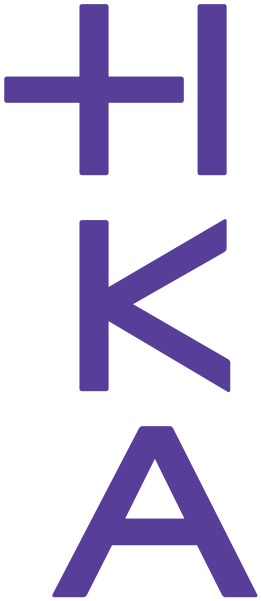
Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik



The Fourteenth International Conference on Advances in Databases, Knowledge, and Data Applications

DBKDA 2022

May 22 - 26, 2022 - Venice, Italy

Tutorial: From Ctrl-F to Information Retrieval - An Excursion into the World of Searching in Text

Andreas Schmidt

Institute for Automation and Applied Informatics
Karlsruhe Institute of Technologie
Germany

Department of Informatics and
Business Information Systems
University of Applied Sciences Karlsruhe
Germany

Latest Version of this Slides

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik

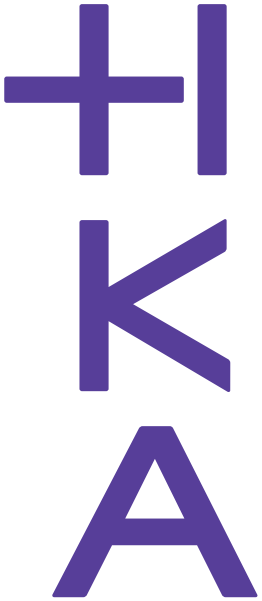
https://www.smiffy.de/dbkda-2022/ [1]

- Slideset
- Exercise
- Command refcard
- Many examples
- Example datasets
- Further resources

1. all materials copyright 2017, 2018, 2019, 2020, 2021, 2022 by andreas schmidt

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik



Better Title for a 1 Hour Session

The Fourteenth International Conference on Advances in Databases, Knowledge, and Data Applications

DBKDA 2022

May 22, 2022 to May 26, 2022 - Venice, Italy

Tutorial: Around the Inverted Index

Andreas Schmidt

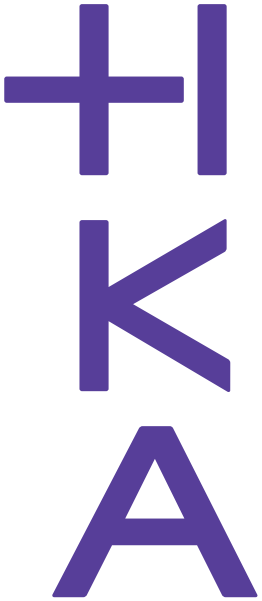
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technologie
Germany

Department of Informatics and
Business Information Systems
University of Applied Sciences Karlsruhe
Germany

Outlook

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik

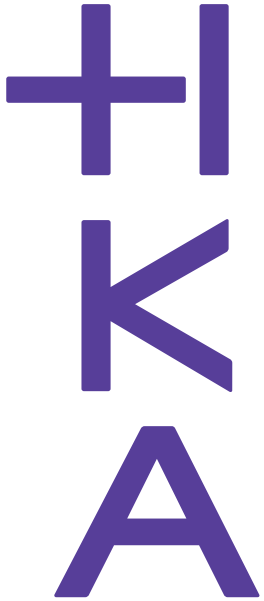


- Boolean Search
- Inverted Index & Extensions
- Term Relevance and Ranking
- Vector Space Model
- Summary

Boolean Retrieval

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik



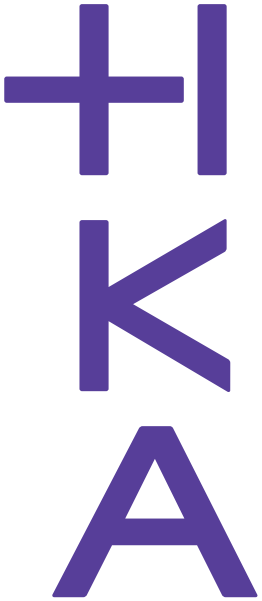
Queries of the Form:

- Q1: Wimbledon
- Q2: Agassi AND Federer
- Q3: Becker OR Boris
- Q4: Federer AND NOT Wimbledon
- Q5: (Federer OR Agassi) AND (Wimbledon OR Rothenbaum)

Term-Document Matrix

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik



	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	...
Agassi	1	1	0	1	0	...
Australlia	1	0	0	1	0	...
Australian Open	0	0	0	1	1	...
Britain	1	1	0	0	0	...
Davenport	0	1	1	0	0	...
Federer	1	0	1	1	1	...
France	0	1	0	0	0	...
Hewitt	0	0	1	0	1	...
Kuznetsova	1	0	0	0	0	
Melbourne	1	1	0	0	0	

Query Processing

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik

In which documents appear the words *Agassi* and *Federer*?

Agassi	**1**	**1**	**0**	**1**	**0**	**...**
			&			
Federer	**1**	**0**	**1**	**1**	**1**	**...**
			=			
Agassi & Federer	**1**	**0**	**0**	**1**	**0**	**...**

- In which documents appear the words *Agassi* and *Federer*, but not *Melbourne*?

Agassi	**0**	**1**	**0**	**1**	**0**	**...**
			&			
Federer	**1**	**0**	**1**	**1**	**1**	**...**
			&			
not(Melbourne)	**0**	**0**	**1**	**1**	**1**	
			=			
Agassi & Federer & not (Melbourne)	**0**	**0**	**0**	**1**	**0**	**...**

Quantitative Aspects

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Document Collection:
  - 1.000.000 documents
  - Average size of a document: ~ 1000 words
  - Average word length: ~6 characters
  - 50.000 - 500.000 distinct words in corpus

=> total volumne: ~ 6 GB

Quantitative Aspects of the Term-Document Matrix

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Term-Document-Matrix:
  - 1.000.000 Columns (the documents)
  - 500.000 Rows (the different words)

=> total volume: 1.000.000 x 500.000 bits = 62.5 GB

- but:
  - about 99.8% of all cells in a row contain ’0’

=> we only store the information about the ’1’cells

Inverted Index: Quantitative Aspects

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- 1.000.000 x 500.000 * 0.2% ~ 1.000.000 entries (Document IDs)
  => 4 MB (with 4 byte for each entry)

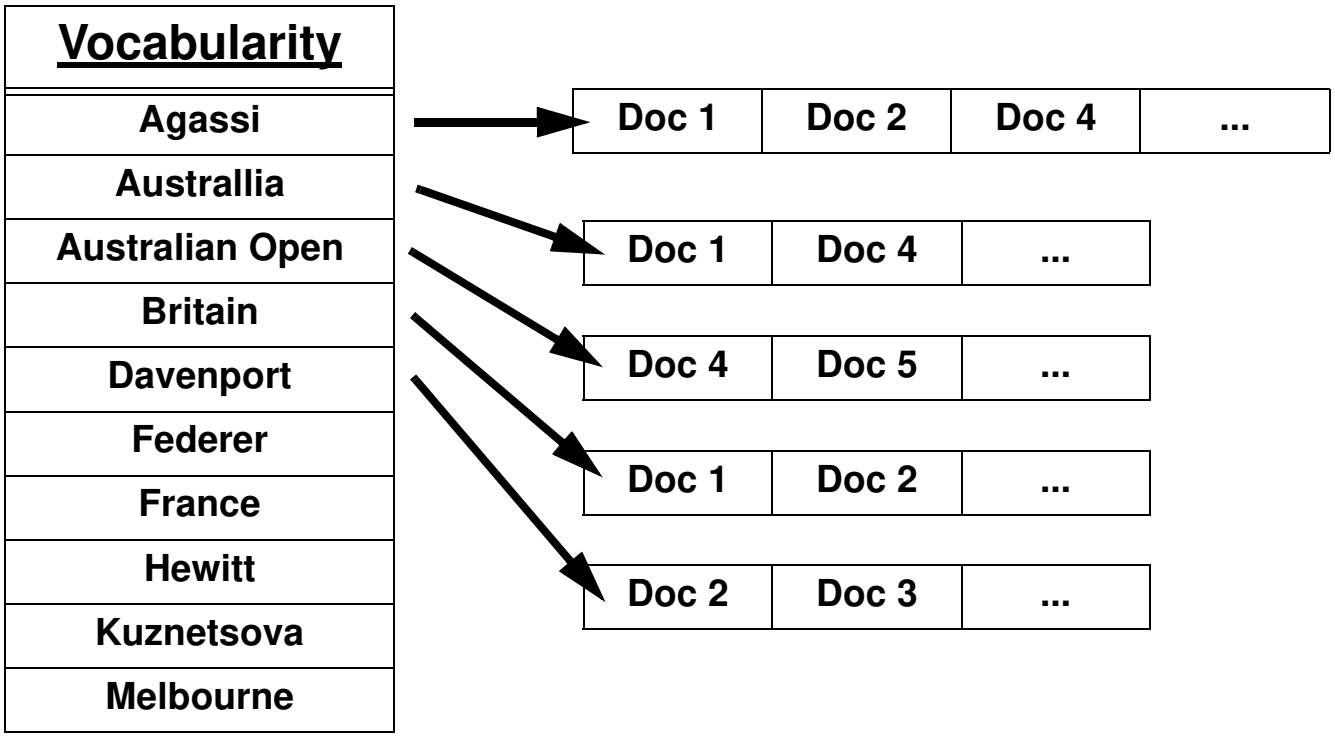
- 500.000 rows (vocabularity)
  - average word length: ~ 8 characters
    => memory requirement: 4 MB
    + additional 2 MB for the pointers to postlinglists

