Better than the Real Thing?
Iterative Pseudo-Query Processing using Cluster-Based Language Models

Oren Kurland, Cornell University
Lillian Lee, Cornell University
Carmel Domshlak, Technion
Ad hoc IR: Short queries are problematic …

• ambiguity (polysemy)
• synonomy
• sparse data problem

Desire a query model that is:

• “richer” (with terms)
• more accurate
One approach: query expansion using pseudo-feedback (PF)

- Apply relevance-feedback methods to top-ranked documents retrieved in response to the query
  - e.g., treat them as relevant and apply Rocchio’s model

- Issues
  - Precision (sensitivity to the quality of initial ranking)
    - Tao&Zhai ‘04
  - Aspect recall
    - Query aspects missing from the initial retrieved list
      (Harman&Buckley ’04)
The goal

Design a method that will

- be less sensitive to the quality of the initial ranking used for PF (e.g., to the number of top retrieved documents used)
- ameliorate the problem of aspects missing from the initial list used for feedback
- take advantage of “corpus structure”
  - for dealing with synonomy and polysemy
The language modeling approach to IR

• Rank documents by the probability of their language models generating the query terms (Ponte&Croft ’98)
  – Principle: retrieve the documents that are the best renderers (generators) of the query

• Motivating hypothesis for our work:
  “Documents that are good renderers of a query may be good alternate renditions of it”
  – cf. the basic idea underlying pseudo-feedback methods
  – PF methods in the LM framework: Lafferty&Zhai ’01, Zhai&Lafferty ’01, Lavrenko&Croft ’01, Tao&Zhai ‘04
A new approach: iterative pseudo-query processing

- Replace query with \textit{pseudo-queries} consisting of the documents that are the query’s best renderers
- Iteratively seek top renderers of pseudo-queries

![Diagram showing iterative pseudo-query processing]

original query: $q^*$
Renderers (Pseudo-Queries)

- Documents
- Clusters
  - Alleviate the “aspect recall” problem by capturing corpus structure
    - Using clusters’ language models was shown to be effective (Liu & Croft ’04, Kurland & Lee ’04)
  - We use nearest-neighbors (and hence overlapping) clusters (Kurland & Lee ’04)
Notation and Conventions

$p_d(\cdot), p_c(\cdot)$: rendition probabilities assigned by doc $d$, cluster $c$

$q^* : original query$

$\hat{Q} = (\hat{q}_1, \hat{w}_1), (\hat{q}_2, \hat{w}_2), (\hat{q}_3, \hat{w}_3),...$ ranked list of pseudo queries and their scores

$TopRen(x)$: set of $k$ items $r$ that yield highest $p_r(x)$

$Score(d) = \hat{w}(d)$, $Score(c) = \hat{w}(c)$
Doc-Audition algorithm: Renderers are documents

\[ \text{Score}_{\text{doc}}(d) \triangleq \sum_{\hat{q}: d \in \text{TopRen}(\hat{q})} \hat{w}(\hat{q}) \cdot p_d(\hat{q}) \]

Diagram: Diagram showing the relationship between queries and documents.
common PF(QE) methods
(cf. Rocchio ’71)

doc-audition
(cf. Noreault ‘79)
Cluster-Audition Algorithm:
Renderers are clusters and documents

\[
Score_{clust}(c) \triangleq \sum_{\hat{q}:c \in \text{TopRen}(\hat{q})} \hat{w}(\hat{q}) \cdot p_c(\hat{q})
\]

\[
Score_{clust}(d) \triangleq \sum_{c:d \in \text{TopRen}(c)} Score_{clust}(c) \cdot p_d(c)
\]
Query Drift

– “The alteration of the focus of a search topic caused by improper expansion” (Mitra et al. ’98)

– Solutions:
  • Low number of iterations
  • Truncated Re-Rank (*Clust, Doc* algorithms)
    – Re-rank final list of retrieved documents by $p_d(q^*)$
  • Interpolation (*Int-Clust, Int-Doc* algorithms)
    – $\lambda \text{Score}(d) + (1 - \lambda) p_d(q^*)$
  • More techniques in the paper
Empirical Results (Avg. Prec.)

• parameters optimized for max avg. precision
Empirical Results (Recall)

- Parameters optimized for max avg. precision
Comparison to PF methods (avg. prec)

- parameters optimized for max avg. precision
Comparison to PF methods (recall)

- parameters optimized for max avg. precision
Effect of # of top-retrieved documents (AP89) Doc, Int-Doc algorithms, Relevance Model
Effect of # of top-retrieved clusters (AP89)
Clust, Int-Clust algorithms, Relevance Model
Coping with Length Issues in Rendition Probabilities

Problem with unigram LMs:
"p(\tilde{w}) = p(w_1) \cdots p(w_n)" will penalize clusters, long docs

• can affect our algorithms (e.g., underflow issues)

\[
p_x(\tilde{w}) \triangleq \exp\left(-D\left(\tilde{p}_w^{MLE}(\bullet) \parallel \tilde{p}_x^{\text{smoothed}}(\bullet)\right)\right)
\]

We use:
\[
\left(\frac{1}{|\tilde{w}|} \cdot \exp\left(H\left(\tilde{p}_w^{MLE}(\bullet)\right)\right)\right)
\]

geometric mean \quad \text{entropy-based term}

Geometric mean of the MLE was used in TDT: Lavrenko et al. ‘02
Future Work

• Compare against vector space representation and metrics
• Mixture of renderers
  – The renderers list of a textual item could include both documents and clusters
Summary

• Novel approach to retrieval using pseudo-queries, clusters, language models and iterative mechanism

• Our algorithms:
  – provide state-of-the-art performance
  – are stable with respect to the number of top retrieved renderers
  – utilize clusters’ language models
  – provide effective means for coping with query drift