Predicting the Best Information Retrieval System Parameter Value: the Per Parameter Learning (PPL) Method

Prof. Josiane Mothe
Mahdi Washha

7-Sep-2017
Outline

• Introduction.
• Problem Statement and Motivation.
• State of Art Solutions.
• Per Parameter Learning Method.
• Experimental Setup.
• Performance Metrics.
• Experimental Results.
• Conclusion and Future Direction.
Introduction

• Information Retrieval (IR) System Components:
  – Indexing: collect, parse, and store data (collection of documents).
  – Retrieval: return list of documents for a query.

• Retrieval process parameters:
  – Stemmer, Matching model, Query expansion, ..., etc.
  – Each has a set of values.
  – Different combinations of systems (S=< P1; O3; …; Z2>).
  – Variation in results with respect to an effectiveness measure (e.g. Average Precision, Precision @10).
Problem Statement and Motivation

- **System variability problem**: S1 performs well on query Q1 but badly on query Q2, while system S2 performs the opposite.

- **Unseen queries problem**: predicting successfully the best system (S) configuration!!

- **Systems space limitation**: perform learning and prediction over a few systems (S1,S2,...,Sn).

- **Query features**: Pre-retrieval Vs Post-retrieval.
State of Art Solutions: Data Fusion

• **Idea**: utilizes system variability problem to combine outputs of different systems [1,2].

• **Combination algorithms**: Linear combination, Power series combination.

• **Drawbacks**: (i) retrieval process takes place many times; (ii) finding the best combination.
State of Art Solutions: Repeated Queries

• **Idea:** store a set of queries with their best systems configuration [3,4].

• **Concept:** the same concept of lookup tables.

• **Drawbacks:** (i) can't handle unseen (not stored) queries; (ii) not scalable; (iii) limited on specific retrieval domains.
State of Art Solutions: Selective Approach

• **Idea:** cluster queries first and then assign system for each system [5,6].

• **New query:** check to which cluster it belongs and then use the corresponding system.

• **Drawbacks:** (i) assigning system performs on average per cluster; (ii) System space limitation (assigned systems only)
Proposed Solution: Per-Parameter Learning (PPL) Method

• **Predict the best system configuration (S) for a query.**
  – Solves system variability problem.
  – Best system optimizes a performance measure (AP, P@10)

• **Learn each system parameter individually (instead of learning one classifier to predict S’s).**
  – Solves system space limitation.
  – Solves unseen queries problem.

• **Represent queries using hand designed features**
  – Solves query representation problem.
Per-Parameter Learning (PPL) Method: Training Phase

System: $S_k = \langle p_1, p_2, \ldots, p_j \rangle$

Training Queries

Q1

Pre-Retrieval and/or Post-Retrieval Features Extraction

Systems Evaluation ($S_1, S_2, \ldots, S_k$)

Best N Systems Selection

Combining Features and Best Systems

Feature Vector Space for system parameter $p_1$

Classification Model $F_1(x)$

Feature Vector Space for system parameter $p_j$

Classification Model $F_j(x)$

Predicting the Best Information Retrieval System Parameter Value: the Per Parameter Learning (PPL) method
Per-Parameter Learning (PPL) Method: Prediction Phase

\[ S_{\text{new}} = \langle F_1(x), \ldots, F_j(x) \rangle \]
Experimental Setup: System parameters and systems generation

- Seven parameters adopted in conducting experiments.
- 82,627 different system configurations generated.
- Terrier platform: recent implementation of indexing, retrieving, and query expansion methods.

<table>
<thead>
<tr>
<th>System Parameter Name</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stemmer</td>
<td>TR Porter Stemmer (TRPS), French Snowball Stemmer (FSS), TR Weak Porter Stemmer (TRWPS), Crop Term (CT), Porter Stemmer (PS), Weak Porter Stemmer (WPS), English Snowball Stemmer (ESS)</td>
</tr>
<tr>
<td>Parameter Free Expansion</td>
<td>TRUE, FALSE</td>
</tr>
<tr>
<td>Query Expansion (QE) Model</td>
<td>KL, Info, KLCp, Bo2, Bo1, KLCt</td>
</tr>
<tr>
<td>Number of Doc. for QE</td>
<td>2, 5, 10, 20</td>
</tr>
<tr>
<td>Number of terms for QE</td>
<td>2, 5, 10, 20</td>
</tr>
<tr>
<td>Min. number of doc. in which terms should occur</td>
<td>2, 5, 10, 20, 50</td>
</tr>
</tbody>
</table>
Experimental Setup: Corpus Description

- TREC Robust collection.

- Consists of half million of documents, set of topics, and relevance assessments.

- 100 topics used in performing the experiments.

- Title part used as a user query.

