Beyond Keywords: Finding Information More Accurately and Easily Using Natural Language

Matt Lease
mlease@cs.brown.edu

Brown Laboratory for Linguistic Information Processing (BLLIP)
Brown University

Center for Intelligent Information Retrieval (CIIR)
University of Massachusetts Amherst
What is the state of recognizing handwriting in today's computer systems?

1st relevant result: Computer Systems Manager Career Overview

DiplomaGuide.com

Job Titles and Careers List » Engineering Tech. » Computer Systems Manager Career Overview

Schools that fit your selection:

- University of Phoenix

Duties and Responsibilities of a Computer Systems Manager

A computer systems manager will be responsible for a wide range of activities, both technical and business related, involving the overall implementation of technology within the organization. Systems managers direct the work of other computer-related employees, similar equipment, which produce precision parts for the aerospace, automobile and machine industries. A computer control tech may start as a CNC machine operator, supervising several working machines at once. Higher on the career ladder is the CNC setup operator, who gets the machine ready at the beginning of a job by downloading the program, installing the correct cutting tools in their holders, positioning the workpiece (the block of material which will be cut) in the machine's workspace.
Searching off the Desktop

Longer and more natural queries emerge in spoken settings [Du and Crestani’06]
Verbosity and Complexity

- Complex information requires complex description
  - Information theory [Shannon’51]
  - Human discourse implicitly respects this [Grice’67]

- Simple searches easily expressed in keywords
  - navigation: “alaska airlines”
  - information: “american revolution”

- Verbosity naturally increases with complexity
  - More specific information needs [Phan et al.’07]
  - Iterative reformulation [Lau and Horvitz’99]
Outline of Talk

- Natural language queries: what, where & why?
- Term-based models for NL queries
  - Problem: query complexity → query ambiguity
- Regression Rank [Lease, Allan, and Croft, ECIR’09]
  - Learning framework independent of retrieval model
- Extensions
  - Modeling term relationships [Lease, SIGIR’09]
  - Relevance feedback: explicit and pseudo [Lease, TREC’08]
Term-Based Retrieval

Standard approaches

▶ Vector-similarity [Salton et al.’60s, Singhal et al.’96]
▶ Document-likelihood [Sparck Jones et al.’00]
▶ Query-likelihood [Ponte and Croft’98]
  ▪ KL-divergence variant [Lafferty and Zhai’01]

Roughly same features and accuracy [Fang et al.’04]

DL       QL under = parameterization [Lease, SIGIR’09]
KL-Divergence Ranking

- **Estimate a unigram** $£^D$ underlying each document
  - Length- & order-independent representation of topicality
  - Smoothing assigns non-zero probability to unseen terms

- $\text{KL}(£^Q || £^D) = ² \phi \log ² + C_Q$

**Key Idea**: $\text{weight}_{QD}(\phi)$ decomposed into $£^Q$ & $£^D$

- $£^D$ fixed for all queries (Dirichlet smoothing)
- $£^Q$ expresses importance of terms for a given query
Verbosity vs. Retrieval Accuracy

TREC Topic 838

Title: “urban suburban coyotes”

Description: “How have humans responded and how should they respond to the appearance of coyotes in urban and suburban areas?”
Verbosity vs. Retrieval Accuracy

TREC Topic 838

**Title**: “urban suburban coyotes”

**Description**: “How have humans responded and how should they respond to the appearance of coyotes in urban and suburban areas?”

---

Average Precision example:

\[
AP = \frac{1/1 + 2/2 + 3/5}{3}
\]
RIA Workshop [Buckley and Harman’04]

- 10-40 hours error analysis per-query, 45 Description queries
- Models failed to emphasize the right terms for ¼ 2/3 queries

Mean Average Precision (MAP): per-query AP averaged across queries
Problem: Query Ambiguity

ML assumes all query tokens equally important to $Q_i$!

- The core information is often obscured
- Details distract rather than inform
Example: Better Estimate £^Q

More important terms should be assigned greater weight in £^Q

How to estimate £^Q??

Matt Lease <mlease@cs.brown.edu>
Outline of Talk

▶ Natural language queries: what, where & why?

