

BM25 Pseudo Relevance Feedback Using Anserini at Waseda University

Zhaohao Zeng

Tetsuya Sakai

Waseda University Tokyo, Japan

@The SIGIR 2019 Open-Source IR Reproducibility Challenge (OSIRRC 2019)
July 25 Paris, France

Introduction

- Implement BM25 Pseudo Relevance Feedback (BM25PRF) [RJ1994] retrieval model with Anserini [YFL2017]
- 2. Test on Robust04.
- Replicable with Docker and the Jig.

BM25PRF (1)

- 1. Initial search using the classic BM25 with the query **q**.
- 2. For each term t_i , calculate its Relevance Weight RW and Offer Weight OW.
- 3. Extract m terms from the top R documents according to OW.

$$RW(t_i) = \log \frac{(r+0.5)(N-n-R+r+0.5)}{(n-r+0.5)(R-r+0.5)}$$

$$OW(t_i) = RW(t_i) \cdot \log(r)$$

where N = # docs n = DF of the term r = DF in the top R

BM25PRF (2)

- 3. Add the new terms (weight w) into the original query.
- 4. Search again with a BM25 variant.
 - Replacing IDF with *RW*

Technical Design

- BM25PRF: implement two JAVA classes in Anserini
 - BM25PrfReranker & BM25PrfSimilarity
 - Has been merged into Anserini master branch
 - Welcome to have a try on any other collections
- Parameter Tuning: A simple python script
 - Train hook of the jig

Results

Table 2: Tuned hyper-parameters.

	<i>K</i> 1	b	$K1_{prf}$	b_{prf}	m	R	w
Tuned Value	0.9	0.2	0.9	0.6	40	10	0.1

Table 3: BM25PRF performance on robust04.

Model	MAP	P@30
BM25 [1]	0.2531	0.3102
BM25PRF (default parameters)	0.2928	0.3438
BM25PRF (tuned parameters)	0.2916	0.3396

- BM25PRF is more effective than BM25 on Robust04
 - By 15% in terms of MAP
- Tuning params on 49 topics does not help.
- The results can be replicated by the Jig easily.



Thank you!

Zhaohao Zeng zhaohao@fuji.waseda.jp