Modeling Query Term Dependencies in IR with Markov Random Fields

Donald Metzler and W. Bruce Croft

University of Massachusetts, Amherst
Center for Intelligent Information Retrieval
Language Modeling for IR

- Assumes there exists some model $\theta$ that generates both queries and documents
- Ranking documents
  - Query likelihood – $P( Q \mid D )$
  - Document likelihood – $P( D \mid Q )$
  - Model comparison – $KL( \theta_Q \parallel \theta_D )$
- Estimating complex joint distributions is difficult due to data sparsity
- Basic idea behind most of these models
  - Create a simple approximation for some complex underlying distribution
Unigram Model

- Bag of words assumption
- Query terms independent
- Easy estimation

\[
P_\hat{\theta}(Q, D) = \int_\theta P(Q | \theta) P(D | \theta) P(\theta)
\]

\[
\approx \prod_{q \in Q} P(q | \hat{\theta}) \prod_{d \in D} P(d | \hat{\theta})
\]
Bigram Model

- Models sequential generation of text
- Term \( j \) dependent on term \( j-1 \)
- Estimation requires back-off due to sparsity

\[
P_{\hat{\theta}}(Q, D) = \int_{\theta} P(Q | \theta) P(D | \theta) P(\theta) \\
\approx \prod_{i} P(q_i | q_{i-1}, \hat{\theta}) \prod_{j} P(d_j | d_{j-1}, \hat{\theta})
\]
Tree Dependence Model

- Method of approximating complex joint distribution
- Compute EMIM between every pair of terms
- Build maximum spanning tree
- Add directionality
- Tree encodes first order dependencies

\[ P(A, B, C, D, E) \approx P(D)P(A \mid D)P(C \mid D)P(E \mid C)P(B \mid C) \]
Tree Dependence Model

- Can captures non-sequential generation of text
- Estimation not as straightforward

\[
P_{\hat{\theta}}(Q, D) = \int_{\theta} P(Q | \theta)P(D | \theta)P(\theta)
\]

\[
\approx \prod_{i} P(q_i | q_{\text{Parent}(i)}, \hat{\theta}) \prod_{j} P(d_j | d_{\text{Parent}(j)}, \hat{\theta})
\]
Pros and Cons

**Pros**
- Easy to implement and use
- Perform relatively well
- Well-studied

**Cons**
- Implicitly use textual features (unigrams, etc.)
- Limited notion of term dependence

**Want to move away from modeling text generation to discriminating between relevant and non-relevant**
Desiderata

- Our desired model should be able to...
  - Support standard IR tasks (ranking, query expansion, etc.)
  - Easily model dependencies between terms
  - Handle wide range of features
  - Consistently and significantly improve effectiveness over bag of words models and existing dependence models

- Proposed solution: Markov random fields
Markov Random Fields

The anatomy of a MRF

- Graph $G$
  - vertices represent random variables
  - edges encode dependence semantics
- Potentials over the cliques of $G$
  - Non-negative functions over clique configurations
  - Measures ‘compatibility’

\[
P_{G,\Lambda}(X) = \frac{1}{Z} \prod_{c \in \text{Clique}(G)} \psi(c; \Lambda) \quad \psi_i(c; \Lambda) = \exp[\lambda_i f_i(c)]
\]
Using the model...

Three steps:

- Create graph $G$ that encodes dependencies between query terms
- Define set of features over cliques sets
- Estimate parameters
Full Independence

- Generalization of unigram model
- Shifts burden from making distributional assumptions on $\theta$ to choosing good potentials
- Potentials measure compatibility of term and document
  - $\psi(w_i, D)$ – “how many times $w_i$ occurs within document $D$”
Sequential Dependence

- Generalization of bigram model
- Potentials can go beyond bigram features and use generalized term proximity features
  - $\psi(w_i, w_{i+1}, D)$ – “how many times ‘$w_i \ w_{i+1}$’ occurs as an exact phrase in document $D$”
Generalized Dependence

- Any dependence structure can be used
- Various possibilities for potentials
  - $\psi(w_i, w_j, D)$ – “how many times $w_i$ and $w_j$ occur within $N$ words of each other in document $D$”
Full Dependence

- Most general model
- Makes no dependence assumptions
Parameter Tying

- In theory, a different potential function can be associated with every clique in a graph.
- Typical solution is to define potentials over *maximal* cliques of $G$.
- Need more fine-grained control over our potentials.
- Use clique sets:
  - Set of cliques that share a parameter and potential function.
  - We identified 7 clique sets that are relevant to IR tasks.
Clique Sets

**Single term document/query cliques**

\[ T_D = \text{cliques w/ one query term + } D \]

\[ \psi(\text{domestic, } D), \psi(\text{adoption, } D), \psi(\text{laws, } D) \]

**Ordered terms document/query cliques**

\[ O_D = \text{cliques w/ two or more contiguous query terms + } D \]

\[ \psi(\text{domestic, adoption, } D), \psi(\text{adoption, laws, } D), \]
\[ \psi(\text{domestic, adoption, laws, } D) \]

