Information Retrieval Assignment 1
Vector-Space Model

Andrew McIsaac

December 7, 2021

Index Construction

- Czech tags: [Geography, Title, Heading, Text]
- English tags: [PH, KH, HD, DH, SE, DL, LD, TE, CP, DC, CR, DP, SM]
- Inverted Index
  - (docID, term) pairs for every document and every term
  - Invert with SPIMI-Index
  - Sorted terms and postings list for documents of every file
  - Merge to final inverted index and write to disk

Baseline Results

- Czech: 0.0597 MAP, 0.084 P_{10}
- English: 0.0445 MAP, 0.084 P_{10}

Preprocessing

- Lemmas - `spacy`, `spacy_udpipe`
- Stemming - Porter Stemmer
- Stopwords - from `spacy`

Table 1: Mean average precision (MAP) and P_{10} precision of the first 10 documents training performance with different preprocessing techniques. wf: word forms, sw: stopwords, l: lemmatization, s: stemming

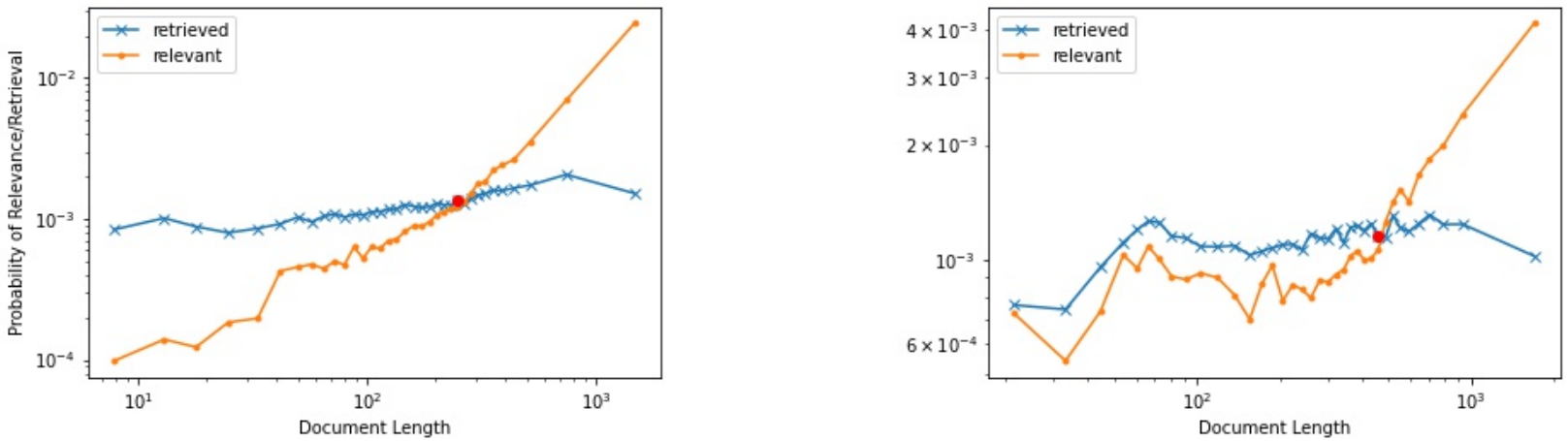
Language		wf	sw	sw+l	sw+s
English	MAP	0.0445	0.1244	0.0834	**0.1751**
	P_{10}	0.084	0.172	0.136	0.252
Czech	MAP	0.0597	0.0770	**0.1553**	0.0672
	P_{10}	0.084	0.100	0.148	0.060

Term/Document Frequency Weighting

Table 2: Mean average precision (MAP) and P_{10} precision of the first 10 documents training performance with different tf-idf weightings. SMART notation tags are applied to both query and document in all cases.

