Open Web Search at LongEval 2023
Reciprocal Rank Fusion on Automatically Generated Query Variants

CLEF 2023, 18–21 September, Thessaloniki

Maik Fröbe, Gijs Hendriksen, Arjen P. de Vries, Martin Potthast

University of Jena  Radboud Universiteit Nijmegen  Leipzig University and ScaDS.AI

@webis_de  www.webis.de
Motivation from Static Test Collections

RMIT at the 2017 TREC CORE Track

Rodger Benham  
RMIT University  
Melbourne, Australia

Luke Gallagher  
RMIT University  
Melbourne, Australia

Joel Mackenzie  
RMIT University  
Melbourne, Australia

Tadele T. Damessie  
RMIT University  
Melbourne, Australia

Ruey-Cheng Chen  
RMIT University  
Melbourne, Australia

Falk Scholer  
RMIT University  
Melbourne, Australia

Alistair Moffat  
The University of Melbourne  
Melbourne, Australia

J. Shane Culpepper  
RMIT University  
Melbourne, Australia
Motivation from Static Test Collections

RMIT at the 2017 TREC CORE Track

- User query variants may substantially improve retrieval effectiveness [Bailey’16, Benham’17, etc.]
- Experts formulate meaningful user query variants on a given topic
- Rank fusion on user query variants
Motivation from Static Test Collections

RMIT at the 2017 TREC CORE Track

- User query variants may substantially improve retrieval effectiveness [Bailey’16, Benham’17, etc.]
- Experts formulate meaningful user query variants on a given topic
- Rank fusion on user query variants

Our Research Question

- Intentions and preferences can change considerably over time [Huang’22]
- Must retrieval pipelines with query variants adapt over time?
Our Approach

- In the best case, we would have manually formulated query variants
Open Web Search at LongEval 2023

Our Approach

- In the best case, we would have manually formulated query variants, but
  - No topic descriptions available
  - Difficult to guess what searchers did look for in retrospect
  - We have 2023, so let's use ChatGPT :)

Open Web Search at LongEval 2023

Our Approach

- In the best case, we would have manually formulated query variants, but
  - No topic descriptions available
  - Difficult to guess what searchers did look for in retrospect
  - We have 2023, so let's use ChatGPT :)

Pipeline

- Step 1: Generate query Variants with ChatGPT
- Step 2: Reciprocal rank fusion over query variants to produce joint ranking
Open Web Search at LongEval 2023

Our Approach

- In the best case, we would have manually formulated query variants, but
  - No topic descriptions available
  - Difficult to guess what searchers did look for in retrospect
  - We have 2023, so let's use ChatGPT :)

Pipeline

- Step 1: Generate query Variants with ChatGPT
- Step 2: Reciprocal rank fusion over query variants to produce joint ranking

Hyperparameters Tuned on the Training Data

- Which prompt for ChatGPT?
- How many query variants?
- Which retrieval model?
<table>
<thead>
<tr>
<th>Number</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>I wanted to search the web for “&lt;ORIGINAL_QUERY&gt;” but I was not able to find relevant documents. Please recommend me up to ten better queries in a numbered list that retrieve relevant documents. All Queries must be English and have at least two words.</td>
</tr>
</tbody>
</table>
**Open Web Search at LongEval 2023**

**Our Best ChatGPT Prompt**

<table>
<thead>
<tr>
<th>Number</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>I wanted to search the web for “&lt;ORIGINAL_QUERY&gt;” but I was not able to find relevant documents. Please recommend me up to ten better queries in a numbered list that retrieve relevant documents. All Queries must be English and have at least two words.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example Input</th>
<th>Example Response</th>
</tr>
</thead>
</table>
Open Web Search at LongEval 2023
Our Best ChatGPT Prompt

<table>
<thead>
<tr>
<th>Number Prompt</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>I wanted to search the web for “&lt;ORIGINAL_QUERY&gt;” but I was not able to find relevant documents. Please recommend me up to ten better queries in a numbered list that retrieve relevant documents. All Queries must be English and have at least two words.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example Input</th>
<th>I wanted to search the web for “4kvideo downloader” but I was not able to find relevant documents. Please recommend me up to ten better queries in a numbered list that retrieve relevant documents. All Queries must be English and have at least two words.</th>
</tr>
</thead>
</table>