- total memory consumption for our index: ~10 MB

Inverted Index

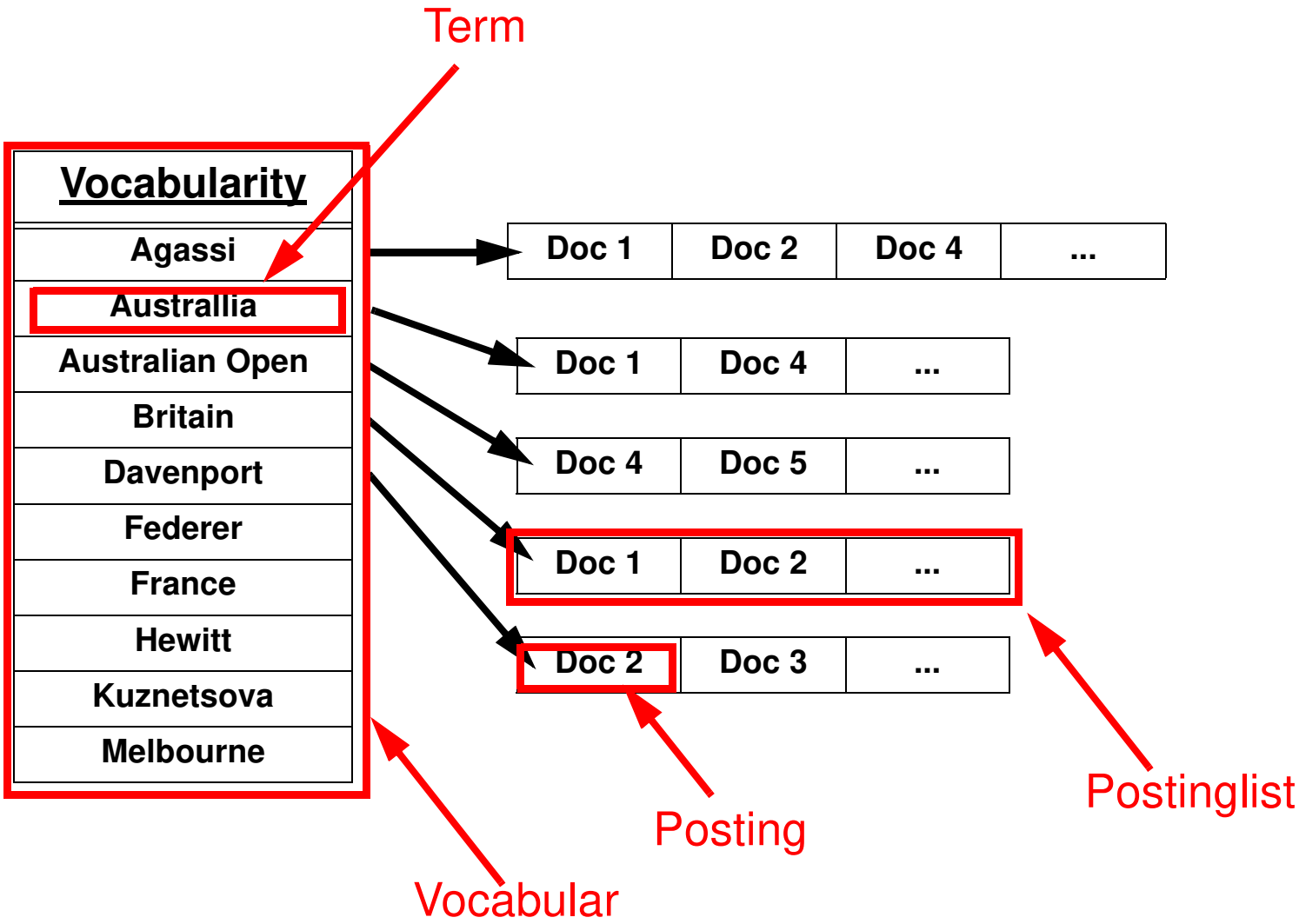
Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik



Inverted Index

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik



Inverted Index: Characteristics

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

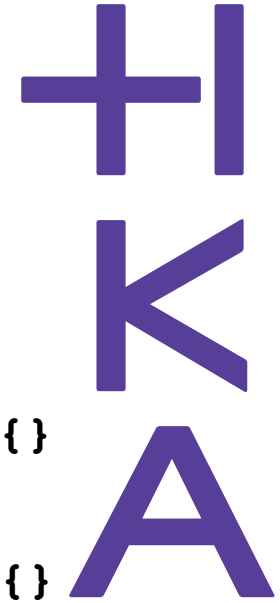
- Fast access to posting list of a term e.g. via hash function, complexity: O(1)
- Postings are sorted by Document ID
- AND, OR operations have linear complexity (number of documents)

- Example Queries:
  - All documents containing the term „Melbourne“
    - trivial, already available
  - All documents containing the terms „Cacablanca“ and „Martini“:

```
Casablanca: 1, 7, 23, 61, 109, 207
Martini:    2, 23, 24, 51, 109, 211, 220
```

Inverted Index: Query Processing

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik



```
Casablanca: 1, 7, 23, 61, 109, 207
Martini:    2, 23, 24, 51, 109, 211, 220
```

L = {}

```
Casablanca: 1, 7, 23, 61, 109, 207
Martini:    2, 23, 24, 51, 109, 211, 220
```

L = {}

```
Casablanca: 1, 7, 23, 61, 109, 207
Martini:    2, 23, 24, 51, 109, 211, 220
```

L = {}

```
Casablanca: 1, 7, 23, 61, 109, 207
Martini:    2, 23, 24, 51, 109, 211, 220
```

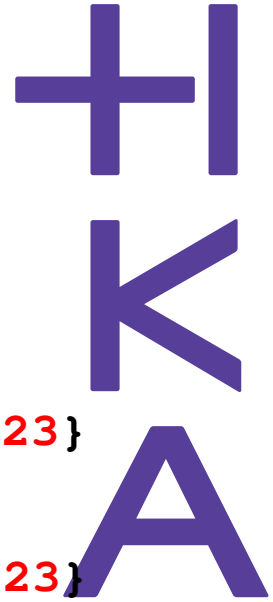
L = {23}

```
Casablanca: 1, 7, 23, 61, 109, 207
Martini:    2, 23, 24, 51, 109, 211, 220
```

L = {23}

Inverted Index: Query Processing

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik



```
Casablanca: 1, 7, 23, 61, 109, 207
Martini:    2, 23, 24, 51, 109, 211, 220
```

L = {23}

```
Casablanca: 1, 7, 23, 61, 109, 207
Martini:    2, 23, 24, 51, 109, 211, 220
```

L = {23}

```
Casablanca: 1, 7, 23, 61, 109, 207
Martini:    2, 23, 24, 51, 109, 211, 220
```

L = {23}

```
Casablanca: 1, 7, 23, 61, 109, 207
Martini:    2, 23, 24, 51, 109, 211, 220
```

L = {23, 109}

```
Casablanca: 1, 7, 23, 61, 109, 207
Martini:    2, 23, 24, 51, 109, 211, 220
```

L = {23, 109}

Inverted Index: Query Processing

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Analog strategy also for OR, NOT

Search query with n (n>2) terms:

- Postingists: L_1, L_2, ..., L_n

L_1 **AND** L_2 **->** L_{12}

L_{12} **AND** L_3 **->** L_{123}

...

