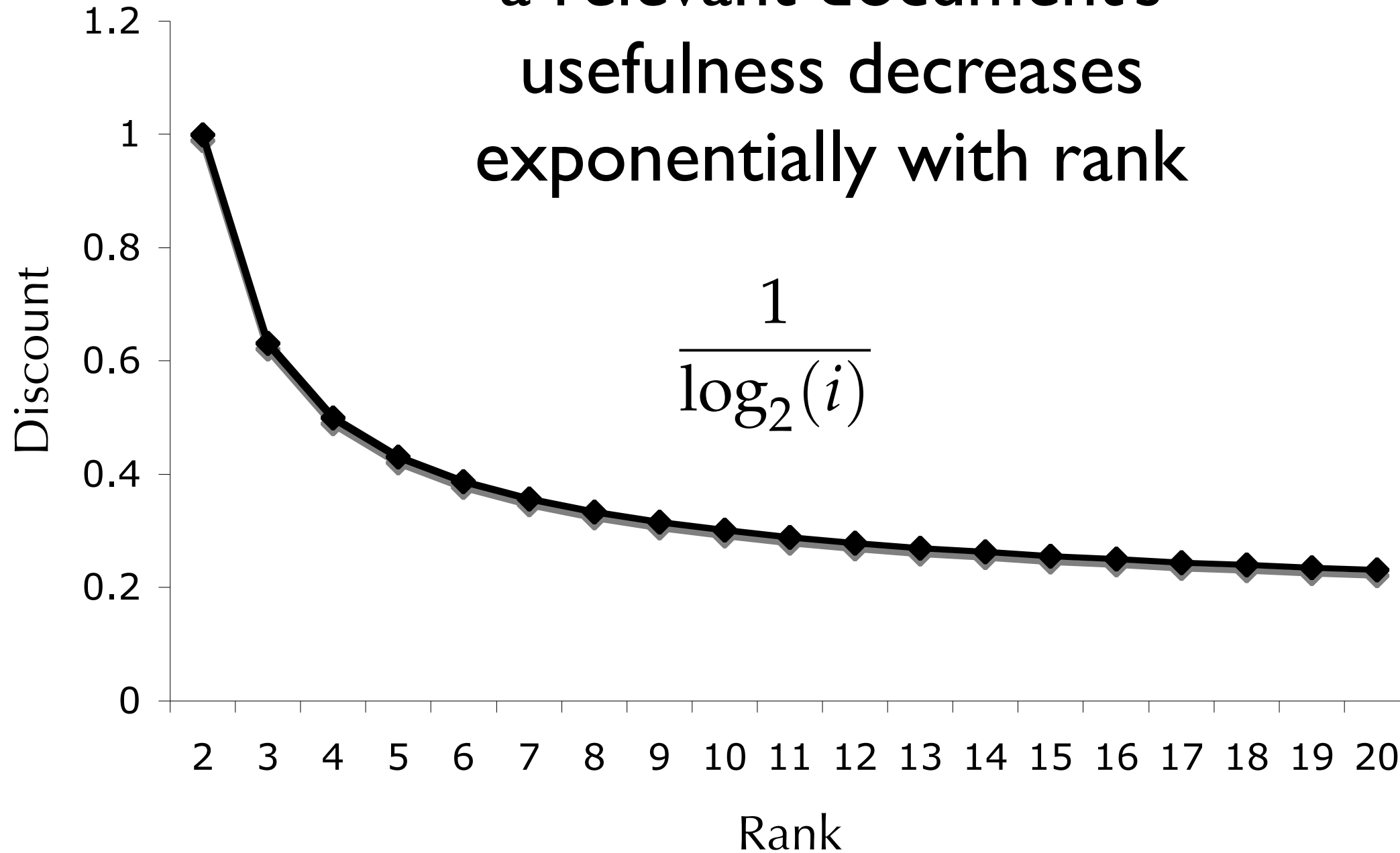
- DCG: discounted cumulative gain

$$DCG@K = \sum_{i=1}^K \frac{REL_i}{\log_2(\max(i, 2))}$$

