

Boolesches Retrieval

Boolesches Retrieval

Anfragen und einfache Datenstrukturen

Dokumente

Dokumente sind die Einheiten des Datenbestandes bezeichnet, die durch das jeweilige Information Retrieval System bearbeitet werden.

Dokumente können beispielsweise Bücher, die Kapitel eines Buchs, Notizen, Emails, aber auch andere **Informationseinheiten** wie digitale Bilder, Videos, Audiodateien sein. Dies hängt immer von der Anwendung ab.

Die Grundmenge an Dokumenten, für die Information Retrieval durchgeführt wird, wird als **Dokumentkollektion** oder auch als Korpus oder – spezifischer – als **Textkorpus** bezeichnet. Wir verwenden üblicherweise das Symbol **D**.

Terme

Als **Terme** oder **Index-Terme** bezeichnet man im Information Retrieval diejenigen Einheiten der Dokumente, die Gegenstand der logischen Repräsentation sind.

Die Terme bilden eine Menge **repräsentativer Stichwörter**. Terme sind meistens Wörter oder Wortkombinationen. Über die Terme kann man einen **Index** als Repräsentation eines oder mehrerer Textdokumente aufbauen.

Historisch wurden die Terme zu einem Dokument von **Bibliothekaren** im Rahmen der **Klassifikation** manuell festgelegt. Heute verwendet man oft (aber nicht immer) automatische Verfahren.

Informationsbedarf und Ad-hoc-Anfragen

Die Formulierung und Beantwortung von **Ad-Hoc-Anfragen** ist eine Standardaufgabe des Information Retrieval:

- Gesucht: Dokumente aus der Dokumentkollektion, die für eine Anfrage „**relevant**“ im Hinblick auf den jeweiligen Informationsbedarf sind. Relevanz durch denjenigen definiert, der Anfrage gestellt hat.
- Algorithmen zur Anfragebeantwortung sollen **effizient** (d.h. schnell ihre Ergebnisse liefern) und **effektiv** (d.h. möglichst genau die Menge der „relevanten“ Dokumente auffinden) sein.

Informationsbedarf:

- Sachverhalt, über den ein Nutzer etwas in Erfahrung bringen möchte.
- Nicht exakt definiert
- unterscheidet sich von einer Anfrage (die Benutzer an das Retrieval-System richtet, um Informationsbedarf zu formulieren).
- Oft Folge von Anfragen notwendig, um Informationsbedarf zu erfüllen.

Boolesches Retrieval-Modell

Das **Boolesche Modell** oder **Boolesche Retrieval-Modell** ist ein Information-Retrieval-Modell der folgenden Art:

- Die logische Repräsentation betrachtet die Dokumente als **Menge von Wörtern**.
- Anfragen werden aus Index-Termen zusammen mit den **Booleschen Operatoren** AND, OR und NOT gebildet.

Beispiel

Shakespeares gesammelte Werke als Dokumentkollektion. Unser (sehr einfacher) Informationsbedarf ist:

In welchen Theaterstücken kommen Brutus und Cäsar vor?

Die entsprechende Anfrage lautet also: **Brutus AND Caesar**

Im Prinzip ist folgender Lösungsansatz denkbar:

Führe einen linearen Scan durch alle Dokumente durch und sammele alle Trefferdokumente auf.

Nach dem UNIX-Shellkommando `grep` wird diese Methode auch als „grepping“ bezeichnet.

Beispiel

Obwohl ein linearer Scan (grepping) bei kleineren Datenvolumina sehr effizient durchführbar ist, gibt es Gründe, nach weiterführenden Methoden zu suchen:

1. **Sehr große Datenbestände**, z.B. das Web;
2. **Komplexe Suchbedingungen**, z.B. „Brutus“ und „Caesar“, aber nicht „Calpurnia“
3. **Flexibleres Matching**, z.B. Ähnlichkeit zu Suchbegriffen oder Nachbarschaft von Wörtern als Kriterium;
4. **Ranking** der Antwortmenge, um die beste oder relevanteste Antwort zu erhalten.

Term-Dokument Inzidenzmatrix

Die **Term-Dokument Inzidenzmatrix** M enthält eine Zeile für jeden betrachteten Term t und eine Spalte für jedes im Grundbestand vorkommende Dokument d .