$L_{12..(n-1)}$ **AND** L_n **->** $L_{12..n}$

- Optimization (for AND): start with shortest list
  (general: sort the postinglists by their length in ascending order)

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

Inverted Index: Construction

- Given::
  - <n> Documents with Document IDs (Document collection, sorted by id)
  - Dictionary with text keys (for the terms) and list based values (post-inglists)

  - Algorithmus:

```
index = {}
foreach (doc in document_collection)
    foreach (word in doc.words.as_set())
       if index[word]
          index[word].append(doc.id)
       else
          index[word] = list(doc.id)
return index
```

bbcsport: Example dataset [1]

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Consists of documents from the BBC Sport website corresponding to sports news articles in five topical areas from 2004-2005.
- Documents: 737, Terms: 4613
- Natural Classes: 5 (athletics, cricket, football, rugby, tennis)

```
$ ls bbcsport/
athletics  cricket  football rugby  tennis  README.TXT

$ ls bbcsport/tennis/
001.txt  009.txt  017.txt  025.txt  033.txt  041.txt  049.txt
057.txt  065.txt  073.txt  081.txt  089.txt  097.txt
002.txt  010.txt  018.txt  026.txt ...
```

[1] D. Greene and P. Cunningham. "Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering", Proc. ICML 2006.

Example dataset (category tennis)

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

```
$ head -n7 bbcsport/tennis/001.txt
Henman overcomes rival Rusedski

Tim Henman saved a match point before fighting back to defeat British rival
Greg Rusedski 4-6 7-6 (8-6) 6-4 at the Dubai Tennis Championships on Tues-
day.

World number 46 Rusedski broke in the ninth game to take a tight opening
set. Rusedski had match point at 6-5 in the second set tie-break after Henman dou-
ble-faulted, but missed his chance and Henman rallied to clinch the set. The Brit-
ish number one then showed his superior strength to take the decider and earn his
sixth win over Rusedski. Serve was held by both players with few alarms until the
seventh game of the final set, when Rusedski's wild volley gave Henman a vital
break. A furious Rusedski slammed his racket onto the ground in disgust and was
warned by the umpire.

Henman, seeded three, then held his serve comfortably thanks to four serve-and-vol-
ley winners to take a clear 5-3 lead. Rusedski won his service game but Henman took
the first of his three match points with a service winner to secure his place in
the second round at Dubai for the first time in three years. It was the first match
between the pair for three years - Henman last ...
```

Construction of the Index using Shell Commands ...

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Tokenization

```
grep -E -o  '[A-Za-z]+' bbcsport/tennis/001.txt
```

- Lowercase text:

```
tr < bbcsport/tennis/001.txt  'A-Z' 'a-z'
```

- Construct Inverted Index (Uppercase words only)

```
mkdir -p InvIndex
grep -o -E '[A-Z][a-z]+' bbcsport/*/*.txt | sort | uniq |\
      awk -F: '{print $1 >> "InvIndex/"$2}'
```

- Query: All documents containing the words *Alex* and *Rusedski*

```
comm InvIndex/Alex InvIndex/Rusedski -1 -2
```

- Query: All documents conatining the word *Rusedski, but not the word Alex*

```
comm InvIndex/Alex InvIndex/Rusedski -1 -3
```

Boolean Retrieval: Summary & Extensions

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Based on Set-of-Word model
- Advantage: mathematically based (set theory)
- Queries are based on boolean expressions with AND, OR, NOT, (,)
- Disadvantage: no order (ranking) of the returned documents.
- Possible Extensions:
  - Partition document into different fields (title, abstract, ..., literature) and query specific fields, e.g.:

    ```
    TITLE:(trump AND impeachement) AND AUTHOR:sanders
    ```

  - Consider frequency of search terms (switch to the bag-of-words model) for ranking the results.
  - Phrase-Search, NEAR Operator, z.B.:

    ```
    „The world is my oyster“
    trump NEAR corona
    ```

Boolean Retrieval: Summary & Extensions

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Based on Set-of-Word model
- Advantage: mathematically based (set theory)
- Queries are based on boolean expressions with AND, OR, NOT, (,)
- Disadvantage: no order (ranking) of the returned documents.
- Possible Extensions:
  - Partition document into different fields (title, abstract, ..., literature) and query specific fields, e.g.:

    `TITLE:(trump AND impeachement) AND AUTHOR:sanders`

  - Consider frequency of search terms (switch to the bag-of-words model) for ranking the results.
  - Phrase-Search, NEAR Operator, z.B.:

    `„The world is my oyster“`
    `trump NEAR corona`

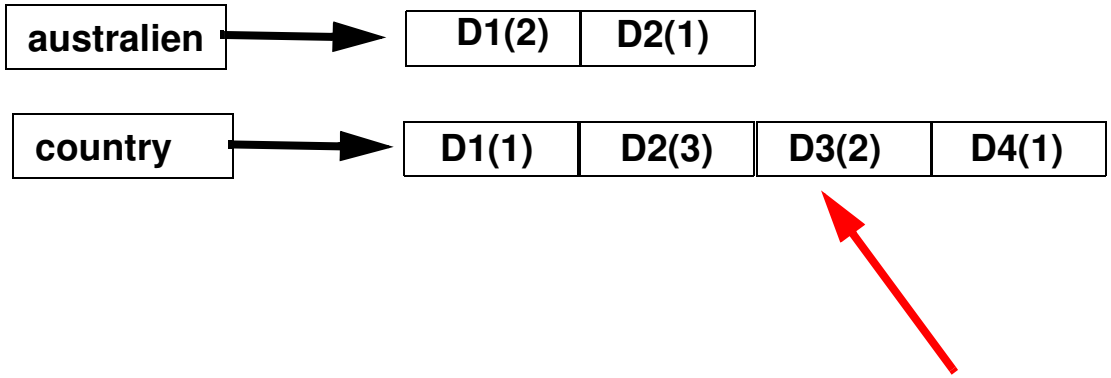
more fine-grained subdivision of documents

requires extension of the inverted index

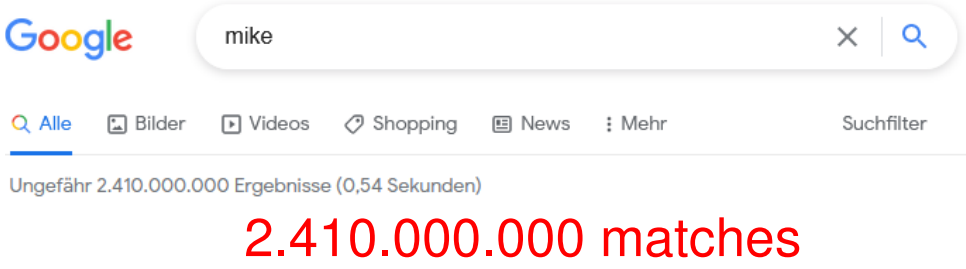
Inverted Index: Extension 1

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Additionally consider the number of times a term occur in a document