# Ranked Retrieval

discounted-cumulative gain

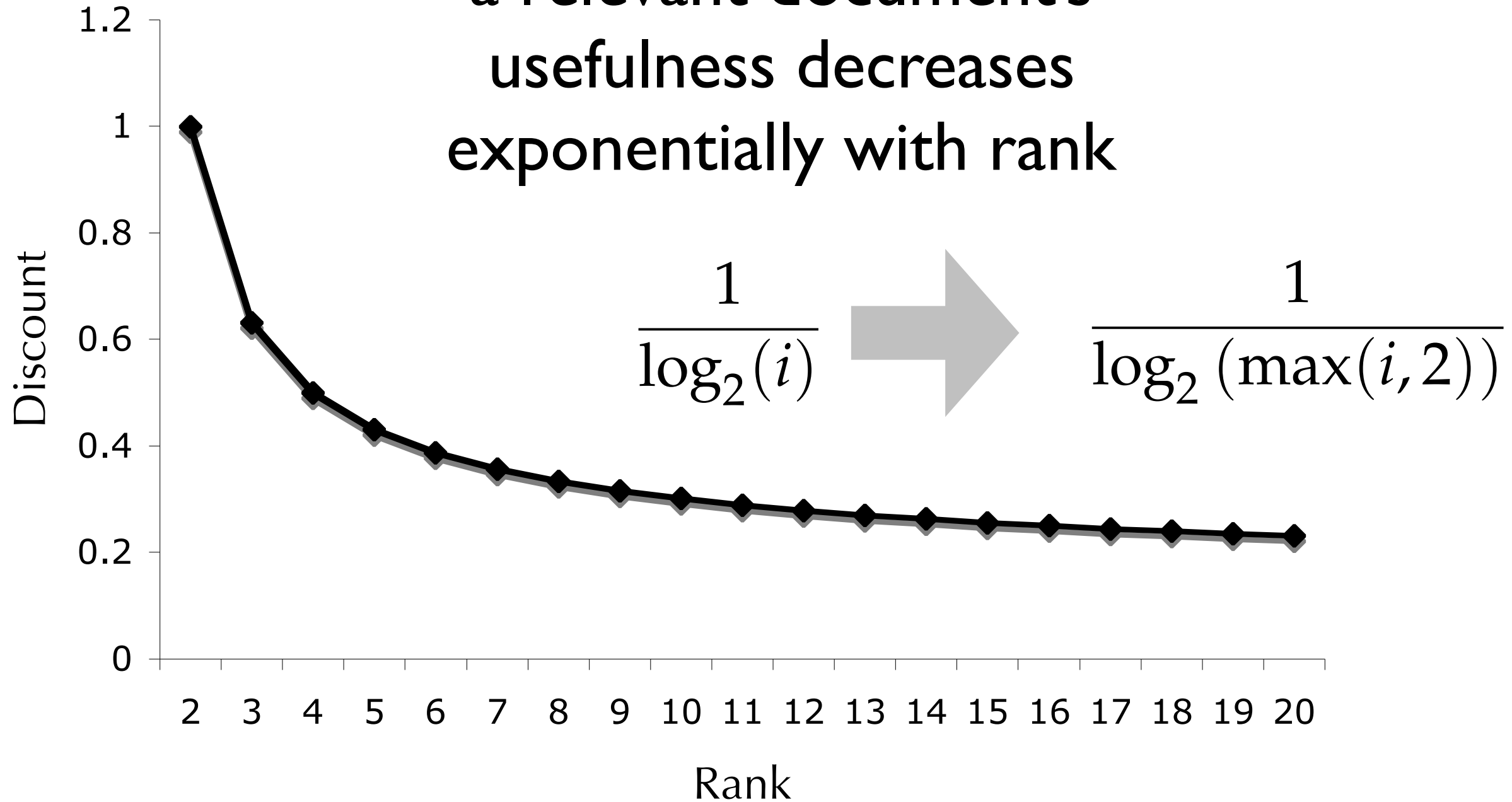
a relevant document's  
usefulness decreases  
exponentially with rank



