Lecture 10: Query Expansion & Relevance Feedback
Recap: Clustering & Cluster in IR

• Clustering
  • Goal: to group similar things together

• “Hard” clustering
  • Each instance (document) belongs to only one cluster
  • e.g., K-means

• “Soft” clustering
  • Each instance (document) can belong to different clusters
  • e.g., topic models such as PLSI and LDA
  • Important components of topic models (PLSI and LDA)
    • Topics (z) and topic representation, e.g., P(w|z)
    • Topic document association, e.g., P(d|z) or P(z|d)
Recap: Cluster-based Document Model

- Smoothing a document $D$’s MLE model by
  - The corpus model
  - The model of the cluster that $D$ belongs to

$$P(t | D) = \lambda_1 P_{MLE}(t | D) + \lambda_2 P(t | cluster) + (1 - \lambda_1 - \lambda_2) \cdot P(t | corpus)$$

- Issues of hard clustering
  - Each document belongs to one cluster
  - A binary weight: 1 or 0
  - Borderline documents get inaccurate smoothing
Recap: LDA-based Document Model

• Smoothing a document $D$’s MLE model by
  • The corpus model
  • The model of the topics that $D$ belongs to

  \[
P(w \mid D) = \lambda \left( \frac{N_d}{N_d + \mu} P_{ML}(w \mid D) + (1 - \frac{N_d}{N_d + \mu}) P_{ML}(w \mid coll) \right)
  + (1 - \lambda) P_{lda}(w \mid D) \tag{7}
\]

• Addressing a few issues of hard clustering
  • LDA can be considered as a soft clustering technique
  • Each document can belong to multiple topics with different weight
  • $P(\text{topic} \mid D)$ can be the weight of the topic in smoothing
  • Outperforms the cluster-based smoothing method
Outline Today

• *Query expansion: an overview*
• Relevance-based language model
• Alternatives of RM
Query Reformulation

• Users often need to use multiple rounds of searches (multiple queries) to satisfy an information need
  • Query reformulation: issue a new query (when there was an old one)

• Many possible reasons for reformulation
  • Limited search performance: to refine query
  • Can be the fault of the system, the user, or both
  • High complexity of search problem, e.g., relevant docs belong to multiple topics such that it is difficult to find all using a single query
  • Sometimes a search strategy: e.g., berry-picking (Bates, 1989)
  • User’s information need is evolving (Belkin, 1982)

• To support users’ need of query reformulation
  • Interactive techniques, e.g., offering suggestions & let users choose
  • Automatic techniques, e.g., query expansion, spelling correction, etc.
Automatic Query Expansion

• Adjusting and reweighting query terms, e.g.,:
  • *Original*: information retrieval
  • *Expanded*: 0.8 information 1.0 retrieval 0.6 search 0.3 BM25 ...

• Based on “semantic” analysis
  • Latent semantic indexing
  • Word co-occurrence, e.g., at document or passage level
  • Using external source, e.g., thesaurus or knowledge base

• Based on users’ relevance feedback
  • *Explicit* feedback, e.g., explicitly marking up which results are relevant and not relevant.
  • *Implicit* feedback, e.g., based on user behavior that may denote relevance information
  • *Pseudo*-relevance feedback, e.g., assuming top-ranked results by an initial search as relevant
Explicit Relevance Feedback

• Explicit relevance feedback
  • Users explicitly assess the relevance of the search results they visited and inform the system
  • The system improves results after receiving users’ feedback
  • IR is a binary classification if we have enough training data ...
  • BIM was proposed to make use of explicit feedback
  • A little bit recap of BIM ...
Recap: BM25 Parameter Estimation

Let \( p_i = P(X_i = 1 \mid R, q) \) and \( q_i = P(X_i = 1 \mid NR, q) \).

RSJ weight: \( w_i = \log \frac{p_i(1-q_i)}{(1-p_i)q_i} = \log \frac{(r+0.5)(N-n-N_R+r+0.5)}{(N_R-r+0.5)(n-r+0.5)} \)

- **Croft & Harper (1979)**
  - **Assumption 3**: \( p_i \) is a constant.
  - **Assumption 4**: the number of relevant documents (\( N_R \) and \( r \)) is a small number comparing to the size of the corpus (\( N \) and \( n \)).

