Corpus structure, language models, and ad hoc information retrieval
(Based on a SIGIR’04 Presentation)

Oren Kurland

Joint work with Lillian Lee

Computer Science Department
Cornell University
The Language Modeling Approach to Ad Hoc IR

- Given a query Q, document D, and induced language models, \( M_Q, M_D \):
  - determine the probability \( p(Q | M_D) \) (Ponte&Croft 98)
  - use a loss function \( KL(M_Q \| M_D) \) (Lafferty&Zhai 01)
Advantages of LM approach

• Generative model with theoretical foundations
• Tools from other areas: machine translation, speech recognition
• Integration of indexing and retrieval models
Problems and partial solutions

• Sparse data problem
  – Smoothing: sensitive to method/parameters
    • Song&Croft 99, Zhai&Lafferty 01, 02, Hiemstra 02

• Synonymy/polysemy
  – Latent variable models: sensitive to outliers
    • Hofmann 99, Blei et al. 03

• Feature selection
  – Stop word removal, stemming: affect performance
The goal

• Design a retrieval approach that will
  – help with polysemy and synonomy
  – benefit from structural analysis but will still handle outliers
  – be robust with respect to
    • Feature selection
    • Smoothing method / parameter settings

• A possible approach: use clustering
Combining clusters and LMs is effective in:

- Distributional clustering
  - Pereira, Tishby and Lee 93
- Modeling long distance dependencies in text
  - Iyer and Ostendorf 99
- Distributed IR
  - Xu and Croft 99
- Topic detection and tracking
  - Allan et al. 98, Spitters and Kraaij 01
- Ad hoc IR?
Clustering and ad hoc IR

- The cluster hypothesis: Closely associated documents tend to be relevant to the same requests (C.J. Van Rijsbergen, 79)
- Cluster-retrieval work
  - Jardine and Rijsbergen 71, Croft 80, Voorhees 85, Hearst and Pederson 96, Tombros et al. 2002
- Not clear whether using offline clustering consistently helps ad hoc information retrieval
- Clustering of retrieved results seemed to be effective in specific settings (e.g. for browsing documents)
When can clustering help?

Query = truck, bus

\[ \text{sim}(t_i, t_j) = |t_i \cap t_j| \]

\(\triangleright\) Rank using documents <

\[ \text{sim}(q, d1) > \text{sim}(q, d5) = \text{sim}(q, d6) \]

\(\text{Ranking} = d1, d3, d4, d2, d6, d5\)

\(\triangleright\) Rank using clusters and docs <

\[ \text{sim}(q, C2) > \text{sim}(q, C1) > \text{sim}(q, C3) \]

\[ \text{sim}(d5, C2) > \text{sim}(d4, C2) = \text{sim}(d3, c2) \]

\(\text{Ranking} = d5, d3, d4, d2, d1, d6\)
A Framework integrating Corpus Structure and Language Models

- Corpus structure = document similarities

**Our Framework**

- *Document d*
  - *doc-based LM*
    - \( p_d(string) \)
  - Accurate representation of document

- *Cluster c*
  - *cluster-based LM*
    - \( p_c(string) \)
  - Potentially good representation of doc context
Using overlapping clusters - motivation

• Finding facets in the corpus
  – Topics, viewpoints, authors, etc …

• Approximating candidate relevance models
  – Cluster hypothesis
Offline Step: Clustering

Set $c_i = k$ nearest neighbors of $d_i$

(metric = KL divergence)

- The language model of a cluster: treat cluster as concatenation of constituent documents
- Overlapping clusters
- Long history in IR (e.g. Griffiths et al. 86)
Algorithmic Framework

Given query $q$ and $N$(the number of docs to retrieve):

1. For each document $d$, 
   - Choose $Facets(q, d) \subseteq Clusters$
   - Score $d$ by a weighted combination of $p_{d}(q)$ and the $p_{c}(q)$'s for all $c \in Facets(q, d)$

2. Set $TopDocs(N)$ to the ranked-ordered list of $N$-top scoring documents

3. Optional: re-rank $d \in TopDocs(N)$ by $p_{d}(q)$

4. Return $TopDocs(N)$
Two Families of Algorithms

- **Selection Algorithms**
  - Clusters select relevant documents: those within
    \[ \text{TopClusters}_q(m) = \text{the } m \text{ clusters with highest } p_c(q) \text{'s} \]
  - Proximity information within clusters is not considered, i.e., \( p_c(d) \) is not used.
  - Algorithms: \textit{set-select, bag-select}

- **Extended Aspect Algorithms (aspect-x)**
  - Clusters select relevant documents and smooth documents’ LMs
  - Algorithms: \textit{aspect-x, interpolation}
The *Set-Select* Algorithm

- Instantiation of the framework:

\[ Facets(q, d) = \{c : d \in c\} \cap \text{TopClusters}_q(m) \]

\[ \text{Score}(d) = p_d(q) \cdot \delta[|Facets(q, d)| > 0] \]

- The procedure

Rank only documents in top retrieved clusters using \( p_d(q) \)
The Bag-Select Algorithm

- Instantiation of the framework:

\[
\text{Facets}(q,d) = \{ c : d \in c \} \cap \text{TopClusters}_q(m)
\]

\[
\text{Score}(d) = p_d(q) \cdot |\text{Facets}(q,d)|
\]

- The procedure

Rank only documents in top retrieved clusters using \( p_d(q) \times \# \text{ of top clusters } d \text{ belongs to} \)
Set-Select, Bag-Select

Facets \( q, d_1 \) = \{c_1\}
Facets \( q, d_{40} \) = \{c_1, c_2\}
Facets \( q, d_2 \) = \{c_1, c_2, c_3\}