Tritt t in dem Dokument d auf, so enthält das Matrixelement (t, d) eine 1, sonst eine 0:

$$M(t, d) = \begin{cases} 1, & \text{falls } t \text{ in } d \text{ vorkommt} \\ 0 & \text{sonst} \end{cases}$$

Die Inzidenzmatrix erlaubt verschiedene Sichtweisen:

- Jede Zeile (t, \cdot) stellt einen Vektor dar, der angibt, in welchen Dokumenten der Term t vorkommt.
- Jede Spalte (\cdot, d) bildet einen Vektor, der angibt, welche Terme in dem Dokument d auftreten.

Beispiel: Shakespeare

Folgendes ist ein Ausschnitt aus der Term-Dokument Inzidenzmatrix für Shakespeare-Dramen und die Terme **Antony**, **Brutus**, **Caesar**, **Calpurnia**, **Cleopatra**, **mercy**, **worser**.

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

Beispiel: Shakespeare

Um mit Hilfe dieser Term-Dokument-Inzidenzmatrix die Anfrage

Brutus AND Caesar AND NOT Calpurnia

zu beantworten, werden die Termvektoren

$$(\mathbf{Brutus}, \cdot) = (1, 1, 0, 1, 0, 0)$$

$$(\mathbf{Caesar}, \cdot) = (1, 1, 0, 1, 1, 1)$$

$$(\mathbf{Calpurnia}, \cdot) = (0, 1, 0, 0, 0, 0) = (1, 0, 1, 1, 1, 1)$$

bitweise konjunktiv verknüpft. Man erhält das Ergebnis

$$(1, 1, 0, 1, 0, 0) \text{ AND } (1, 1, 0, 1, 1, 1) \text{ AND } (1, 0, 1, 1, 1, 1) = (1, 0, 0, 1, 0, 0)$$

Das Anfrageergebnis umfasst also die Stücke *Antony and Cleopatra* sowie *Hamlet*.

Hoher Speicherbedarf der Inzidenzmatrix

Sei ein Dokumentenkörper D von 1 Million Dokumenten mit je 1000 Wörtern gegeben. Insgesamt gebe es 500000 Index-Terme. Dann enthält die Inzidenzmatrix M

$$|M| = 5 \cdot 10^5 \cdot 10^6 = 5 \cdot 10^{11}$$

Elemente mit Werten 0 oder 1.

Da jedes Dokument nur 1000 Wörter umfasst, kann es je Dokument höchstens 1000 Einträge in der Inzidenzmatrix geben, die von 0 verschieden sind. Insgesamt enthält M also höchstens

$$10^3 \cdot 10^6$$

von 0 verschiedene Einträge entsprechend einem Anteil von

$$\frac{10^9}{5 \cdot 10^{11}} = \frac{1}{5 \cdot 10^2} = 0,002$$

Der Anteil der 0-Einträge in der Inzidenzmatrix beträgt also

$$99,8\%$$

Invertierter Index

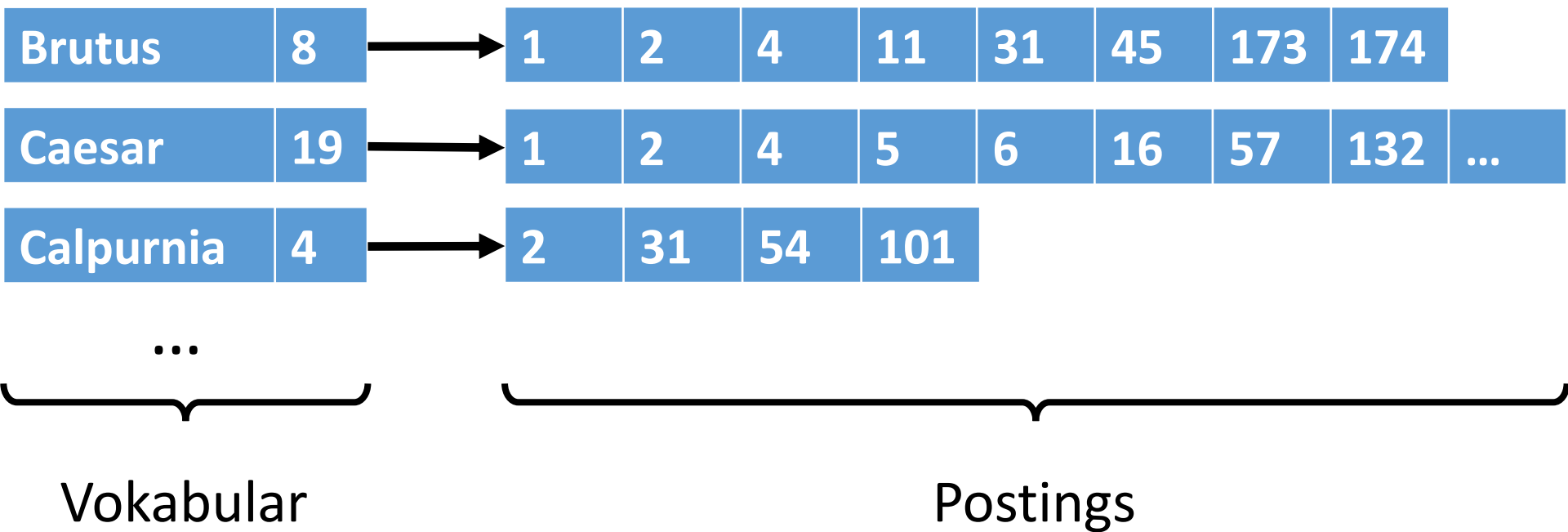
Ein **invertierter Index** oder **invertierte Datei** für eine Dokumentenkollektion D besteht aus einem **Vokabular** (Dictionary) und den **Positionen** (Postings).