A3: \( \frac{p_i}{(1-p_i)} \) is constant and does not affect ranking

A4: \( \frac{N-n-N_R+r+0.5}{n-r+0.5} \approx \frac{N-n+0.5}{n+0.5} \), very similar to IDF \( \log \frac{N}{n} \)

Without any relevance information, we can estimate \( w_i = \log \frac{N-n+0.5}{n+0.5} \)

Rank results by \( \sum_{t \in D \cap q} \log \frac{N-n_t+0.5}{n_t+0.5} \) or \( \sum_{t \in q} TF_{binary}(t,d) \cdot \log \frac{N-n_t+0.5}{n_t+0.5} \)
Explicit Relevance Feedback

- **Explicit relevance feedback**
  - Users explicitly assess the relevance of the search results they visited and inform the system
  - The system improves results after receiving users’ feedback
  - IR is a binary classification if we have enough training data ...
  - BIM was proposed to make use of explicit feedback

- **A few practical issues**
  - Increases cost of users; users are unwilling to assess
  - Requires a certain amount of judgments to work
  - Improve after interaction? Novelty of results?
  - Can improve over the current query, but users may formulate a new query, which may outperform the improvements
Rocchio Algorithm

• Consider query and document vectors
• A set of relevance judgments R and NR
• Modifying the original query Q by
  • Positive Feedback: moving the query vector closer to the centroid of the judged relevant documents
  • Negative Feedback: keeping the query vector away from the centroid of the judged non-relevant documents
• Can be applied to VSM and BM25

\[ Q' = Q + \alpha \frac{1}{|R|} \sum_{D_j \in R} D_j - \beta \frac{1}{|NR|} \sum_{D_j \in NR} D_j \]
Rocchio Algorithm: An Example

Original Query: 
(5, 0, 3, 0, 1)

Document $D_1$, Relevant: 
(2, 1, 2, 0, 0)

Document $D_2$, Non-relevant: 
(1, 0, 0, 0, 2)

$\alpha = 0.50$, $\beta = 0.25$

Relevance Feedback Formula:

$$Q' = Q + \alpha \frac{1}{|R|} \sum_{D_j \in R} D_j - \beta \frac{1}{|NR|} \sum_{D_j \in NR} D_j$$

$Q' = Q + 0.5 \, D_1 - 0.25 \, D_2$

$$= (5, 0, 3, 0, 1) + 0.5 \, (2, 1, 2, 0, 0) - 0.25 \, (1, 0, 0, 0, 2)$$

$$= (5.75, 0.50, 4.00, 0.0, 0.5)$$
Implicit Relevance Feedback

- **User behaviors that may indicate relevance**
  - Click vs. skip: users’ preference of results based on snippets
  - Click dwell time: the longer the time spent on a webpage is, the more likely the webpage is relevant (often consider $t > 30s$)
  - Past search queries
  - Social website share; social bookmark; ...
  - ...
  - The key (real) secret for IR system improvements since 2005

- **When to apply**
  - While users issue a query, directly apply previous queries’ relevance feedback
  - Requires search log ...

- **Lecture 20 (10/30)**
An Example: Context-sensitive Feedback

- KL-divergence model; estimate a query model using
  - The current search query
  - Past search queries $P(w|H_Q)$
  - Past clicked results $P(w|H_C)$

$$p(w|\theta_k) = \alpha p(w|Q_k) + (1 - \alpha)[\beta p(w|H_C) + (1 - \beta)p(w|H_Q)]$$

$$p(w|Q_i) = \frac{c(w, Q_i)}{|Q_i|}$$

$$p(w|H_Q) = \frac{1}{k-1} \sum_{i=1}^{k-1} p(w|Q_i)$$

$$p(w|C_i) = \frac{c(w, C_i)}{|C_i|}$$

$$p(w|H_C) = \frac{1}{k-1} \sum_{i=1}^{k-1} p(w|C_i)$$

$$p(w|H) = \beta p(w|H_C) + (1 - \beta)p(w|H_Q)$$

$$p(w|\theta_k) = \alpha p(w|Q_k) + (1 - \alpha)p(w|H)$$

Pseudo-relevance Feedback

• Pseudo-relevance feedback (PRF); blind feedback; ...
  • Do an initial search using a regular approach, such as QL
  • Assume the top \( k \) ranked results as relevant
  • Perform relevance feedback based on the top \( k \) results
  • Normally by query expansion
  • Re-run the query

