IS2140 Information Storage and Retrieval

Unit 6: Evaluation of IR Systems

Daqing He
School of Information Sciences
University of Pittsburgh

October 8, 2018

Muddiest Points

- Efficiency in Vector Space Model
  - When we rank with Cosine values, we normally implement it by heap sort which takes O(nlogk). I'm wondering why cannot we cluster documents first so that similar documents stay in one cluster (assume the cluster includes m documents). After that we implement heap sort which only takes O(mlogk). Is it a faster algorithm?

Muddiest Points

- Language Models vs Vector Space Model
  - Why do we have to calculate the probability for each word when we have the method to make cosine similarity more efficiently?
  - Can you give us two circumstances that the Language Model and the Space Vector model should be applied for? In other words, Is there any specific situation that one is better than another?
  - Do we implement both methods (probability and similarity) together in implementation or there would be a preference in different situation?

Your comments

- One of your comments
  - So far we have talked about the Vector Space Model and the Language Model, and I found the two models share a somehow similar thought in defining the weights. In other words, the LM uses a weighting scheme that have the same effect as TF-IDF by different perspective. The weights of TF-IDF is positive related to the term frequency and the document length is controlled by implement normalization or penalization, while for the weights of LM, it's also positive related to the term frequency and use normalization as the method to remove the effect from document length. The difference is that the sum of the probabilities will always be equal to 1 after normalization but the similarities in vector space don't have this character. Is my understanding correct?

Muddiest Points

- Prediction in Language Models
  - In our lecture slides P37: In a language model for Chinese, p4 > p3 > p1 > p2.
    - p1 = P("a happy running dog")
    - p2 = P("dog running a happy")
    - p3 = P("一条 happy running dog")
    - p4 = P("一条快乐地奔跑的狗")
    - Just take unigram language model for Chinese as an example, the model is actually a dictionary only contains Chinese words as key words. So I think p3 = p1 = p2 = 0. Since the probability for any English words should be ZERO, why p4 > p3 > p1 > p2?

Muddiest Points

- Estimate a Language Model
  - Why don't remove stop words in Unigram Language Model Estimation?
  - In ppt 41, in the unigram language model, when calculating the probability of each word, would 'like' and 'likes' considered be the same word.
  - The point is about language model. Since the granularity of language model can be one or more than document, how can we decide the best granularity of a language model and how can we decide the topic (language model) of a new document?

Muddiest Points

- Queries in Language Models
  - For the statistical language model, should we treat each term in query as same weight, or give the word like "a", "the" smaller weight than other? Or we need to remove the stop word like in vector space model?
  - We know that Multinomial model is good for sequence of events, Multiple-Bernoulli model is good for existence of events. But in real life, how do we choose from these two models? Because when we need to find a article or document which are relevant to the text we searched, both models are useful.?

Agenda

- Evaluation of IR systems
  - Cranfield method and test collections, TREC
  - Evaluation Measures
  - Statistical Testing

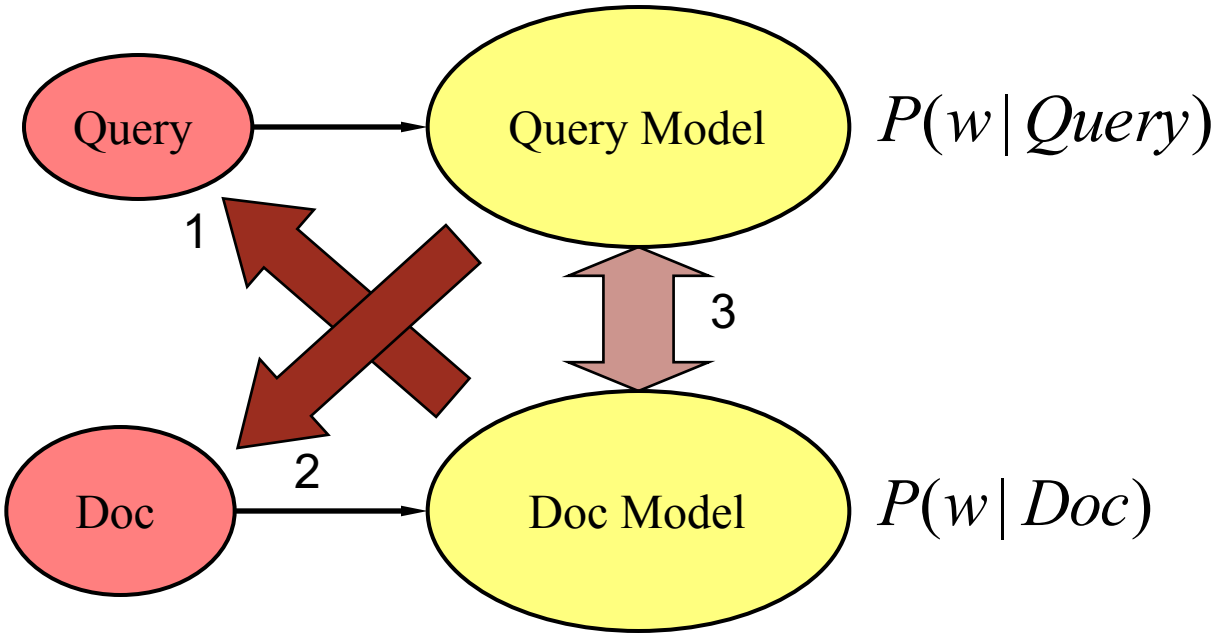
Class Goals

- After this class, you should be able to
  - know the basic ideas of cranfield evaluation framework
  - Know the advantages and limitations of the measures, and is able to calculate mean average precisions
  - Familiar with running an evaluation on a retrieval system using Cranfield evaluation framework

Statistical Language Models

Many slides are based on James Allan , Jimmy Lin and Lavrenko’s related courses

Retrieval Using Language Models



Retrieval: Query likelihood (1), Document likelihood (2), Model comparison (3)

Language Models

- Query Likelihood Language Model

$$P(Q \mid M_D) = P(q_1 \ldots q_k \mid M_D) = \prod_{i=1}^{k} P(q_i \mid M_D)$$

- Document Likelihood Language Models
  - Model 1: $P(D \mid M_Q) = \prod_{w \in D} P(w \mid M_Q)$
  - Model 2: $P(M_Q \mid D) = \frac{P(M_Q)P(D \mid M_Q)}{P(D)} \approx \frac{c\prod_{w \in D} P(w \mid M_Q)}{\prod_{w \in D} P(w \mid GE)}$

Model Comparison: Method 3

- Estimate query and document models and compare
- Suitable measure is KL divergence $D(M_Q \| M_D)$

$$KL(M_Q \| M_D) = \sum_{w \in V} M_Q(w) \log \frac{M_Q(w)}{M_D(w)}$$

dependent to query only

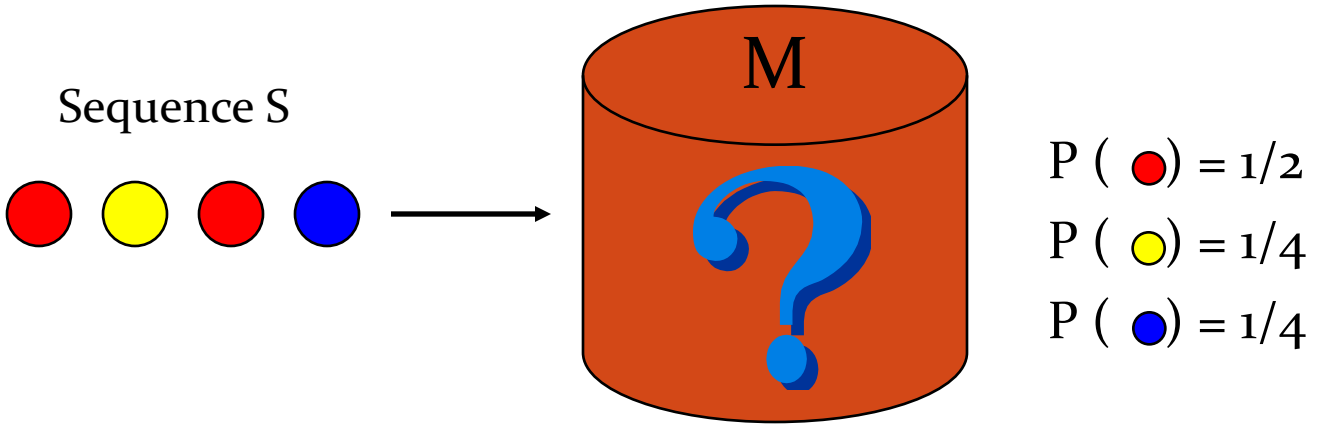
$$= -(\sum_{w \in V} M_Q(w) \log M_D(w)) + (-\sum_{w \in V} M_Q(w) \log M_Q(w)))$$

Use this part for ranking

- Better results than query-likelihood or document-likelihood approaches

Estimation

- Want to estimate M_Q and/or M_D from Q and/or D
- Maximum Likelihood Estimate (MLE or ML): for example estimate M_D is to simply count the frequencies in the document



$$P_{MLE}(w|M_S) = \#(w,S) \,/\, |S|$$

$\#(w,S)$ = number of times *w* occurs in S
$|S|$ = length of S

Exercise

- Suppose we have 3 document collection

Doc1: wrecked roads and bridges in chile hinder rescue effort in chile earthquake

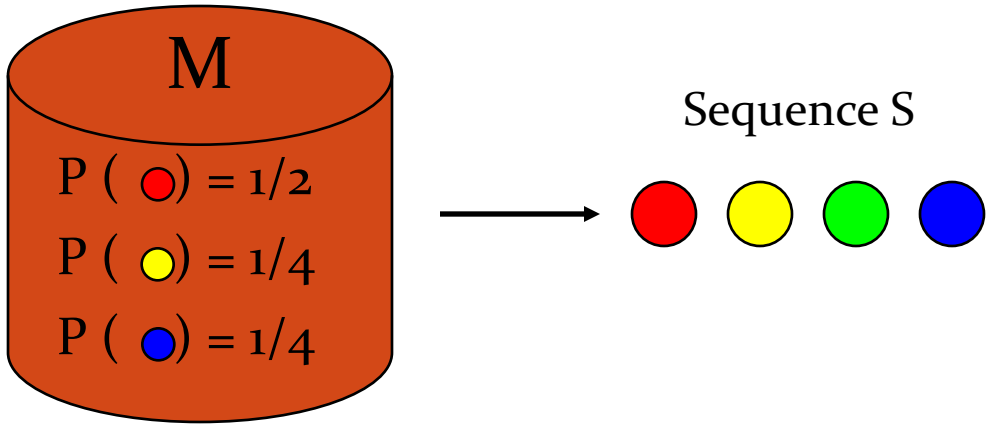
Doc2: test for chile's coalition in presidential election in chile

Doc3: frantic rescue efforts in chile as troops seek to keep order for rescue

- We know the length of doc 1 is 12, that of doc 2 is 9, that of doc 3 is 13
- Suppose the query is chile,
- What is the probability under query likelihood and MLE p(chile | doc 1)?

Zero-Frequency Problem

- Suppose some event is not in our observation S
  - Model will assign zero probability to that event



P ( ) = P ( ) × P ( ) × P ( ) × P ( )

= (1/2) × (1/4) × 0 × (1/4) = 0 !!

Why is this a Bad Idea?

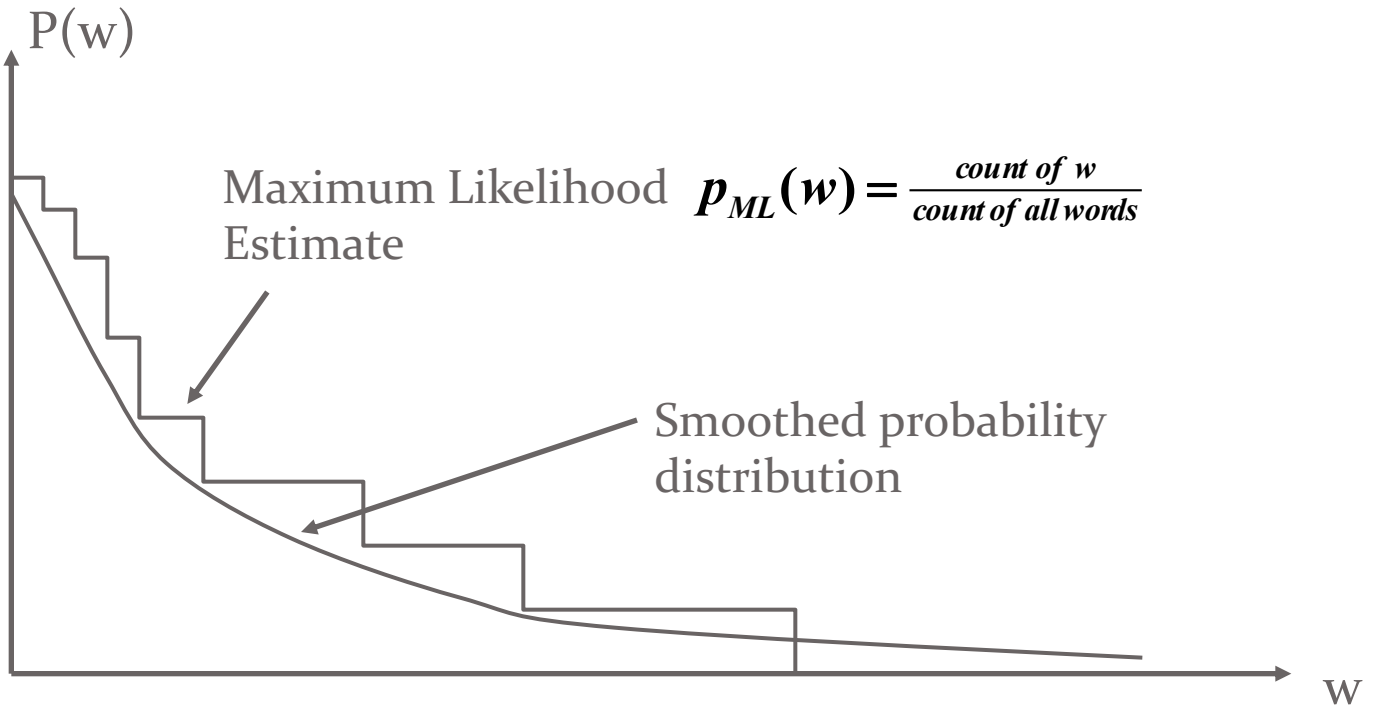
- Think of the document model as a topic
  - There are many documents that can be written about a single topic
  - We're trying to figure out what the model is based on just one document
- Modeling a document
  - Just because a word didn't appear doesn't mean it'll never appear…
  - But safe to assume that unseen words are rare

Analogy: fishes in the sea

- Practical effect: assigning zero probability to unseen words forces exact match
  - But partial matches are useful also!

