RMIT at the TREC 2016 LiveQA Track

Joel Mackenzie, Ruey-Cheng Chen, and J. Shane Culpepper

1. Introduction

- RMIT fielded four systems for the 2016 LiveQA challenge, all built upon a two-stage retrieval architecture.
  - Stage 1: Retrieve a set of candidate snippets, or passages.
  - Stage 2: Rerank the candidates, or generate a summary of the candidates.

2. Research Questions

Research Question (1): Which strategy produces better passages, retrieving from a local test collection or pulling content from a commercial search engine?

Research Question (2): Which strategy produces better answers, locating the best passages directly or generating a succinct summary from the top passages?

Research Question (3): Does the efficiency of the first-stage retrieval module contribute to the failure rate on longer questions?

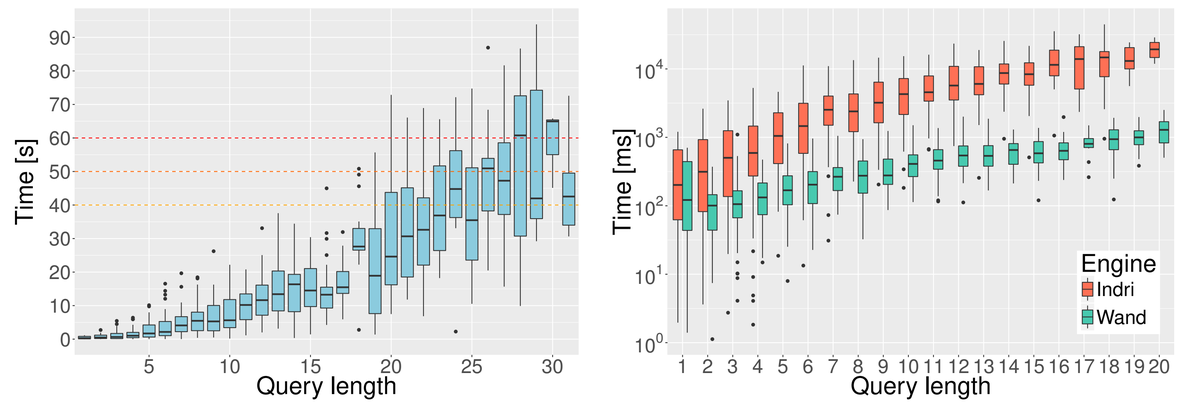
3. Systems

1. **RMIT-1 (automatic):** A WAND bag-of-words passage retrieval using all of the terms in the question title, with answers generated from top-k passages by using a Learning-to-Rank model. Avg. Score (0-3) = 0.723, All runs = 0.577.
2. **RMIT-2 (automatic):** Bing Search API snippets using all of the terms in the question title, with answers generated from top-k passages by using a Learning-to-Rank model. Avg. Score (0-3) = 0.422.
3. **RMIT-11 (automatic):** A WAND bag-of-words passage retrieval using all of the terms in the question title, with answers generated from top-k passages by using a coverage-based summarization algorithm. Avg. Score (0-3) = **0.786**.
4. **RMIT-12 (automatic):** Bing Search API snippets using all of the terms in the question title, with answers generated from top-k passages by using an optimization-based summarization algorithm. Avg. Score (0-3) = 0.447.

4. Apriori Analysis

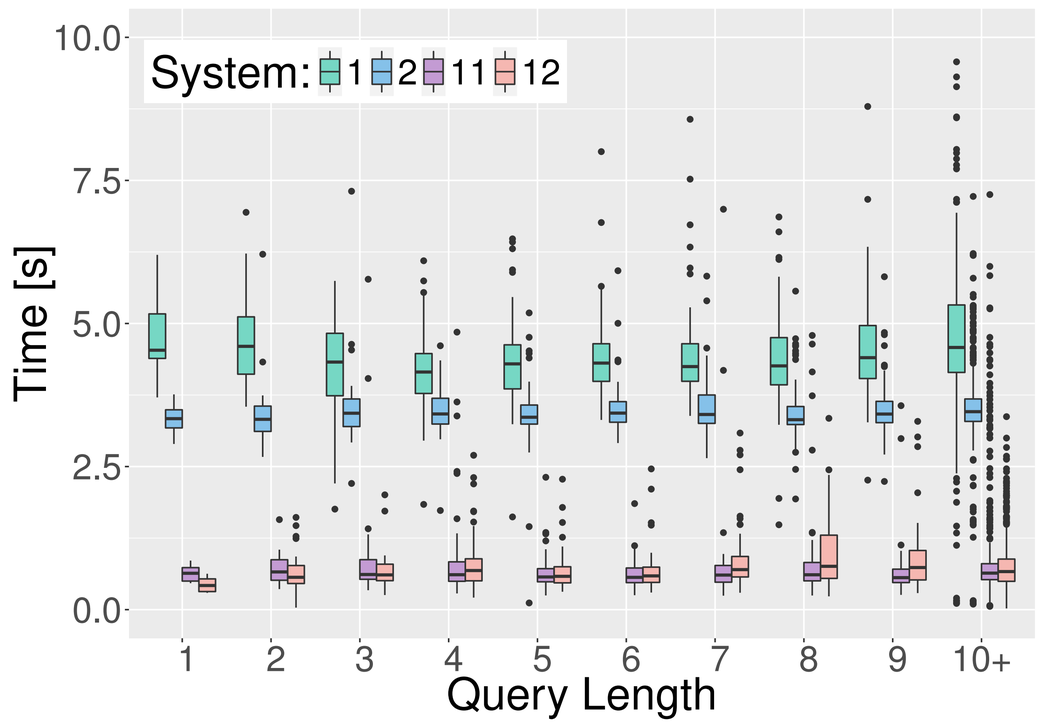
- Title vs Title+Desc queries: Tests using the LiveQA2015 collection showed that Title queries were more effective (as bag-of-words) queries than Title+Desc queries.
- Documents vs Paragraphs: Instead of indexing documents, we opted to index paragraphs. This allowed bag-of-words similarity models such as Okapi BM25 to be used for top-k candidate retrieval.
- Indri vs WAND: Since we no longer needed to extract the top-k passages using Indri, we used our own implementation of Weak-AND, which allowed us to retrieve the top-k candidate paragraphs efficiently.

5. 2015-2016 Efficiency Comparisons



System performance comparisons between Indri and the RMIT WAND system. We can see on the left that using the passage function in Indri is very sensitive to query length. For queries longer that 25 terms, the candidate generation stage can dominate total performance costs, and even result in queries not being processed within the 60 second limit. On the right, we can see that partitioning the documents into paragraphs before indexing improves performance in both systems, but the RMIT system is significantly more efficient for longer queries.

6. 2016 Overall System Efficiency



Systems were much more efficient compared to last year, with the slowest recorded response taking under 10 seconds. Our LtR late stage is less efficient than our summarizer.

7. Failure Analysis

Cause	# Queries
Local collection errors	
Navigational intent	1
Formatting (i.e., answer in HTML table)	1
Query drift caused by question body	2
Irrelevant answer	7
Assessor disagreement	4
Bing snippets errors	
Result filled with query terms but no answer text	13
Answer truncated	12
Assessor disagreement	7

When comparing the RMIT-11 and RMIT-12 systems, we see some reasons each failed. The main reason for the Bing snippets failing was due to the answer snippet containing the query terms, but no actual answer text.

Funding

This work was supported by the Australian Research Council's *Discovery Projects* Scheme (DP140102655 and DP140103256). Shane Culpepper is the recipient of an Australian Research Council DECRA Research Fellowship (DE140100275).