• A few practical issues
  • The assumption ...
  • Efficiency concerns: expand a short query (2-3 words) into a long one (e.g., \(~50\) words)
  • Practically effective for improving overall search effectiveness (in terms of the mean values of effectiveness metrics)
  • Our focus today
Outline Today

- Query expansion: an overview
- **Relevance-based language model**
- Alternatives of RM
Relevance-based Language Model

• Lavrenko & Croft, SIGIR ’01
• Usually called Relevance Model or RM
  • Four variants: RM1, RM2, RM3 (the most popular), and RM4
• RM is a pseudo-relevance feedback (PRF) approach
  • Assumes the top-ranked results by QL as relevant
  • Estimates a query language model $P(t|q)$ based on top results
  • The de facto standard PRF approach for language modeling IR
  • Works reasonably well (in terms of improving search effectiveness metrics such as AP)

Relevance-based Language Model

• Assumes there exists some relevance LM $P(t|R)$
  • $R$ generates both the query and the relevant documents
  • Based on an observed query $q$, make our best estimation of $P(t|R) \Rightarrow P(t|R,q)$
  • We can simply consider $P(t|R,q)$ as a query model $P(t|q)$

$$P(t|R,q) = \frac{P(t,q|R)}{P(q|R)} \propto P(t,q|R)$$
\textbf{RM1} \quad P(t, q \mid R)

\[
= \sum_{D \in \{D_R\}} P(D \mid R) P(t, q \mid D, R)
\]

\[
= \sum_{D \in \{D_R\}} P(D \mid R) P(t \mid D, R) P(q \mid D, R)
\quad \text{A3}
\]

\[
= \sum_{D \in \{D_R\}} P(D \mid R) P(t \mid D, R) \prod_{q_i \in q} P(q_i \mid D, R)
\quad \text{A4}
\]

\[
\propto \sum_{D \in \{D_R\}} P(t \mid D) \prod_{q_i \in q} P(q_i \mid D)
\quad \text{A1, A2}
\]

• Assumptions
  
  • A1: \( P(D \mid R) \) is uniform  
  • A2: \( P(t \mid D, R) = P(t \mid D) \) and \( P(q_i \mid D, R) = P(q_i \mid D) \)  
  • A3: \( P(t, q \mid D, R) = P(t \mid D, R) P(q \mid D, R) \)  
  • A4: \( P(q \mid D, R) = \prod_{q_i \in q} P(q_i \mid D, R) \)
RM1

RM1: \( P(t \mid q, R) \propto P(t, q \mid R) \propto \sum_{D \in \{D_R\}} P(t \mid D) \prod_{q_i \in q} P(q_i \mid D) \)

- **Computation**
  - Iterate over each feedback document (source) \( D \)
  - Assign a weight \( P(q \mid D) = \prod_{q_i \in q} P(q_i \mid D) \) to \( D \)
    - In terms of PRF, we just retrieve top \( k \) results by QL and weight each document by QL probability
    - Higher-ranked results get more weights
  - Expand a term \( t \) from \( D \) by the weight \( P(t \mid D)P(q \mid D) \)
  - Sum up terms’ weights in each feedback document \( D \)
  - Normalize the terms’ weights to probability

\[
P(t \mid q, R) = \frac{\sum_{D \in \{D_R\}} \left( P(t \mid D) \prod_{q_i \in q} P(q_i \mid D) \right)}{\sum_{t_j} \sum_{D \in \{D_R\}} \left( P(t_j \mid D) \prod_{q_i \in q} P(q_i \mid D) \right)}
\]
\textbf{RM2}

\[
P(t,q \mid R) = P(t \mid R)P(q \mid t, R)
\]

\[
= P(t \mid R) \prod_{q_i \in q} P(q_i \mid t, R) \quad \text{A5}
\]

\[
= P(t \mid R) \prod_{q_i \in q} \sum_{D \in \{D_R\}} P(q_i \mid D, t, R)P(D \mid t, R)
\]

\[
= P(t \mid R) \prod_{q_i \in q} \sum_{D \in \{D_R\}} P(q_i \mid D, R) \frac{P(t \mid D, R)P(D \mid R)}{P(t \mid R)} \quad \text{A6}
\]

\[
= \left( \sum_{D_j} P(t \mid D_j, R)P(D_j \mid R) \right) \cdot \prod_{q_i \in q} \sum_{D \in \{D_R\}} P(q_i \mid D, R) \frac{P(t \mid D, R)P(D \mid R)}{\sum_{D_j} P(t \mid D_j, R)P(D_j \mid R)}
\]

\[
\propto \left( \sum_{D_j} P(t \mid D_j) \right) \cdot \prod_{q_i \in q} \sum_{D \in \{D_R\}} P(q_i \mid D) \frac{P(t \mid D)}{\sum_{D_j} P(t \mid D_j)} \quad \text{A1, A2}
\]

