Beyond Bags of Words
Effectively Modeling Dependence and Features in Information Retrieval

Donald Metzler
University of Massachusetts Amherst

Outline

• Introduction
• MRF Model
• Parameter Estimation
• Evaluation
• Automatic Feature Selection
• Latent Concept Expansion
• Conclusions and Future Work

The Problem

• Bag of words models are overly simplistic
• Previous attempts to go beyond bag of words models have failed to significantly and consistently improve effectiveness
• How more complex features and term dependencies be incorporated into a robust, highly effective retrieval model?
The Historical Perspective

Classical Probabilistic Model
Reference Network Models
Language Modeling
Linear Feature-Based Models

Binary
Independence
Tree

Poisson
BM25
Inference
Network
Model
Indri
Language Modeling
Multinomial
Unigram
Dependence

Paradigm Shifts

Contributions

• Robust retrieval model
  – Formally motivated
  – Handles term dependencies and the combination of arbitrary features
• Better understanding of features for information retrieval
  – Phrases and term proximity
  – Document priors
  – Large collection effects
• Parameter estimation insights
  – Formal look at parameter space of linear feature-based retrieval models
  – Direct maximization techniques
• Automatic model learning
  – Supervised feature selection algorithm
  – Learns highly effective models
• Concept-based query expansion
  – Expansion using dependencies and arbitrary features
  – Multi-term concept query expansion and generation
• State of the art retrieval effectiveness
  – Consistent and significant improvements on ad hoc retrieval and web search tasks
Outline

- Introduction
- MRF Model
- Parameter Estimation
- Evaluation
- Automatic Feature Selection
- Latent Concept Expansion
- Conclusions and Future Work

The Event Space

- Event space is $U \times Q \times D$
  - $U$ – User representation
  - $Q$ – Query representation
  - $D$ – Document representation
- Ignore time, other factors
- Define random variable $R$ (relevance) that is a deterministic function of $U, Q, D$
  - Relevance is typically treated as binary

Probability Ranking Principle

- Under the PRP, documents should be ranked according to:
  $$P(R = 1 | U, Q, D) = \frac{P(U, Q, D | R = 1)P(R = 1)}{P(U, Q, D)}$$
- However, users are difficult to model, so we explain them away and actually model:
  $$P(R = 1 | Q, D) \approx \frac{\max P(U, Q, D | R = 1)}{P(U, Q, D)}$$
- How do we model $P(Q, D | R=1)$?

Markov Random Fields

- MRFs provide a general, robust way of modeling a joint distribution
- 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’
  - Joint probability mass function
    - $P_{G,\lambda}(X_1, \ldots, X_n) = \prod_{c \in C(G)} \psi(c; \lambda)$
MRFs for Information Retrieval

(A) Three ways to model P(Q,D)
   - Option (A) is too coarse
   - Option (C) is too fine
   - Option (B) is our best choice

Building MRFs

• Build MRFs bottom-up, feature by feature
• Three steps:
  - Choose graph structure
  - Choose which cliques to apply feature to
  - Choose which feature to apply
• Canonical form:
  \[ (\text{dependence model type}, \text{clique set type}, \text{weighting function})_1 : \lambda_1 \]
  \[ (\text{dependence model type}, \text{clique set type}, \text{weighting function})_2 : \lambda_2 \]
  \[ \ldots \]
  \[ (\text{dependence model type}, \text{clique set type}, \text{weighting function})_n : \lambda_n \]

Dependence Model Type

Three generalized dependence model structures: full independence (left), sequential dependence (middle), and full dependence (right)

Clique Sets

Example clique sets for the query \( q_1, q_2, q_3 \) under full dependence model.
Weighting Functions for $T_{QD}$, $O_{QD}$, and $U_{QD}$

Summary of language modeling (LM) and BM25 weighting functions for the $T_{QD}$, $O_{QD}$, and $U_{QD}$ clique sets. Both $M$ and $N$ are parameters that control how matching is done.

Weighting Functions for $T_{QI}$, $O_{QI}$, and $U_{QI}$

Summary of inverse collection frequency (ICF) and inverse document frequency (IDF) weighting functions for the $T_{QI}$, $O_{QI}$, and $U_{QI}$ clique sets. Both $M$ and $N$ are parameters that control how matching is done.