Which Retrieval Model

- We had 10 lexical retrieval models from PyTerrier in the comparison
- BM25, LGD, PL2, . . .
### Open Web Search at LongEval 2023

#### Our Best ChatGPT Prompt

<table>
<thead>
<tr>
<th>Number Prompt</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prompt</td>
<td>I wanted to search the web for “&lt;ORIGINAL_QUERY&gt;” but I was not able to find relevant documents. Please recommend me up to ten better queries in a numbered list that retrieve relevant documents. All Queries must be English and have at least two words.</td>
</tr>
</tbody>
</table>

| Example Input | I wanted to search the web for “4kvideo downloader” but I was not able to find relevant documents. Please recommend me up to ten better queries in a numbered list that retrieve relevant documents. All Queries must be English and have at least two words. |


### Which Retrieval Model

- We had 10 lexical retrieval models from PyTerrier in the comparison
- BM25, LGD, PL2, ...

### How Many Query Variants? We tested 3, 5, and 10
Open Web Search at LongEval 2023
Let's look at the Evaluation
Open Web Search at LongEval 2023

Let's look at the Evaluation

TLDR: Query variants could be more effective

<table>
<thead>
<tr>
<th>Approach / Run</th>
<th>nDCG@10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>June</td>
</tr>
<tr>
<td>ows-pl2-10-variants-prompt-2</td>
<td>0.163</td>
</tr>
<tr>
<td>BM25</td>
<td>0.163</td>
</tr>
</tbody>
</table>
Open Web Search at LongEval 2023

Let's look at the Evaluation

TLDR: Query variants could be more effective

<table>
<thead>
<tr>
<th>Approach / Run</th>
<th>nDCG@10</th>
<th>Unjudged@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ows-pl2-10-variants-prompt-2</td>
<td>0.163</td>
<td>0.164</td>
</tr>
<tr>
<td>BM25</td>
<td>0.163</td>
<td>0.180</td>
</tr>
</tbody>
</table>
Let's look at the Evaluation

TLDR: Query variants could be more effective

<table>
<thead>
<tr>
<th>Approach / Run</th>
<th>nDCG@10</th>
<th>Unjudged@10</th>
<th>Cond. nDCG@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ows-pl2-10-variants-prompt-2</td>
<td><strong>0.163</strong> 0.164 0.167</td>
<td>0.801 0.799 0.788</td>
<td>0.440 <strong>0.480</strong> <strong>0.488</strong></td>
</tr>
<tr>
<td>BM25</td>
<td><strong>0.163</strong> 0.180 0.184</td>
<td>0.782 0.771 0.760</td>
<td><strong>0.446</strong> 0.476 0.478</td>
</tr>
</tbody>
</table>
Open Web Search at LongEval 2023

If there is some time left...
I would like to share some ideas for future work (I.e., what we would have submitted with a bit more time, including some advertisements)
“People think that computer science is the art of geniuses but the actual reality is the opposite, just many people doing things that build on each other, like a wall of mini stones.”

Donald Knuth
Open Web Search at LongEval 2023

Our Goal with TIREx
Open Web Search at LongEval 2023

Our Goal with TIREx

We want to simplify collaboration by promoting software submissions

- Submit software instead of run files (currently 34 retrieval datasets)
- Preferrably small reusable components of retrieval pipelines
- Many components must only be executed “once in a lifetime”
Open Web Search at LongEval 2023
How would this look like?

- Reusability: Approach implemented against ir_datasets
- Reproducibility: self-contained Docker image executed in a sandbox
Open Web Search at LongEval 2023

How would this look like?

- Reusability: Approach implemented against `ir_datasets`
- Reproducibility: self-contained Docker image executed in a sandbox

Advantages of TIREx

- Improved Reusability + Reproducibility
- Blinded experimentation on new corpora possible
Query Segmentation Revisited

Matthias Hagen       Martin Potthast       Benno Stein       Christof Bräutigam
Example: Query Segmentation

Input

Query: hubble telescope achievements

Output

Query segments: hubble telescope | achievements
Example: Query Segmentation

Query Segmentation Revisited

Matthias Hagen      Martin Potthast      Benno Stein      Christof Bräutigam

- Input
  
  Query: hubble telescope achievements

- Output
  
  Query segments: hubble telescope | achievements

Currently, I dockerize and submit such components to TIREx.
Open Web Search at LongEval 2023

The paper and code is published. Why Docker, sandboxing, etc?
Open Web Search at LongEval 2023
The paper and code is published. Why Docker, sandboxing, etc?

