The Pearl Retriever: Two-Stage Retrieval for Pairs of Argumentative Sentences

Sebastian Schmidt, Jonas Probst, Bianca Bartelt, and Alexander Hinz

08.09.2022
Retrieval is split into separate pipelines for arguments and sentences

**Arguments**
- 60,000 non-argumentative „arguments“ were discarded based on quality scores
  - Classification based on the Webis Argument Quality Corpus 2020 (Gienapp, L., et al. 2020)
  - Index on premises and the conclusion
- DirichletLM (Zhai, C., Lafferty, J., 2017)
- Highly relevant results with little variance (Potthast, M. et al. 2019)

**Sentences**
- Individual sentences
- Index on sentence content
- DPH-Retrieval (Amati, G., 2006)
- Highly relevant results (Potthast, M. et al. 2019)
- Focus on specific query terms
First, a selection of relevant arguments and sentences is retrieved.
Then, arguments are filtered according to their quality.
The quality scores are calculated on BERT encodings

[Diagram showing the process of argument/sentence analysis with BERT Tokenizer, BERT Encoder, and Quality Scorer]

**Arguments**
- Webis Argument Quality Corpus 2020 (Gienapp, L., et al. 2020)
- Scores between -4 and 3
- Arguments with a score < 1 are discarded
- Validation performance was consistent with Team Yeagerists in Touché 2021 (Bondarenko, A. et al. 2021)

**Sentences**
- IBM Debater - IBM-ArgQ-Rank-30kArgs (Gretz, S. et al. 2019)
- Scores between 0 and 1
- Sentences with a score < 0.7 are discarded
- Validation-Performance consistent with baseline by Gretz, S. et al. (2019)

- Architecture based on Gretz, S. et al. (2019)
- Implemented with PyTorch and the Hugging Face Library
ArgRanks are used to rerank the remaining arguments

Query

Sentence Index

DirichletLM
Zhai, C., Lafferty, J. (2017)

Argument Index

Argument Retriever → Quality Filter → ArgRank Reranking

ArgRank
Wachsmuth, H. et al. (2017)

Sentence Retriever

DPH

Sentences are filtered and sorted based on their source arguments
After filtering on quality, the final sentence pairs are formed.
Two approaches to sentence matching were evaluated (I/II)

The first approach was inspired by Maximal Marginal Relevance (MMR)\(^1\)

Choose partner \(S_j\) for sentence \(S_i\), that fulfills:

\[
\max_{S_j \in \mathbb{R}\setminus\{S_i\}} \left[ \lambda \cdot \text{sim}_1(S_i, S_j) - (1 - \lambda) \cdot \text{sim}_2(S_i, S_j) \right]
\]

- \(\text{sim}_1(S_i, S_j)\): Next Sentence Prediction (L2-Normalized)
- \(\text{sim}_2(S_i, S_j)\): Cosine similarity of \(S_i\) and \(S_j\)

- No sentence is matched with multiple other sentences
- \(\lambda=0.5\) leads to nDCG = 0.2801
- \(\lambda=1\) (Next Sentence Prediction) leads to nDCG = 0.4255

---

\(^1\) Carbonell, J. & Goldstein, J. (1998)
Two approaches to sentence matching were evaluated (II/II)

The second approach forms pairs within existing arguments

Neighbor Matching
- Quality scores are calculated using the sentence quality model
- The optimal neighbor was precalculated for each sentence
- $\text{nDCG} = 0.6593$

Extension with a blocklist
- Sentences that include passages like “My opponent” are discarded
- Improvement to $\text{nDCG} = 0.6914$

Quality scores are calculated for pairs with the preceding and following sentence

$Q(S_{i-1}, S_i) = 0.74$  \hspace{1cm} $Q(S_i, S_{i+1}) = 0.85$
Our evaluation was based on three metrics

Every retrieval result was evaluated on

- Argumentativeness \( r_a \in \{-2, 0, 1, 2, 3\} \)
- Sentence Coherence \( r_c \in \{-2, 0, 1, 2, 3\} \)
- Argument Representation \( r_r \in \{-2, 0, 1, 2, 3\} \)

The final nDCG for each model is calculated as the average nDCG over 10 queries and three metrics:

\[
nDCG = \frac{nDCG_a + nDCG_c + nDCG_r}{3}
\]

The optimal ranking was chosen for each query and metric individually
The final nDCG is based on the average of the three metrics

<table>
<thead>
<tr>
<th></th>
<th>Prototype</th>
<th>Neighbor Matching</th>
<th>Blocklist</th>
<th>ArgRank</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Argumentativeness</strong></td>
<td>0.3997</td>
<td>0.5814</td>
<td>0.6281</td>
<td>0.6168</td>
</tr>
<tr>
<td><strong>Sentence Coherence</strong></td>
<td>0.3966</td>
<td>0.7782</td>
<td>0.7814</td>
<td>0.7792</td>
</tr>
<tr>
<td><strong>Argument Representation</strong></td>
<td>0.6967</td>
<td>0.6184</td>
<td>0.6648</td>
<td>0.6873</td>
</tr>
<tr>
<td><strong>nDCG@10</strong></td>
<td>0.4977</td>
<td>0.6593</td>
<td>0.6914</td>
<td>0.6944</td>
</tr>
<tr>
<td><strong>Variance</strong></td>
<td>0.029710</td>
<td>0.010935</td>
<td>0.006412</td>
<td>0.006628</td>
</tr>
</tbody>
</table>