Example

Query: new york city

Example

Query: new york city
Example
Query: new york city

\( \psi(\text{new, york}, D) = \exp[\lambda_1 f_{LM,OS}(\text{new, york}, D)] \)
\( \psi(\text{york, city}, D) = \exp[0] \)

Example
Query: new york city

\( \psi(\text{new, york}, D) = \exp[\lambda_1 f_{LM,OS}(\text{new, york}, D)] \)
\( \psi(\text{york, city}, D) = \exp[\lambda_1 f_{LM,OS}(\text{york, city}, D)] \)
Example

Query: new york city

Ranking

- As we showed before, ranking according to $P(Q, D|R=1)$ satisfies the PRP.
- After taking the logarithm and dropping all document independent terms, we get the following ranking function:

$$P_{LM}(Q, D) = \frac{1}{g^k} \exp\left( \lambda_1 f_{LM,Q}(new, york, D) + \lambda_2 f_{LM,Q}(york, city, D) + \lambda_3 f_{LM}(D) \right)$$

- Weighted combination of functions defined over different cliques.
Outline

- Introduction
- MRF Model
- Parameter Estimation
- Evaluation
- Automatic Feature Selection
- Latent Concept Expansion
- Conclusions and Future Work

Linear Feature-Based Models

- Family of ranking functions known as linear feature-based models:
  \[ S = (S_\lambda (D; Q)) : \exists \lambda \text{ s.t. } l \text{ is strictly monotonically increasing and } \lambda \text{ produces the same ranking} \]
  \[ l(S_\lambda (D; Q)) = \lambda^T f(D, Q) + Z \]
- Includes several previous retrieval models and the MRF model when exponential potentials are used
- Ranking functions in this family have special properties

Properties of the Parameter Space

- Parameter space is typically treated as \( \mathbb{R}^d \)
- What about rank equivalence?
- If we assume that \( \lambda_i \geq 0 \) for all \( i \), then it can be shown that the parameter space can be represented efficiently as \( \mathbb{P}^{d-1} \), the simplex over \( d \) outcomes
- Realistic assumption, since most features in IR provide “positive” evidence
Parameter Estimation

- Estimation problem:
  \[ \hat{\lambda} = \arg \max_{\lambda} E(\mathcal{R}_{\lambda}; T) \]
  s.t. \[ \mathcal{R}_{\lambda} \sim A^T f(D, Q) + Z \]
  \( \Lambda \in \mathcal{M}_\Lambda \)

- \( M_{\Lambda} \) can be either \( R^d \) or \( P^{d-1} \)
- Find the set of parameters that ranks documents in such a way as to maximize the underlying metric

Metric Divergence

- What is it?
  - When model parameters are tuned to maximize some metric that does not agree well with the actual metric under consideration [Morgan et al. '04]
  - Especially problematic in IR because of the large number of evaluation metrics
- Examples
  - Logistic regression [Gey '94]
  - Support vector machines [Nallapati '04]

Direct Optimization

- Grid Search
  - Unbounded, infinite computation, over \( R^d \)
  - Bounded, exponential computation, over \( P^{d-1} \)
  - Guaranteed to find global optimum
- Coordinate Ascent/Descent
  - Little difference when optimizing over \( R^d \) or \( P^{d-1} \)
  - Use single dimensional line search or finite difference derivatives for ascent/descent
  - Not guaranteed to find global optimum

Optimizing Surrogate Functions

- Perceptron Learning [Gao et al. '05]
- RankNet [Burges et al. '05]
- SVM-based Optimization
  - Precision at k [Joachims '05]
  - nDCG [Le and Smola '07]
  - Mean average precision [Yue et al. '07]
### Parameter Estimation Summary

<table>
<thead>
<tr>
<th>Direct Optimization</th>
<th>Surrogate Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>YES</td>
<td>YES/NO</td>
</tr>
<tr>
<td>YES/NO</td>
<td>YES/NO</td>
</tr>
<tr>
<td>YES/NO</td>
<td>YES/NO</td>
</tr>
</tbody>
</table>

### Outline

- **Introduction**
- **MRF Model**
- **Parameter Estimation**
- **Evaluation**
  - **Automatic Feature Selection**
  - **Latent Concept Expansion**
- **Conclusions and Future Work**

### Basic MRF Models for Ad Hoc Retrieval

- **Ad hoc retrieval**
  - Find all documents that are topically relevant to a given query
  - Typically many documents relevant per query
- **Basic MRF models**
  - Hand built MRF models
  - Uses bag of words and term proximity features inspired by previous research
  - Used to show usefulness of MRF framework