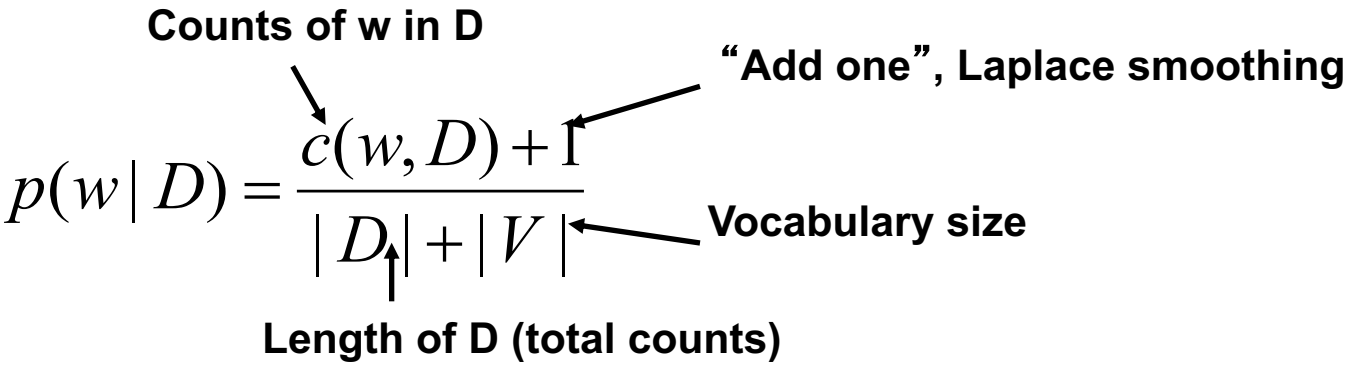
Smoothing

- All smoothing methods try to
  - discount the probability of words seen in a document
  - re-allocate the extra counts so that unseen words will have a non-zero count

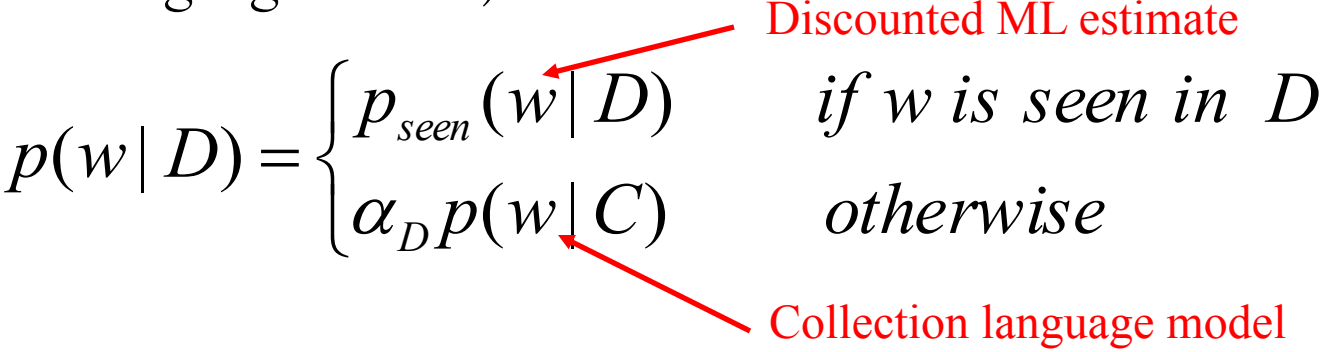


How to Smooth?

- A simple method (additive smoothing): **Add a small constant to the counts of each word**



- Another method using a reference model (collection language model) to discriminate unseen words



Linear interpolation Smoothing

- Also called Jelinek-Mercer smoothing
- Mixes the probability from the document with the general collection
  - "Shrink" uniformly toward p(w | Mc)

$$p(w|D) = \lambda p(w|M_D) + (1-\lambda) p(w|M_C)$$

- λ often is set around 0.8
- Methods for identifying optimal λ
  - Split data into training, held-out, and test
  - Train model on training set
  - Use held-out to test different values and pick the ones that works best (i.e., maximize the likelihood of the held-out data)
  - Test the model on the test data

Dirichlet Prior Smoothing

- Dirichlet prior smoothing is one particular smoothing method that follows the general smoothing scheme

$$p(w|D) = \frac{c(w,D) + \mu p(w|REF)}{|D| + \mu}$$

$$= \frac{|D|}{|D| + \mu} \frac{c(w,D)}{|D|} + \frac{\mu}{|D| + \mu} p(w|REF)$$

The optimal prior μ seems to vary from collection to collection, though in most cases, it is around 2,000.

Exercise

- Suppose we have 3 document collection

Doc1: wrecked roads and bridges in chile hinder rescue effort in chile earthquake

Doc2: test for chile's coalition in presidential election in chile

Doc3: frantic rescue efforts in chile as troops seek to keep order for rescue

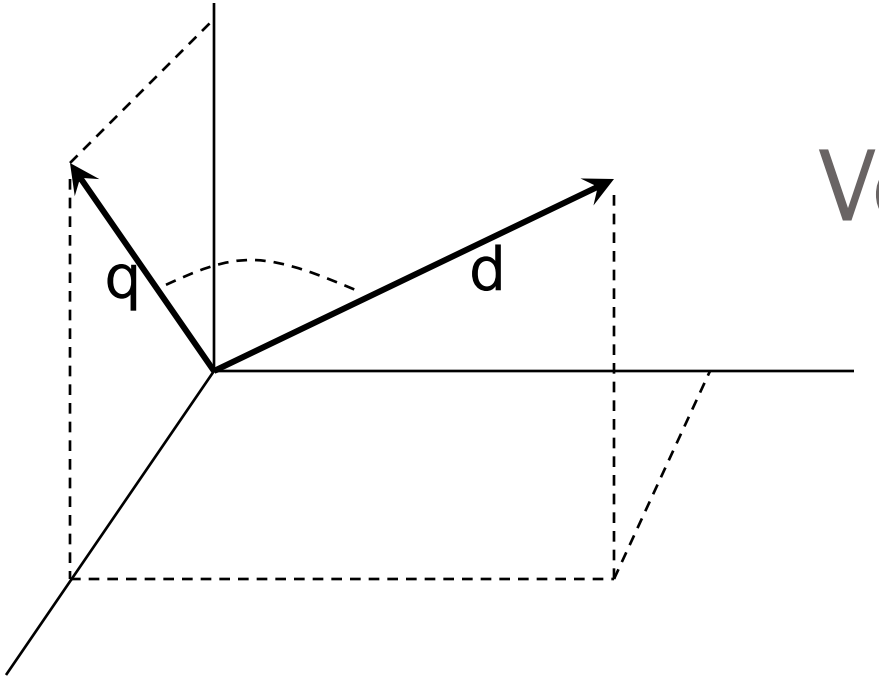
- If we use JM smoothing $p(t|M_D) = \lambda * p(t|D) + (1-\lambda) * p(t|C)$, where C is the whole collection and λ is 0.8, what is the probability p(earthquake | doc 3) and p(rescue | doc 3)?

Major Issues in Applying LM

- What kind of language model should we use?
  - Unigram or higher-order models
  - Multinomial or multiple-Bernoulli?
- How can we use the model for ranking?
  - Query-likelihood
  - Document-likelihood
  - Divergence of query and document models
- Many other issues, e.g.
  - how can we estimate model parameters?
  - How to model relevance
  - How to model relevance feedback

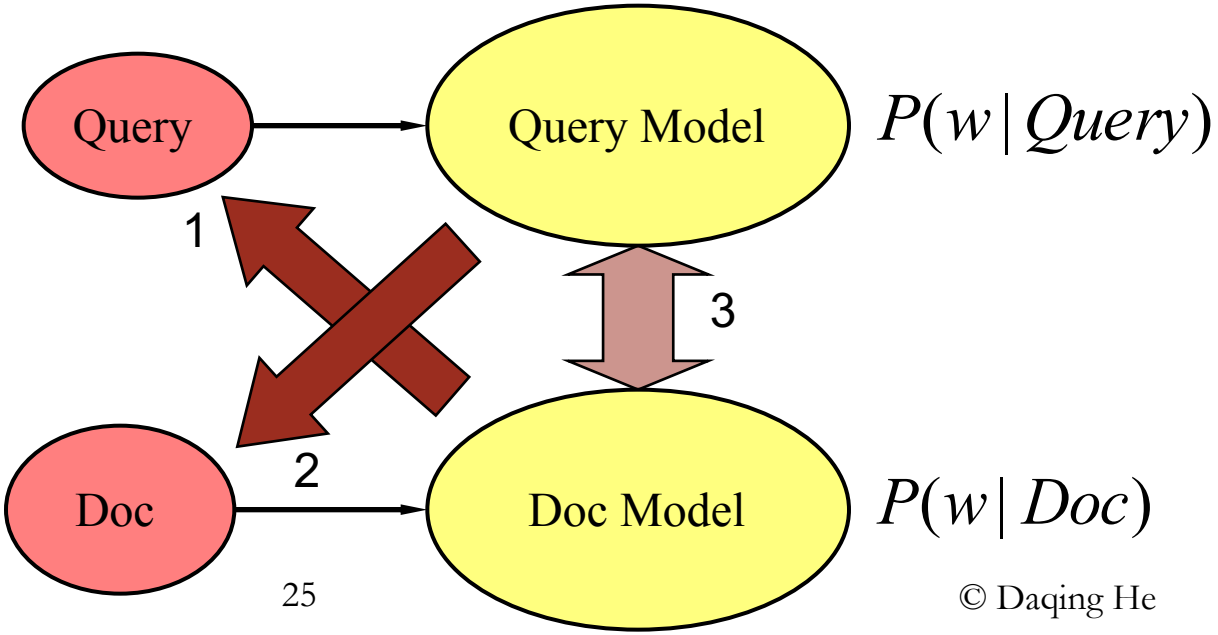
Language Modeling: pros and cons

- Pros
  - Formal mathematical model
  - Simple, well-understood framework
  - Integrates both indexing and retrieval models
  - Natural use of collection statistics, no heuristics
  - Avoid tricky issues of "relevance", "aboutness", etc
- Cons
  - Difficult to incorporate notions of "relevance", user preferences
  - Relevance feedback / query expansion not straightforward
  - Can't accommodate phrases, passages, Boolean operator
  - But there are recent LM works that address these issues



Vector Space

Language Modeling .



Language Modeling vs Vector Space

- Similarities
  - Term weights based on frequency
  - Terms often used as if they were independent
  - Inverse document/collection frequency used
  - Some form of length normalization useful

- Differences
  - Based on probability rather than similarity
    - Intuitions are probabilistic rather than geometric
  - Details of use of document length and term, document, and collection frequency differ

Reference for Language Modelings

1. J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. Proceedings of ACM-SIGIR 1998, pages 275-281.
2. J. M. Ponte. A language modeling approach to information retrieval. Phd dissertation, University of Massachusets, Amherst, MA, September 1998.
3. D. Hiemstra. Using Language Models for Information Retrieval. PhD dissertation, University of Twente, Enschede, The Netherlands, January 2001.
4. D. R. H. Miller, T. Leek, and R. M. Schwartz. A hidden Markov model information retrieval system. Proceedings of ACM-SIGIR 1999, pages 214-221.
5. F. Song and W. B. Croft. A general language model for information retrieval. In Proceedings of Eighth International Conference on Information and Knowledge Management (CIKM 1999)
6. S. F. Chen and J. T. Goodman. An empirical study of smoothing techniques for language modeling. In Proceedings of the 34th Annual Meeting of the ACL, 1996.
7. C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. Proceedings of the ACM-SIGIR 2001, pages 334-342.
8. V. Lavrenko and W. B. Croft. Relevance-based language models. Proceedings of the ACM SIGIR 2001, pages 120-127.
9. V. Lavrenko and W. B. Croft, Relevance Models in Information Retrieval, in Language Modeling for Information Retrieval, W. Bruce Croft and John Lafferty, ed., Kluwer Academic Publishers, chapter 2.

