Round 1 results — Run cu_dbmi_bm25_2 submitted from columbia_university_dbmi

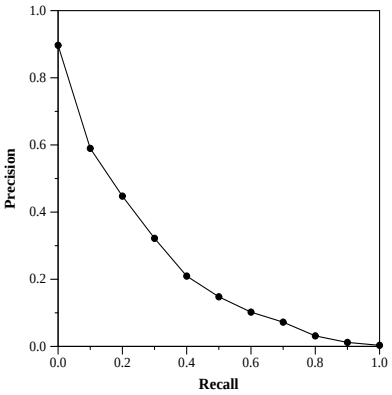
Run Description

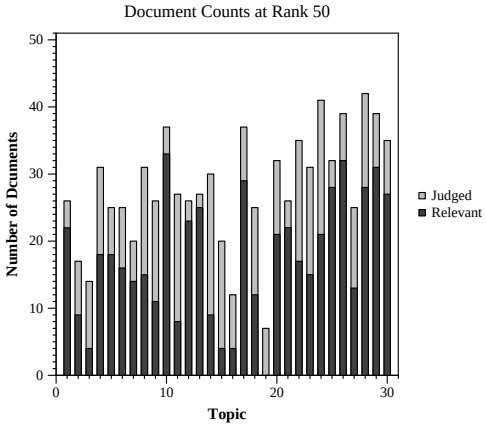
Define COVID-19 key words To find COVID-19 related articles, we have defined a list of key words, the article is considered COVID-19 related if, any of these fields (title, abstract and full text) has any of the key word mentions. To make sure we have included all the key words for COVID-19, we trained a word2vec model on all full texts for phrase embeddings, then we tried to find all synonyms for COVID-19 from the word2vec model. We used an iterative approach, where we start looking for synonyms of one key word and add new phrases or words to the key word list, then use the newly found key word to repeat the same process until there is no new key word found anymore. Here is the list of synonyms for COVID-19. ['ncov', 'covid19', 'covid-19', 'sars cov2', 'sars cov-2', 'sars-cov-2', 'sars coronavirus 2', '2019-ncov', '2019 novel coronavirus', '2019-ncov sars', 'cov-2', 'cov2', 'novel coronvirus', 'coronavirus 2019-ncov'] Retrieve relevant articles for COVID-19 (BM25) We use a python library called whoosh as the indexing engine to enable fast search in title, abstract, and full_text across all documents. The standard tokenizer and the stemmer analyzer are applied during indexing. We retrieve relevant articles using the BM25 algorithm. https://whoosh.readthedocs.io/en/latest/index.html. We construct the search query for each topic using query, question and narrative fields provided in the topic document as the following demonstrates, lower case words, remove puncuation marks and stop words from query, question and narrative there are two parts defined in the construction of the seach query – main query and subquery main query is constructed using the query and question fields following the pattern ((query) OR (question)), the OR operator allows us to retrieve the maximum number of documents related to the main topic. The purpose of the main query is to decide the "scope" of the search. the subquery is constructed using narrative only, we run spaCy to extract the noun phrases and construct the subquery using an OR operator following the pattern (phrase_1 OR phrase_2 OR phrase_3 OR .... phrase_n), the purpose of the subquery is to decide the priorities of the relevant documents. Obviously the more keywords a document contains, the higher score it will receive. while generating the subquery, the PMI (Pointwise Mutual Information) measure is applied to filter out common narrative keywords that contain less information related to the query. The PMI score is computed for each collocation pair of query and narrative keyword. Any keywords with PMI higher than the overall median will be kept in the subquery and others are removed. main_query and subquery are assembled together using the AND operator ((query) OR (question)) AND ((query) OR (question) phrase_1 OR phrase_2 OR phrase_3 OR .... phrase_n). Noted that a copy of the main query is also added to the subquery because we don't want to lose any relevant documents that do not contain any of the phrases extracted from the narrative.

Summary Statistics	
Run ID	cu_dbmi_bm25_2
Topic type	manual
Contributed to judgment sets?	yes

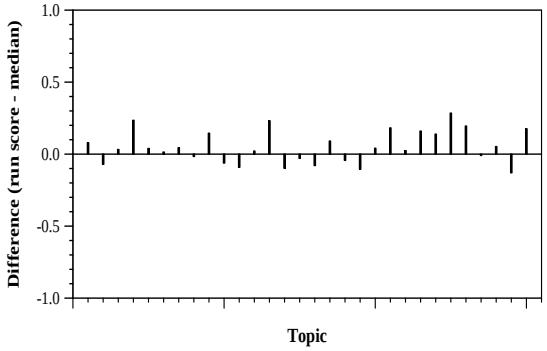
Overall measures	
Number of topics	30
Total number retrieved	12568
Total relevant	2352
Total relevant retrieved	1148
MAP	0.2308
Mean Bpref	0.3740
Mean NDCG@10	0.5819

Document Level Averages	
	Precision
At 5 docs	0.7000
At 10 docs	0.6133
At 15 docs	0.5356
At 20 docs	0.4967
At 30 docs	0.4233
R-Precision	
Exact	0.2924

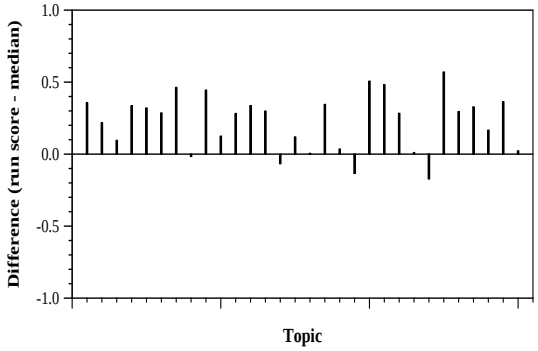




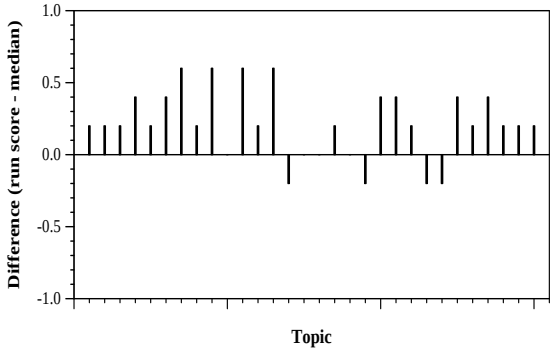
Round 1 results — Run cu_dbmi_bm25_2 submitted from columbia_university_dbmi



Per-topic difference from median bpref for all Round 1 runs



Per-topic difference from median NDCG@10 for all Round 1 runs



Per-topic difference from median P@5 for all Round 1 runs