### Full Independence Model (MRF-FI)

- Uses simple bag of words features
- Rank equivalent to unigram language modeling

\[
P_{G,A}(D|Q) \overset{\text{rank}}{=} \sum_{(q,d) \in T_{G,A}} \log \frac{f_{T_{G,A}}(q,d) + \mu^A T_{G,A}}{|D| + \mu^A}
\]

**Canonical Form:**
- \((F I, T_{G,A}, LM) : \lambda_{T_D}
- \((FI, T_D, ICF) : \lambda_{T_D} \)**
Sequential Dependence Model (MRF-SD)

- Pair wise dependencies
- Three feature types
  - Single terms
  - Ordered window
  - Unordered window

$$P_{C \rightarrow D}(Q) = \sum_{(\theta, \omega) \in Q} \lambda_{\theta \omega} \prod_{(i, j) \in \theta} \log \frac{f_{\theta, \omega}^{(i,j)} + \mu^{(\theta, \omega)}}{[D_i + \mu^{(\theta)}][D_j + \mu^{(\omega)}]} +$$

$$\lambda_{\omega} \sum_{(i, j) \in \omega} \log \frac{f_{\omega}^{(i,j)} + \mu^{(\omega)}}{D_i + \mu^{(\omega)}} +$$

$$\lambda_{\theta} \sum_{(i, j) \in \theta} \log \frac{f_{\theta}^{(i,j)} + \mu^{(\theta)}}{D_i + \mu^{(\theta)}}$$

Full Dependence Model (MRF-FD)

- All dependencies
- Three feature types
  - Single terms
  - Ordered window
  - Unordered window

$$P_{C \rightarrow D}(Q) = \sum_{(\theta, \omega) \in Q} \lambda_{\theta \omega} \prod_{(i, j) \in \theta} \log \frac{f_{\theta, \omega}^{(i,j)} + \mu^{(\theta, \omega)}}{[D_i + \mu^{(\theta)}][D_j + \mu^{(\omega)}]} +$$

$$\lambda_{\omega} \sum_{(i, j) \in \omega} \log \frac{f_{\omega}^{(i,j)} + \mu^{(\omega)}}{D_i + \mu^{(\omega)}} +$$

$$\lambda_{\theta} \sum_{(i, j) \in \theta} \log \frac{f_{\theta}^{(i,j)} + \mu^{(\theta)}}{D_i + \mu^{(\theta)}}$$

Ad Hoc Retrieval Results

Test set mean average precision. The † and ‡ indicate statistically significant improvements over MRF-FI and MRF-SD, respectively.

<table>
<thead>
<tr>
<th></th>
<th>MRF-FI</th>
<th>MRF-SD</th>
<th>MRF-FD</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>0.2077</td>
<td>0.2147</td>
<td>0.2128</td>
</tr>
<tr>
<td>WSJ</td>
<td>0.3258</td>
<td>0.3425</td>
<td>0.3429</td>
</tr>
<tr>
<td>ROBUST04</td>
<td>0.2500</td>
<td>0.3096</td>
<td>0.3062</td>
</tr>
<tr>
<td>WT10g</td>
<td>0.1861</td>
<td>0.2053</td>
<td>0.2149</td>
</tr>
<tr>
<td>GOV2</td>
<td>0.2894</td>
<td>0.3325</td>
<td>0.3360</td>
</tr>
</tbody>
</table>

Collection Size

Relationship between the number of documents in a collection and the relative improvement in mean average precision of the MRF-FD model over unigram language modeling. Note that the x-axis is log scaled.
Using BM25 Weighting Functions

- Same as MRF-SD model, except uses BM25 weighting functions

\[ P_{BM25}(D|Q) = \lambda_D \sum_{(D,F) \in D,F} \frac{(1 + r_f) r_d}{1 + \sum_{(D',F) \neq (D,F)} r_{D'} r_{D,F}} \]  

Canonical Form:

- MRF
- BM25

Test set results for the MRF-BM25 model. The †, ‡, and * indicate statistically significant improvements over the MRF-BM25 and MRF-SD models, respectively.

- Bigram Language Model vs. MRF-SD Model

Test set results for the bigram language model. The † indicates a statistically significant improvement over the MRF-BM25 model and the ↓ indicates a statistically significant decrease in effectiveness compared to the MRF-SD model.