Example topic description:

<title> sick building syndrome
<desc> Identify documents that discuss sick building syndrome or building-related illnesses
<narr> A relevant document would contain any data that refers to the sick building or building-related illnesses, including illnesses caused by asbestos, air conditioning, pollution controls. Work-related illnesses not caused by the building, such as carpal tunnel syndrome, are not relevant
Experimental Setup: Features

- Combination of pre-retrieval and post-retrieval features.

- **30** features extracted from **four** query difficulty predictors:
  
  - **Wordnet sense number (pre-retrieval):** measures the ambiguity of the query.
  
  - **Frequency Inverse (IDF) (pre-retrieval):** measures whether the terms of the query are rare in a corpus or not.
  
  - **Standard deviation (STD) (post-retrieval):** measures the deviation in the scores of the ranked lists.
  
  - **Feedback of the query (QF) (post-retrieval):** measures the overlap between two list ranked documents (before expansion and after expansion).
Experimental Setup: Classifiers

- Support vector machine (SVM) used to learn system parameters.
- kernel trick methods applied (data are not linearly separable).
- Kernel methods:
  - Radial Bias Function (RBF) (gamma values: 1 to 10).
  - Sigmoid (gamma values: 1 to 10).
  - Polynomial (degree: 1 to 5).
- 10-cross validation used to select the best kernel over different values of best systems $N$.

<table>
<thead>
<tr>
<th>System Param</th>
<th>$N=1$</th>
<th>$N=5$</th>
<th>$N=10$</th>
<th>$N=20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stemmer</td>
<td>RBF, gamma=1</td>
<td>poly, deg=3</td>
<td>RBF, gamma=1</td>
<td>RBF, gamma=1</td>
</tr>
<tr>
<td>Retrieving model</td>
<td>RBF, gamma=1</td>
<td>RBF, gamma=1</td>
<td>RBF, gamma=1</td>
<td>RBF, gamma=1</td>
</tr>
<tr>
<td>Parameter Free expansion</td>
<td>RBF, gamma=1</td>
<td>RBF, gamma=1</td>
<td>RBF, gamma=1</td>
<td>RBF, gamma=1</td>
</tr>
<tr>
<td>Query Expansion (QE) Model</td>
<td>poly, deg=2</td>
<td>RBF, gamma=1</td>
<td>poly, deg=1</td>
<td>poly, deg=3</td>
</tr>
<tr>
<td>Number of Documents for QE</td>
<td>RBF, gamma=1</td>
<td>poly, deg=2</td>
<td>RBF, gamma=1</td>
<td>RBF, gamma=1</td>
</tr>
<tr>
<td>Number of terms for QE</td>
<td>RBF, gamma=1</td>
<td>poly, deg=5</td>
<td>poly, deg=3</td>
<td>RBF, gamma=1</td>
</tr>
<tr>
<td>Minimum Number of documents</td>
<td>RBF, gamma=1</td>
<td>RBF, gamma=1</td>
<td>RBF, gamma=1</td>
<td>RBF, gamma=1</td>
</tr>
</tbody>
</table>
Performance Metrics

• **Accuracy**: Check whether the predicted system falls in the top \( x \) actual systems of the submitted query.

\[
Acc_x = \frac{1}{L} \sum_{i=1}^{L} IN(M, S_i, x)
\]

- \( L \): number of testing queries.
- \( IN \): Boolean function returns 1 when the predicted system “\( S_i \)” belonging to the top “\( x \)” systems of the query \( Q_i \).

• **Baselines**: Performance of the predicted systems is compared with two baselines.
  - Hard baseline: average performance of the best system of each query in the corpus.
  - Weak baseline: best system performance performs on average over all queries.
Experimental Evaluation: Pre-defined System Vs PPL (Average Precision)

- 10-folds cross validation experiments using AP as effectiveness measure, learning over different values top systems per query.

Pre-defined systems
One classifier

PPL method
Seven classifiers

![Graphs showing comparison between Pre-defined systems and PPL method](image-url)
Experimental Evaluation: Pre-defined System Vs PPL (P@10)

- 10-folds cross validation experiments using P@10 effectiveness measure, learning over different values top systems per query.

<table>
<thead>
<tr>
<th>Pre-defined systems</th>
<th>PPL method</th>
</tr>
</thead>
<tbody>
<tr>
<td>One classifier</td>
<td>Seven classifiers</td>
</tr>
</tbody>
</table>

![Graphs showing accuracy vs. number of best systems for Pre-defined systems and PPL method.](image-url)
Baselines Analysis

- PPL method predicts systems close to ground truth baseline (hard baseline).
- Pre-defined system approach predicts systems close to the (weak baseline).
Conclusion and Future Direction

- **PPL** method can be adopted as an efficient solution for all queries, regardless the difficulty of the queries.

- **PPL** solves all drawbacks that were mentioned in the state of art solutions.

- As a future work, extracting new features such as “deep features” from queries may improve the prediction performance.
Predicting the Best Information Retrieval System Parameter Value: the Per Parameter Learning (PPL) method
References


References