Classic Probabilistic Models

Many slides are based on James Allan's related courses and Manning's Introduction to IR book

A few words on BIR model and its extensions

- One of the oldest retrieval model
  - Give a firm theoretical foundations in 1970
  - Originally designed for short catalog records
  - Do not pay attention to term frequency and document length
    - So do not really suitable for full text retreival
- Revised in recent studies so it is among the best retrieval models
  - BM25 weighting scheme (also called Okapi weighting scheme)

$$RSV_d = \sum_{t \in q} \left[\log \frac{N}{\mathrm{df}_t} \right] \cdot \frac{(k_1 + 1)\mathrm{tf}_{td}}{k_1((1 - b) + b \times (L_d / L_{\mathrm{ave}})) + \mathrm{tf}_{td}} \cdot \frac{(k_3 + 1)\mathrm{tf}_{tq}}{k_3 + \mathrm{tf}_{tq}}$$

where k1 and k3 has a value between 1.2 and 2, and b = 0.75

Evaluation

IR is an Empirical Discipline

- Goal is practical
  - Help people find relevant information they want
- Process is complicated with many decision points
  - During indexing of documents
  - During processing of queries
  - During matching queries against documents
- The goal determines which decision to take
  - Need careful and thorough evaluation
- So evaluation becomes the center issue in IR
  - All novel techniques need demonstrate superior performance on representative document collections

Important Issues in Evaluation

- Focus of evaluation
  - Retrieval process or retrieval algorithm
- If focused on retrieval process, aspects for evaluation are
  - Assistance in formulating queries
  - Aspects of retrieval algorithm
  - Resource required
  - Presentation of search results
  - Ability to find relevant documents
  - Appealing to users
- If focused on retrieval algorithm, aspects for evaluation are
  - Efficiency or effectiveness

Important Issues in Evaluation - II

- Types of evaluation
  - Objective evaluation: how well a system performs
  - Comparative evaluation: how one compares to others
- In IR, most time we see evaluations are
  - Focus on retrieval algorithms
  - Look at effectiveness of retrieval
  - Comparative evaluation
- But we do see more and more evaluations
  - Look at the whole retrieval process
  - Keep users in the loop

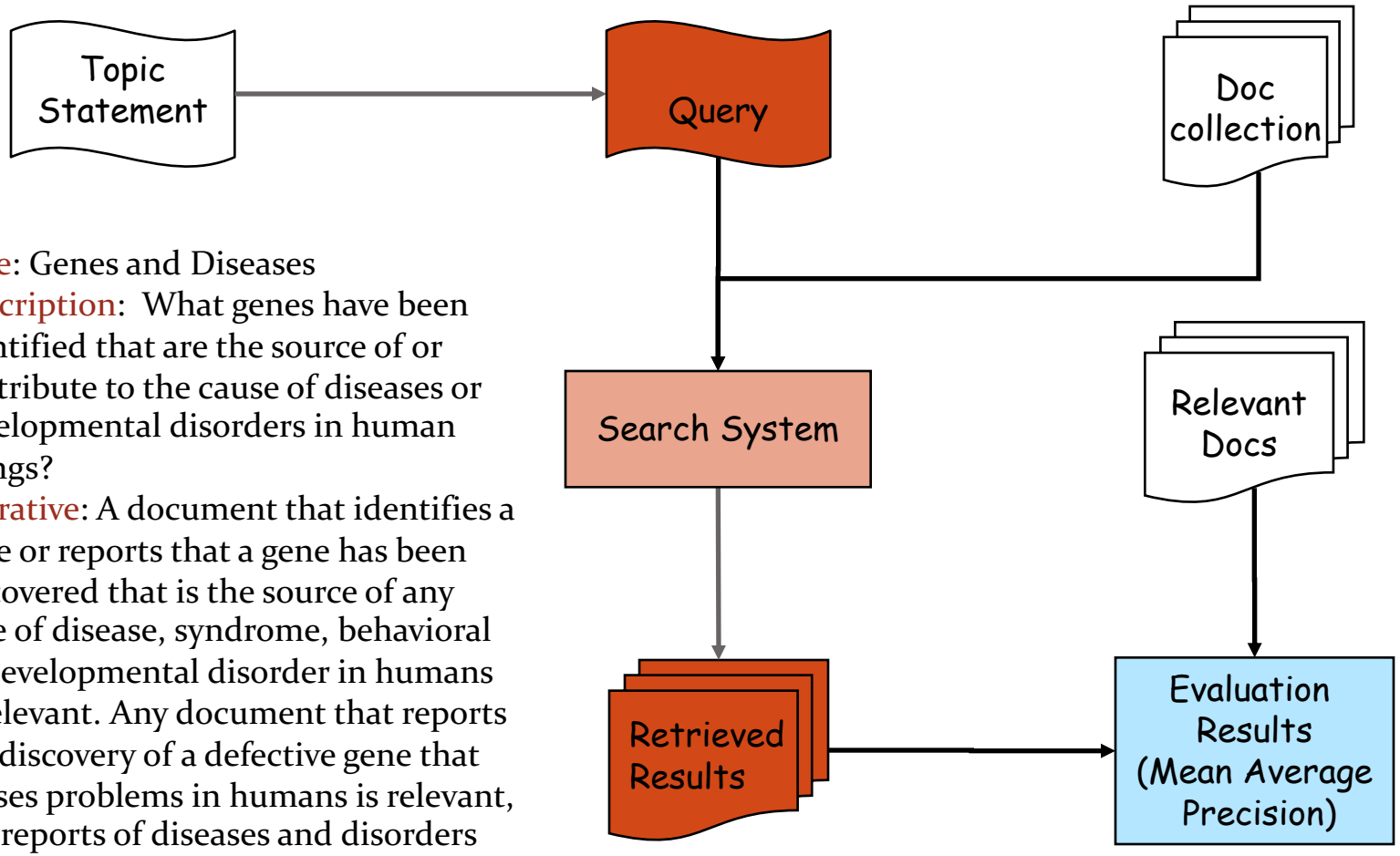
The Cranfield Methodology

- Basics
  - Early and influential studies on IR effectiveness
  - Goal was to compare human and automatic indexing methods
  - Basics of the 1967 experiment
    - Data: 1400 documents (titles and abstracts only), 225 descriptions of information needs (usually in sentence length), Human decision about which documents satisfy which information needs
    - Measures: precision and recall
    - Methodology: use different index to retrieval documents, use precision and recall to judge the results.
- Importance: experimental methodology influential even today
  - **Test collection** that consists of **a document collection**, a set of **information needs (search requests)**, and **the ground truth** between the documents and the needs
  - Cross-site/platform comparison: since the experiments are on a static test collection that is always available, it is possible for experiment results to be compared across different time period or different sites.

Steps in Cranfield Methodology

- Obtain a test collection
  - Obtain a corpus of documents
  - Obtain a set of search requests
  - Obtain the ground truth judgments
- Conduct retrievals using different experiment conditions
  - Obtain several search results
- Measure the effectiveness of the searches by comparing the relevant documents in ground truth and those in the search results
- To truly compare different retrieval algorithms, may need multiple test collections
  - WHY?

TREC Batch Retrieval Evaluation



Title: Genes and Diseases
Description: What genes have been identified that are the source of or contribute to the cause of diseases or developmental disorders in human beings?
Narrative: A document that identifies a gene or reports that a gene has been discovered that is the source of any type of disease, syndrome, behavioral or developmental disorder in humans is relevant. Any document that reports the discovery of a defective gene that causes problems in humans is relevant, but reports of diseases and disorders that are caused by the absence of a gene are not relevant.

TREC

- Text REtrieval Conference
- Established in 1992 to evaluate large-scale IR
  - Has run continuously since then, 2008 is 17th meeting
- Run by NIST information access division
  - Initially supported by DARPA as part of Tipster program
  - Now supported by many agencies, including DARPA, ARDA and NIST
- Probably the most well-known IR evaluation framework
  - A forum for exchanging research ideas and developing research methodology
- Proceedings available on-line (http://trec.nist.gov)

TREC Tracks and Participants

Table 1: Number of participants per track and total number of distinct participants in each TREC

	TREC														
Track	'92	'93	'94	'95	'96	'97	'98	'99	'00	'01	'02	'03	'04	'05	'06
Ad Hoc	18	24	26	23	28	31	42	41							
Routing	16	25	25	15	16	21									
Interactive			3	11	2	9	8	7	6	6	6				
Spanish			4	10	7										
Confusion				4	5										
Merging				3	3										
Filtering				4	7	10	12	14	15	19	21				
Chinese					9	12									
NLP					4	2									
Speech						13	10	10	3						
XLingual						13	9	13	16	10	9				
High Prec						5	4								
VLC							7	6							
Query							2	5	6						
QA								20	28	36	34	33	28	33	31
Web								17	23	30	23	27	18		
Video										12	19				
Novelty											13	14	14		
Genomics												29	33	41	30
HARD												14	16	16	
Robust												16	14	17	
Terabyte													17	19	21
Enterprise														23	25
Spam														13	9
Legal															6
Blog															16
Participants	22	31	33	36	38	51	56	66	69	87	93	93	103	117	107

- http://trec.nist.gov/tracks.html

Test Collections Available

- ClueWeb 12 collection http://lemurproject.org/clueweb12/
  - 733,019,372 web pages
  - 5.54 TB compressed. (27.3TB, uncompressed.)
- ClueWeb 09 collection http://lemurproject.org/clueweb09/
  - 1,040,809,705 web pages, in 10 languages
  - 5 TB, compressed. (25 TB, uncompressed.)
- Some earlier collections

Characteristic	Cranfield	CACM	TREC-2	RCV1	WT10g	GOV2
Size (docs)	1.4 K	3.2 K	742 K	807 K	1.7 M	25 M
Size	1.5 MB	2.3 MB	2.2 GB	2.5 GB	11 GB	427 GB
Year Created	1968	1983	1993	2000	2000	2004
Stems	8.2 K	5.5 K	1 M	557 K	4.7 M	51.2 M
Stem Occurrences	123 K	117 K	244 M	203 M	1 B	22.8 B
Avg Doc Length	88	37	328	252	606	905
Queries	225	50	100	50	100	100

Ad Hoc Topics

- In TREC, a statement of information need is called a *topic*

  Title: Health and Computer Terminals

  Description: Is it hazardous to the health of individuals to work with computer terminals on a daily basis?

  Narrative: Relevant documents would contain any information that expands on any physical disorder/problems that may be associated with the daily working with computer terminals. Such things as carpel tunnel, cataracts, and fatigue have been said to be associated, but how widespread are these or other problems and what is being done to alleviate any health problems.

How to Collect Relevance Judgments?

- In test collection, the most expensive part is the ground truth
  - Relevance judgments have to be done manually
  - The quality of relevance judgments can affect the evaluation results
- In small collections, can review all documents for all queries
- But it is not practical in large collections
  - TREC collections contain millions of documents, too expensive
- Solutions
  - Search-guided relevance assessment
  - Known-item judgments
  - Pooling

How to Find Relevant Documents?

- Search-guided relevance assessment
  - Iterate between topic research/search/assessment
    - Use manually guided search to retrieve documents from the collection
    - Read returned documents for relevant judgment
    - Until convinced all relevant documents have found
- Known-item judgments have the lowest cost
  - Tailor queries to retrieve a single known document
  - Useful as a first cut to see if a new technique is viable

Pooling Method

- Retrieve documents using several techniques or systems
- use top n documents from each search result to build a pool for judgment
  - Single pool, duplicates removed, arbitrary order
  - Judged by the person who developed the topic
- Relevant document set is the union of all relevant documents from each result
  - Treat unevaluated documents as not relevant
- To make pooling work:
  - Systems must do reasonable well
  - Systems must not all "do the same thing"

Does pooling work?

- But judgments can't possibly be exhaustive!

  It doesn't matter: relative rankings remain the same!

  Chris Buckley and Ellen M. Voorhees. (2004) Retrieval Evaluation with Incomplete Information. SIGIR 2004.

- But this is only one person's opinion about relevance

  It doesn't matter: relative rankings remain the same!

  Ellen Voorhees. (1998) Variations in Relevance Judgments and the Measurement of Retrieval Effectiveness. SIGIR 1998.

- But what about hits 101 to 1000?

  It doesn't matter: relative rankings remain the same!

- But we can't possibly use judgments to evaluate a system that didn't participate in the evaluation!

  Actually, we can!

  Justin Zobel. (1998) How Reliable Are the Results of Large-Scale Information Retrieval Experiments? SIGIR 1998.

Kappa measure for inter-judge agreement

- Kappa measure
  - Agreement measure among judges
  - Designed for categorical judgments
  - Corrects for chance agreement
- Kappa = [P(A) – P(E)] / [1 – P(E)]
- P(A) – proportion of time judges agree
- P(E) – what agreement would be by chance
- Kappa = 0 for chance agreement, 1 for total agreement.

P(A)? P(E)?

Kappa Measure: Example

Number of docs	Judge 1	Judge 2
300	Relevant	Relevant
70	Nonrelevant	Nonrelevant
20	Relevant	Nonrelevant
10	Nonrelevant	relevant

Kappa Example

Number of docs	Judge 1	Judge 2
300	Relevant	Relevant
70	Nonrelevant	Nonrelevant
20	Relevant	Nonrelevant
10	Nonrelevant	relevant