Summary of Other Ad Hoc Retrieval Results

- Smoothing
- The Role of Features
- Long Queries
- Generalization
Web Search

- Three types of Web queries
  - Content-based (ad hoc retrieval)
    - Many relevant documents
    - Document structure (HTML tags) and other features, such as PageRank, often do not help
  - Navigational
    - Typically only one document on entire web is relevant for known-item search queries
    - Use document structure, PageRank, etc. do help
  - Transactional
    - Difficult to evaluate

An MRF Model for Navigational Web Search

Outline

- Introduction
- MRF Model
- Parameter Estimation
- Evaluation
- Automatic Feature Selection
- Latent Concept Expansion
- Conclusions and Future Work

<table>
<thead>
<tr>
<th>LM-Mixture</th>
<th>MRR</th>
<th>S910</th>
<th>Not Found</th>
<th>MRR</th>
<th>S910</th>
<th>Not Found</th>
</tr>
</thead>
<tbody>
<tr>
<td>TREC 2005</td>
<td>0.414</td>
<td>0.563</td>
<td>0.175</td>
<td>0.441</td>
<td>0.583</td>
<td>0.171</td>
</tr>
<tr>
<td>TREC 2006</td>
<td>0.472</td>
<td>0.657</td>
<td>0.133</td>
<td>0.512</td>
<td>0.696</td>
<td>0.138</td>
</tr>
</tbody>
</table>

Web Search Results

Canonical form for the MRF-NP model and summary of web search results.
The Need for Feature Selection

- Basic MRF models used manually chosen features
- New tasks and data sets often require the use of different features
- Infeasible to manually select best feature set for all new data set and tasks
- Other advantages
  - Reduce number of unnecessary/redundant features
  - Limit overfitting
  - Increase training speed

Related Work

- Feature selection/induction for machine learning
  - Markov random fields [Della Pietra et al. ‘97]
  - Conditional random fields [McCallum ‘03]
- Feature selection/induction for information retrieval
  - Genetic algorithms [Fan et al. ‘04]
  - Result fusion [Fox and Shaw ‘03]

Feature Selection for Linear Feature-Based Models

- Proposed algorithm:
  - Input: pool of features
  - Begin with an empty model
  - Until some stopping criteria is met,
    - For each feature in the pool,
      - Temporarily add feature to the current model
      - Holding all other parameters fixed, train the augmented model to maximize/minimize some metric
      - Score feature according to how much it improved underlying metric
    - Add best feature to the model and remove from feature pool
    - Retrain the entire model (optional)

Feature Selection Results

- Training vs. No Retraining
- Number of Features
- Automatic vs. Manual Feature Selection
Training vs. No Retraining

Training and test set mean average precision values for retraining and no retraining.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No Retrain Train</th>
<th>No Retrain Test</th>
<th>Retrain Train</th>
<th>Retrain Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>0.1864</td>
<td>0.2206</td>
<td>0.1865</td>
<td>0.2216</td>
</tr>
<tr>
<td>WSJ</td>
<td>0.2706</td>
<td>0.3553</td>
<td>0.2703</td>
<td>0.3543</td>
</tr>
<tr>
<td>ROBUST04</td>
<td>0.2587</td>
<td>0.3079</td>
<td>0.2391</td>
<td>0.3065</td>
</tr>
<tr>
<td>WT10G</td>
<td>0.2344</td>
<td>0.2129</td>
<td>0.2357</td>
<td>0.2140</td>
</tr>
</tbody>
</table>

No significant change in training or test set effectiveness.
There is little evidence that retraining is useful.

Feature Selection Analysis

The order that features were selected, and their respective weights, for the WSJ and WT10G data sets.

WSJ
- (FI, T_{QD}, BM25) : 0.5864
- (SD, U_{QD}, BM25-U-1) : 0.3746
- (FD, U_{QD}, BM25-U-32) : 0.0193
- (FI, T_{QD}, LM) : 0.0196
- (FD, U_{QD}, BM25-U-unlimited) : 0.0001

WT10G
- (FI, T_{QD}, BM25) : 0.8138
- (FD, U_{QD}, LM-U-8) : 0.0001
- (SD, U_{QD}, BM25-U-unlimited) : 0.0090
- (FD, U_{QD}, BM25-U-8) : 0.1575
- (SD, O_{QD}, BM25-O-8) : 0.0196