Das **Vokabular** enthält alle Index-Terme zu D . Die **Positionen-Tabelle** enthält zu jedem Term aus dem Vokabular alle Dokument-IDs und ggf. weitere Informationen, z.B. Positionen innerhalb von Dokumenten, an denen er auftritt. Die Positionsliste eines Terms heißt auch **invertierte Liste** des Terms.

Es gelten folgende weitere Vereinbarungen:

- Jedes Dokument $d \in D$ besitzt einen (auf D) eindeutigen **Identifikator** $DocId$, beispielsweise eine eindeutige Dokumentnummer oder seine URI im Web.
- Jedem Index-Term t wird seine **Dokumenthäufigkeit** oder **Document Frequency** zugeordnet, die angibt, in wie vielen Dokumenten t vorkommt. Im betrachteten Fall ist die Dokumenthäufigkeit eines Terms t gleich der Länge seines Positionsvektors.
- Die Positionen jedes Terms werden **nach $DocId$ sortiert**.

Beispiel: Invertierter Index



Einfache konjunktive Boolesche Anfragen

Seien **t1** und **t2** Index-Terme. Sei ein invertierter Index **I** für die Dokumentkollektion gegeben. Die Verarbeitung einer einfachen konjunktiven Booleschen Anfrage

t1 AND t2

erfolgt dann in den folgenden Schritten:

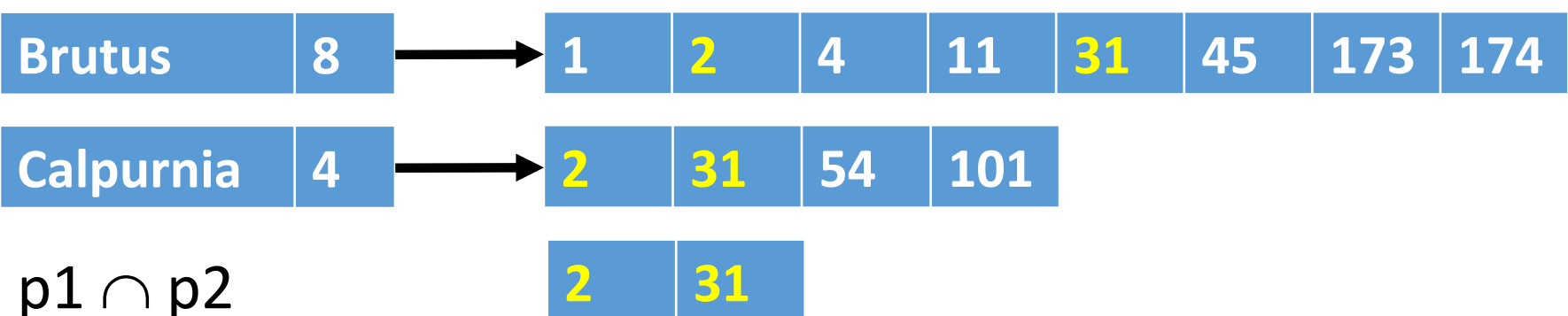
1. Finde den Index-Term **t1** im Vokabular von **I**.
2. Seien **p1** die Positionen von **t1**.
3. Finde **t2** im Vokabular von **I**.
4. Seien **p2** die Positionen von **t2**.
5. Der **Durchschnitt** **p1** \cap **p2** repräsentiert die Treffermenge.