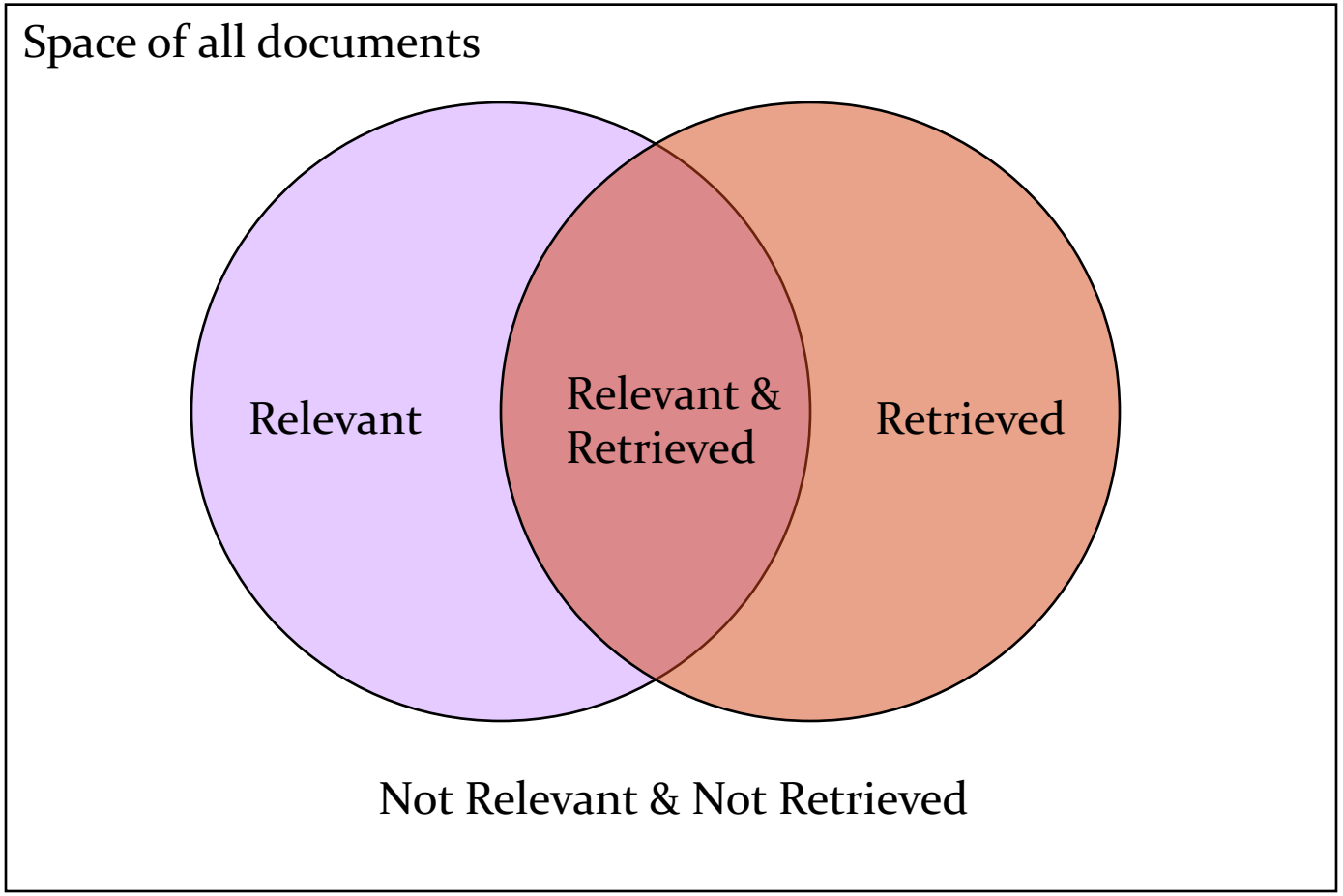
- P(A) = 370/400 = 0.925
- P(nonrelevant) = (10+20+70+70)/800 = 0.2125
- P(relevant) = (10+20+300+300)/800 = 0.7878
- P(E) = 0.2125^2 + 0.7878^2 = 0.665
- Kappa = (0.925 – 0.665)/(1-0.665) = 0.776

- Kappa > 0.8 = good agreement
- 0.67 < Kappa < 0.8 -> "tentative conclusions" (Carletta'96)
- Depends on purpose of study
- For >2 judges: average pairwise kappas

Evaluation Measures

- Good Effectiveness Measures
  - Should capture some aspect of what the user wants
    - That is, the measure should be meaningful
  - Should have predictive value for other situations
    - What happens with different queries on a different document collection?
  - Should be easily replicated by other researchers
  - Should be easily comparable
    - Optimally, expressed as a single number
- Many measures
  - View search results as a set: precision, recall, F measure, …
  - View search results as a ranked list: mean average precision, NDCG …

A Graphic View of Documents



Set-Based Basic Measures

	Relevant	Not relevant
Retrieved	A	B
Not retrieved	C	D

Collection size = A+B+C+D
Relevant = A+C
Retrieved = A+B

- **Precision** = A / (A+B)
- **Recall** = A / (A+C)
- **Miss** = C / (A+C)
- **False alarm** = B / (B+D)
- A: true positive, B: false positive, C: false negative, D: true negative

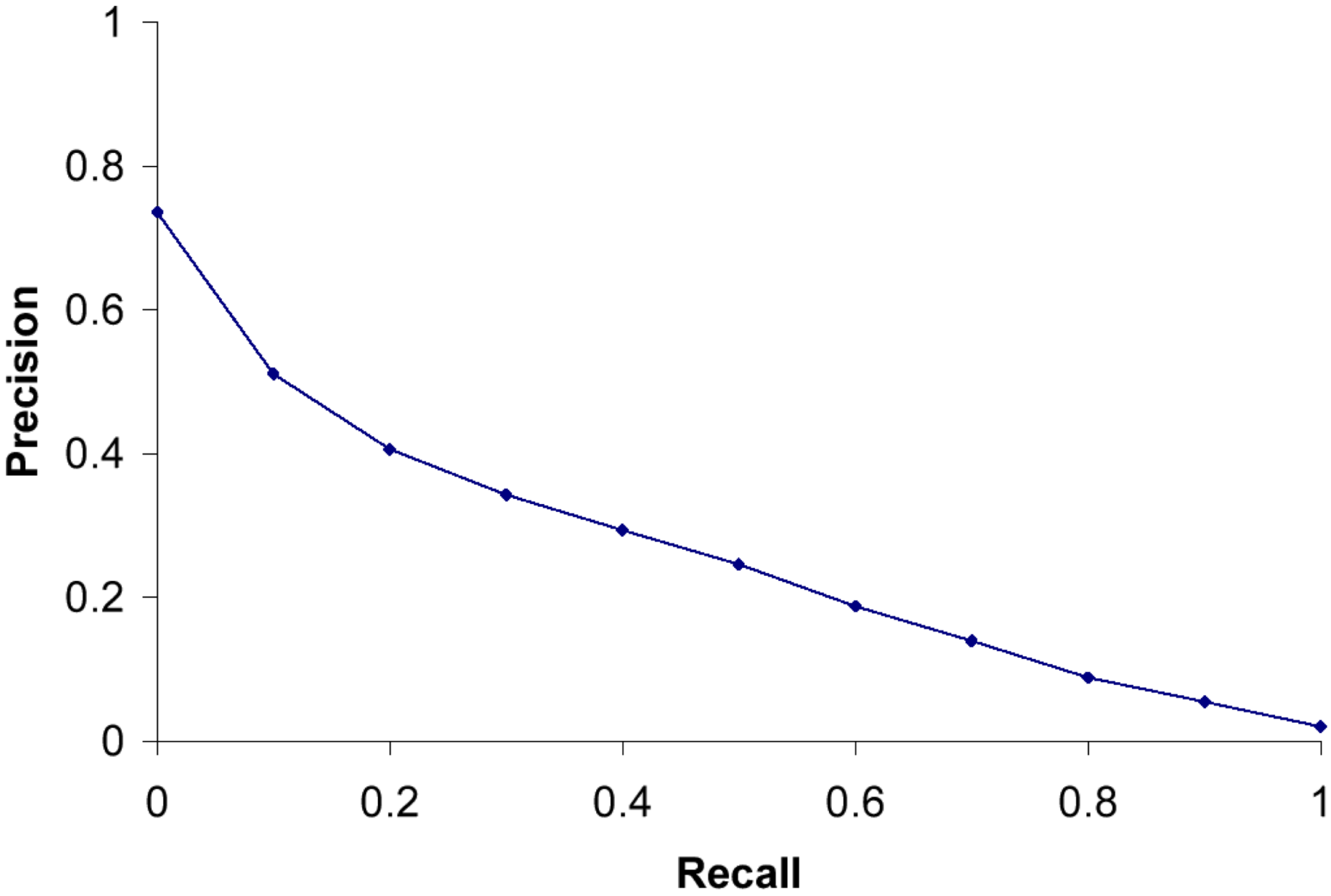
Precision and Recall

- Precision and Recall are two measures that should be viewed together
  - Precision is the bullseye or the needle in the haystack – you only try to get highly relevant hits (understanding that you may miss other relevant documents.)
  - Recall is the kitchen sink – you try to get all the relevant documents possible (understanding that you may get many non-relevant documents as well.)
- Problematic if consider only one of them
  - A perfect precision retrieval system
  - A prefect recall retrieval system

Some Exercise on Precision and Recall

- A collection has 1000 documents, and there are 200 relevant documents to topic X. a search on topic X returns 100 documents, among which 80 are relevant. What is precision and what is recall? What is F1?

- A search engine for a given search topic returns 20 relevant documents and 30 nonrelevant documents, after examining the collection, the missed relevant not being returned are 40 documents, so what is the precision and recall of this search?



F-Measure

- Harmonic mean of recall and precision
- Beta controls relative importance of precision and recall
  - Beta = 1, precision and recall equally important, called F1
  - Beta > 1, recall is more important than precision
  - Beta < 1, precision is more important than recall

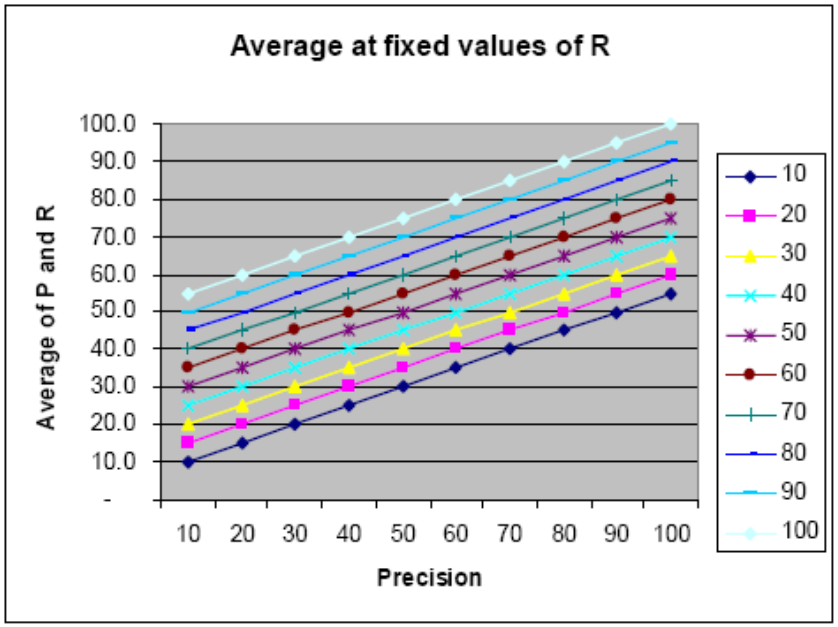
$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R} \quad \text{where} \quad \beta^2 = \frac{1-\alpha}{\alpha}$$

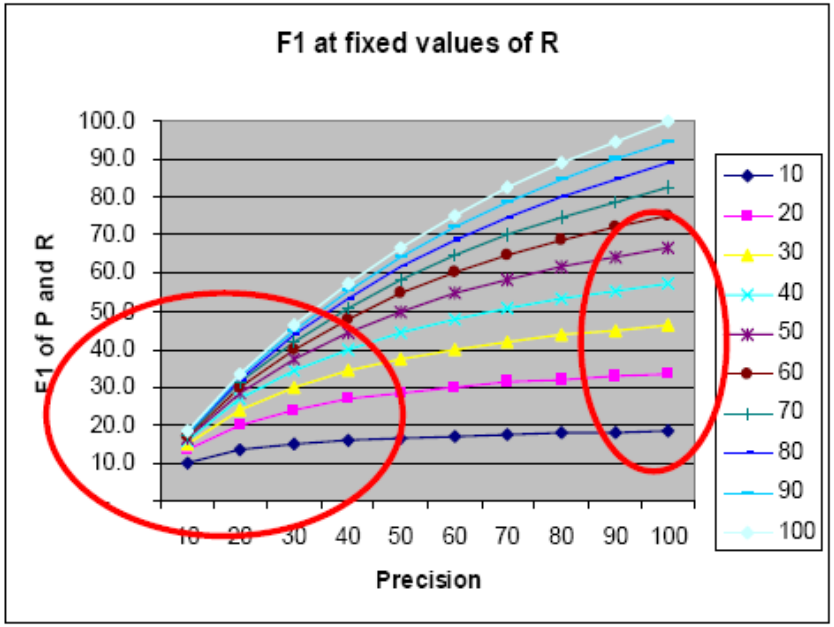
F measure as Harmonic Mean

- Harmonic mean of P and R
  - Inverse of average of their inverses

$$F_1 = \frac{2PR}{P+R} = \frac{1}{\frac{1}{2}(\frac{1}{R}+\frac{1}{P})}$$

- Heavily penalizes low values of P or R
  - Compared to standard average (P+R)/2