- Handling of multiple search terms?
  - Example: *mike stonebraker*



2.410.000.000 matches

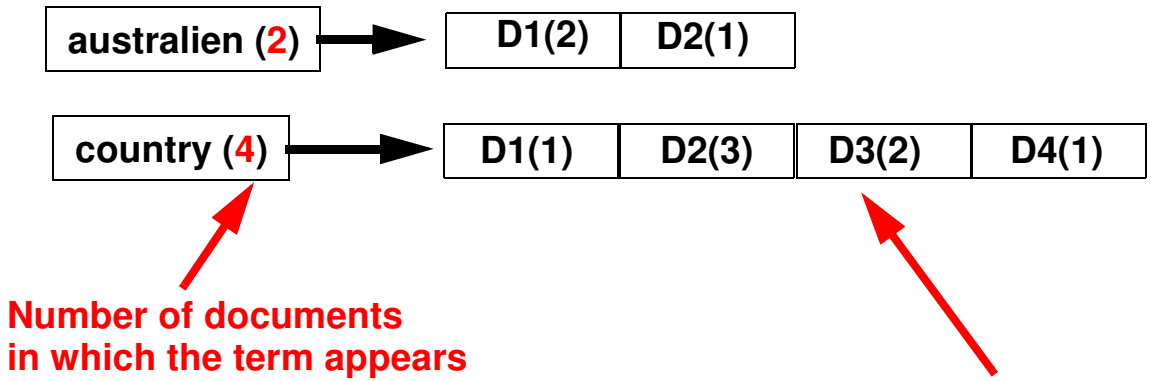
6.780.000 matches

Inverted Index: Extension 1b

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik

- Additionally consider the number of times a term occur in a document



- Handling of multiple search terms?
  - Example: *mike stonebraker*

the weighting of „stonebraker“ should be higher, like that of „mike“

Term Weighting

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- How often a term occurs in a document (Term Frequency - TF)
- In how many documents of the collection does a term occur (Document Frequency - DF)

- Intuitive solution:
  - A document should get a high weight if a search term occurs very often in the document
  - Search terms themselves have different weights: A search term that occurs in many documents in the document collection is not as important as a term that occurs in relatively few documents.

=> see Vector-Model later for concrete calculation

Construction of Inverted Index with TF

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Inverted Index with additional Term Frequency (in 2nd column):

```
$ grep -o -E '[A-Z][a-z]+' bbcsport/tennis/*.txt |sort| \
    uniq -c | tr -s ' ' : | \
    awk -F: '{print $3"\t"$2 >> "InvIndexTf/"$4}'
```

- Calculate Document Frequency

```
wc -l InvIndex/*|head -n -1| sed 's#InvIndex/##'| \
   awk '{print $2"\t"$1}'
```

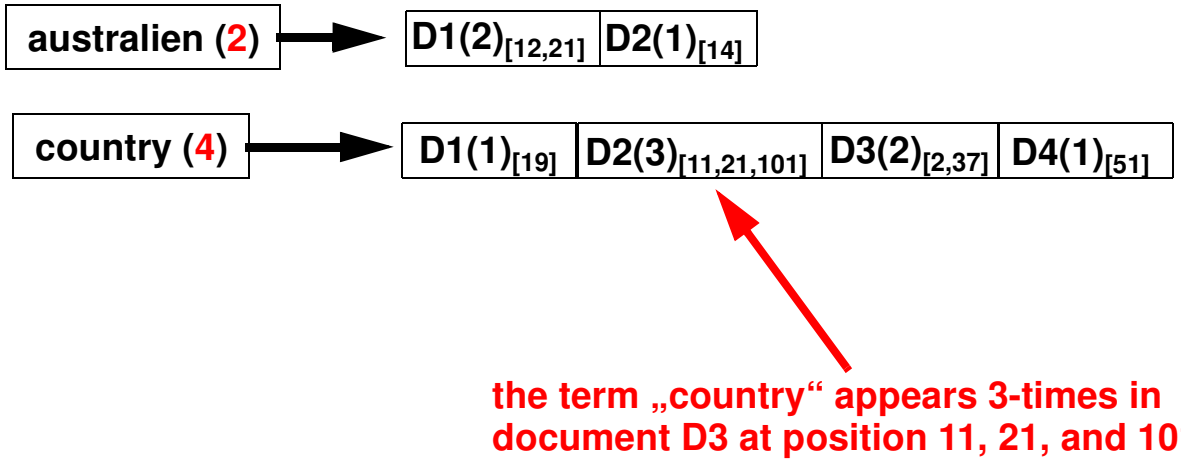
- Create Stopword-list (most common 50 words in corpus)

```
cat bbcsport/*/*.txt | tr 'A-Z' 'a-z' | grep -o -E '[a-z]+' |\
   sort | uniq -c | sort -nr| head -n50
```

Inverted Index: Extension 2

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Additionally consider the position(s) of a term in a document



- Allows phrase queries, i.e.
- „mike stonebraker“
  => term[*docid*, *pos*]=“mike“ AND term[*docid*, *pos+1*]=“stonebraker“

Vectorspace Model

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- General idea:
  - A document represents a vector in a high-dimensional vector space (dimension: |vocabulary|)
  - The query is also represented by a vector in the high-dimensional vector space
  - One can define various similarity measures on vectors, which can then be used for ranking documents.
  - Typically cosine measure is used to determine the similarity of the documents to the request
  - Can also be used to find similar documents to a given document (show similar documents)

Again: Term-Document Matrix

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik

	tf(D1)	tf(D2)	tf(D3)	tf(D4)
australia	2	1		
canada				2
continent	2		1	
country	1	3	2	1
player		2		
participant		2		
world			1	1

... and now ?