Language		nnc	ntc	npc	ltc	apc
English	MAP	0.1751	0.2244	**0.2248**	0.1320	0.0469
	P_{10}	0.252	0.328	0.328	0.172	0.092
Czech	MAP	0.1553	0.1933	**0.1947**	0.0879	0.0922
	P_{10}	0.148	0.224	0.228	0.096	0.100

Pivoted Document Length Normalization



(a) Czech

(b) English

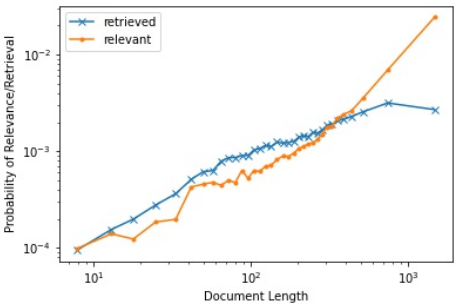
Figure 1: Document lengths for retrieval using cosine normalization compared to relevance.

Pivoted Document Length Normalization

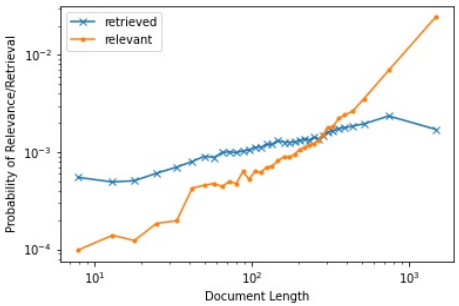
Table 3: MAP training performance of pivoted document length normalization with different values of scaling factor *a*

	0.6	0.7	0.8	0.85	0.9	cosine
English	0.2428	0.2442	0.2446	**0.2453**	0.2450	0.2248
Czech	0.1830	0.1869	0.1932	0.1938	**0.2053**	0.1947

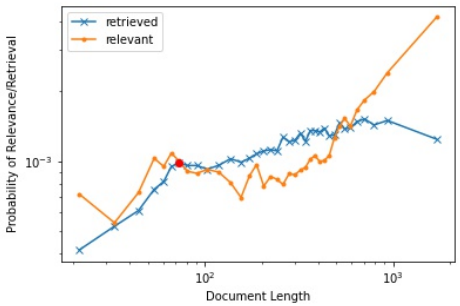
Pivoted Document Length Normalization



(a) Czech, $a = 0.6$



(b) Czech, $a = 0.9$



(c) English, $a = 0.85$

Figure 2: Document lengths for retrieval using pivoted document length normalization compared to relevance.

Problems

- Not enough RAM so index construction would crash

Problems

- Not enough RAM so index construction would crash
  - Close Firefox, Spotify

Problems

- Not enough RAM so index construction would crash
  - Close Firefox, Spotify
- Queries took a very long time (>20 minutes)

Problems

- Not enough RAM so index construction would crash
  - Close Firefox, Spotify
- Queries took a very long time (>20 minutes)
  - Use `pd.DataFrame` to index by doc name instead
  - Now much quicker

Problems

- Not enough RAM so index construction would crash
  - Close Firefox, Spotify
- Queries took a very long time (>20 minutes)
  - Use `pd.DataFrame` to index by doc name instead
  - Now much quicker
- `nltk` has no Czech stemmer

Problems

- Not enough RAM so index construction would crash
  - Close Firefox, Spotify
- Queries took a very long time (>20 minutes)
  - Use `pd.DataFrame` to index by doc name instead
  - Now much quicker
- `nltk` has no Czech stemmer
  - Find it online
  - But maybe it wasn't very good...

Problems

- Not enough RAM so index construction would crash
  - Close Firefox, Spotify
- Queries took a very long time (>20 minutes)
  - Use `pd.DataFrame` to index by doc name instead
  - Now much quicker
- `nltk` has no Czech stemmer
  - Find it online
  - But maybe it wasn't very good...
- `spacy` has no Czech lemmatizer

Problems

- Not enough RAM so index construction would crash
  - Close Firefox, Spotify
- Queries took a very long time (>20 minutes)
  - Use `pd.DataFrame` to index by doc name instead
  - Now much quicker
- `nltk` has no Czech stemmer
  - Find it online
  - But maybe it wasn't very good...
- `spacy` has no Czech lemmatizer
  - Use `spacy_udpipe`
  - But it was much slower...