Beispiel

Sei die Anfrage

Brutus AND Calpurnia

gegeben. Dann erfolgt die Ermittlung der Antwort durch die Berechnung des **Durchschnitts der zugehörigen Positionslisten**.



Disjunktive Boolesche Anfragen

Disjunktive Boolesche Anfragen

t1 OR t2 OR ... OR tn

werden durch **Vereinigung der Positionslisten** verarbeitet.

Sei die Anfrage

Brutus OR Calpurnia

gegeben. Dann erfolgt die Ermittlung der Antwort durch die Berechnung der Vereinigung der zugehörigen Positionslisten.

Brutus	8	→	1	2	4	11	31	45	173	174
--------	---	---	---	---	---	----	----	----	-----	-----

Calpurnia	4	→	2	31	54	101
-----------	---	---	---	----	----	-----

$p1 \cup p2$

1	2	4	11	31	45	54	...
---	---	---	----	----	----	----	-----

Negierte Boolesche Anfragen

Werden negierte Anfragen der Form

NOT t

zugelassen, so treten verschiedene Probleme auf:

- Die Antwort kann sehr groß werden; denn alle Dokumente, in denen **t** nicht vorkommt, gehören zum Anfrageergebnis.
- Die Antwort ist in meistens nicht gewünschter Weise abhängig vom Datenbestand: Kommt ein neues Dokument hinzu, in dem **t** nicht auftritt, zählt es zur Anfrageantwort.

Ähnliche Phänomene sind in Datenbanken im Zusammenhang mit unsicheren Anfragen bekannt.

In Information Retrieval Systemen wird deshalb in der Regel **kein unärer Negationsoperator** zur Verfügung gestellt. Stattdessen gibt es eine „geschützte“ Negation **AND NOT**, die mit der Negation zwingend eine Konjunktion verknüpft.

Negierte Boolesche Anfragen

Die Verarbeitung einfacher konjunktiver Anfragen mit Negation

t1 AND NOT t2

erfolgt in den Schritten

1. Ermittle die Positionsliste **p1** für **t1**;
2. Ermittle die Positionsliste **p2** für **t2**;
3. Entferne aus **p1** alle Einträge, die auch in **p2** enthalten sind.

Allgemeine konjunktive Anfragen können dadurch verarbeitet werden, dass durch **Umordnung** zunächst die Treffer für alle positiven (nicht-negierten) Anfrageteile berechnet und anschließend aus dieser Trefferliste die Ergebnisse für die negativen Anfrageteile entfernt werden.

Beispiel

Sei die Anfrage

Brutus AND NOT Calpurnia

gegeben. Dann erfolgt die Ermittlung der Antwort wie folgt:

Brutus	8	→	1	2	4	11	31	45	173	174
--------	---	---	---	---	---	----	----	----	-----	-----

Calpurnia	4	→	2	31	54	101
-----------	---	---	---	----	----	-----

p1 \ p2	1	4	11	45	173	174
---------	---	---	----	----	-----	-----

Boolesches Retrieval

Vorverarbeitung von Dokumenten und Indexierung

Indexierung

Die klassischen Dokumentmodelle abstrahieren ein Dokument auf eine Menge von sogenannten **Indextermen** oder **Deskriptoren**.

Idealerweise sollten Indexterme so gewählt sein, dass sie

1. den **Inhalt** der einzelnen Dokumente adäquat repräsentieren,
2. eine möglichst klare **Abgrenzung** der einzelnen Dokumente gewährleisten,
3. die Verknüpfung von **thematisch ähnlichen** Dokumenten ermöglichen.

Der Prozess der Auswahl von Indextermen heißt **Indexierung**.