Automatic vs. Manual Feature Selection

Comparison of test set mean average precision for language modeling (MRF-FI), BM25, MRF model using language modeling weighting (MRF-FD), MRF model using BM25 weighting (MRF-BM25), and MRF learned using our proposed feature selection algorithm (MRF-FS).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MRF-FI</th>
<th>BM25</th>
<th>MRF-FD</th>
<th>MRF-BM25</th>
<th>MRF-FS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>0.2077</td>
<td>0.2149</td>
<td>0.2147</td>
<td>0.2210</td>
<td>0.2266</td>
</tr>
<tr>
<td>WSJ</td>
<td>0.3258</td>
<td>0.3332</td>
<td>0.3425</td>
<td>0.3512</td>
<td>0.3553</td>
</tr>
<tr>
<td>ROBUST04</td>
<td>0.2920</td>
<td>0.2892</td>
<td>0.3086</td>
<td>0.3101</td>
<td>0.3079</td>
</tr>
<tr>
<td>WT10G</td>
<td>0.1864</td>
<td>0.1948</td>
<td>0.2053</td>
<td>0.2129</td>
<td>0.2129</td>
</tr>
<tr>
<td>GOV2</td>
<td>0.2984</td>
<td>0.2971</td>
<td>0.3325</td>
<td>0.3476</td>
<td>0.3398</td>
</tr>
</tbody>
</table>

MRF-FS > all other models for AP, WSJ
MRF-FS > MRF-FI and BM25 for ROBUST04, WT10G, and GOV2
Outline

• Introduction
• MRF Model
• Parameter Estimation
• Evaluation
• Automatic Feature Selection
• Latent Concept Expansion
• Conclusions and Future Work

Lavrenko’s Relevance Models

• A query is an inaccurate representation of a true information need
• Relevance models build richer query representations by mixing term probabilities computed over top ranked documents

\[
P(w|Q) = \frac{\sum_{D \in Q} P(w,Q,D)}{\sum_{D \in Q} P(w,Q,D)}
\]

\[
\approx \frac{\sum_{D \in Q} P(w|D) \prod_{q \in Q} P(q|D)}{\sum_{D \in Q} \prod_{q \in Q} P(q|D)}
\]

\[
\propto \sum_{D \in Q} \exp \left[ \log P(w|D) + \sum_{q \in Q} \log P(q|D) \right]
\]

Related Query Expansion Work

• Expansion with Single Terms
  – Rocchio (Rocchio ’71)
  – Model-Based Feedback (Zhai and Lafferty ’02)
  – Relevance Models (Lavrenko and Croft ’01)
• Expansion with Multi-Term Concepts
  – Local Context Analysis (Lavrenko and Croft ’00)
• Expansion with Term Dependence
  – Very little work (Harper and van Rijsbergen ’78)
• Expansion with Arbitrary Features
  – Nothing (?)

Lavrenko’s Relevance Models

• Pros
  — Relatively straightforward to implement
  — State of the art retrieval effectiveness
• Cons
  — Makes unnecessary distributional assumptions (e.g., multinomial LM)
  — No notion of term dependence
  — No easy way to incorporate arbitrary features
Latent Concept Expansion: Improving upon Relevance Models

• Proposed approach
  — Replace unigram language model with MRF model

• Pros
  — Less strict distributional assumptions
  — Allows modeling of term dependencies
  — Allows arbitrary textual and non-textual features

• Cons
  — Slightly more complex to implement, depending on features used

• Effectiveness
  — Will modeling term dependence help?
  — Will more complex features help?

Latent Concept Expansion

• Assumption
  — There exists some set of latent concepts \( \{E_1 \ldots E_k\} \) that the user had in mind when formulating the query

• Goal is to find these latent concepts

• Uses
  — Generating concepts
  — Query expansion (relevance / pseudo-relevance feedback)

Latent Concept Expansion

Step 1: Build Expansion MRF

Original MRF (\( G \))

Expansion MRF (\( H \))
**Latent Concept Expansion**

**Step 2: Find k Most Likely Concepts**

Score concepts according to:

\[ P_{H,A}(w_1, w_2 | Q = \text{train, station, security}) \]

**Computing Concept Likelihood**

- Same idea as Relevance Models, except we use a more complex joint distribution.
- Only sum over known-relevant or pseudo-relevant documents.
- No explicit assumption that \( P(D) \) is uniform.