- A1: \(P(D \mid R)\) is uniform; A2: \(P(t \mid D, R) = P(t \mid D)\) and \(P(q_i \mid D, R) = P(q_i \mid D)\)
- A5: \(P(q \mid t, R) = \prod_{q_i \in q} P(q_i \mid t, R)\); A6: \(P(q_i \mid D, t, R) = P(q_i \mid D, R)\)
RM2

RM2: $P(t \mid q, R) \propto \left( \sum_{D_j} P(t \mid D_j) \right) \cdot \prod_{q_i \in q} \sum_{D \in \{D_R\}} P(q_i \mid D) \frac{P(t \mid D)}{\sum_{D_j} P(t \mid D_j)}$

- **Computation**
  - Iterate over each query term $q_i$
    - Iterate over each feedback document $D$
    - Assign a weight $P(q_i \mid D)$ to $D$
    - Expand a term $t$ from $D$ by $P(t \mid D)P(q_i \mid D)$: if both $t$ and $q_i$ occur frequently in $D$, $t$ gets a greater weight
  - Sum up the weight in each document
  - Multiply the expansion weight for each $q_i$
  - Normalize the terms’ weights to probability
RM1 vs. RM2

- All assumptions
  - A1: \( P(D|R) \) is uniform
  - A2: \( P(t|D, R) = P(t|D) \) and \( P(q_i|D, R) = P(q_i|D) \)
  - A3: \( P(t, q|D, R) = P(t|D, R)P(q|D, R) \)
  - A4: \( P(q|D, R) = \prod_{q_i \in q} P(q_i|D, R) \)
  - A5: \( P(q|t, R) = \prod_{q_i \in q} P(q_i|t, R) \)
  - A6: \( P(q_i|D, t, R) = P(q_i|D, R) \)

- RM1: A1, A2, A3, A4
  - Only assumes the independence between \( t \) and \( q \) (as a whole)

- RM2: A1, A2, A5, A6
  - Assumes the independence between \( t \) and each \( q_i \)
  - Separately expand by each individual terms in the original query
  - (Many believe this is) a stronger independence assumption
RM3 and RM4

- Sometimes the original query terms do not have the highest weights in the expanded query model ...
  - Seems risky and problematic
- Interpolate the expanded model with the query’s MLE
  - RM3: original query’s MLE + RM1
  - RM4: original query’s MLE + RM2

\[
P_{RM3} (t \mid q) = (1 - \lambda) \cdot P_{MLE} (t \mid q) + \lambda \cdot P_{RM1} (t \mid q)
\]

\[
P_{RM4} (t \mid q) = (1 - \lambda) \cdot P_{MLE} (t \mid q) + \lambda \cdot P_{RM2} (t \mid q)
\]
Model Estimation Details

- \( P(w|d) \) and \( P(q_i|d) \) can have different settings
  - E.g., using different smoothing parameters
  - Many suggest that keeping \( P(q_i|d) \) the same as QL
  - Often do not use smoothing for \( P(w|d) \) – to avoid assigning a high probability to the common terms

- A few practical settings
  - RM3 is the most popular one
  - Top \( k \) results: \( k \) typically ranges from 5 – 50
  - Only using the top \( n \) terms; \( n \) typically 5 – 100
    - Related to search efficiency issues if using the expanded query for a new search
  - Reranking the original results or a new search?
  - Needs careful training
Outline Today

• Query expansion: an overview
• Relevance-based language model
• Alternatives of RM
Alternatives of RM

• The query model used in the original KL-divergence model paper by Lafferty & Zhai in SIGIR ’01.