# Ranked Retrieval

discounted-cumulative gain

a relevant document's  
usefulness decreases  
exponentially with rank



# Ranked Retrieval

## discounted-cumulative gain

$$DCG@K = \sum_{i=1}^K \frac{REL_i}{\log_2(\max(i, 2))}$$

rank (i)	REL_i	
1	4	This is given!
2	3	
3	4	the result at rank 1 is perfect
4	2	the result at rank 2 is excellent
5	0	the result at rank 3 is perfect
6	0	...
7	0	the result at rank 10 is bad
8	1	
9	1	
10	0	

# Ranked Retrieval

## discounted-cumulative gain

$$DCG@K = \sum_{i=1}^K \frac{REL_i}{\log_2(\max(i, 2))}$$

rank (i)	REL_i	discount factor
1	4	1.00
2	3	1.00
3	4	0.63
4	2	0.50
5	0	0.43
6	0	0.39
7	0	0.36
8	1	0.33
9	1	0.32
10	0	0.30

Each rank is associated with a discount factor

$$\frac{1}{\log_2(\max(i, 2))}$$

rank 1 is a special case!



# Ranked Retrieval

## discounted-cumulative gain

$$DCG@K = \sum_{i=1}^K \frac{REL_i}{\log_2(\max(i, 2))}$$

rank (i)	REL_i	discount factor	gain
1	4	1.00	4.00
2	3	1.00	3.00
3	4	0.63	2.52
4	2	0.50	1.00
5	0	0.43	0.00
6	0	0.39	0.00
7	0	0.36	0.00
8	1	0.33	0.33
9	1	0.32	0.32
10	0	0.30	0.00

multiply  $REL_i$   
by the  
discount  
factor  
associated  
with the  
rank!

# Ranked Retrieval

## discounted-cumulative gain

$$DCG@K = \sum_{i=1}^K \frac{REL_i}{\log_2(\max(i, 2))}$$

rank (i)	REL_i	discount factor	gain	DCG_i
1	4	1.00	4.00	4.00
2	3	1.00	3.00	7.00
3	4	0.63	2.52	9.52
4	2	0.50	1.00	10.52
5	0	0.43	0.00	10.52
6	0	0.39	0.00	10.52
7	0	0.36	0.00	10.52
8	1	0.33	0.33	10.86
9	1	0.32	0.32	11.17
10	0	0.30	0.00	11.17

# Ranked Retrieval

discounted-cumulative gain

$$DCG_{10} = 11.17$$

rank (i)	REL_i	discount factor	gain	DCG_i
1	4	1.00	4.00	4.00
2	3	1.00	3.00	7.00
3	4	0.63	2.52	9.52
4	2	0.50	1.00	10.52
5	0	0.43	0.00	10.52
6	0	0.39	0.00	10.52
7	0	0.36	0.00	10.52
8	1	0.33	0.33	10.86
9	1	0.32	0.32	11.17
10	0	0.30	0.00	11.17

# Ranked Retrieval

discounted-cumulative gain

$$DCG_{10} = 10.17$$

rank (i)	REL_i	discount factor	gain	DCG_i
1	3	1.00	3.00	3.00
2	3	1.00	3.00	6.00
3	4	0.63	2.52	8.52
4	2	0.50	1.00	9.52
5	0	0.43	0.00	9.52
6	0	0.39	0.00	9.52
7	0	0.36	0.00	9.52
8	1	0.33	0.33	9.86
9	1	0.32	0.32	10.17
10	0	0.30	0.00	10.17

changed top result from **perfect** instead of **excellent**

# Ranked Retrieval

discounted-cumulative gain

$$DCG_{10} = 11.17$$

rank (i)	REL_i	discount factor	gain	DCG_i
1	4	1.00	4.00	4.00
2	3	1.00	3.00	7.00
3	4	0.63	2.52	9.52
4	2	0.50	1.00	10.52
5	0	0.43	0.00	10.52
6	0	0.39	0.00	10.52
7	0	0.36	0.00	10.52
8	1	0.33	0.33	10.86
9	1	0.32	0.32	11.17
10	0	0.30	0.00	11.17