\[
P(c|Q) = \frac{\sum_{D \in R_Q} P(c, Q, D)}{\sum_{D \in R_Q} P(c, Q, D)} \\
\approx \frac{\sum_{D \in R_Q} P(c, Q, D)}{\sum_{D \in R_Q} P(c, Q, D)}
\]

**Latent Concept Expansion**

**Step 3: Construct Augmented MRF**

Augmented MRF \((G')\)

**Step 4: Rerank using Augmented MRF**

Rank documents according to:

\[ P_{H,A}(\text{train, station, security, security, camera}, \ldots, \text{train, safety} | D) \]
Relationship to Relevance Models

\[ P_{R,M}(w|Q) \propto \sum_{D \in \mathcal{D}} \exp \left[ \lambda_{R} \sum_{w \in Q} \log \left( 1 - \alpha \frac{t_{w,D}}{|D|} + \alpha \frac{c_{w}}{|C|} \right) \right] + \lambda_{M} \sum_{w \in Q} \log \left( 1 - \beta \frac{t_{w,D}}{|D|} + \beta \frac{c_{w}}{|C|} \right) + \lambda_{O} \sum_{w \in Q} \log \left( \frac{1 - \alpha \frac{t_{w,D}}{|D|} + \alpha \frac{c_{w}}{|C|}}{\frac{1 - \beta \frac{t_{w,D}}{|D|} + \beta \frac{c_{w}}{|C|}}{\frac{|D|}{|C|}} \frac{1}{N_{Q}} \frac{N_{D}}{N_{Q}}} \right) \]

Relevance Model

\[ P_{R,M}(w|Q) \propto \sum_{D \in \mathcal{D}} \exp \left[ \lambda_{R} \sum_{w \in Q} \log \left( 1 - \alpha \frac{t_{w,D}}{|D|} + \alpha \frac{c_{w}}{|C|} \right) \right] + \lambda_{M} \sum_{w \in Q} \log \left( 1 - \beta \frac{t_{w,D}}{|D|} + \beta \frac{c_{w}}{|C|} \right) + \lambda_{O} \sum_{w \in Q} \log \left( \frac{1 - \alpha \frac{t_{w,D}}{|D|} + \alpha \frac{c_{w}}{|C|}}{\frac{1 - \beta \frac{t_{w,D}}{|D|} + \beta \frac{c_{w}}{|C|}}{\frac{|D|}{|C|}} \frac{1}{N_{Q}} \frac{N_{D}}{N_{Q}}} \right) \]

Query Score

Term Score

Relationship to Relevance Models

\[ P_{R,M}(w|Q) \propto \sum_{D \in \mathcal{D}} \exp \left[ \lambda_{R} \sum_{w \in Q} \log \left( 1 - \alpha \frac{t_{w,D}}{|D|} + \alpha \frac{c_{w}}{|C|} \right) \right] + \lambda_{M} \sum_{w \in Q} \log \left( 1 - \beta \frac{t_{w,D}}{|D|} + \beta \frac{c_{w}}{|C|} \right) + \lambda_{O} \sum_{w \in Q} \log \left( \frac{1 - \alpha \frac{t_{w,D}}{|D|} + \alpha \frac{c_{w}}{|C|}}{\frac{1 - \beta \frac{t_{w,D}}{|D|} + \beta \frac{c_{w}}{|C|}}{\frac{|D|}{|C|}} \frac{1}{N_{Q}} \frac{N_{D}}{N_{Q}}} \right) \]

Relevance Model

\[ P_{R,M}(w|Q) \propto \sum_{D \in \mathcal{D}} \exp \left[ \lambda_{R} \sum_{w \in Q} \log \left( 1 - \alpha \frac{t_{w,D}}{|D|} + \alpha \frac{c_{w}}{|C|} \right) \right] + \lambda_{M} \sum_{w \in Q} \log \left( 1 - \beta \frac{t_{w,D}}{|D|} + \beta \frac{c_{w}}{|C|} \right) + \lambda_{O} \sum_{w \in Q} \log \left( \frac{1 - \alpha \frac{t_{w,D}}{|D|} + \alpha \frac{c_{w}}{|C|}}{\frac{1 - \beta \frac{t_{w,D}}{|D|} + \beta \frac{c_{w}}{|C|}}{\frac{|D|}{|C|}} \frac{1}{N_{Q}} \frac{N_{D}}{N_{Q}}} \right) \]

