PACRR: A Position-Aware Neural IR Model for Relevance Matching

Kai Hui¹, Andrew Yates¹, Klaus Berberich¹, Gerard de Melo²

¹Max Planck Institute for Informatics
{khui, kberberi, ayates}@mpi-inf.mpg.de

²Rutgers University, New Brunswick

gdm@demelo.org
Motivation

- Decades of research in ad-hoc retrieval provides useful measures to boost the performance

- Unigram matching signals have been successfully incorporated in neural IR models [2,4]

- How to incorporate positional matching information remains unclear
Matching Information to Incorporate

**QUERY**

computer science course Denmark

**DOCUMENT**

1. Institutes in **Denmark** provide graduate-level **courses** in **computer science**.
2. PCHandle is an online portal for purchasing personal **computers** in **Denmark**.

- **Unigram matching**: matching individual terms independently
- **Term dependency**: “computer science”
- **Query proximity**: the proximity between different matchings
Model Unigram Matching by Counting

- Given a query $Q$ and a document $D$
- Compute the semantic similarity between each term pair, where one term is from $Q$ and another is from $D$ (via word2vec)
- Group such similarity into bins and model the relevance between $Q$ and $D$ with a histogram [2]

![Diagram showing the computation of relevance between terms and the histogram for the relevance score Rel(Q, D)]

Computer

Science

Course

Denmark

Bag-of-word assumption (independence among terms)
Beyond Unigram Matching: Model Positional Information

1) Retain the similarity into the similarity matrix, keeping both similarity and their relative positions [1,3,5]
2) Matching could be modeled based on different local patterns in the similarity matrix
3) Individual text windows only include one salient matching pattern
Beyond Unigram Matching: Model Positional Information

<table>
<thead>
<tr>
<th></th>
<th>institute</th>
<th>Denmark</th>
<th>provide</th>
<th>graduate</th>
<th>level</th>
<th>course</th>
<th>computer</th>
<th>science</th>
</tr>
</thead>
<tbody>
<tr>
<td>computer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>blue</td>
<td>dark</td>
</tr>
<tr>
<td>science</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>dark</td>
<td></td>
</tr>
<tr>
<td>course</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Denmark</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4) Only retain the salient matching signals for individual query terms
PACRR: Position-Aware Convolutional Recurrent Relevance Matching

(1) CNN layers with different sizes: 2X2, 3X3, 4X4, etc..

(2) Max-pooling among filters

(3) K-max pooling: retain the k most salient signals for each query term

(4) LSTM layer for combination
PACRR: Position-Aware Convolutional Recurrent Relevance Matching

- CNN kernels (dozens of filters) in different sizes, corresponding to text windows with different length.

<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

computer science, science course, etc..

<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

computer science course, science course Denmark, etc..
PACRR: Position-Aware Convolutional Recurrent Relevance Matching

- Max pooling different filters for individual kernels (individual text windows at most include one matching pattern)
K-max pooling for individual query terms, retaining the k most salient signals for individual query terms

- K=2, 2X2 kernel
- K=2, 3X3 kernel
A LSTM layer combines signals on different query terms
Evaluation

- Based on TREC Web Track ad-hoc task 2009-2014, including 300 queries, 100k judgments and approx. 50 runs in each year

- Measures: ERR@20
  - A real value measure summarizing the quality of a ranking
  - The higher the better

- Baseline models: MatchPyramid [1], DRMM [2], local model in DUET [3], and K-NRM [4]
Training and Validation

- Employ five years (250 queries) for training and validation
- Randomly reserve 50 queries from the 250 queries for validation, and the model selection is per ERR@20
- Test on the remaining year (50 queries)
The training loss, ERR@20 and nDCG@20 per iteration on validation data. The x-axis denotes the iterations. The y-axis indicates the ERR@20/nDCG@20 (left) and the loss (right).
Result: RerankSimple

-----How good a neural IR model can achieve by reranking QL baseline?

- The Neural IR model is employed as a re-ranker, making improvements by re-ranking top-k (e.g., top-30) search results from initial ranker

- Initial ranker can access the whole collection of documents

- Re-rank search results from a simple ranker, namely, query-likelihood model (QL)
Result: **RerankSimple**

---How good a neural IR model can achieve by reranking QL baseline?

- All neural IR models can improve based on QL search results.
- PACRR can achieve top-3 by solely re-ranking the search results from query-likelihood model.
Result: PairAccuracy

----How many doc pairs a neural IR model can rank correctly?

- Evaluate on pairwise ranking benchmark. Given (q, d₁, d₂), d₁ is more relevant or d₂ is more relevant?

- Cover all document pairs that are being predicted

- Calculate the accuracy: the ratio of the concordant pairs
The average accuracy for PACRR among different label pairs is 72%
As reference, human accessors agree with each other by 74–77% according to literature
Reference


Thank You!

code: https://github.com/khui/repacrr
contact: khui@mpi-inf.mpg.de