Score \( d_1 \) = \( p_{d_1}(q) \)
Score \( d_{40} \) = \( p_{d_{40}}(q) \)
Score \( d_2 \) = \( p_{d_2}(q) \)

Score \( d_1 \) = \( p_{d_1}(q) \)
Score \( d_{40} \) = \( 2 \times p_{d_{40}}(q) \)
Score \( d_2 \) = \( 3 \times p_{d_2}(q) \)
The *aspect-x* Algorithm

- **Instantiation of the framework:**

  \[ Facets(q,d) = \{ c : d \in c \} \cap TopClusters_q(m) \]

  \[ \text{Score}(d) = \sum_{c \in Facets(q,d)} p_c(q) \cdot p_c(d) \]

- **Probabilistic motivation:**

  \[ \hat{p}(q \mid d) = \sum_c p(q,c \mid d) = \alpha \sum_c p(q \mid c,d) p(d \mid c) = \alpha \sum_c p(q \mid c) p(d \mid c) \]

  Assuming uniform priors for clusters and documents

  Assuming query is independent of document given cluster

  \[ \text{aspect-x requires a post re-rank phase} \]
The interpolated Model

• Instantiation of the framework

\[ \text{Facets}(q, d) = \{ c : d \in c \} \cap \text{TopClusters}_q(m) \]

\[ \text{Score}(d) = \lambda p_d(q) + (1 - \lambda) \sum_{c \in \text{Facets}(q, d)} p_c(q) \cdot p_c(d) \]

• Probabilistic motivation:

\[ \hat{p}(q \mid d) = \alpha \sum_c p(q \mid c, d) p(d \mid c) = \alpha \sum_c \left[ \lambda p(q \mid d) + (1 - \lambda) p(q \mid c) \right] p(d \mid c) \]

Dropping the independence assumption \(\rightarrow\) no need for post re-rank
Aspect-x models (Summary)

• Clusters play a dual role of smoothing and selection
• Re-rank phase is implemented for the aspect-x algorithm
  – Deal with overgeneralization
• Re-rank is not needed for the interpolated algorithm
• Relatively robust with respect to the number of top–retrieved clusters
Empirical Results (Avg. Prec.)

- Baseline
- Set-s
- Bag-s
- Aspect-x
- Interp

* parameters optimized for max avg. precision

- AP89
- AP88+89
- LA+FR

\{ \} — S
\{ \} — A
Empirical Results (Recall)

- parameters optimized for max avg. precision
## Feature Selection (Stemming)

<table>
<thead>
<tr>
<th></th>
<th>AP89</th>
<th>S-Baseline</th>
<th>U-Baseline</th>
<th>S-interp</th>
<th>U-interp</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Avg. Prec.</strong></td>
<td>21.03%</td>
<td>19.56%</td>
<td>24.9%*</td>
<td>24.08%*</td>
<td></td>
</tr>
<tr>
<td><strong>Recall</strong></td>
<td>48.67%</td>
<td>44.99%</td>
<td>63.62%*</td>
<td>60.66%</td>
<td></td>
</tr>
</tbody>
</table>
Effect of interpolation (AP89)
Additional Issues

• Smoothing
  – Default smoothing parameter settings (Zhai & Lafferty, 2002) provide very good performance
  • No need for training or optimization for corpora tested
  – Different smoothing methods (Jelinek Mercer, Absolute Discounting) w/ default settings show significant improvement with respect to the optimized baselines

• Size of clusters
  – Small clusters (5-10 documents) yield very good performance

• Feature selection
  – Using clusters provides very good performance with and without stemming/stop-word removal
Is it all due to language models?

Experiments with the selection algorithms using td.idf representation and inner product:

<table>
<thead>
<tr>
<th></th>
<th>LA+FR</th>
<th>Baseline</th>
<th>set-select</th>
<th>bag-select</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Prec.</td>
<td>16.43%</td>
<td>17.18%*</td>
<td>16.68%*</td>
<td></td>
</tr>
<tr>
<td>Recall</td>
<td>47.45%</td>
<td>57.58%</td>
<td>51.98%</td>
<td></td>
</tr>
</tbody>
</table>

**Conclusion**: Modeling corpus structure with overlapping clusters is effective for retrieval.
Discussion

– Is smoothing of documents’ LMs the main reason for improved performance?
  • Results of smoothing documents’ LMs with their nearest neighbors’ LMs indicate that no

– Is cluster-based selection the main reason for improved performance?
  • Superiority of aspect-x algo. to selection algo. indicates that no

– Our algorithms don’t depend on relevant documents being clustered together
Xiaoyong Liu and W. Bruce Croft, SIGIR ‘04

- Experimented with
  - Hard clustering both for offline and query-dependent clustering
  - Using clusters for either selecting relevant documents or smoothing documents’ LMs

- Their conclusions:
  - Using clusters only for smoothing is effective
    - Consistent with our findings for smoothing with nearest neighbors’ LMs
  - Cluster selection doesn’t yield improvements when using query-dependent clusters
    - In our framework it does when using offline overlapping clusters
  - Cluster-based smoothing outperforms cluster-selection (for query-dependent clusters)
    - In our framework (for offline clusters) selection outperforms smoothing; integration yields the best results
  - Potential for using overlapping (or soft clusters): model document-cluster relationship using language models
Summary

• A novel approach which combines language models and corpus structure
• An algorithmic framework; our instantiations:
  – Have superior performance to the baselines’
  – Are robust with respect to many factors
    • Smoothing method
    • Smoothing parameters
    • Feature selection
    • Similarity metric
Future work

• Study whether overlapping of clusters is critical
• Use different clustering algorithms with the retrieval framework (e.g., compare to Azzopardi et al. 04)
• Use the same algorithms on query-dependent clusters
• Combine relevance models with our framework