Ordered Window Score

Unordered Window Score

Expansion Term IDF

\[ P_{R,M}(w|Q) \propto \sum_{D \in \mathcal{D}} \exp \left[ \lambda_{R} \sum_{w \in Q} \log \left( 1 - \alpha \frac{t_{w,D}}{|D|} + \alpha \frac{c_{w}}{|C|} \right) \right] + \lambda_{M} \sum_{w \in Q} \log \left( 1 - \beta \frac{t_{w,D}}{|D|} + \beta \frac{c_{w}}{|C|} \right) + \lambda_{O} \sum_{w \in Q} \log \left( \frac{1 - \alpha \frac{t_{w,D}}{|D|} + \alpha \frac{c_{w}}{|C|}}{\frac{1 - \beta \frac{t_{w,D}}{|D|} + \beta \frac{c_{w}}{|C|}}{\frac{|D|}{|C|}} \frac{1}{N_{Q}} \frac{N_{D}}{N_{Q}}} \right) \]
Evaluation

• Generating concepts
  – Illustrative examples
• Query expansion
  – One term concepts
  – One and two term concepts
• Robustness

Concept Generation Example

<table>
<thead>
<tr>
<th>One term concepts</th>
<th>Two term concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>telescope</td>
<td>telescope</td>
</tr>
<tr>
<td>hubble</td>
<td>hubble</td>
</tr>
<tr>
<td>new</td>
<td>new</td>
</tr>
<tr>
<td>mirror</td>
<td>mirror</td>
</tr>
<tr>
<td>space</td>
<td>space</td>
</tr>
<tr>
<td>universe</td>
<td>universe</td>
</tr>
<tr>
<td>two</td>
<td>two</td>
</tr>
<tr>
<td>observatory</td>
<td>observatory</td>
</tr>
<tr>
<td>nano</td>
<td>nano</td>
</tr>
<tr>
<td>optical</td>
<td>optical</td>
</tr>
<tr>
<td>test</td>
<td>test</td>
</tr>
<tr>
<td>new</td>
<td>new</td>
</tr>
<tr>
<td>cost</td>
<td>cost</td>
</tr>
<tr>
<td>test</td>
<td>test</td>
</tr>
<tr>
<td>world</td>
<td>world</td>
</tr>
<tr>
<td>year</td>
<td>year</td>
</tr>
<tr>
<td>power</td>
<td>power</td>
</tr>
<tr>
<td>light</td>
<td>light</td>
</tr>
<tr>
<td>astronomy</td>
<td>astronomy</td>
</tr>
<tr>
<td>two</td>
<td>two</td>
</tr>
<tr>
<td>launch</td>
<td>launch</td>
</tr>
<tr>
<td>cost</td>
<td>cost</td>
</tr>
<tr>
<td>million</td>
<td>million</td>
</tr>
<tr>
<td>project</td>
<td>project</td>
</tr>
<tr>
<td>work</td>
<td>work</td>
</tr>
<tr>
<td>europe</td>
<td>europe</td>
</tr>
<tr>
<td>star</td>
<td>star</td>
</tr>
<tr>
<td>telescope</td>
<td>telescope</td>
</tr>
<tr>
<td>hubble</td>
<td>hubble</td>
</tr>
<tr>
<td>telescope</td>
<td>telescope</td>
</tr>
<tr>
<td>hubble</td>
<td>hubble</td>
</tr>
<tr>
<td>telescope</td>
<td>telescope</td>
</tr>
<tr>
<td>hubble</td>
<td>hubble</td>
</tr>
<tr>
<td>telescope</td>
<td>telescope</td>
</tr>
</tbody>
</table>

Pseudo-Relevance Feedback with One Term Concepts

<table>
<thead>
<tr>
<th></th>
<th>LM</th>
<th>MRF</th>
<th>RM3</th>
<th>LCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSJ</td>
<td>.3258</td>
<td>.4255</td>
<td>.3493</td>
<td>.3963</td>
</tr>
<tr>
<td>AP</td>
<td>.2077</td>
<td>.2147</td>
<td>.2518</td>
<td>.2692</td>
</tr>
<tr>
<td>ROBUST</td>
<td>.2920</td>
<td>.3066</td>
<td>.3382</td>
<td>.3601</td>
</tr>
<tr>
<td>WT10g</td>
<td>.1891</td>
<td>.2054</td>
<td>.1944</td>
<td>.2209</td>
</tr>
<tr>
<td>GOV2</td>
<td>.3234</td>
<td>.3529</td>
<td>.3656</td>
<td>.3924</td>
</tr>
</tbody>
</table>