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik

- Other view on Term-Document Matrix:

	tf(D1)	tf(D2)	tf(D3)	tf(D4)
australia	2	1		
canada				2
continent	2		1	
country	1	3	2	1
player		2		
participant		2		
world			1	1

tf*idf

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Answer to the question: how important is a word to a document in a collection

	df(t)	idf(t)	tf(D1)	tf(D2)	tf(D3)	tf(D4)
australia	2	1	2	1		
canada	1	2				2
continent	2	1	2		1	
country	4	0	1	3	2	1
player	1	2		2		
participant	1	2		2		
world	2	1			1	1

with $idf(t) = \log_2(\frac{|N|}{df(t)})$ |N|: Number of documents in corpus

	df(t)	idf(t)	$\vec{d_1}$	$\vec{d_2}$	$\vec{d_3}$	$\vec{d_4}$
australia	2	1	2	1	0	0
canada	1	2	0	0	0	4
continent	2	1	2	0	1	0
country	4	0	0	0	0	0
player	1	2	0	4	0	0
participant	1	2	0	4	0	0
world	2	1	0	0	1	1

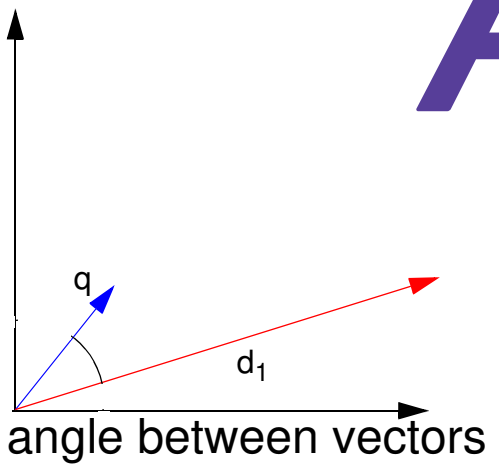
with d(t) = tf(t,d) * idf(t)

Example Query:

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Query: country australia
  - Vector of query: q = (1 0 0 1 0 0 0)
  - Vector document d_1 = (2 0 2 0 0 0 0)
  - Cosine similarity measure

$$sim(q, d) = \frac{\sum_{i=1}^{n} q_i d_i}{\sqrt{\sum_{i=1}^{n} q_i^2} \sqrt{\sum_{i=1}^{n} d_i^2}},$$



- sim(q, D1) = (1*2 + 0*0 + 0*2 + 1*0 + 0*0 + 0*0 + 0*0) /
  (sqrt(1^2 + 1^2) * sqrt(2^2 + 2^2)) = 2/4 = 0.5
- sim(q, D2) = 0.123091
- sim(q, D3) = sim(q, D4) = 0

Vectorspace Model: Summary

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Document is represented as a vector in high-dimensional vector space
- The query will also be represented as a vector
- Cosine measure as similarity measure between vectors
- tf.idf: The number of occurrences of a term in a document and the number of documents in which the term occurs determine the relevance of a term

Preprocessing of Documents

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- What happens before the Inverted Index is build?
  - Adaptation Character Sets
  - Removal of formatting
  - Tokenization (Breakdown into individual terms)
  - Normalisation (lowercase, handling dates, abbreviations, reduction, lematisation, stemming, stopwords)
  - Further optional steps:
    - synonyms
    - integration of taxonomies, class-instance relationships
    - ...

Summary

Hochschule Karlsruhe
University of
Applied Sciences
Fakultät für
Informatik und
Wirtschaftsinformatik

- Inverted Index
  - Mapping from term to documents
  - Allows fast answer which document contain a single/multiple words
  - Extensions for ranking, phrase matches
- Vector Space Model
  - Every document is a vector
  - Query is a vector
  - tf*idf to handle number of occurences/relevance of terms
  - Cosine measure as similarity measure
- The Shell
  - More powerful than we thought
  - Filter and Pipe Architecture
  - Typically incremental approach to program development