Chapter III.2: Basic ranking & evaluation measures

1. TF-IDF and vector space model
    1.1. Term frequency counting with TF-IDF
    1.2. Documents and queries as vectors
2. Evaluating IR results
    2.1. Evaluation methods
    2.2. Unranked evaluation measures

Based on **Manning/Raghavan/Schütze,** Chapters 6 and 8

TF-IDF and vector-space model

- In Boolean case we considered each document as a set of words
  - A query matches all documents where terms appear at least once
- But if a term appears more than once is it not more important?
- Instead of Boolean queries, use free text queries
  - E.g. normal Google queries
  - Query seen as a set of words
  - Document is a better match if it has the query words more often

TF and IDF

- **Term frequency** of term t in document d, $\mathrm{tf}_{t,d}$, is just the number of times t appears in d
  - Naïve scoring: score of document d for query q is the sum of $\mathrm{tf}_{t,d}$s for terms t in query q
  - But some terms appear overall more often than others
- **Document frequency** of term t, df_t, is the number of documents in which t appears
- **Inverse document frequency** of term t, idft, is

$$\mathrm{idf_t} = \log \frac{\mathrm{N}}{\mathrm{df_t}}$$

where N is the total number of documents

TF-IDF

- **The TF-IDF** of term t in document d is

  $$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t$$

- $\text{tf-idf}_{t,d}$ is high when t occurs often in d but rarely in other documents
- $\text{tf-idf}_{t,d}$ is smaller if either
  - t occurs fewer times in d or
  - t occurs more often in other documents
- Slightly less naïve scoring

$$\text{Score}(q, d) = \sum_{t \in q} \text{tf-idf}_{t,d}$$

Variations to TF-IDF

- *Sublinear tf scaling* addresses the problem that 20 occurrences of a word is probably not 20 times more important than 1 occurrence

$$\mathrm{wf}_{t,d} = \begin{cases} 1 + \log \mathrm{tf}_{t,d} & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$$

- *Maximum tf normalization* tries to overcome the problem that longer documents yield higher tf-idf scores

$$\mathrm{ntf}_{t,d} = a + (1 - a)\frac{\mathrm{tf}_{t,d}}{\mathrm{tf}_{\max}(d)}$$

  - $\mathrm{tf}_{\max}(d)$ is the largest term frequency in document d
  - a is a smoothing parameter, $0 < a < 1$ (typically $a = 0.4$)

Documents as vectors

- Documents can be represented as M-dimensional vectors
  - M is the number of term in vocabulary
  - Vector values are the tf-idf (or similar) values
  - Does not store the order of the terms in documents
- Document collection can be represented as M-by-N matrix
  - Each document is a column vector
- Queries can also be considered as vectors
  - Each query is just a short document

The vector space model

- The similarity between two vectors can be computed using **cosine similarity**

$$\mathrm{sim}(\mathrm{d}_1, \mathrm{d}_2) = \langle \boldsymbol{v}(\mathrm{d}_1), \boldsymbol{v}(\mathrm{d}_2) \rangle$$

  - $\boldsymbol{v}(d)$ is the normalized vector representation of document d
  - A normalized version of vector $\boldsymbol{v}$ is $\boldsymbol{v}/\|\boldsymbol{v}\|$
  - Thus cosine similarity is equivalently

$$\mathrm{sim}(\mathrm{d}_1, \mathrm{d}_2) = \frac{\langle \mathbf{V}(\mathrm{d}_1), \mathbf{V}(\mathrm{d}_2) \rangle}{\|\mathbf{V}(\mathrm{d}_1)\| \, \|\mathbf{V}(\mathrm{d}_2)\|}$$

  - Cosine similarity is the cosine of the angle between $\boldsymbol{v}(d_1)$ and $\boldsymbol{v}(d_2)$

Finding the best documents

Using cosine similarity and vector space model, we can obtain the best document d^* with respect to query q from document collection D as

$$d^* = \arg\max_{d \in D} \langle \boldsymbol{v}(q), \boldsymbol{v}(d) \rangle.$$

This can be easily extended to top-k documents.

