RE-PACRR: A Context and Density-Aware Neural Information Retrieval Model

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 insights about the effective measures to boost the performance.

- Implementation of such insights into neural IR models is under-explored.

- More importantly, building blocks to encode different insights should work together.
Insights to Incorporate

Query: **Jaguar SUV price**

- **Unigram matching.**
  All occurrences of “jaguar”, “suv” or “price” are regarded as relevance signals.

- **Vocabulary mismatch and sense mismatch (e.g., ambiguity).**
  Occurrences of “F-face”, “sport cars” or “discount” could also lead to relevance signals; “jaguar” referring to one kind of big cat should not be considered as relevant.

- **Positional information, e.g., term dependency and query proximity.**
  Co-occurrences of “jaguar price” or “jaguar suv price” indicate stronger signals.

- **Query coverage.**
  “jaguar”, “suv” and “price” should all be covered by a relevant document.

- **Cascade reading model.**
  Earlier occurrences of relevant information are preferred, given that users are impatient, resulting in information in the end being neglected due to an early stop.
Insights to Incorporate

- **Unigram matching.**
  - Counting, as in DRMM and K-NRM.

- **Vocabulary mismatch and sense mismatch** (e.g., ambiguity).
  - Similarity in place of exact match, as in DUET distributed model etc.

- **Positional information**, e.g., term dependency and **query proximity**.
  - CNN filters as in DUET, MatchPyramid and PACRR.

- **Query coverage.**
  - Combination of relevance signals from different query terms, as in DRMM etc.

- **Cascade reading model.**
  - ?
Recap PACRR Model

- Four building blocks are proposed and plugged into an established neural IR model: PACRR (Hui et al., 2017).

Kai Hui, Andrew Yates, Klaus Berberich, Gerard de Melo: PACRR: A Position-Aware Deep Model for Relevance Matching. EMNLP 2017
Design of Modular

- Sense mismatch (e.g., ambiguity).
  For individual relevance signals, examine whether their contexts are also relevant, e.g., if context of “jaguar” is distant with a car but close to an animal, …

- Query proximity.
  Consider co-occurrences of multiple query terms in a large text window.

- Query coverage.
  Cover of all query terms, meanwhile, assume relevance signals for individual query terms are independent, so that the relevance signals could be shuffled before combination.

- Cascade reading model.
  Max-pool salient signals in cascade manners.
Design of Modular

- Please refer to our paper and poster for more technical details.

Sense mismatch: context checker

Shuffle the query terms: better generalization

Large CNN kernel: query proximity

Cascade max-k-pooling: cascade reading model
Evaluation

- Based on TREC Web Track ad-hoc task 2009-2014.
- Measures: nDCG@20 and ERR@20.

- Benchmarks:
  - **RerankSimple**: re-rank search results from a simple ranker, namely, query-likelihood model.
  - **RerankALL**: re-rank different runs from TREC, examining the applicability and the improvements.
  - **PairAccuracy**: cast as classification problems on individual document pairs.

- Baseline models: DRMM, local model in DUET, PACRR and MatchPyramid.
Training and Validation

- Split the six years into four years for training, one year for validation and one year for test.

- In total, there are 15 such train/validation/test combinations.

- For each year, there are five predictions based on different training/validation combinations.

- Significant tests are based on these five predictions for individual comparisons.
All neural IR models can improve based on QL search results (omitted here).
RE-PACRR can achieve top-1 by solely re-ranking the search results from query-likelihood model.
Result: **RerankALL**

---How many runs could be improved by a neural IR model?