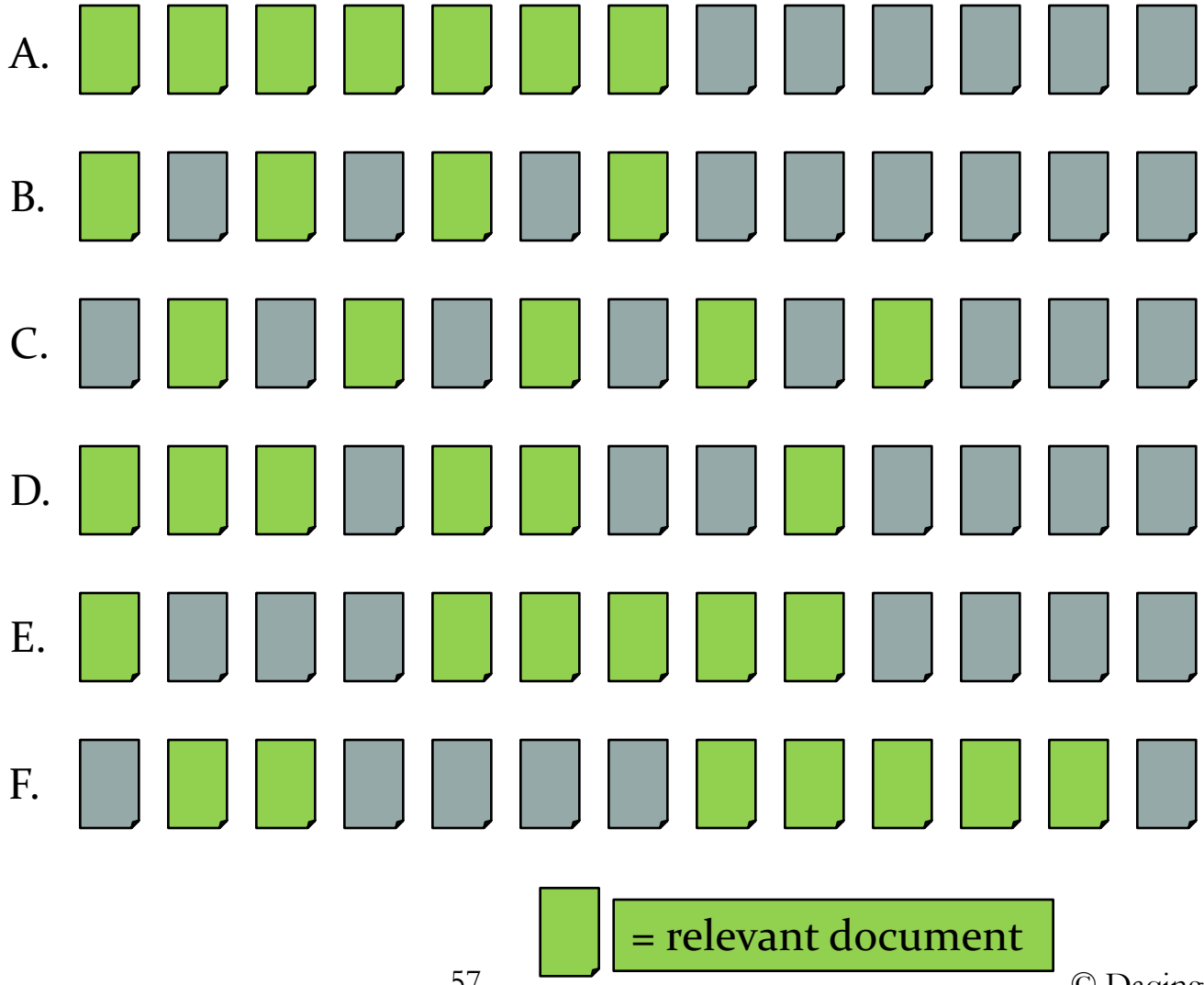




Measures for Ranked Retrieval

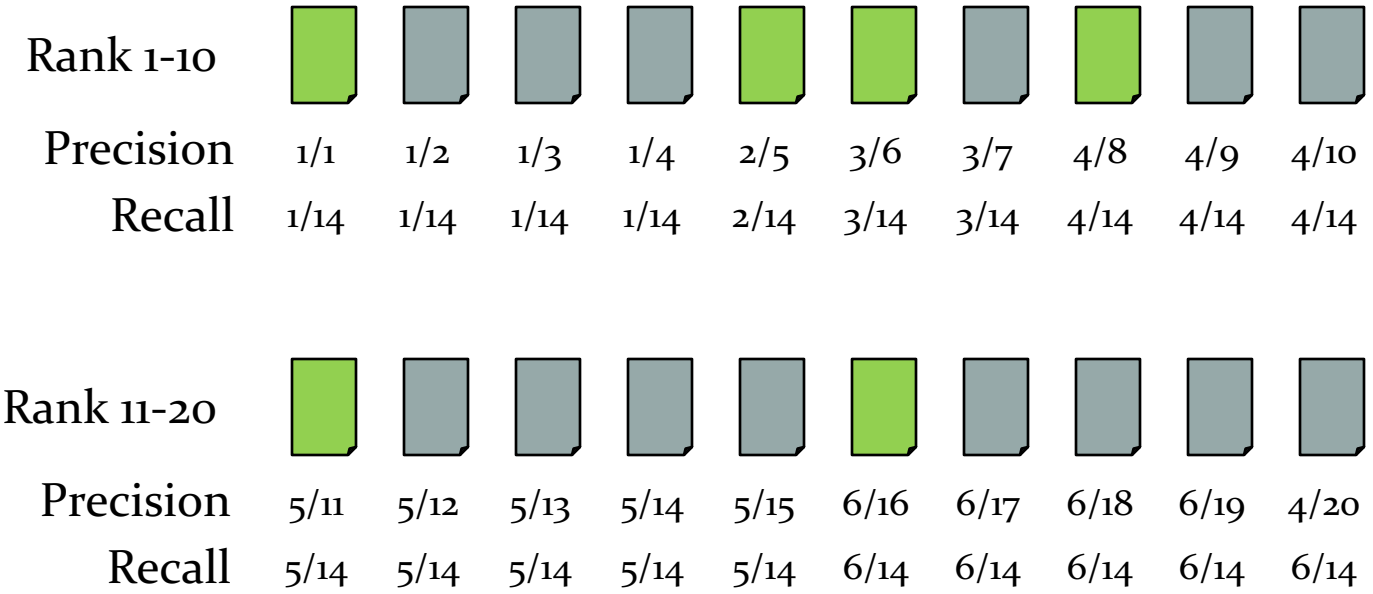
- Returned documents in relevant or not difference, but at the same time the ranking of each relevant document is also important
  - Instead of viewing the returned documents as a set
  - In theory, all documents in the collection are ranked
- Precision and Recall calculations
  - Compute the precision and recall value for each relevant document
  - Compute the precision at fixed recall points (e,g. P at 40% recall)
  - Compute the precision at fixed rank points (e.g., P at rank 20)

Which is the Best Rank Order?



Measuring Precision and Recall

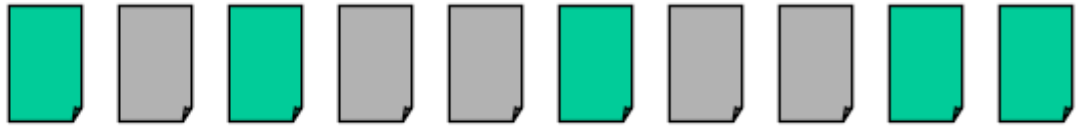
Assume there are a total of 14 relevant docs in a collection of 20 docs



Comparing two Ranked Lists

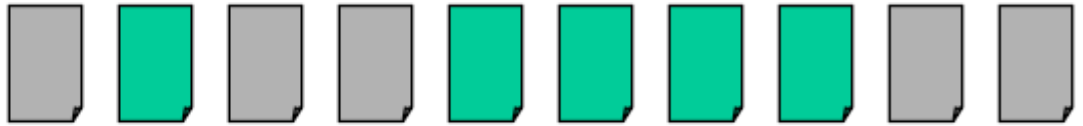
= All relevant documents

Ranking #1



Recall	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	1.0
Precis.	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

Ranking #2

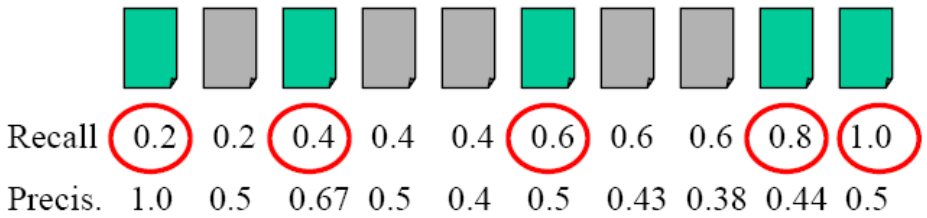


Recall	0.0	0.2	0.2	0.2	0.4	0.6	0.8	1.0	1.0	1.0
Precis.	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.63	0.55	0.5

Average Precision

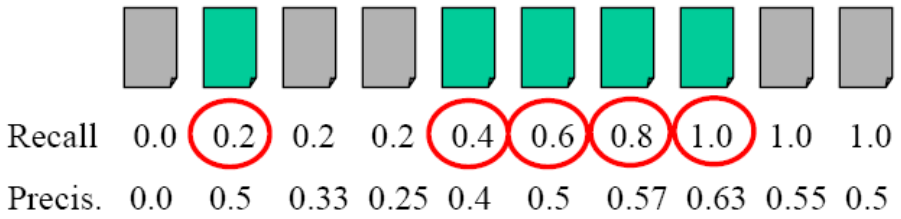
- Often want a single value to indicate the effectiveness in ranked retrieval
- Commonly used measure is average precision
  - Average precisions when recall increases (when meet each relevant document)

Ranking #1



AvgPrec = (1+0.67+0.5+0.44+0.5)/5 = 0.622

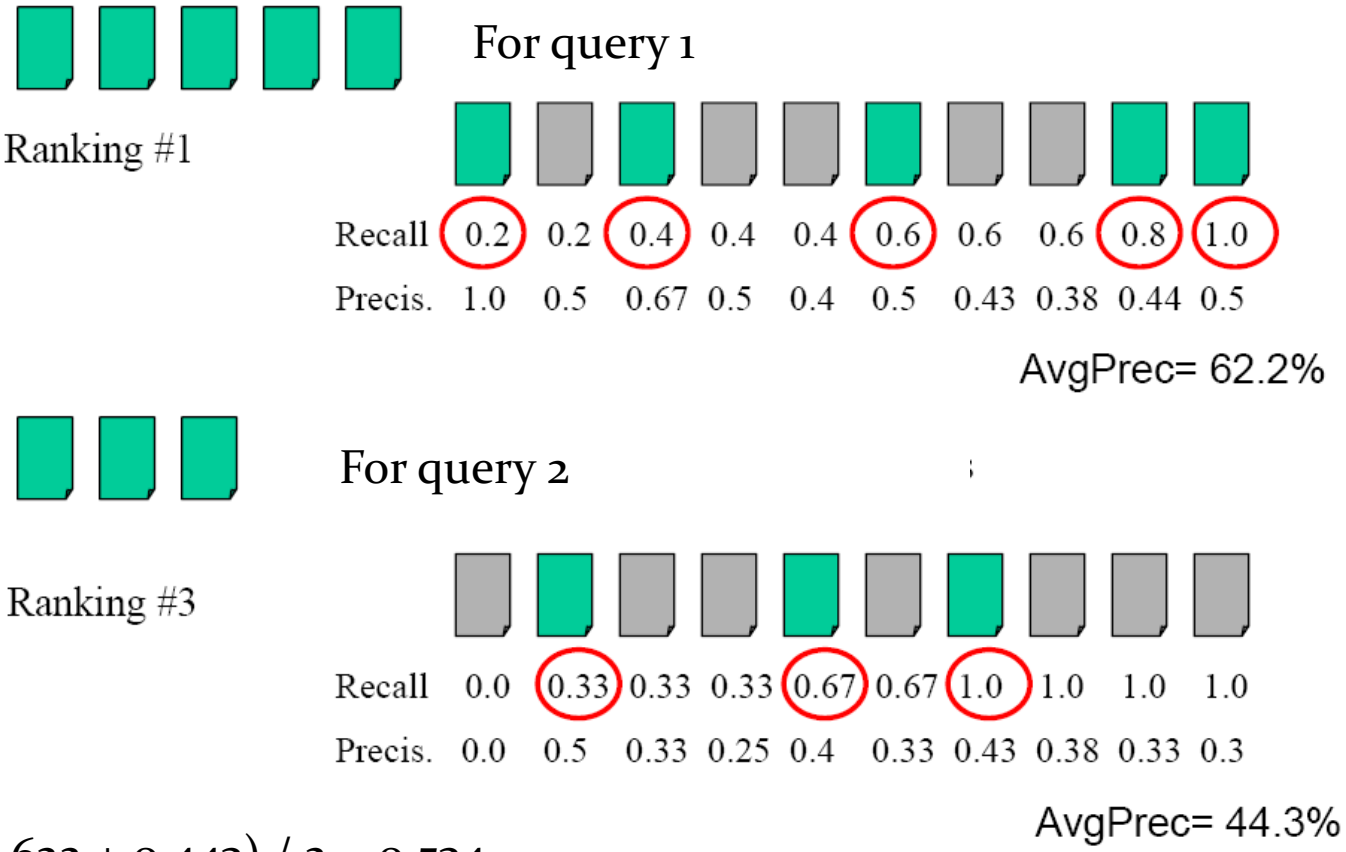
Ranking #2



AvgPrec = (0.5+0.4+0.5+0.57+0.63)/5 = 0.52

Mean Average Precision (MAP)

- Mean over the average precision of multiple queries.



MAP = (0.622 + 0.443) / 2 = 0.534

Mean Average Precision (MAP) - II

- The formula for MAP

$$\mathrm{MAP}(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j}\mathrm{Precision}(R_k)$$

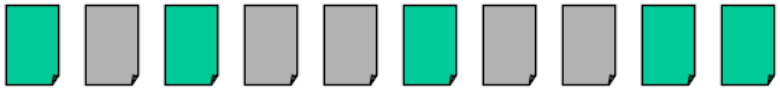
the set of relevant documents for an information need $q_j \in Q$ is $\{d_1, \ldots d_{m_j}\}$ and R_k is the set of ranked retrieval results from the top result until you get to document d_k, then

- Question: what happens if not all relevant documents are returned
  - Because average precision assumes that there is a precision value for every recall point
- Solutions:
  - Put the unretrieved documents at the bottom of the extension
  - Essentially, assume that the precision value for the unretreived docs is 0

Therefore

- If we know that there are totally 10 relevant documents, what is the average precision then?

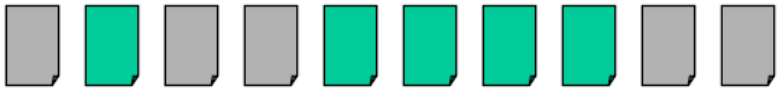
Ranking #1



AvgPrec = ?

Precis. 1.0 0.5 0.67 0.5 0.4 0.5 0.43 0.38 0.44 0.5

Ranking #2



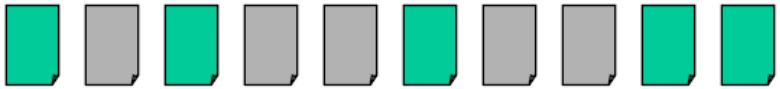
AvgPrec = ?

Precis. 0.0 0.5 0.33 0.25 0.4 0.5 0.57 0.63 0.55 0.5

Therefore

- If we know that there are totally 10 relevant documents, what is the average precision then?