▶ Term-based models for NL queries
  - Problem: query complexity → query ambiguity
  - Regression Rank [Lease, Allan, and Croft, ECIR’09]
    - Learning framework independent of retrieval model
  - Extensions
    - Modeling term relationships [Lease, SIGIR’09]
    - Relevance feedback: explicit and pseudo [Lease, TREC’08]
Supervised Learning of £^Q$

► **Training data: document relevance**
  - Known relevance: documents manually assessed
  - Inferred relevance: query log “click-through” data

► **Potential benefits**
  - Data-driven: let examples guide estimation
  - Lifetime learning: continually improve with more data
  - Expressiveness: keep terms, replace estimation

► **Challenge: sparsity**
  - One parameter per vocabulary term [cf. Mei et al.’07]
  - Existing *Learning To Rank* methods don’t address this
Regression Rank [Lease et al.’09]

**Idea:** Predict £^Q_ using fewer parameters

- Find features correlated with £^Q_ (term importance)
- Predict £^Q_ from these features

![Graph showing the relationship between Log(DF) and £^Q_]

<table>
<thead>
<tr>
<th>Log(DF)</th>
<th>£^Q_</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.13</td>
<td>0.03</td>
</tr>
<tr>
<td>3.48</td>
<td>0.3</td>
</tr>
<tr>
<td>3.83</td>
<td>0.11</td>
</tr>
<tr>
<td>3.73</td>
<td>0.16</td>
</tr>
<tr>
<td>3.23</td>
<td>0.40</td>
</tr>
</tbody>
</table>
Training

Input Query: $Q_n$

Feature Extraction

Features $F = \{f_1, f_2, f_3\}$

Regression Prediction

$\alpha \cdot F = \ell^Q$

Predicted $\ell^Q$

Run-time

Training Examples

Estimate "gold" $\ell^Q$’s

"gold" $\ell^Q$’s

Feature Extraction

Feature Weights $\alpha = \{\alpha_1, \alpha_2, \alpha_3\}$

Regression Training
Regression Rank: Estimation

Goal: optimize £Q for rank-based metric (e.g. AP)

- Challenge: non-differentiable, non-convex
- Simpler metrics to optimize, but diverge from goal

Grid search (sampling)

- [cf. Metzler and Croft’05]
- Embarrassingly parallel
- Exponential # samples
Query: [human suburban urban]

<table>
<thead>
<tr>
<th>Sub-query</th>
<th>ML $\mathbf{f}_Q$</th>
<th>AP($\mathbf{f}_Q$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_1$: human</td>
<td>$[1, 0, 0]$</td>
<td>0.3859</td>
</tr>
<tr>
<td>$Q_2$: suburban</td>
<td>$[0, 1, 0]$</td>
<td>0.2992</td>
</tr>
<tr>
<td>$Q_3$: urban</td>
<td>$[0, 0, 1]$</td>
<td>0.4897</td>
</tr>
</tbody>
</table>
Feature Extraction: define features correlated with term importance

Training Examples

Feature Extraction
F = \{f_1, f_2, f_3\}

Training
Regression Rank: Features

Features

- Traditional IR statistics: e.g. term frequency, document frequency
  - source: document collection & large external corpora
- Position: integer index of term in query
- Lexical context: Preceding/following terms and punctuation
- Syntactic part-of-speech: e.g. is term a noun / verb / other?

Feature normalization: set mean=0 & standard deviation=1

Feature selection: prune features occurring <12 times
**Estimation**

Estimation

**Feature Extraction**

Feature Extraction

**Regression**

Regression

---

**Training**

Training Examples

Estimate “gold” £^Q^’s

“gold” £^Q^’s

**Features**

F = \{f_1, f_2, f_3\}

---

**Regression Training**

Feature Weights

\( \theta = \{\theta_1, \theta_2, \theta_3\} \)

---

Estimate "gold" £^Q^’s

---

Training Examples

---

Matt Lease <mlease@cs.brown.edu>
Regression Rank: Ridge regression (L2 regularization of least-squares)

- Consistently better than ML, Lasso (L1), and others
- Metric divergence (squared-loss vs. AP)
Regression Rank: Strengths

- Learning framework is independent of retrieval model
  - e.g. Predict weights for term-interactions rather than terms
  - Similar to Probabilistic Indexing
    - Fuhr and Buckley ‘91

- Can learn context-dependent term weights
  - Model richer context than just query length

- Together: query-specific LTR
  - e.g. Dynamically-weighted mixture model
    - Geng et al. ‘08
Key Concepts [Bendersky and Croft’08]

- Annotate “key” NP for each query, train a classifier
- Weight NPs by classifier confidence, and mix with ML £^Q

![Graphs showing Precision@5 and Mean Average Precision for different datasets: Robust04, W10g, GOV2, comparing Description and Key Concepts.](image-url)
Regression Rank: Results

Fully-predicts all parameters (no mixing/tying)
Can optimize model accuracy for any metric
Lifetime learning from query log

![Bar charts showing Precision@5 and Mean Average Precision for Robust04, W10g, and GOV2 categories. The results indicate that Regression Rank outperforms Description and Key Concepts across all categories.]
Example: Predicted £$^Q$

TREC Topic 838

How have humans responded and how should they respond to the appearance of coyotes in urban and suburban areas?