Evaluating IR results

- We need a way to evaluate different IR systems
  - What pre-processing should I do?
  - Should I use tf-idf or wf-idf or ntf-idf?
  - Is cosine similarity good similarity measure, or should I use something else?
  - etc.
- For this we need evaluation data and evaluation metrics

IR evaluation data

- Document collections with documents labeled *relevant* or *irrelevant* for different information needs
  - Information need is not a query; it is turned into a query
    - E.g. ”What plays of Shakespeare have characters Caesar and Brutus, but not character Calpurnia?”
- For tuning parameters, document collections are divided to *development* (or *training*) and *test sets*
- Some real-world data sets exist that are commonly used to evaluate IR methods

Classifying the results

- The retrieved documents can be either relevant or irrelevant and same for not retrieved documents
  - We would like to retrieve relevant documents and not retrieve irrelevant ones
  - We can classify all documents into four classes

	relevant	irrelevant
retrieved	true positives (tp)	false positives (fp)
not retrieved	false negatives (fn)	true negatives (tn)

Unranked evaluation measures

Precision, P, is the fraction of retrieved documents that are relevant

$$P = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp}}$$

Recall, R, is the fraction of relevant documents that are retrieved

$$R = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fn}}$$

Accuracy, acc, is the fraction of correctly classified documents

$$acc = \frac{\mathrm{tp} + \mathrm{tn}}{\mathrm{tp} + \mathrm{fp} + \mathrm{tn} + \mathrm{fn}}$$

Unranked evaluation measures

Precision, *P*, is the fraction of retrieved documents that are relevant

$$P = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp}}$$

Recall, *R*, is the fraction of relevant documents that are retrieved

$$R = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fn}}$$

Accuracy, *acc*, is the fraction of correctly classified documents

$$acc = \frac{\mathrm{tp} + \mathrm{tn}}{\mathrm{tp} + \mathrm{fp} + \mathrm{tn} + \mathrm{fn}}$$