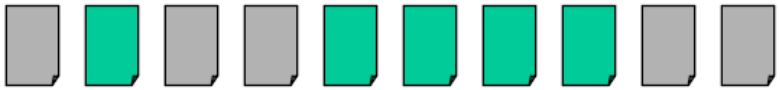
Ranking #1



Precis. 1.0 0.5 0.67 0.5 0.4 0.5 0.43 0.38 0.44 0.5

AvgPrec = (1+0.67+0.5+0.44+0.5)/10 = 3.11/10

Ranking #2



Precis. 0.0 0.5 0.33 0.25 0.4 0.5 0.57 0.63 0.55 0.5

AvgPrec = (0.5+0.4+0.5+0.57+0.63)/10 = 2.6/10

Measures for Ranked Retrieval

- Returned documents in relevant or not difference, but at the same time the ranking of each relevant document is also important
- Precision and Recall calculations
  - Compute the precision and recall value for each relevant document
  - Compute the precision at fixed recall points (e,g. P at 40% recall)
  - Compute the precision at fixed rank points (e.g., P at rank 20)
- Fdifferent queries, different relevant documents in the collection. Precision at fixed rank points cannot perfectly describe the search engine's retrieval ability.
  - e.g., for first query, 5 relevant documents in total; for second query, 10 relevant documents in total. P at rank 20 is unfair.

R- Precision

- Precision at the R-th position in the ranking of results for a query that we know has R relevant documents in the collection

n	doc #	relevant
1	588	x
2	589	x
3	576	
4	590	x
5	986	
6	592	x
7	984	
8	988	
9	578	
10	985	
11	103	
12	591	
13	772	x
14	990	

R = # of relevant docs = 6

R-Precision = 4/6 = 0.67

Mean Reciprocal Rank (MRR)

- MRR is the mean of Reciprocal Rank of a set of queries
  - Reciprocal Rank is the reciprocal of the first relevant document's rank in the ranked list
  - E.g.
    - The first relevant document is at rank 5
    - RR = 1/5 = 0.2
- Which type of searches is MRR a good measure?

Cumulative Gain

- Cumulative Gain (CG) is the sum of the graded relevance values of all results in a search result list. The CG at a particular rank position p is defined as:

$$\mathrm{CG_p} = \sum_{i=1}^{p} rel_i$$

where rel_i is the graded relevance of the result at position i.

- E.g. For ranked documents $D_1, D_2, D_3, D_4, D_5, D_6$ with relevance scores as 3,2,3,0,1,2. CG is

$$\mathrm{CG_p} = \sum_{i=1}^{p} rel_i = 3 + 2 + 3 + 0 + 1 + 2 = 11$$

Discounted Cumulative Gain

- CG is not really sensitive to the ranking
- But we know ranked list follow two assumptions:
  - Highly relevant documents are more useful when appearing earlier in a search engine result list (have higher ranks)
  - Highly relevant documents are more useful than marginally relevant documents, which are in turn more useful than irrelevant documents.
- Therefore, we would use Discounted Cumulative Gain (DCG)

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log_2(1+i)}$$

DCG example

- For ranked documents $D_1, D_2, D_3, D_4, D_5, D_6$ with relevance scores as 3,2,3,0,1,2.

i	Reli	$\frac{2^{rel_i}-1}{log_2(1+i)}$
1	3	(2^3-1)/log(1+1) =7
2	2	(2^2-1)/log(1+2)= 1.89
3	3	(2^3-1)/log(1+3) = 3.5
4	0	(2^0-1)/log(1+4)= 0
5	1	(2^1-1)/log(1+5) = 0.39
6	2	(2^2-1)/log(1+6) = 1.07

DCG_6 = 7+1.89 + 3.5 + 0 + 0.39+ 1.07 = 13.85

Normalized DCG

- DCG has problems in aggregating among different topics
- Better normalized DCG to a fixed range of values
- Ideal (Perfect) ranking: order documents to be the monotonically decreasing sort of the relevance judgment scores
  - E.g. for all documents to be ranked $D_1, D_2, D_3, D_4, D_5, D_6$ with relevance scores as 3,2,3,0,1,2, and ideal ranking would be 3,3,2,2,1,0
    - i.e., one idea ranking is D1,D3,D2,D6,D5,D4, but there could be more idea ranking
- For an ideal ranking, we can have DCG too, it is called IDCG
  - $IDCG_6 =$

IDCG

- For a topic with all relevant documents D_1, D_3, D_2, D_6, D_5 with relevance scores as 3,3,2,2,1

i	Reli	$\frac{2^{rel_i}-1}{log_2(1+i)}$
1	3	$(2^3-1)/\log(1+1) = 7$
2	3	$(2^3-1)/\log(1+2) = 4.42$
3	2	$(2^2-1)/\log(1+3) = 1.5$
4	2	$(2^2-1)/\log(1+4) = 1.29$
5	1	$(2^1-1)/\log(1+5) = 0.39$

$IDCG_6$ = = (7+ 4.42+ 1.5+ 1.29+ 0.39) = 14.6

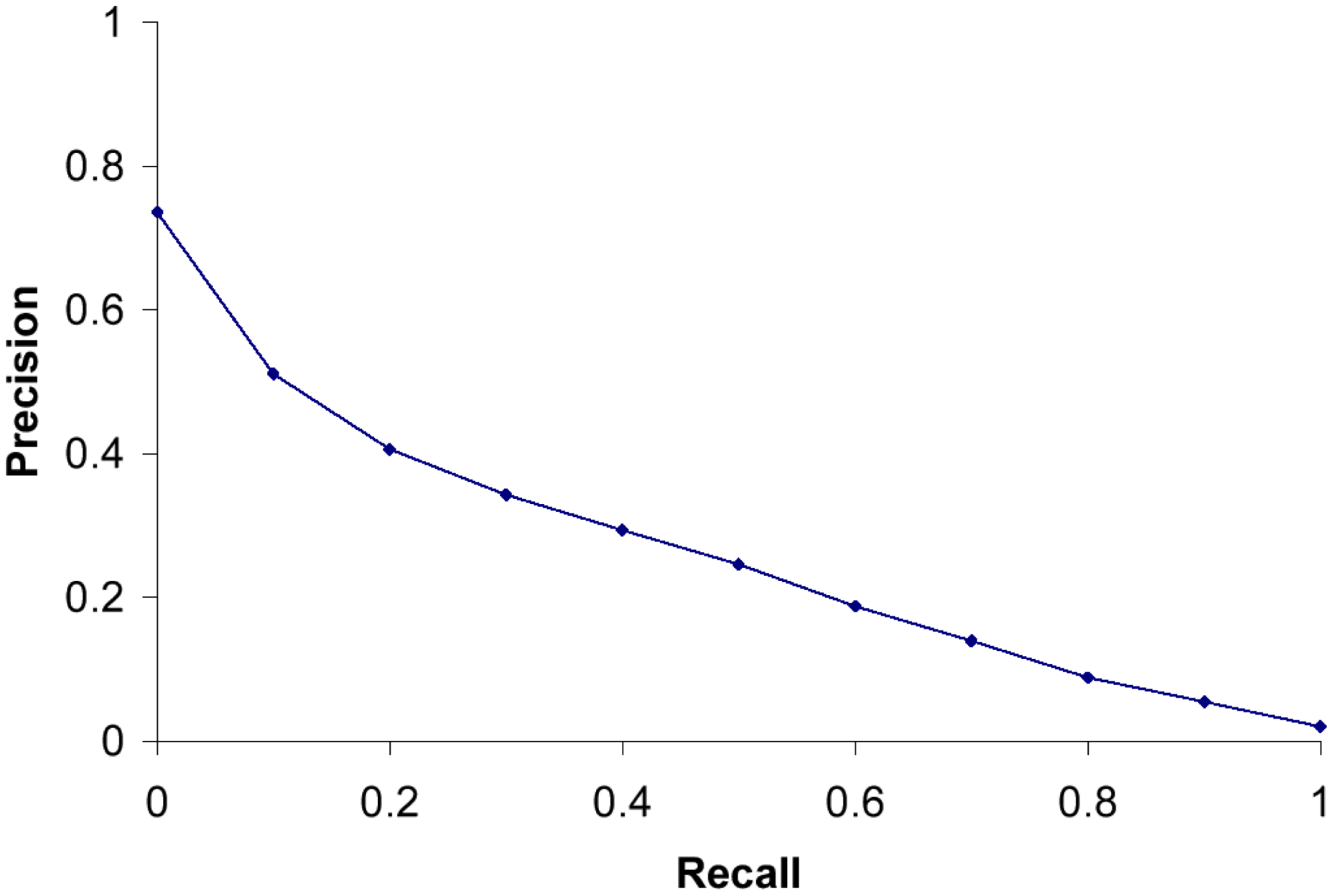
NDCG

- Normalized DCG therefore is

$$\mathrm{nDCG_p} = \frac{DCG_p}{IDCGp}$$

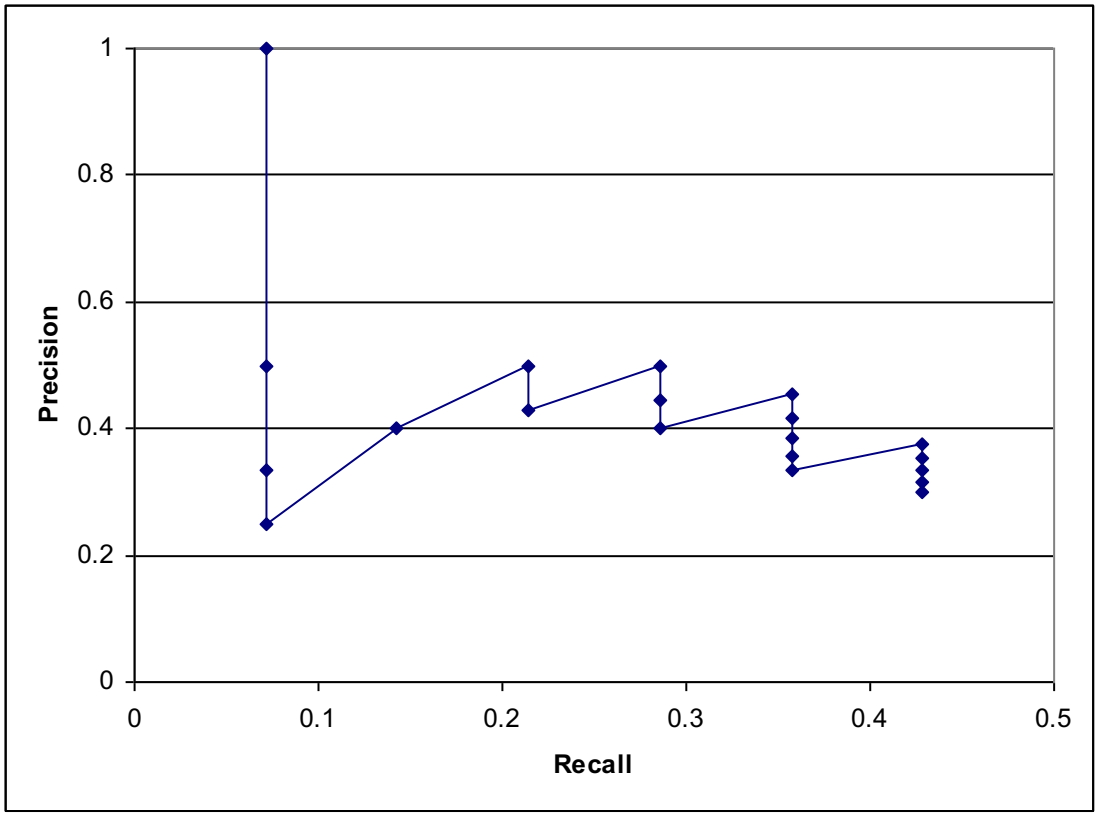
- IIR use the following formula where Z_k is the normalization factor

$$\mathrm{NDCG}(Q, k) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} Z_k \sum_{m=1}^{k} \frac{2^{R(j,m)} - 1}{\log(1+m)},$$



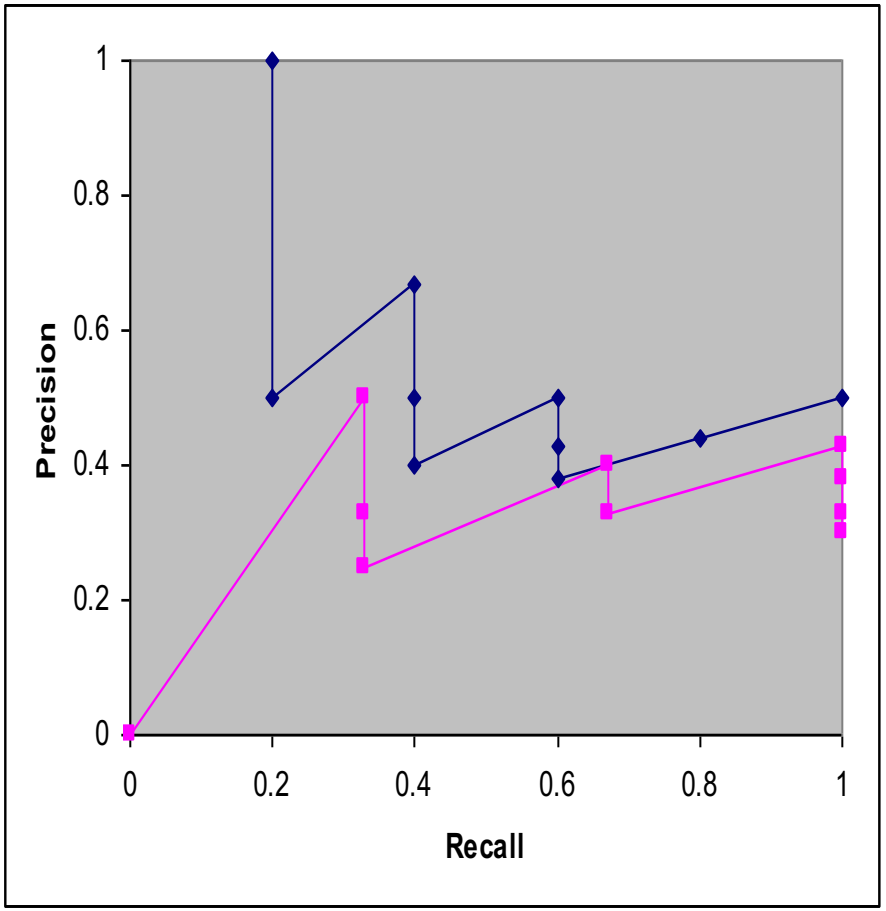
Precision and Recall Graph

- Average precision only gives a value, good summary, but
  - Sometimes, want to see the precision/recall tradeoff
  - Plot each (recall, precision) point on a graph