**Unordered terms document/query cliques**

\[ U_D = \text{cliques w/ two or more query terms (in any order) + } D \]

\[ \psi(\text{domestic, adoption, } D), \psi(\text{adoption, laws, } D), \]
\[ \psi(\text{domestic, laws, } D), \psi(\text{domestic, adoption, laws, } D) \]
**Clique Sets**

<table>
<thead>
<tr>
<th><strong>Single term query cliques</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_Q =$ cliques w/ one query term</td>
</tr>
<tr>
<td>$\psi$(domestic), $\psi$(adoption), $\psi$(laws)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Ordered terms query cliques</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_Q =$ cliques w/ two or more contiguous query terms</td>
</tr>
<tr>
<td>$\psi$(domestic, adoption), $\psi$(adoption, laws), $\psi$(domestic, adoption, laws)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Unordered terms query cliques</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_Q =$ cliques w/ two or more query terms (in any order)</td>
</tr>
<tr>
<td>$\psi$(domestic, adoption), $\psi$(adoption, laws), $\psi$(domestic, laws), $\psi$(domestic, adoption, laws)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Document clique</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$D =$ singleton clique w/ $D$</td>
</tr>
<tr>
<td>$\psi(D)$</td>
</tr>
</tbody>
</table>
Using these clique sets, the joint distribution simplifies to:

\[
\ln P_{G,\Lambda}(Q, D) = \lambda_{T_D} \sum_{c \in T_D} f_{T_D}(c) + \lambda_{O_D} \sum_{c \in O_D} f_{O_D}(c) + \lambda_{U_D} \sum_{c \in U_D} f_{U_D}(c) + \\
\lambda_{T_Q} \sum_{c \in T_Q} f_{T_Q}(c) + \lambda_{O_Q} \sum_{c \in O_Q} f_{O_Q}(c) + \lambda_{U_Q} \sum_{c \in U_Q} f_{U_Q}(c) + \\
\lambda_D f_D(D) - \log Z_{\Lambda}
\]
Ranking with the Model

Documents are ranked according to:

\[ P_{G,A}(D|Q) \overset{\text{rank}}{=} \lambda_{T_D} \sum_{c \in T_D} f_{T_D}(c) + \lambda_{O_D} \sum_{c \in O_D} f_{O_D}(c) + \lambda_{U_D} \sum_{c \in U_D} f_{U_D}(c) + \lambda_D f_D(D) \]

Can use P(Q | D) for query expansion (ongoing work)
Features

- Features can be either textual or non-textual
- Should measure compatibility of clique configuration
- Examples:
  - TF – how common is this term in this document?
  - IDF – how rare is this term?
  - Named entities – is this a name? location? date?
  - Term proximity – where do the query terms occur?
  - Stylistic – is the matched term in a heading? table? title?
  - Document length – does this document contain 1 or 1M words?
  - PageRank – how ‘popular’ is this web document?
  - Readability – how readable is this document?
Parameter Estimation

Given a set of relevance judgments $T$, we want the maximum *a posteriori* estimate:

$$\hat{\Lambda} = \arg \max_{\Lambda} P(\Lambda | T)$$

What is $P(\Lambda | T)$? $P(T | \Lambda)$ and $P(\Lambda)$?
- Depends on how model is being evaluated!
- Want $P(\Lambda | R)$ to be peaked around the parameter setting that maximizes the metric we are interested in
Direct Maximization

- Must search for maximum
  - Can be shown that intrinsic parameter space is multinomial manifold (a simplex)
- Most IR metrics are not differentiable with respect to the model parameters
- Perform coordinate ascent over the simplex
- Feasible since we have a small number of parameters
Evaluation Metric Surfaces
Application: Ad Hoc Retrieval

- Use Indri search engine (http://www.lemurproject.org/indri)
- Standard document preprocessing
  - Stopword removal
  - Stemming
- Use simple phrase and term proximity features

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Size</th>
<th># Docs</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSJ</td>
<td>Wall Street Journal '87-'92</td>
<td>510MB</td>
<td>173,252</td>
</tr>
<tr>
<td>AP</td>
<td>Associated Press '88-'90</td>
<td>730MB</td>
<td>242,918</td>
</tr>
<tr>
<td>ROBUST04</td>
<td>Collection of news sources</td>
<td>860MB</td>
<td>528,155</td>
</tr>
<tr>
<td>WT10g</td>
<td>TREC Web Track collection</td>
<td>11GB</td>
<td>1,692,096</td>
</tr>
<tr>
<td>GOV2</td>
<td>Crawl of .GOV domain circa 2004</td>
<td>427GB</td>
<td>25,205,179</td>
</tr>
</tbody>
</table>
Ad Hoc Retrieval Features

\[ f_{T_D}(q_i, D) = \log \left[ (1 - \alpha_D) \frac{tf_{q_i,D}}{|D|} + \alpha_D \frac{cf_{q_i}}{|C|} \right] \]

\[ f_{O_D}(q_i, ..., q_{i+k}, D) = \log \left[ (1 - \alpha_D) \frac{tf_{#1(q_i, ..., q_{i+k}),D}}{|D|} + \alpha_D \frac{cf_{#1(q_i, ..., q_{i+k})}}{|C|} \right] \]

\[ f_{U_D}(q_i, ..., q_j, D) = \log \left[ (1 - \alpha_D) \frac{tf_{#uwN(q_i, ..., q_j),D}}{|D|} + \alpha_D \frac{cf_{#uwN(q_i, ..., q_j)}}{|C|} \right] \]