Dokumente und Terme

Bisher haben wir angenommen, dass wir wissen,

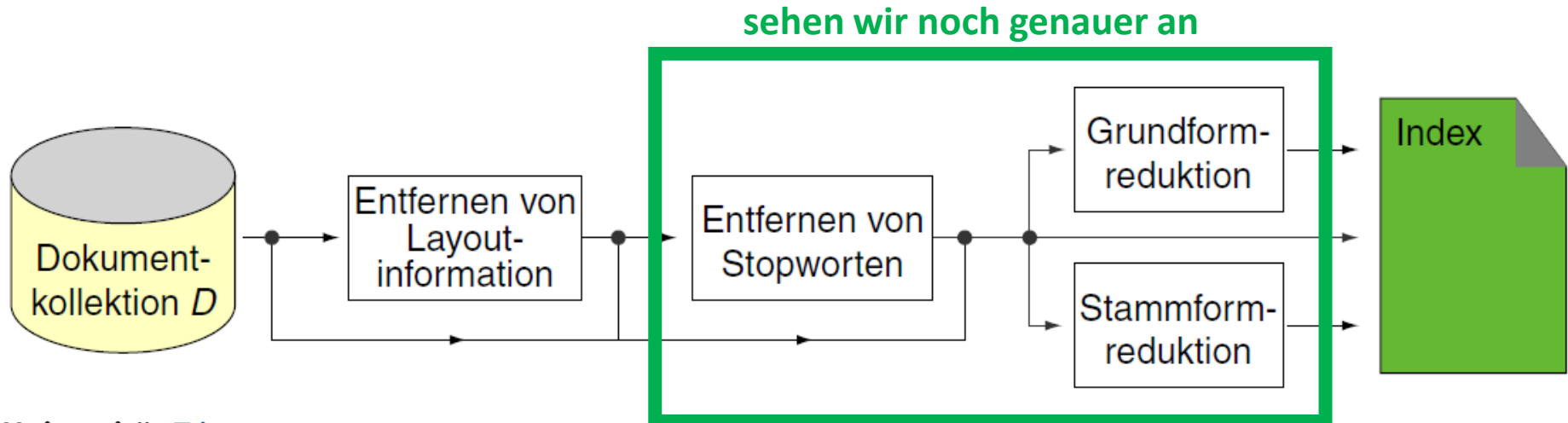
- was ein **Dokument** ist,
- was ein **Term** ist.

In realen Anwendungen kann beides überraschend schwierig werden.

Konstruktion eines invertierten Indexes

Schritte zur Konstruktion eines invertierten Index:

1. Identifikation und Aufsammeln der zu indizierenden Dokumente
`Friends, Romans, countrymen. So let it be with Caesar...`
2. Repräsentation jedes Dokuments als Liste von **Tokens** (Tokenizing)
`Friends Romans countrymen So...`
3. Optionale Normalisierung der Tokenliste durch **linguistische Vorverarbeitung**; Resultat: **Index-Terme**
`friend roman countryman so . . .` Teil der VL im Sommer
4. **Aufbau des invertierten Index aus Vokabular und Positionslisten.**



Dokumente

Vor einer Analyse und Vorverarbeitung der Terme sind Fragen zur **Art des Dokuments** zu beantworten:

- Welches **Format**? pdf, word, excel, html . . .
- Welche **Sprache**? Liegt Mehrsprachigkeit vor?
- Welche Fonts? Welches **Encoding-Schema**? UTF-8? UTF16? ISO-8859-1?

Diese Fragen können auf verschiedene Weise behandelt werden:

- Angabe durch den Benutzer;
- Angabe durch die Metadaten des Dokuments;
- Heuristiken;
- durch einen Klassifikationsalgorithmus

Dokumente

Welches ist die geeignete **Einheit** für die Indizierung?

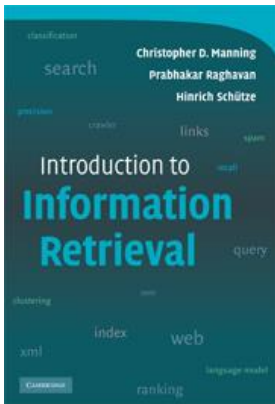
- Einzelne Dateien?
- Dateien mit ihren Versionen?
- Eine Gruppe von Dateien, z.B. LaTeX mit zugehörigen PDF- und HTML-Fassungen?
- Alle Dateien in einem Ordner?
- Eine Email-Nachricht?
- Eine Email mit ihren Anhängen?
- Ein ganzer Email-Kommunikationsstrang?

Je nach Anwendung und Informationsbedarf können auch viele **unterschiedliche Einheiten gleichzeitig** benötigt werden.

Granularität

Bei sehr großen Dokumenten ist die **Granularität**, die der Indizierung zugrunde liegen soll, ein weiterer nicht-trivialer Parameter.

- Ganzes Dokument?
- Einzelne Kapitel?
- Kapitelgruppen?
- einzelne Sätze?



vs.

“Information Retrieval (IR) is **finding material** (usually documents) of an **unstructured** nature (usually text) that satisfies an **information need** from within **large collections** (usually stored on computers).”