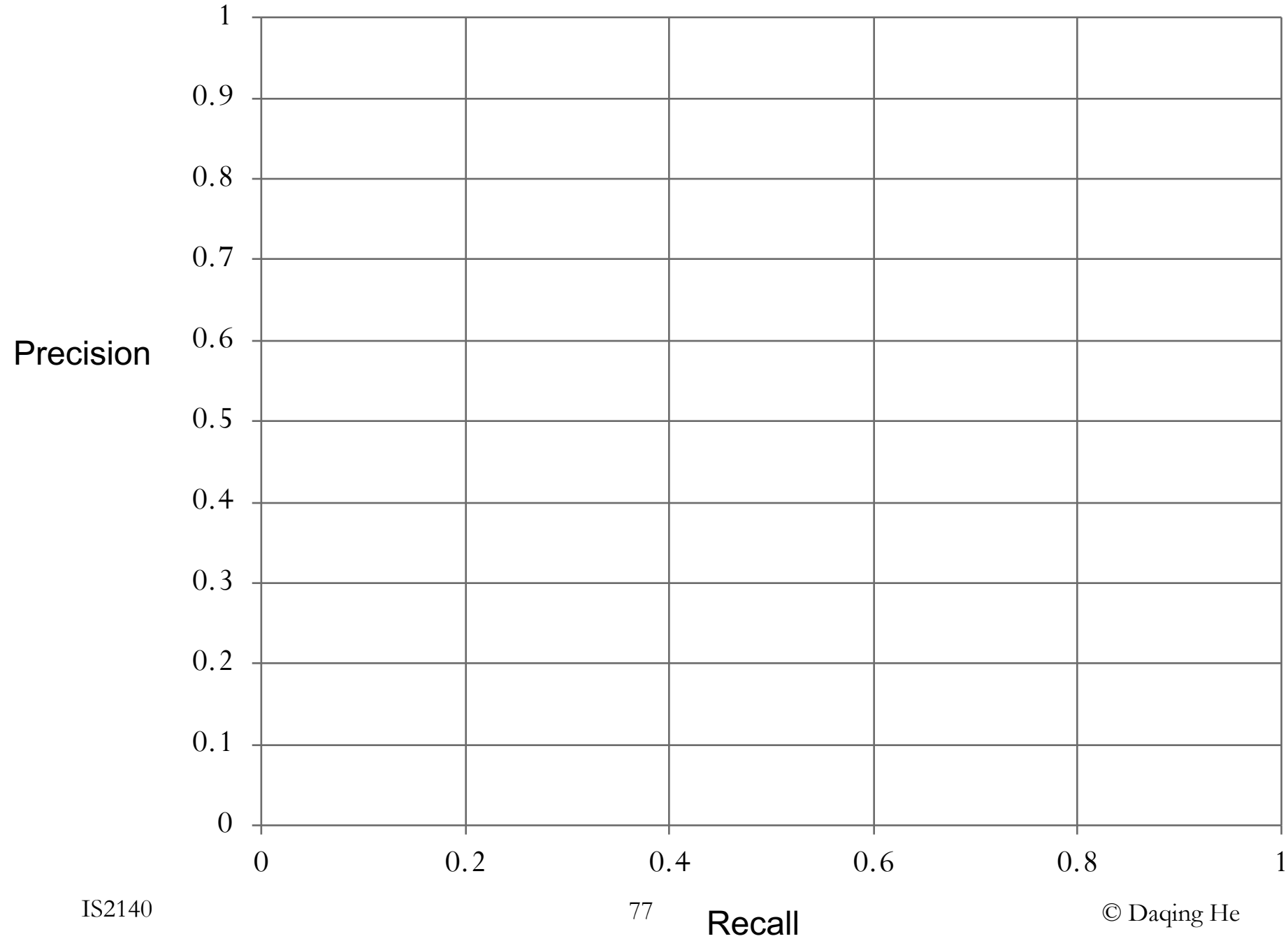


Averaging over Graph

- Recall/Precision graph has the odd sawtooth shape when plot directly
  - Why?
- Problems
  - How do you compare performance across queries?
  - How to obtain precision value when recall is at, say, 30%
  - Can the sawtooth shape describe what's going on with the retrieval result?

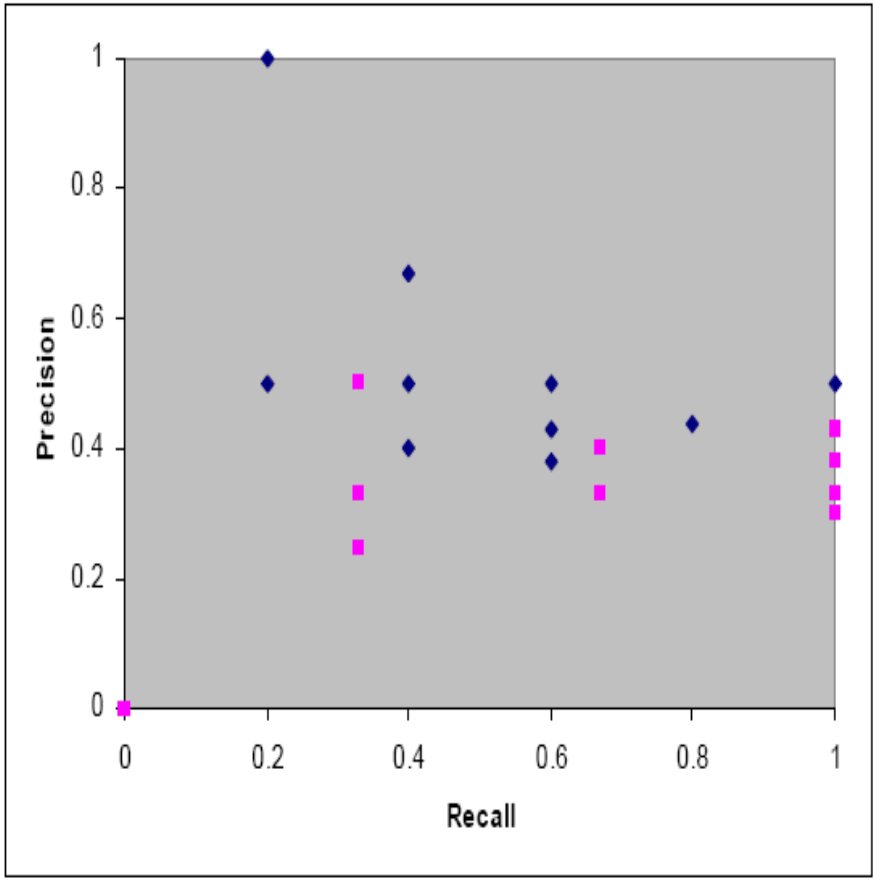


Solution: Interpolation!, but how



Averaging over Graph

- Recall/Precision graph has the odd sawtooth shape when plot directly
  - Why?
- Problems
  - How do you compare performance across queries?
  - How to obtain precision value when recall is at, say, 30%
  - Can the sawtooth shape describe what's going on with the retrieval result?



Solution: Interpolation!, but how

Interpolation

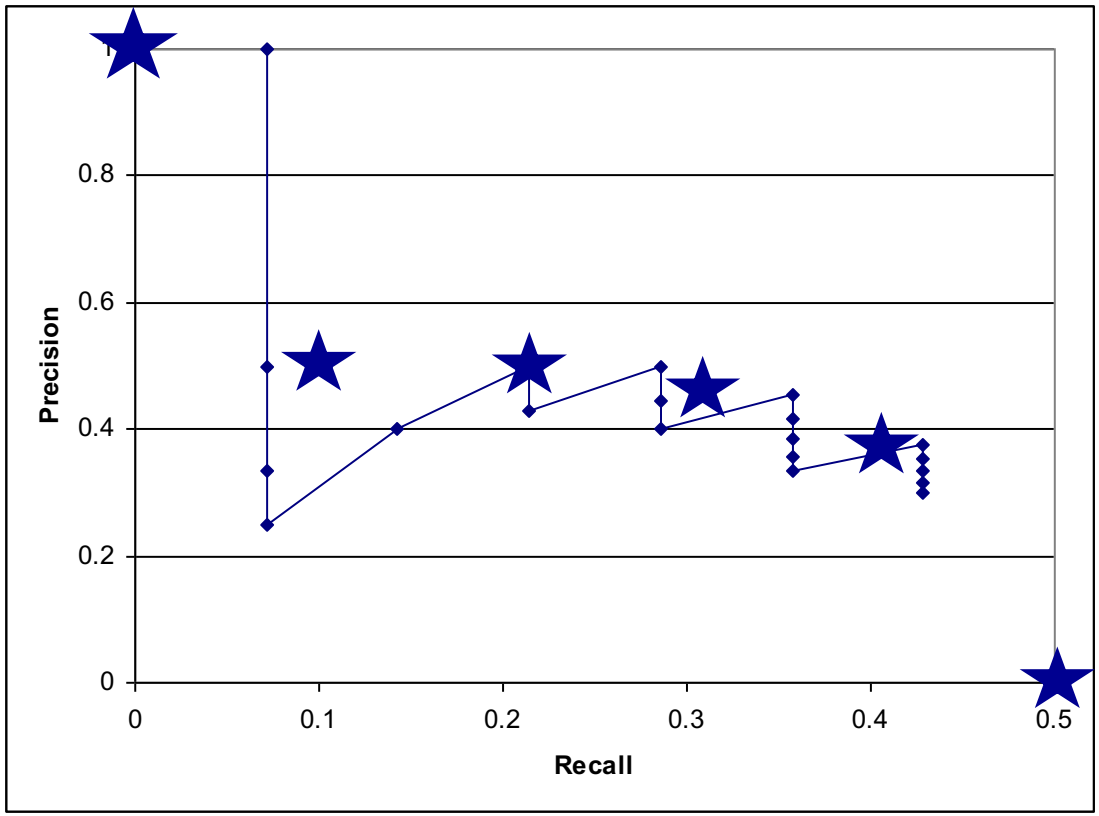
- It is an empirical fact that *on average* as recall increases, precision decreases
  - Verified time and time again
  - *On average*
- Seems reasonable to aim for an interpolation that makes function monotonically decreasing
- One approach:

$$P(R) = \max\{P' \ : \ R' \geq R \wedge (R', P') \in S\}$$

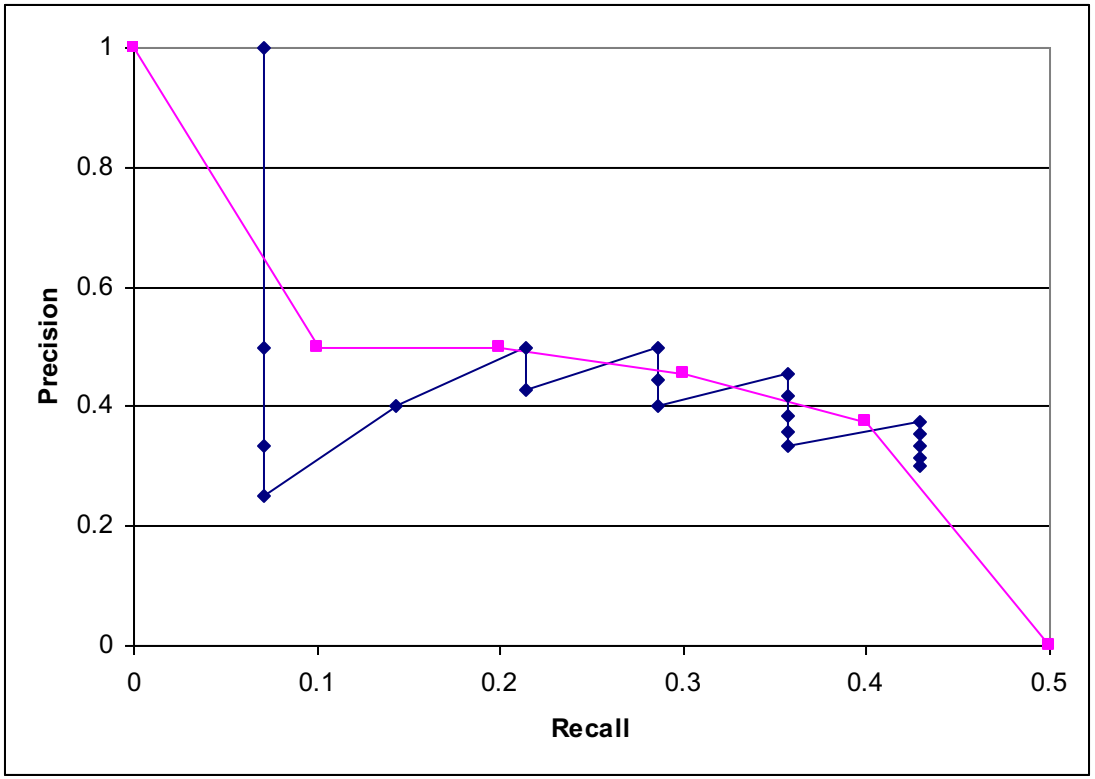
where S is the set of observed (R,P) points