<table>
<thead>
<tr>
<th>Measures</th>
<th>Year</th>
<th>RE-PACRR</th>
<th>PACRR</th>
<th>MatchPyramid</th>
<th>DUETL</th>
<th>DRMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR@20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wt09</td>
<td></td>
<td>91% (D↑L↑)</td>
<td>92% (D↑L↑m↑)</td>
<td>86% (p↓D↑l↑)</td>
<td>77% (P↓m↓)</td>
<td>72% (P↓M↓)</td>
</tr>
<tr>
<td>wt10</td>
<td></td>
<td>98% (P↑D↑L↑M↑)</td>
<td>95% (D↑L↑)</td>
<td>95% (D↑L↑)</td>
<td>69% (P↓D↓M↓)</td>
<td>91% (P↓L↑M↓)</td>
</tr>
<tr>
<td>wt11</td>
<td></td>
<td>98% (P↑D↑L↑M↑)</td>
<td>69% (D↑L↑M↑)</td>
<td>43% (P↓L↑)</td>
<td>26% (P↓D↓M↓)</td>
<td>49% (P↓L↑)</td>
</tr>
<tr>
<td>wt12</td>
<td></td>
<td>98% (P↑d↑L↑M↑)</td>
<td>92% (L↑)</td>
<td>93% (L↑)</td>
<td>68% (P↓D↓M↓)</td>
<td>95% (L↑)</td>
</tr>
<tr>
<td>wt13</td>
<td></td>
<td>94% (P↑D↑L↑M↑)</td>
<td>85% (L↑M↑)</td>
<td>64% (P↓d↓)</td>
<td>61% (P↓D↓)</td>
<td>83% (L↑m↑)</td>
</tr>
<tr>
<td>wt14</td>
<td></td>
<td>96% (P↑D↑L↑M↑)</td>
<td>84% (L↑M↑)</td>
<td>58% (P↓l)</td>
<td>52% (P↓)</td>
<td>68%</td>
</tr>
</tbody>
</table>

Percentage of runs that get improved.

- RE-PACRR significantly outperforms all baselines on five years.
- More than 95% of runs are improved by RE-PACRR.
Result: RerankALL

---By how much a neural IR model can improve?

<table>
<thead>
<tr>
<th>Measures</th>
<th>Year</th>
<th>RE-PACRR</th>
<th>PACRR</th>
<th>MatchPyramid</th>
<th>DUEITL</th>
<th>DRMM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>wt09</td>
<td>43% (D↑L↑M↑)</td>
<td>40% (D↑L↑M↑)</td>
<td>31% (P↓D↑L↑)</td>
<td>22% (P↓M↓)</td>
<td>20% (P↓M↓)</td>
</tr>
<tr>
<td></td>
<td>wt10</td>
<td>98% (P↑D↑L↑M↑)</td>
<td>74% (D↑L↑M↑)</td>
<td>54% (P↓d↑L↑)</td>
<td>23% (P↓M↓)</td>
<td>44% (P↓m↓)</td>
</tr>
<tr>
<td></td>
<td>wt11</td>
<td>33% (P↑D↑L↑M↑)</td>
<td>11% (D↑L↑M↑)</td>
<td>-4% (P↓L)</td>
<td>-11% (P↓D↓)</td>
<td>-0% (P↓L↑)</td>
</tr>
<tr>
<td></td>
<td>wt12</td>
<td>89% (P↑D↑L↑)</td>
<td>66% (L↑)</td>
<td>68% (L↑)</td>
<td>22% (P↓D↓M↓)</td>
<td>70% (L↑)</td>
</tr>
<tr>
<td></td>
<td>wt13</td>
<td>36% (P↑D↑L↑M↑)</td>
<td>27% (L↑M↑)</td>
<td>9% (P↓D↓)</td>
<td>8% (P↓D↓)</td>
<td>20% (L↑M↑)</td>
</tr>
<tr>
<td></td>
<td>wt14</td>
<td>29% (P↑D↑L↑M↑)</td>
<td>16% (d↑L↑M↑)</td>
<td>5% (P↓L)</td>
<td>2% (P↓L)</td>
<td>8% (p↓)</td>
</tr>
</tbody>
</table>

Average differences on all runs between the measure scores before and after re-ranking.

- RE-PACRR significantly outperforms all baselines on four years.
- At least 29% of improvements are observed on individual years.
Result: PairAccuracy

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

<table>
<thead>
<tr>
<th>Label Pair</th>
<th>Volume (%)</th>
<th>Queries</th>
<th>Year</th>
<th>RE-PACRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>HRel-NRel</td>
<td>23.1%</td>
<td>262</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HRel-Rel</td>
<td>8.4%</td>
<td>257</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rel-NRel</td>
<td>63.5%</td>
<td>290</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- RE-PACRR performs better on Hrel-NRel and Rel-NRel, and gets close to other models on Hrel-Rel.
- The overall accuracy is beyond 70%.
Thank You!