All other features set to 0 (potential function = 1)
### Ad Hoc Retrieval Results

**Mean Average Precision**

<table>
<thead>
<tr>
<th>Collection</th>
<th>FI</th>
<th>SD</th>
<th>FD</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>0.1775</td>
<td>0.1867*  (+5.2%)</td>
<td>0.1866*  (+5.1%)</td>
</tr>
<tr>
<td>WSJ</td>
<td>0.2592</td>
<td>0.2776† (+7.1%)</td>
<td>0.2738*  (+5.6%)</td>
</tr>
<tr>
<td>WT10g</td>
<td>0.2032</td>
<td>0.2167*  (+6.6%)</td>
<td>0.2231** (+9.8%)</td>
</tr>
<tr>
<td>GOV2</td>
<td>0.2502</td>
<td>0.2832*  (+13.2%)</td>
<td>0.2844*  (+13.7%)</td>
</tr>
</tbody>
</table>

**Precision @ 10**

<table>
<thead>
<tr>
<th>Collection</th>
<th>FI</th>
<th>SD</th>
<th>FD</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>0.2912</td>
<td>0.2980  (+2.3%)</td>
<td>0.3068*  (+5.4%)</td>
</tr>
<tr>
<td>WSJ</td>
<td>0.4327</td>
<td>0.4427  (+2.3%)</td>
<td>0.4413   (+2.0%)</td>
</tr>
<tr>
<td>WT10g</td>
<td>0.2866</td>
<td>0.2948  (+2.9%)</td>
<td>0.3031   (+5.8%)</td>
</tr>
<tr>
<td>GOV2</td>
<td>0.4837</td>
<td>0.5714* (+18.1%)</td>
<td>0.5837*  (+20.7%)</td>
</tr>
</tbody>
</table>

* stat. sig. better than FI
** stat. sig. over SD and FI
† stat. sig. over FI and FD
### Ad Hoc Retrieval Results

**Mean Average Precision**

<table>
<thead>
<tr>
<th></th>
<th>Unigram</th>
<th>Bigram</th>
<th>MRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>0.1824</td>
<td>0.1857 (+1.8%)</td>
<td>0.1896* (+3.9%)</td>
</tr>
<tr>
<td>WSJ</td>
<td>0.2741</td>
<td>0.2773 (+1.2%)</td>
<td>0.2857** (+4.2%)</td>
</tr>
<tr>
<td>ROBUST04</td>
<td>0.2512</td>
<td>0.2549 (+1.5%)</td>
<td>0.2616** (+4.1%)</td>
</tr>
<tr>
<td>WT10G</td>
<td>0.2008</td>
<td>0.2194* (+9.3%)</td>
<td>0.2238* (+11.5%)</td>
</tr>
<tr>
<td>GOV2</td>
<td>0.2957</td>
<td>-</td>
<td>0.3228* (+9.2%)</td>
</tr>
</tbody>
</table>

* = significant over unigram  
** = significant over unigram and bigram  
(one tailed, paired t-test w/ p < 0.05)
Application: Web Search

- Three flavors of web search
  - Content-based search (ad hoc retrieval)
  - Homepage search
  - Named page finding

- Typically only one document on entire web is relevant for known-item search queries
- Use document structure (HTML tags) and other features such as PageRank
Named-Page Finding Results

- **Textual features**
  - Multinomial features for each field (title, anchor, heading)
- **Non-textual features**
  - PageRank
  - Inlink count
- Results for 272 queries over GOV2 collection

<table>
<thead>
<tr>
<th>Variant</th>
<th>MRR</th>
<th>S@10</th>
<th>Not Found</th>
</tr>
</thead>
<tbody>
<tr>
<td>FI</td>
<td>0.414</td>
<td>0.563</td>
<td>0.175</td>
</tr>
<tr>
<td>FD</td>
<td>0.441</td>
<td>0.583</td>
<td>0.171</td>
</tr>
<tr>
<td>Best TREC</td>
<td>0.463</td>
<td>0.595</td>
<td>0.179</td>
</tr>
</tbody>
</table>
Conclusions

- Markov random fields are robust models for information retrieval

- Benefits:
  - Easily model term dependencies
  - Ability to incorporate textual and non-textual features into the model
  - Query expansion framework
  - State of the art retrieval effectiveness
Questions?

Contact:
http://ciir.cs.umass.edu/~metzler
metzler@cs.umass.edu