• Query expansion using global word co-occurrence
  • $t(q_i \mid w)$: the probability of “translating” a word $w$ into each $q_i$
  • Often estimated based on word co-occurrence & context similarity

$$P(w \mid q) \propto P(q \mid w)P(w) = \prod_{q_i \in q} t(q_i \mid w)P(w)$$

<table>
<thead>
<tr>
<th>$q$</th>
<th>$t(q \mid w)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>everest</td>
<td>0.439</td>
</tr>
<tr>
<td>climb</td>
<td>0.057</td>
</tr>
<tr>
<td>climber</td>
<td>0.045</td>
</tr>
<tr>
<td>whittaker</td>
<td>0.039</td>
</tr>
<tr>
<td>expedition</td>
<td>0.036</td>
</tr>
<tr>
<td>float</td>
<td>0.024</td>
</tr>
<tr>
<td>mountain</td>
<td>0.024</td>
</tr>
<tr>
<td>summit</td>
<td>0.021</td>
</tr>
<tr>
<td>highest</td>
<td>0.018</td>
</tr>
<tr>
<td>reach</td>
<td>0.015</td>
</tr>
<tr>
<td>$w =$ everest</td>
<td>$w =$ wildlife</td>
</tr>
<tr>
<td>wildlife</td>
<td>0.705</td>
</tr>
<tr>
<td>fish</td>
<td>0.038</td>
</tr>
<tr>
<td>acre</td>
<td>0.012</td>
</tr>
<tr>
<td>species</td>
<td>0.010</td>
</tr>
<tr>
<td>forest</td>
<td>0.010</td>
</tr>
<tr>
<td>environment</td>
<td>0.009</td>
</tr>
<tr>
<td>habitat</td>
<td>0.008</td>
</tr>
<tr>
<td>endangered</td>
<td>0.007</td>
</tr>
<tr>
<td>protected</td>
<td>0.007</td>
</tr>
<tr>
<td>bird</td>
<td>0.007</td>
</tr>
</tbody>
</table>
Alternatives of RM

- The query model used in the original KL-divergence model paper by Lafferty & Zhai in SIGIR ’01.
- Query expansion using local PRF documents
  - Similar to RM1, but uses $P(w)P(d|w)$ instead of $P(w|d)$

$$P(w|q) \propto P(w)P(q|w) = P(w) \sum_{d_i} P(q|d_i)P(d_i|w)$$

$$P(d_i|w) = \frac{P(w|d_i)P(d_i)}{\sum_{d_j} P(w|d_j)P(d_j)}$$

| $w$              | $p(w|q)$ |
|------------------|---------|
| virus            | 0.275   |
| ebola            | 0.197   |
| hoax             | 0.051   |
| viruses          | 0.034   |
| outbreak         | 0.034   |
| fever            | 0.033   |
| disease          | 0.024   |
| haemorrhagic     | 0.023   |
| gabon            | 0.022   |
| infected         | 0.019   |
| aids             | 0.016   |
| security         | 0.014   |
| monkeys          | 0.013   |
| hiv              | 0.011   |
| zaire            | 0.011   |

$q = ebola$ $virus$ (Web)

| $w$      | $p(w|q)$ |
|----------|---------|
| star     | 0.361   |
| wars     | 0.217   |
| rpg      | 0.058   |
| trek     | 0.033   |
| starwars | 0.032   |
| movie    | 0.023   |
| episode  | 0.020   |
| movies   | 0.015   |
| war      | 0.014   |
| character| 0.013   |
| tv       | 0.013   |
| film     | 0.012   |
| fan      | 0.012   |
| reviews  | 0.012   |
| jedi     | 0.008   |

$q = star$ $wars$ (Web)
Alternatives of RM

- Zhai & Lafferty, CIKM ’01.
- A Simple mixture model (SMM) approach
  - Similar to the parsimonious language model last week ...
  - Factor out corpus model (an approximation of NR) from pseudo-relevant documents’ language models using EM

\[
\log p(F|\theta_F) = \sum_{w \in V} c(w, F) \log((1 - \lambda)p(w|\theta_F) + \lambda p(w|C))
\]
Alternatives of RM

- Zhai & Lafferty, CIKM ’01.
- Divergence Minimization Model (DMM)
  - Similar to Rocchio algorithm (the language model version)
  - Approximate non-relevant (NR) using the corpus model

\[
p(w|\theta_F) \propto \exp \left[ \frac{1}{1 - \lambda} \left( \frac{1}{|F|} \sum_{i=1}^{F} \log p(w|\theta_i) - \lambda \log p(w|C) \right) \right]
\]
Comparing different approaches