Forschungsgebiet: fokussierte Suche in großen Dokumenten
(Büchern, Artikeln, Bibliographien)

Token und Terme

- Ein **Token** ist die Instanz einer begrenzten Zeichenreihe (Character-String), die in dem gegebenen Dokument auftritt und zu einer für die Weiterverarbeitung semantisch sinnvollen Einheit gruppiert ist.
Ein Token kann in einem Dokument mehrfach auftreten.
- Ein **Typ** ist die Klasse aller Token, die dieselbe Zeichenreihe enthalten.
- Ein **Term** ist ein (ggf. „normalisierter“) Typ, der in das Vokabular aufgenommen werden kann.
Die Normalisierung kann z.B. hinsichtlich Groß-/ Kleinschreibung, Morphologie (Wortart, Flexionsform etc.), Rechtschreibung erfolgen.

Lexikalische Analyse, Tokenisierung: Zerlegung eines Textes in Token

Beispiel: Tokenisierung

Input (Dokument):

Friends, Romans, countrymen, lend me your ears.

Output (Tokens):

Friends Romans countrymen lend me your ears

Tokenisierung ist schwierig (sogar in Englisch)

Wie ist die Tokenisierung dieses Satzes?

Mr. O'Neill thinks that the boys' stories about Chile's capital aren't amusing.

1. Neill ?
2. ONeill ?
3. O'Neill ?
4. O' Neill ?
5. O Neill ?

1. aren't ?
2. arent ?
3. are n't ?
4. aren t ?

Tokenisierung und Normalisierung

Problembereiche (beeinflusst Größe des Wortschatz):

- Satzzeichen
- Binde- bzw. Trennstriche
- Groß-/Kleinschreibung
- Ziffern/Zahlen
- Wortlänge
- zusammengesetzte Wörter
- Umlaute etc.
- Schreibfehler
- Sprache

Tokenisierung

- Satzzeichen

. , ; : ? ! ' " : üblicherweise ignoriert

O'Hara als „O“ und „Hara“

U.S. als „US“, nicht als „U“ und „S“

- Binde- bzw. Trennstriche

up-to-date als „up“ und „date“, oder „uptodate“, oder „up-to-date“?

u-haul als „u“ und „haul“?

- Trennung am Zeilenende

Tren-
nung \Rightarrow Trennung

HSV-
fan \nRightarrow HSVfan

Tokenisierung

Zusammengesetzte Wörter (Hyphens, Compounds)

- Hewlett-Packard
- State-of-the-art
- co-education
- the hold-him-back-and-drag-him-away maneuver
- data base
- San Francisco
- Los Angeles-based company
- cheap San Francisco-Los Angeles fares
- New York University vs. the new York University
- Computerlinguistik
- Lebensversicherungsgesellschaftsangestellter
- tusaatsiarunnanngittualuujunga (in Inuit: I can't hear very well.)

Verwandte Suchanfragen zu hold-him-back-and-drag-him-away maneuver

tokenisierung

Chinesischer Text

- **Fehlende Leerzeichen**, damit Tokenisierung sehr schwierig

莎拉波娃现在居住在美国东南部的佛罗里达。今年4月9日，莎拉波娃在美国第一大城市纽约度过了18岁生日。生日派对上，莎拉波娃露出了甜美的微笑。

- **Ambiguität** von Symbolen, Bedeutung und Segmentierung hängt vom Kontext ab

和尚

Die beiden Zeichen können interpretiert werden

- als ein Wort mit der Bedeutung „Mönch“ oder
- als zwei Wörter mit der Bedeutung „und“ „immer noch“.

Japanischer Text

ノーベル平和賞を受賞したワンガリ・マータイさんが名誉会長を務めるMOTTAI NA I キャンペーンの一環として、毎日新聞社とマガジンハウスは「私の、もったいない」を募集します。皆様が日ごろ「もったいない」と感じて実践していることや、それにまつわるエピソードを800字以内の文章にまとめ、簡単な写真、イラスト、図などを添えて10月20日までにお送りください。大賞受賞者には、50万円相当の旅行券とエコ製品2点の副賞が贈られます。

Es gibt **vier Arten** von Schriftzeichen:

- Kanji (chinesische Schriftzeichen);
- Hiragana Silbenschrift (Syllabar) für Flexionsendungen und Funktionswörter (im Deutschen sind das z.B. Artikel, Hilfsverben, Präpositionen, . . .);
- Katakana Silbenschrift, u.a. für die Transkription von Fremdwörtern;
- Lateinische Zeichen.

Es werden **keine Leerzeichen** verwendet.

Arabischer Text

استقلت الجزائر في سنة 1962 بعد 132 عاما من الاحتلال الفرنسي.

← → ← →

← START

‘Algeria achieved its independence in 1962 after 132 years of French occupation.’

