

Index Construction

Sven Fiergolla
1252732
s4svfier@uni-trier.de

4. Mai 2018

1 Einführung

Information Retrieval, zu deutsch Informationsrückgewinnung, ist ein Fachgebiet, welcher sich mit computergestütztem Suchen nach komplexen Inhalten befasst. Der häufigste Anwendungsbereich ist das Suchen von Dokumenten aus einer Sammlung, die für einen Anwender relevant sind. Würde dies auf der unveränderten Datenmenge geschehen, müsste bei jeder Suchanfrage, die komplette Menge an Rohdaten durchsucht werden. Dies ist ein sehr ressourcenintensiver Prozess, daher wird zunächst auf die Hardware selbst eingegangen. Anschließend werden einige Verfahren zum Erstellen eines Index vorgestellt und auf deren Vor- und Nachteile eingegangen.

2 Hardware Constraints

Typische Systemeigenschaften eines Desktop PC im Consumerbereich:

- *clock rate* 2-4 GHz, 4-8 Kerne
- *main memory* 4-32 Gb
- *disk space* ≤ 1 TB SSD oder ≥ 1 TB HDD
 - HDD (hard disk drive)
 - * *average seek time* zwischen 2 und 10 ms
 - * *transfer* 150 - 300 MB/s
 - SSD (solid state disk)
 - * *average seek time* zwischen 0.08 und 0.16 ms
 - * *transfer* Lesen: 545 MB/s, Schreiben: 525 MB/s

Die zu durchsuchende Datensammlung, häufig auch Korpus genannt, befindet sich auf der Festplatte, welche deutlich langsamer arbeitet als der Ar-

beitsspeicher oder die CPU. Daher stellt die Festplatte in diesem Fall ein Bottleneck¹ dar.

Um die intensive Nutzung der Festplatte zu vermeiden, wird eine alternative Datenstruktur benötigt. Je nach Anforderung kann eine andere Struktur verwendet werden. Für klassisches „Document Retrieval“ wird jedoch in der Regel ein so genannter Invertierter Index konstruiert (siehe [Invertierter Index](#)). Da mit sich der SSD eine neue Speichertechnologie verbreitet hat, welche deutlich bessere Zugriffszeiten hat als eine HDD, wird diese separat betrachtet (siehe [Indexierung mit Solid State Drives](#)).

3 Index Construction

Um das ressourcenintensive Durchsuchen der Rohdaten zu vermeiden, existieren geeignete Datenstrukturen um Anfragen beantworten zu können, ohne alle Daten durchsuchen zu müssen. Eine einfache geeignete Struktur wäre eine „Term-Dokument-Matrix“, welche spaltenweise die Dokumente auflistet und zeilenweise das Vorkommen einzelner Wörter. Auf einer solchen Matrix lässt sich mit einfachen booleschen Anfragen arbeiten, jedoch hat diese Datenstruktur auch einige Nachteile. Beispielsweise wächst die Matrix zu stark an für große Sammlungen, es lassen sich keine komplexeren Anfragen stellen und ein Ranking der Dokumente ist ebenfalls nicht möglich. Daher betrachten wir im Folgenden lediglich den invertierten Index als geeignete Datenstruktur.

3.1 Invertierter Index

Um aus einer Sammlung von Rohdaten (1) die relevanten Informationen zu gewinnen, müssen vorher einige Anpassungen getätigt werden. Besteht

¹Wörtlich: „Flaschenhals“ oder „Engpass“. Gemeint ist ein Engpass beim Transport von Daten, der maßgeblichen Einfluss auf die Arbeitsgeschwindigkeit hat.