# Ranked Retrieval

discounted-cumulative gain

$$DCG_{10} = 12.08$$

rank (i)	REL_i	discount factor	gain	DCG_i
1	4	1.00	4.00	4.00
2	3	1.00	3.00	7.00
3	4	0.63	2.52	9.52
4	2	0.50	1.00	10.52
5	0	0.43	0.00	10.52
6	0	0.39	0.00	10.52
7	0	0.36	0.00	10.52
8	1	0.33	0.33	10.86
9	1	0.32	0.32	11.17
10	3	0.30	0.90	12.08

changed 10th result from **bad** to **excellent**

# Ranked Retrieval

normalized discounted-cumulative gain

- DCG is not 'bounded'
- In other words, it ranges from zero to .....
- Makes it problematic to average across queries
- NDCG: normalized discounted-cumulative gain
- "Normalized" is a fancy way of saying, we change it so that it ranges from 0 to 1

# Ranked Retrieval

normalized discounted-cumulative gain

- $NDCG_i$ : normalized discounted-cumulative gain
- For a given query, measure  $DCG_i$
- Then, divide this  $DCG_i$  value by the best possible  $DCG_i$  for that query



# Ranked Retrieval

normalized discounted-cumulative gain

- **Given:** a query has two 4's, one 3, and the rest are 0's
- **Question:** What is the best possible ranking for  $i = 1$

# Ranked Retrieval

normalized discounted-cumulative gain

- **Given:** a query has two 4's, one 3, and the rest are 0's
- **Question:** What is the best possible ranking for  $i = 1$
- All these are equally good:
  - ▶ 4, 4, 3, ....
  - ▶ 4, 3, 4, ....
  - ▶ 4, 0, 0, ....
  - ▶ ... anything with a 4 as the top-ranked result

# Ranked Retrieval

normalized discounted-cumulative gain

- **Given:** a query has two 4's, one 3, and the rest are 0's
- **Question:** What is the best possible ranking for  $i = 2$

# Ranked Retrieval

normalized discounted-cumulative gain

- **Given:** a query has two 4's, one 3, and the rest are 0's
- **Question:** What is the best possible ranking for  $i = 2$
- All these are equally good:
  - ▶ 4, 4, 3, ....
  - ▶ 4, 4, 0, ....

# Ranked Retrieval

normalized discounted-cumulative gain

- **Given:** a query has two 4's, one 3, and the rest are 0's
- **Question:** What is the best possible ranking for  $i = 3$

# Ranked Retrieval

normalized discounted-cumulative gain

- **Given:** a query has two 4's, one 3, and the rest are 0's
- **Question:** What is the best possible ranking for  $i = 3$
- All these are equally good:
  - ▶ 4, 4, 3, ....

# Ranked Retrieval

normalized discounted-cumulative gain

- $NDCG_i$ : normalized discounted-cumulative gain
- For a given query, measure  $DCG_i$
- Then, divide this  $DCG_i$  value by the best possible  $DCG_i$  for that query
- Measure  $DCG_i$  for the best possible ranking for a given value  $i$

# Metric Review

- **set-retrieval evaluation:** we want to evaluate the set of documents retrieved by the system, without considering the ranking
- **ranked-retrieval evaluation:** we want to evaluate the ranking of documents returned by the system



# Metric Review

## set-retrieval evaluation

- **precision:** the proportion of retrieved documents that are relevant
- **recall:** the proportion of relevant documents that are retrieved
- **f-measure:** harmonic-mean of precision and recall
  - ▶ a difficult metric to “cheat” by getting very high precision and abysmal recall (and vice-versa)

# Metric Review

## ranked-retrieval evaluation

- **P@K**: precision under the assumption that the top-K results is the 'set' retrieved
- **R@K**: recall under the assumption that the top-K results is the 'set' retrieved
- **average-precision**: average of P@K values for every K where recall increases
- **DCG**: ignores recall, considers multiple levels of relevance, and focuses on the top ranks
- **NDCG**: make DCG range between 0 and 1

# Which Metric Would You Use?



bing

twitter



eHarmony

PANDORA

Google



match.com

mapquest m<sup>q</sup>



YAHOO! ANSWERS



LinkedIn

flickr



Picasa

Westlaw

The New York Times

yelp

YouTube

Broadcast Yourself™

LexisNexis