Bidirektionalität: Die Hauptleserichtung arabischer Texte ist von rechts nach links; Zahlen werden jedoch umgekehrt gelesen.

Bidirektionalität stellt kein Problem dar, wenn der Text in Unicode vorliegt.

Vorverarbeitung

Nicht alle Wörter, die in einem Dokument auftreten, haben die gleiche Signifikanz.

Meistens wird daher ein Dokument einer **Vorverarbeitung** unterzogen, um die tatsächlich zu verwendenden Index-Terme zu ermitteln. Im Zuge der Vorverarbeitung können weitere Verarbeitungsschritte ausgeführt werden, u.a.

- Normalisierung
- Reduktion auf Wortstämme und Lemmatisierung
- Thesaurusbildung
- Elimination von Stoppwörtern

Normalisierung

In der Regel möchte man auch bei gewissen **Abweichungen** zwischen den Dokumenttermen und den Anfragetermen gültige Anfrageergebnisse erzielen.

Beispielsweise sollten die Terme 'USA' und 'U.S.A.' als „äquivalent“ interpretiert werden.

Normalisierung ist der Prozess **der Kanonisierung von Token**, damit irrelevante Abweichungen nicht ins Gewicht fallen. Terme sind also die „Normalformen“ von Token. Es muss gelten:

Die Normalisierung muss für Dokumente und Anfragen auf die gleiche Weise erfolgen und zu den gleichen Normalformen führen.

Das wichtigste Kriterium für die Art der Normalisierung sind die **Anfragegewohnheiten der Benutzer**.

Normalisierung

Das gängige Normalisierungsverfahren besteht in der Bildung von **Äquivalenzklassen von Token**. Ein ausgewählter Repräsentant der Äquivalenzklasse ist dann der resultierende Indexterm.

Alternativ ist es auch möglich, eine sog. **asymmetrische Expansion** durchzuführen, die jedoch eine geringere Effizienz zur Folge hat.

- window → window, windows
- windows → Windows, windows
- Windows (no expansion)

Normalisierung

Ziffern, Zahlen, Daten müssen für ein gutes Retrievalergebnis (speziell Recall) ebenfalls segmentiert und in ein Standardformat gebracht werden

- 3/12/91 vs. 12/3/91 vs. 12.03.1991 vs. Mar 12, 1991
- B-52 vs. B52 vs. B 52
- 100,000 vs. 100.000 vs. 100000 vs. 100 000
- 5% vs. 0.05 vs. 5/100 vs. 5 Prozent
- 20\$ vs. 20 US-Dollar vs. 20 Dollar vs. 2000 Cent
- 10€ vs. 10 Euro vs. 1000 Eurocent vs. 1000 Cent
- (800) 234-2333 vs. 800.234.2333 vs. +1 800 234 2333

Man sieht sofort, dass auch die Normalisierung von Zahlen, Daten, Telefonnummern etc. ein großes Problem ist.

Normalisierung

- **Umlaute und Sonderzeichen**

- München vs. Muenchen vs. Munchen
- Oeszu vs. Öszu vs. Oszu
- fenêtre vs. fenetre
- In4matik vs. Informatik

Normalisierung ist sprachabhängig

- **Schreibfehler**

Jetes valsch geschrievene Vort is en neus Word



Donald J. Trump  @realDonaldTrump · 11m

Despite the constant negative press covfefe



5.9K



7.6K



9.3K



Case Folding

Der Übergang zur Kleinschreibung (**case folding**) stellt einen weiteren möglichen Schritt zur Normalisierung dar.

- Nicht immer ist case folding adäquat. Beispielsweise sollte die **Großschrift bei Eigennamen** möglicherweise beibehalten werden, um Verwechslungen vorzubeugen.
- Manchmal wird case folding daher so eingerichtet, dass bestimmte Wörter oder Wörter an bestimmten Positionen nicht verändert werden.
- Andererseits erscheint der generelle Übergang zur Kleinschreibung oft trotzdem gerechtfertigt, weil die **Benutzer in ihren Anfragen** ohnehin Groß- und Kleinschreibung nicht unterscheiden.
- Wissensbasierte Methoden sind generell leistungsfähiger als pauschale (heuristische) Regeln.

Reduktion auf Grundformen

Durch die Reduktion von Wörtern auf eine **Grundform** oder auf einen **Wortstamm** können Äquivalenzklassen gebildet werden. Dadurch lässt sich die **Größe von Indexen** und die **Komplexität von Anfragen** stark reduzieren.