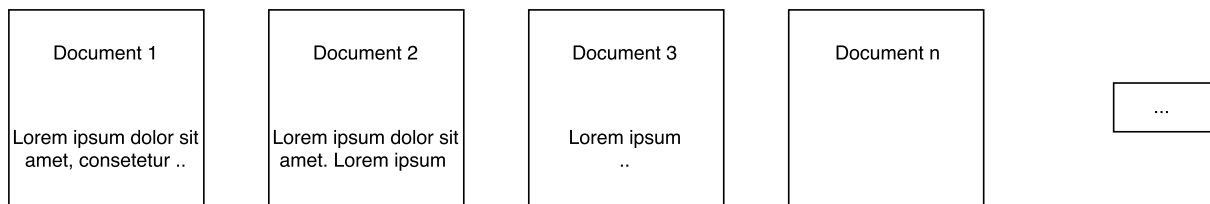


Abbildung 1: beispielhafter Dokumenten-Korpus

die Sammlung beispielsweise aus Webseiten, müssen die Informationen vor dem Indizieren aus der *.html*-Datei geparkt und von html-tags befreit werden. In der Regel wird ebenfalls Tokenization und Stemming auf den Text angewendet. Bei der Tokenisierung wird Dokument in kleinste Einheiten (meistens einzene Wörter) geteilt. Stemming erlaubt, dass verschiedene morphologische Varianten eines Wortes auf ihren gemeinsamen Wortstamm zurückgeführt werden (beispielsweise die Deklination von Wortes oder Wörter zu Wort und Konjugation von gesehen oder sah zu seh).

ference on Very Large Data Bases, September 13-17, 2010, Singapore.

3.2 Blocked sort-based indexing (BSBI)

3.3 Single-pass in-memory indexing (SPIMI)

3.4 Distributed indexing

3.5 Dynamic indexing

3.6 andere Indexierungsverfahren

4 Indexierung mit Solid State Drives

5 Fazit

Literatur

- [1] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze „**Introduction to Information Retrieval**“ Cambridge University Press 2008, pp. 1-18 and 67-84.
- [2] Ian H. Witten, Alistair Moffat, Timothy C. Bell „**Managing Gigabytes: Compressing and Indexing Documents and Images**“ Morgan Kaufman Publishers 1999, pp. 223-261.
- [3] Yinan Li, Bingsheng He, Robin Jun Yang, Qiong Luo, Ke YiTree (Hong Kong University of Science and Technology) „**Indexing on Solid State Drives**“ The 36th International Con-

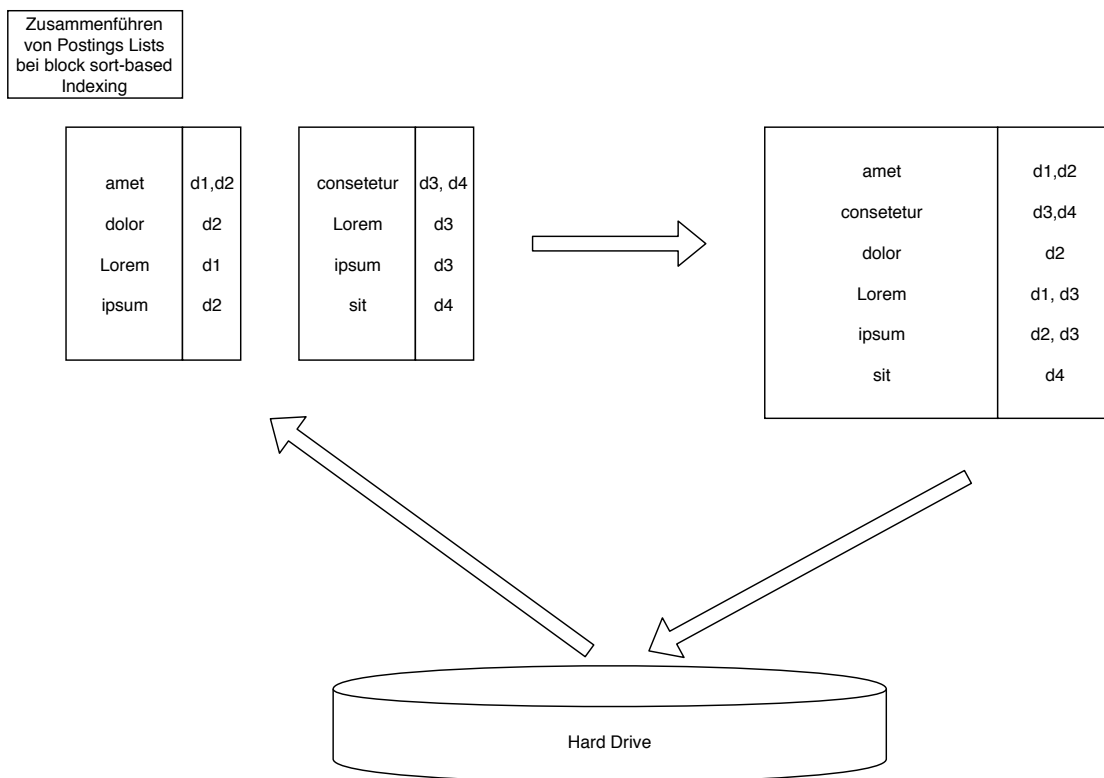


Abbildung 3: beispielhafter Dokumenten-Korpus