- Lv & Zhai, CIKM ’09

<table>
<thead>
<tr>
<th>S.w.</th>
<th>Metric</th>
<th>MLE</th>
<th>RM3</th>
<th>RM4</th>
<th>DMM</th>
<th>SMM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trained on AP1 and Tested on AP2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/</td>
<td>AvgPr</td>
<td>0.220</td>
<td>0.295</td>
<td>0.301</td>
<td>0.290</td>
<td>0.304</td>
</tr>
<tr>
<td></td>
<td>Pr@10</td>
<td>0.386</td>
<td>0.408</td>
<td>0.418</td>
<td><strong>0.422</strong></td>
<td>0.400</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>3074</td>
<td>3810</td>
<td>3892</td>
<td>3681</td>
<td><strong>3933</strong></td>
</tr>
<tr>
<td>w/o</td>
<td>AvgPr</td>
<td>0.231</td>
<td>0.312</td>
<td>0.321</td>
<td>0.289</td>
<td><strong>0.324</strong></td>
</tr>
<tr>
<td></td>
<td>Pr@10</td>
<td>0.398</td>
<td>0.436</td>
<td><strong>0.448</strong></td>
<td>0.424</td>
<td>0.432</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>3154</td>
<td>3913</td>
<td>3908</td>
<td>3674</td>
<td>3921</td>
</tr>
<tr>
<td></td>
<td>Trained on TREC6 and Tested on TREC78</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/</td>
<td>AvgPr</td>
<td>0.217</td>
<td>0.249</td>
<td>0.242</td>
<td>0.235</td>
<td><strong>0.251</strong></td>
</tr>
<tr>
<td></td>
<td>Pr@10</td>
<td>0.437</td>
<td>0.438</td>
<td>0.426</td>
<td>0.443</td>
<td>0.443</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>5114</td>
<td>5805</td>
<td>5739</td>
<td>5476</td>
<td><strong>5821</strong></td>
</tr>
<tr>
<td>w/o</td>
<td>AvgPr</td>
<td>0.217</td>
<td>0.251</td>
<td>0.243</td>
<td>0.235</td>
<td><strong>0.252</strong></td>
</tr>
<tr>
<td></td>
<td>Pr@10</td>
<td>0.434</td>
<td><strong>0.454</strong></td>
<td>0.446</td>
<td>0.433</td>
<td>0.441</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>5107</td>
<td>5799</td>
<td>5776</td>
<td>5500</td>
<td><strong>5896</strong></td>
</tr>
<tr>
<td></td>
<td>Well-Tuned on WT2G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/</td>
<td>AvgPr</td>
<td>0.293</td>
<td><strong>0.338</strong></td>
<td>0.319</td>
<td>0.327</td>
<td>0.330</td>
</tr>
<tr>
<td></td>
<td>Pr@10</td>
<td>0.450</td>
<td><strong>0.500</strong></td>
<td>0.470</td>
<td>0.494</td>
<td>0.496</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>1830</td>
<td>1822</td>
<td>1806</td>
<td>1843</td>
<td><strong>1856</strong></td>
</tr>
<tr>
<td>w/o</td>
<td>AvgPr</td>
<td>0.306</td>
<td><strong>0.344</strong></td>
<td>0.328</td>
<td>0.326</td>
<td>0.331</td>
</tr>
<tr>
<td></td>
<td>Pr@10</td>
<td>0.456</td>
<td><strong>0.490</strong></td>
<td><strong>0.490</strong></td>
<td>0.476</td>
<td>0.476</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>1870</td>
<td>1862</td>
<td>1879</td>
<td>1873</td>
<td><strong>1889</strong></td>
</tr>
</tbody>
</table>
Pseudo-relevance Feedback

• “Usually” believed to be a useful technique
• But somewhat controversial …
  • Recall oriented; limited improvements in precision at the top
  • Making good queries bad; making bad queries worse
  • “Overall” improvements: average values of metrics?
  • But improving bad/difficult queries may be more important
  • Search efficiency concern
  • Difficult to control; unpredictable for the user

• Difficult to improve in noisy corpus (such as web corpus)
  • Using some clean corpus for query expansion, e.g., Wikipedia
Next two weeks

- Machine learning for IR (learning-to-rank)
- Representation learning, e.g., word embeddings
- Deep neural nets for IR ranking