Not appropriate for IR problems

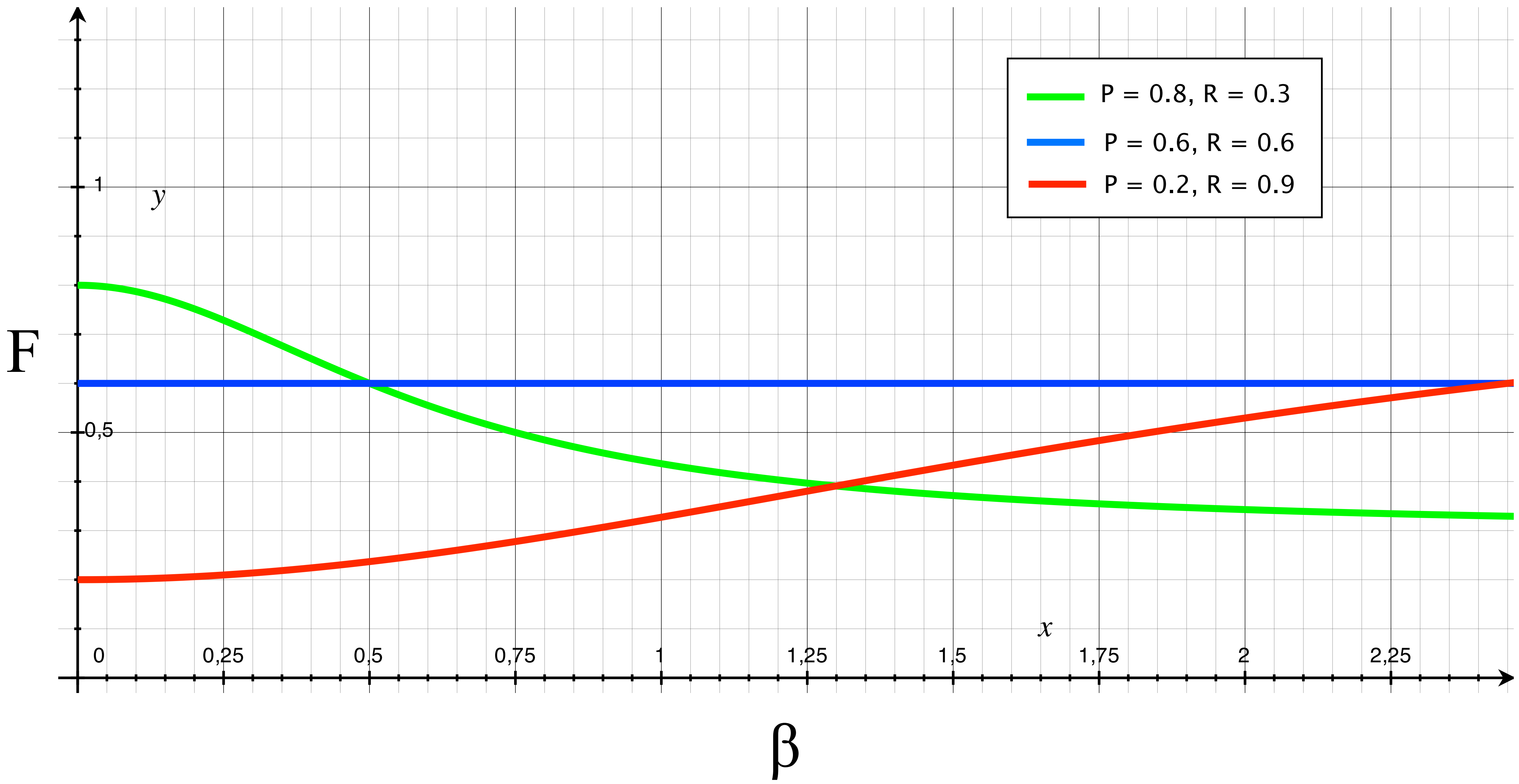
The F measure

- Different tasks may emphasize precision or recall
  - Web search, library search, ...
- But usually some type of balance is sought
  - Maximizing either one is usually easy if other can be arbitrarily low
- The **F measure** is a trade-off between precision and recall

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

  - The β is a trade-off parameter: $\beta = 1$ is *balanced F*, $\beta < 1$ emphasize precision, and $\beta > 1$ emphasize recall

F measure example



Ranked evaluation measures

- Precision as a function of retrieval $P(r)$
  - What is the precision after we have obtained 10% of relevant documents in ranked results?
  - *Interpolated precision* at recall level r is $\max_{r' \geq r} P(r')$
  - Precision–recall curves
- Precision at k ($P@k$)
  - The precision after we have obtained top-k documents (relevant or not)
    - Typically k=5, 10, 20
  - E.g. web search
- $F_\beta@k = ((\beta^2 + 1)P@k \times R@k)/(\beta^2 P@k + R@k)$

Mean Average Precision

- Precision, recall, and F measure are unordered measures
- **Mean average precision** (MAP) averages over different information needs and ranked results
  - Let $\{d_1, \ldots, d_{m_j}\}$ be the set of relevant documents for $q_j \in Q$
  - Let R_{jk} be the set of ranked retrieval results of q_j from top result until you get to document d_k
  - MAP is

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

Mean Average Precision

- Precision, recall, and F measure are unordered measures
- **Mean average precision** (MAP) averages over different information needs and ranked results
  - Let $\{d_1, \ldots, d_{m_j}\}$ be the set of relevant documents for $q_j \in Q$
  - Let R_{jk} be the set of ranked retrieval results of q_j from top result until you get to document d_k
  - MAP is

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

Average precision

Measures for weighted relevance

- Non-binary relevance
  - 0 = not relevant, 1 = slightly relevant, 2 = more relevant, ...
- **Discounted cumulative gain** (DCG) for information need *q*:
  - *R(q,d)* ∈ {0, 1, 2, ...} is the relevance of document *d* for query *q*

$$\mathrm{DCG}(q,k) = \sum_{m=1}^{k} \frac{2^{R(q,m)} - 1}{\log(1+m)}$$

- **Normalized discounted cumulative gain** (NDCG):

$$\mathrm{NDCG}(q,k) = \frac{\mathrm{DCG}(q,k)}{\mathrm{IDCG}(q,k)}$$

Ideal discounted cumulative gain (IDCG)

- Let rank levels be {0, 1, 2, 3}
- Order rankings in descending order
  $\boldsymbol{I}_q = (3, 3, ..., 3, 2, 2, ..., 2, 1, 1, ..., 1, 0, 0, ..., 0)$

$$\mathrm{IDCG}(q,k) = \sum_{m=1}^{k} \frac{2^{I_q(m)} - 1}{\log(1+m)}$$

  - IDCG(q, k) is the maximum value that DCG(q, k) can attain
  - Therefore, NDCG(q, k) ≤ 1