INSTRUCTIONS

STEP 1: WRITE $RUN...
Open Web Search at LongEval 2023

The paper and code is published. Why Docker, sandboxing, etc?

INSTRUCTIONS

STEP 1: WRITE $RUN...

...USING UNICORN BLOOD
Outcome

- Query segmentation might be useful for query rewriting
  - hubble telescope achievements ⇒ "hubble telescope" achievements
Outcome

- Query segmentation might be useful for query rewriting
- hubble telescope achievements ⇒ "hubble telescope" achievements

PyTerrier Pipeline

```python
query_segmentation = tira.pt.transform_queries('webis-query-segmentation/hyb-a', dataset)

pipeline = query_segmentation >> rewrite_segments_to_phrases >> bm25
```

- Improves nDCG@10 from 0.00 to 0.31 for hubble telescope achievements
- Use cached outputs from TIREx if available, otherwise use Docker image
Outcome

- Query segmentation might be useful for query rewriting
  - hubble telescope achievements ⇒ "hubble telescope" achievements

PyTerrier Pipeline

```python
query_segmentation = tira.pt.transform_queries('webis-query-segmentation/hyb-a', dataset)
pipeline = query_segmentation >> rewrite_segments_to_phrases >> bm25
```

- Improves nDCG@10 from 0.00 to 0.31 for hubble telescope achievements
- Use cached outputs from TIREx if available, otherwise use Docker image

Hopefully Collaborative Effort to Collect Components

- Query processing: Query intent, entity linking, expansion, reduction, ...
- Document processing: Genre, Spam, expansion, reduction, ...
Outcome

- Query segmentation might be useful for query rewriting
- hubble telescope achievements ⇒ "hubble telescope" achievements

PyTerrier Pipeline

```python
query_segmentation = tira.pt.transform_queries('webis-query-segmentation/hyb-a', dataset)
pipeline = query_segmentation >> rewrite_segments_to_phrases >> bm25
```

- Improves nDCG@10 from 0.00 to 0.31 for hubble telescope achievements
- Use cached outputs from TIREx if available, otherwise use Docker image

Hopefully Collaborative Effort to Collect Components

- Query processing: Query intent, entity linking, expansion, reduction, …
- Document processing: Genre, Spam, expansion, reduction, …

Time is of the Essence

- Very easy if you do this early or from the start
  - Excellent support for self-contained development containers in VS Code
- Might be difficult in some years: Dependencies/binaries become unavailable
Open Web Search at LongEval 2023

Conclusions and Takeaways

- Is the retrieval effectiveness of generated query variants stable over time?
  - No substantial improvement
  - But: retrieval effectiveness was stable :)

- Fusing query variants with lexical retrieval did not improve effectiveness
  - Maybe not enough effort into prompt engineering?
  - Maybe more details, like topic description/narrative required?
Open Web Search at LongEval 2023

Conclusions and Takeaways

- Is the retrieval effectiveness of generated query variants stable over time?
  - No substantial improvement
  - But: retrieval effectiveness was stable :)

- Fusing query variants with lexical retrieval did not improve effectiveness
  - Maybe not enough effort into prompt engineering?
  - Maybe more details, like topic description/narrative required?

Work in Progress and Future Work

- We hope to collect a pool of diverse retrieval components
  - We currently dockerize unavailable components:
    query segmentation, keyphrase extraction, genre classification, etc.

- In the best case, this is a big collaborative effort
  - Please do not hesitate to dockerize your favorite retrieval component :)
Open Web Search at LongEval 2023

Conclusions and Takeaways

- Is the retrieval effectiveness of generated query variants stable over time?
  - No substantial improvement
  - But: retrieval effectiveness was stable :)

- Fusing query variants with lexical retrieval did not improve effectiveness
  - Maybe not enough effort into prompt engineering?
  - Maybe more details, like topic description/narrative required?

Work in Progress and Future Work

- We hope to collect a pool of diverse retrieval components
  - We currently dockerize unavailable components:
    query segmentation, keyphrase extraction, genre classification, etc.

- In the best case, this is a big collaborative effort
  - Please do not hesitate to dockerize your favorite retrieval component :)
  - I am happy to help

Thank You!