Pseudo-Relevance Feedback with One and Two Term Concepts

• Expansion using both one and two term concepts yielded mixed results
• Analysis shows that two term concepts are highly redundant with single term concepts
  – Example: price fixing
  – Single terms: price, fixed, market, report, vote, ...
  – Two terms: fixed pricing, price fixing, market price, ...
• Different features may be necessary in order to choose more diverse / better set of two term concepts for expansion

NOTES
Queries: title only
Training: maximization of mean avg. precision on training set
ROBUST04

LCE Conclusions

- LCE is a novel technique that makes use of term dependencies and arbitrary features
- Term dependence + pseudo-relevance feedback = amplifying effect
- LCE provides robust, state of the art retrieval effectiveness, especially for large collections
- More work needed to better understand query expansion with multi-term concepts

Outline

- Introduction
- MRF Model
- Parameter Estimation
- Evaluation
- Automatic Feature Selection
- Latent Concept Expansion
- Conclusions and Future Work

Putting the MRF to Practice
Summary of Results

Test set mean average precision across a range of retrieval models. The model parameters were trained to maximize mean average precision. Bold values indicate the best model that does not make use of pseudo-relevance feedback.

<table>
<thead>
<tr>
<th>Model</th>
<th>AP</th>
<th>WSJ</th>
<th>ROBUST04</th>
<th>WT10G</th>
<th>GOV2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM</td>
<td>0.2077</td>
<td>0.3258</td>
<td>0.2920</td>
<td>0.1861</td>
<td>0.2094</td>
</tr>
<tr>
<td>BM25</td>
<td>0.2119</td>
<td>0.3332</td>
<td>0.2992</td>
<td>0.1948</td>
<td>0.2071</td>
</tr>
<tr>
<td>BIGRAM</td>
<td>0.2116</td>
<td>0.3319</td>
<td>0.3012</td>
<td><strong>0.2165</strong></td>
<td>0.3371</td>
</tr>
<tr>
<td>MRF-SD</td>
<td>0.2147</td>
<td>0.3325</td>
<td>0.3006</td>
<td>0.2053</td>
<td>0.3257</td>
</tr>
<tr>
<td>MRF-FD</td>
<td>0.2128</td>
<td>0.3290</td>
<td>0.3092</td>
<td>0.2140</td>
<td>0.3360</td>
</tr>
<tr>
<td>MRF-BM25</td>
<td>0.2210</td>
<td>0.3512</td>
<td><strong>0.3101</strong></td>
<td>0.2129</td>
<td><strong>0.3476</strong></td>
</tr>
<tr>
<td>MRF-FS</td>
<td><strong>0.2264</strong></td>
<td>0.3505</td>
<td>0.3079</td>
<td>0.2077</td>
<td>0.3398</td>
</tr>
<tr>
<td>RM3</td>
<td>0.2518</td>
<td>0.4493</td>
<td>0.3382</td>
<td>0.1944</td>
<td>0.3656</td>
</tr>
<tr>
<td>LCE</td>
<td>0.2092</td>
<td>0.3943</td>
<td>0.3601</td>
<td>0.2269</td>
<td>0.3024</td>
</tr>
</tbody>
</table>

Contributions

- Robust retrieval model
  - Formally motivated
  - Handles term dependencies and the combination of arbitrary features
- Better understanding of features for information retrieval
  - Phrases and term proximity
  - Document priors
  - Large collection effects
- Parameter estimation insights
  - Formal look at parameter space of linear feature-based retrieval models
  - Direct maximization techniques
- Automatic model learning
  - Supervised feature selection algorithm
  - Learns highly effective models
- Concept-based query expansion
  - Expansion using dependencies and arbitrary features
  - Multi-term concept query expansion and generation
- State of the art retrieval effectiveness
  - Consistent and significant improvements on ad hoc retrieval and web search tasks

Future Work

- Feature engineering
  - Can we go beyond phrase and proximity features?
  - What other document-dependent features can be useful?
- Better understanding of multi-term concept expansion
  - Can multi-term concept expansion be improved?
  - What makes a good multi-term expansion concept?
- Unified learning to rank framework
  - Can the best aspects of direct optimization and surrogate function optimization be combined?
- Efficient indexing architectures
  - What indexing strategies are best suited for these feature-based types of models?
  - How can these queries be evaluated efficiently?