Example: Predicted £$^Q$
Room for Further Improvement

- Expectation below restricted to query vocabulary
  - Expand vocabulary: feedback documents
  - Model more than terms: e.g. term-interactions

![Mean Average Precision Chart]

Matt Lease <mlease@cs.brown.edu>
Outline of Talk

- Natural language queries: what, where & why?
- Term-based models for NL queries
  - Problem: query complexity $\rightarrow$ query ambiguity
- Regression Rank [Lease, Allan, and Croft, ECIR’09]
  - Learning framework independent of retrieval model
- Extensions
  - Modeling term relationships [Lease, SIGIR’09]
  - Relevance feedback: explicit and pseudo [Lease, TREC’08]
Sequential Dependency Model

- [Metzler and Croft’05]
  - Simple, efficient, & consistently beats unigram
- Consecutive query terms are scored 3 ways
  - Individual occurrence: **unigram**
  - Co-occurrence: **adjacency** (ordered) & **proximity**
- Example

*What research is ongoing for new **fuel sources**?*

Document = “fuel source fuel source”

- **unigram**
  - ![unigram](image)
- **adjacency**
  - ![adjacency](image)
- **proximity**
  - ![proximity](image)
Better Estimation of SD Unigram

- Estimate SD Unigram by Regression Rank
  - Adjacency and Proximity still use ML
  - Consistent improvement [Lease, SIGIR’09]
What research is ongoing for new fuel sources?

{research, ongoing} {ongoing, new} {new, fuel} {fuel, sources}
Oracle Experiment [Lease, SIGIR’09]

- Rank dependencies by expected weight
- Successively add them in rank order
Next: Estimate Dependency Weights

- Apply current features like TF/IDF
- Add new term relationship features
  - Syntax, collocations, named-entities, etc.
Outline of Talk

- Natural language queries: what, where & why?
- Term-based models for NL queries
  - Problem: query complexity → query ambiguity
- Regression Rank [Lease, Allan, and Croft, ECIR’09]
  - Learning framework independent of retrieval model
- Extensions
  - Modeling term relationships [Lease, SIGIR’09]
  - Relevance feedback: explicit and pseudo [Lease, TREC’08]
Relevance Feedback (Explicit & Pseudo)

[Rochio’71, Lavrenko and Croft’01, Lafferty and Zhai’01]

**Idea:** Better estimate $\ell^Q$ using related documents
- Particularly valuable for finding other related terms

**Explicit:** Given examples of relevant documents
- Compute average $\ell^D$, mix with query $\ell^Q$

**Pseudo:** Blind expansion
- Score documents with $\ell^Q$
- Compute expected $\ell^D$, mix with query $\ell^Q$

**How can we apply supervised learning here?**
Preliminaries: TREC’08 RF Track

- Varied feedback: none (ad-hoc) to many documents
- Approach: RF + PRF + Sequential Term Dependencies
- Best results in track [Lease’08] (GOV2)

TREC’08 RF Track: Average P@10

TREC’08 RF Track: Average MAP
Step 1: Supervised £^Q + PRF

- Are supervision and PRF complementary?
- Yes, and dependencies too! [Lease, SIGIR’09]

**Without PRF**

<table>
<thead>
<tr>
<th>Mean Average Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression Rank</td>
</tr>
</tbody>
</table>

**With PRF**

<table>
<thead>
<tr>
<th>Mean Average Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRank + PRF</td>
</tr>
</tbody>
</table>

Matt Lease <mlease@cs.brown.edu>
Outlook: Supervised RF/PRF

[Cao et al. ‘08]

- Standard PRF: only 17% terms help, **26-37%** hurt
- Classify terms as good/bad, weight by confidence
- Some details of approach can be improved

Future work: apply Regression Rank

- Feedback document(s) just more verbosity
- Apply better learning, more features
Summary

- **Natural language queries**: what, where & why?
- **Term-based models for NL queries**
  - Problem: query complexity → query ambiguity
- **Regression Rank** [Lease, Allan, and Croft, ECIR’09]
  - Learning framework independent of retrieval model
- **Extensions**
  - Modeling term relationships [Lease, SIGIR’09]
  - Relevance feedback: explicit and pseudo [Lease, TREC’08]