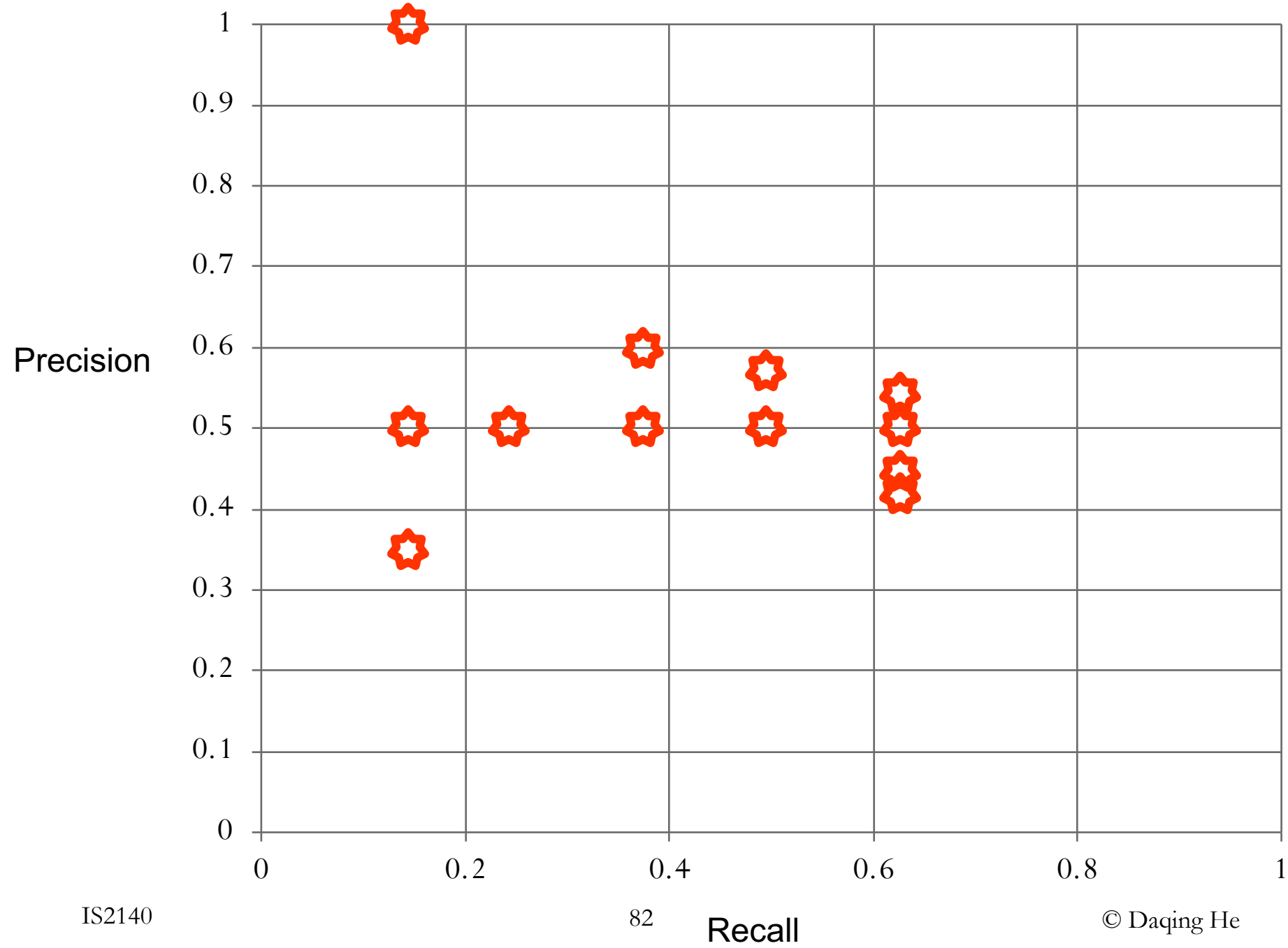
Interpolation

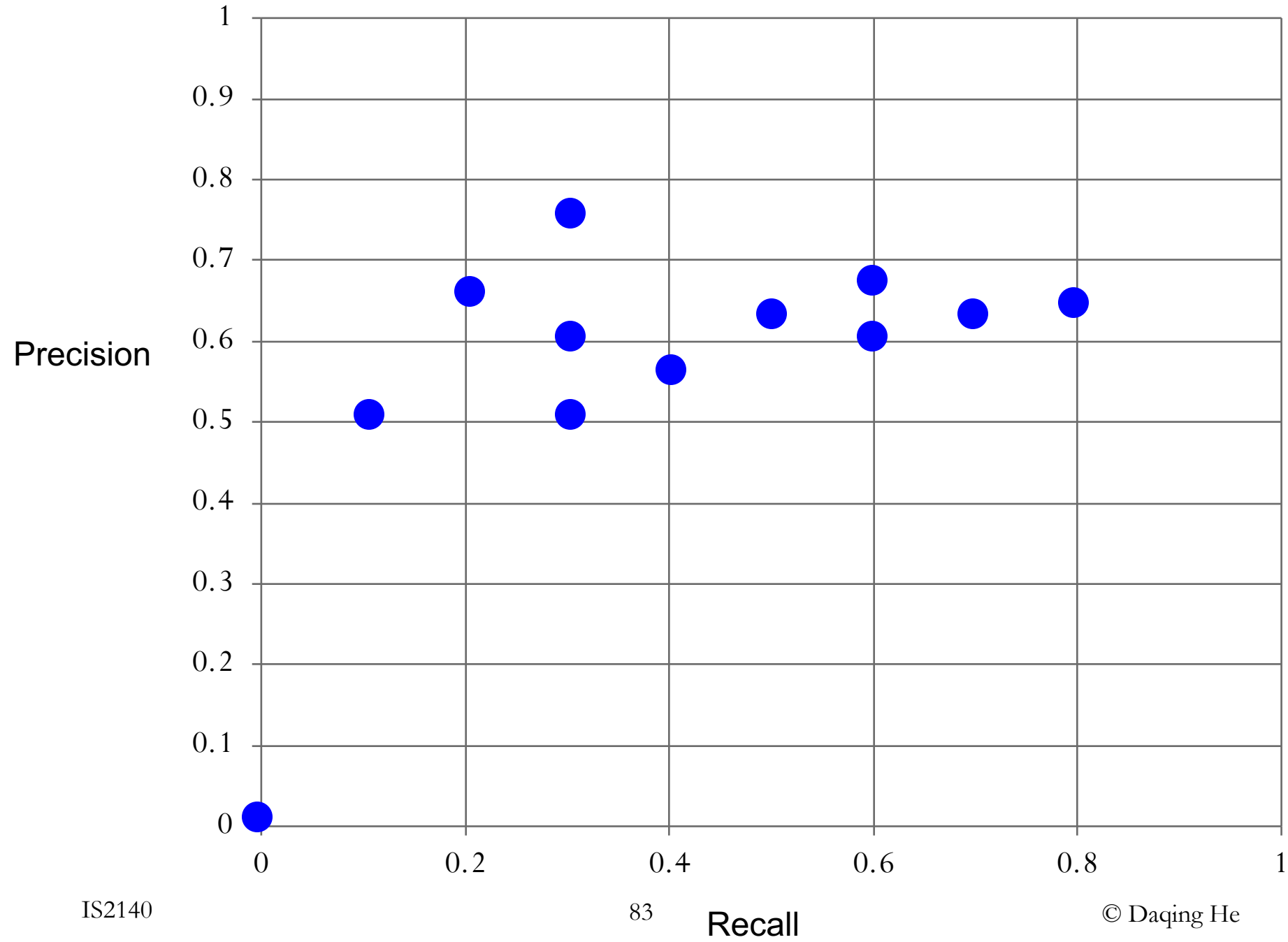
- How to perform interpolation?



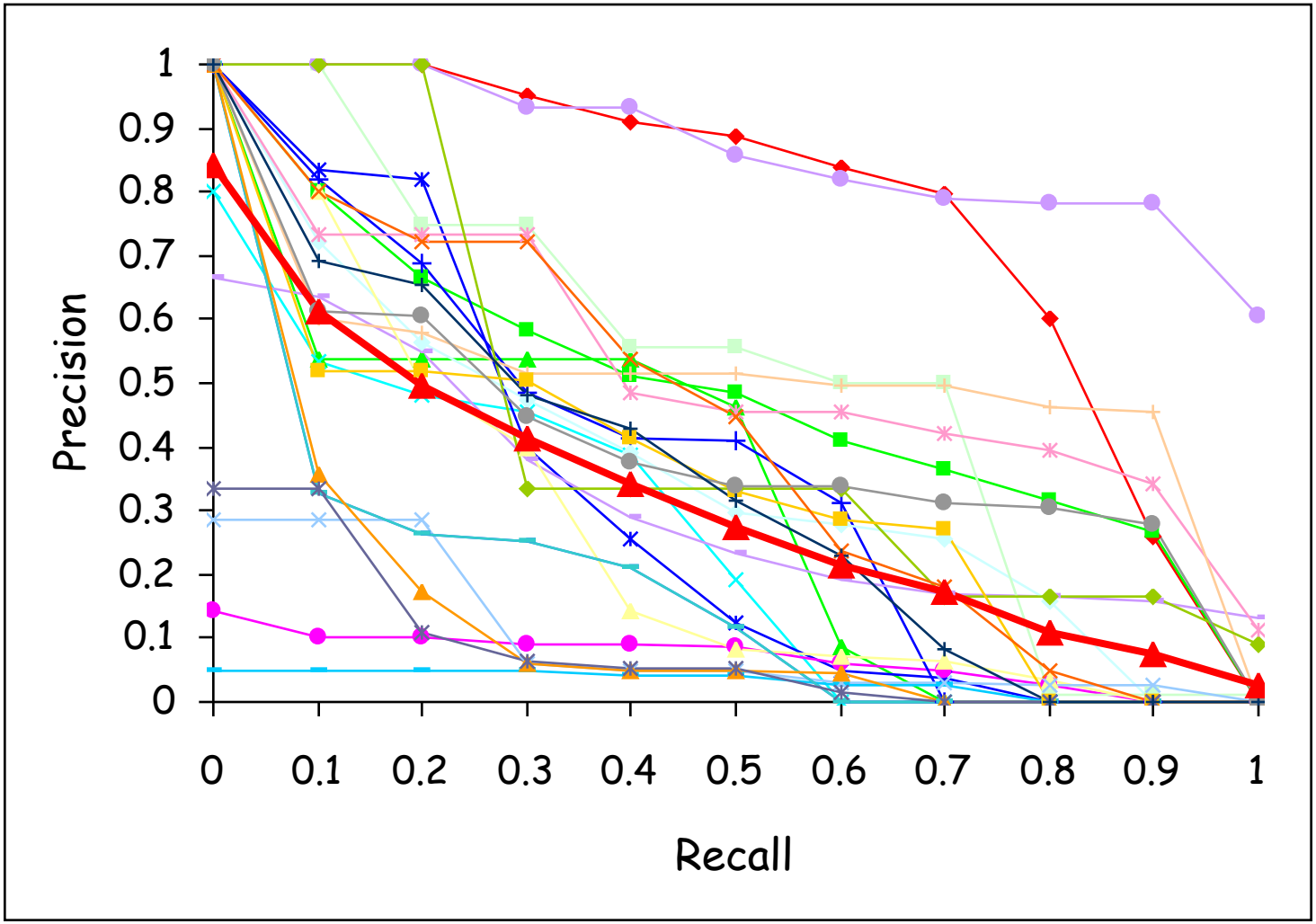
Result of Interpolation







What to do with the curves?



Adapted from a presentation by Ellen Voorhees at the University of Maryland, March 29, 1999

trec_eval

- trec_eval is the standard evaluation software for ad-hoc retrieval
  - Written by Chris Buckley while at Cornell
  - http://trec.nist.gov/trec_eval/
- What trec_eval reports:
  - Retrieved: Number of documents retrieved by your program
  - Relevant: The number of relevant documents in the database
  - Rel_ret: The number of relevant documents your program found
  - Interpolated Recall at 11 Recall points
  - Average precision over all relevant documents
  - Precision at ranks 5, 10, 15, 20, 30, 100, 200, 500, and 1000
  - R-Precision: Precision at rank r, where r is the number of relevant
  - And others
- Statistics provided on a by-query or by-query-set basis

Why significance tests?

- System A and B identical on all but one query:
  - Is it just a lucky query for System A?
  - Need A to beat B frequently to believe it is really better
  - Need as many queries as possible

    Empirical research suggests 25 is minimum needed
    TREC tracks generally aim for at least 50 queries

- System A beats system B on every query:
  - But only does so by 0.00001%
  - Does that mean much?
- Significance tests consider those issues

Averages Can Deceive

Experiment 1

Query	System A	System B
1	0.20	0.40
2	0.21	0.31
3	0.22	0.42
4	0.19	0.25
5	0.17	0.27
6	0.20	0.30
7	0.21	0.22
Average	0.20	0.31

Experiment 2

Query	System A	System B
1	0.20	0.20
2	0.21	0.21
3	0.22	0.22
4	0.19	0.96
5	0.17	0.17
6	0.20	0.20
7	0.21	0.21
Average	0.20	0.31

How Much is Enough?

- Measuring improvement
  - Achieve a meaningful improvement

    Guideline: 0.05 is noticeable, 0.1 makes a difference (in MAP)

  - Achieve reliable improvement on "typical" queries

    Wilcoxon signed rank test for paired samples
- Sign test or Wilcoxon signed-ranks test are typical
  - Do not require that data be normally distributed
  - Sign test answers how often
  - Wilcoxon answers how much
  - Sign test is crudest but most convincing

Evaluating Ranked Retrieval

Test collection: topics, documents, relevance judgments

Topic T1 for System A

Rank	Doc#	Score	Rel?	Prec.
1	FR05	0.97	R	1.00
2	FR03	0.91	R	1.00
3	FR02	0.88		
4	FR10	0.82		
5	FR07	0.80	R	0.60
6	FR04	0.77		
7	FR06	0.63	R	0.57
8	FR08	0.62		
9	FR09	0.55		
10	FR01	0.51	R	0.50
Avg. Prec. (AP):				0.73

System	A	B	Wilcoxon Test	
Topic	AP	AP	A-B	Signed Rank
T1	0.73	0.50	+0.23	+9
T2	0.45	0.38	+0.07	+3.5
T3	0.56	0.36	+0.20	+8
T4	0.00	0.09	-0.09	-5
T5	0.13	0.10	+0.03	+1
T6	1.00	0.83	+0.17	+7
T7	0.24	0.28	-0.04	-2
T8	0.47	0.20	+0.27	+10
T9	0.53	0.41	+0.12	+6
T10	0.23	0.30	-0.07	-3.5
MAP:	0.43	0.35		W+ = 44.5 W- = 10.5

If min(W+, W-) < 8 ➔ difference is not significant (two-tailed, p=0.05)

Assignment 3 is Out

- See details in Courseweb