*Each model includes the methods of the previous level*
We draw three main takeaways from our experiments

Simple solutions were able to achieve good results
• Neighbor-Matching nDCG = 0.6593 vs. Next Sentence Prediction nDCG = 0.4255
• The blocklist led to a noticeable improvement on nDCG

ArgRank had only very little influence in our experiments
• Possible explanation: Low edge density (44,250 edges for roughly 300,000 arguments)

DPH’s strong focus on specific query terms can be disadvantageous
• „9/11 was an inside job“ is retrieved for the query „Should Insider Trading Be Allowed?“
• This influence is reduced by the more stable argument retrieval using Dirichlet
The full retrieval architecture

Argument Retriever → Quality Filter → ArgRank Reranking

DirichletLM
Zhai, C., Lafferty, J. (2017)

ArgRank
Wachsmuth, H. et al. (2017)

Sentence Retriever → Arg Matching & Ranking → Quality Filter

DPH

Quality Filter

ArgRank

Sentence Matching

Query

Argument Index

Sentence Index
Backup
## Results achieved on the official evaluation

<table>
<thead>
<tr>
<th>Metric</th>
<th>Blocklist</th>
<th>ArgRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality</td>
<td>0.670</td>
<td>0.678</td>
</tr>
<tr>
<td>Sentence Coherence</td>
<td>0.392</td>
<td>0.398</td>
</tr>
<tr>
<td>Relevance</td>
<td>0.481</td>
<td>0.479</td>
</tr>
<tr>
<td>nDCG@10</td>
<td>0.5143</td>
<td>0.5183</td>
</tr>
</tbody>
</table>
The argument graph (ArgGraph) forms the basis for ArgRank

Conclusion: "Grey imports limit a company's control over its own products"
Discussion: “Allow retailers to import for resale ‘grey’ goods from abroad.”
Stance: CON

1. Identification of premises from arguments with the same stance in the same discussion
2. Calculation of encodings for conclusion and premises with MPNET [Song, K. et al. (2020)]
3. Creation of edges for arguments with cosine similarity > 0.7

Premise: “Grey imports result in the manufacturer/ distributor effectively losing some, and often most, control of their pricing and retailing strategy in the importing country.”
Similarity = 0.80131
Premises = 5

Premise: “The loss of revenue from grey imports can mean that production is limited or even halted going forward, even though there is market demand for more products from the manufacturer.”
Similarity = 0.79121
Premises = 4

The ArgRank was calculated based on the identified graph

Interpretation: Arrow from $A_1$ to $A_2$:
Argument $A_1$ uses the conclusion of $A_2$ as premise

Adjusted calculation of the ArgRank for argument $i$:

$$
p(c_i) = \frac{(1 - \alpha)}{|A|} + \alpha \sum_j \frac{p(c_j)}{|P_j|} \times \text{sim}(c_i, p_{jk})
$$

- $p(c_i)$: ArgRank of argument
- $|A|$: Number of arguments in the corpus
- $p(c_j)$: ArgRank of argument $j$, that uses the conclusion $c_i$
- $|P_j|$: Number of premises of argument $j$
- $\text{sim}(c_i, p_{jk})$: Similarity between $c_i$ and the premise [Optional]

ArgGraph for cosine similarity > 0.7

The ArgRank was calculated based on the identified graph

Interpretation: Arrow from $A_1$ to $A_2$: Argument $A_1$ uses the conclusion of $A_2$ as premise

Adjusted calculation of the ArgRank for argument $i$:

$$p(c_i) = \frac{(1 - \alpha)}{|A|} + \alpha \sum_j \frac{p(c_j)}{|P_j|} \ast \text{sim}(c_i, p_{jk})$$

Several versions of the ArgGraph were created based on cosine similarity:

- Similarity > 0.9
- Similarity > 0.8
- Similarity > 0.7

The highest nDCGs were achieved for

- $\alpha = 0.3$ and similarity > 0.75 (nDCG = 0.6944)
- $\alpha = 0.4$ and similarity > 0.8 (nDCG = 0.6924)

ArgGraph for cosine similarity > 0.7

Sources (1/3)

• Amati, G. (2003):
  • Probability models for information retrieval based on divergence from randomness (Doctoral dissertation, University of Glasgow).


  • A Large-scale Dataset for Argument Quality Ranking: Construction and Analysis. arXiv:1911.11408.

• Carbonell, J. & Goldstein, J. (2017):
  • The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. SIGIR Forum 51, 2, 209–210. DOI: https://doi.org/10.1145/3130348.3130369.


• Song, K., Xu, T., Qin, T., Lu, J., Liu, T.-Y. (2020):
Sources (3/3)

• Wachsmuth, H., Stein, B., Ajjour, Y. (2017):

• Zhai, C., Lafferty, J. (2017):