Im Wesentlichen gibt es zwei Ansätze

- **Lemmatisierung** und
- **Stemming**

Lemmatisierung

Die Grundform eines Wortes, wie man sie beispielsweise in Lexika findet, wird auch als **Lemma** bezeichnet. Als **Lemmatisierung** wird die Reduktion von Wörtern auf ihre Grundform nach linguistisch gültigen Regeln bezeichnet.

Beispiel:

- am, are, is → be
- car, cars, car's, cars' → car
- ist, bin, sind, war → sein



Lemmatisierung beachtet die Regeln der **Flexion** (Beugung) und **Derivation** (Wortableitung) und berücksichtigt die dadurch hervorgerufenen Wortvarianten.

Beispiel:

- Flexion: cutting → cut
- Derivation: destruction → destroy

Stemming

Als **Stemming** wird eine heuristische Methode zur Reduktion von Wörtern auf einen **Wortstamm** bezeichnet. Durch Stemming werden Wortenden abgeschnitten, um zu Äquivalenzklassen mit gleicher oder ähnlicher Bedeutung zu gelangen.

Im Gegensatz zu Lemmatisierung wird Stemming von Linguisten nicht als gültiges Verfahren akzeptiert.

- Die zugrunde liegende Methode folgt keinen linguistisch abgesicherten Regeln, sondern ist **rein heuristisch** begründet.
- Stemming ist sprachabhängig.
- Stemming mischt beugungs- und ableitungs-induzierte Reduktion.

Beispiel:

automate(s), automatic, automation werden alle zu automat reduziert.

Simple Stemming-Heuristiken

Manuelle Verfahren

Anfrageformulierung mit Wild-Cards, regulären Ausdrücken etc.

aber: zu ungenau

Tabellen Look-Up

Bei Indexierung und/oder Anfrageverarbeitung wird zu jedem Wort der zugehörige Wortstamm in einer Tabelle nachgeschaut.

aber: solche Tabellen existieren praktisch nicht

n-gram-Stemmer

Bei Übereinstimmung **hinreichend vieler n-Gramme** gelten zwei Wörter als morphologisch ähnlich.

Aber: keine Stammbildung im linguistischen Sinne

Beispiel (2-Gramme, Bigramme):

- statistics \Rightarrow st ta at ti is st ti ic cs
unterschiedliche Bigramme: at cs ic is st ta ti
- statistical \Rightarrow st ta at ti is st ti ic ca al
unterschiedliche Bigramme: al at ca ic is st ta ti

Gemeinsame Bigramme: at ic is st ta ti

n-gram-Stemmer

Ähnlichkeitsmaß S ("Dice coefficient"):

$$S(A,B) := 2 * |A \cap B| / (|A| + |B|)$$

A: Menge der Bi-Gramme im ersten Wort

B: Menge der Bi-Gramme im zweiten Wort

Beispiel:

Für "statistics" und "statistical": $S = 2 * 6 / (7 + 8) = 0,8$

Aufbau einer Ähnlichkeitsmatrix

n-gram-Stemmer

- Aufbau einer **Ähnlichkeitsmatrix**:
Berechnung von S für alle Paare indexierter Wörter und Eintrag in (Dreiecks-)Matrix.
- Anwendung eines Cutoff-Wertes (praktikabel: etwa 0,6)
- **Clusterbildung** (hierarchisch, single link):
In jedem Schritt Verknüpfung der beiden Wörter, die am ähnlichsten sind und noch nicht demselben Cluster angehören.

Alle Wörter innerhalb desselben Clusters werden als zum gleichen Stamm gehörig betrachtet.

Stemming mit Successor Variety

Sei α ein Wort der Länge n und α_k ein nichtleerer Präfix von α mit der Länge k .

Sei ferner D ein Text-Corpus und $D_{\alpha k} \subseteq D$ die Menge der Wörter mit Präfix α_k .

Die **Nachfolger-Varietät** $S_{\alpha k}$ von α_k ist dann definiert als die Anzahl verschiedener Zeichen (einschl. dem leeren Zeichen), die in den in $D_{\alpha k}$ enthaltenen Wörtern die Position $k+1$ belegen.

Ein Testwort der Länge n hat n Nachfolger-Varietäten (eine für jeden nichtleeren Präfix).

<http://dblp.org/rec/html/journals/ipm/HaferW74>

Beispiel: Stemming mit Successor Variety

Korpus: ABLE, APE, BEATABLE, FIXABLE, READ,
READABLE, READING, READS, RED, RIPE, ROPE

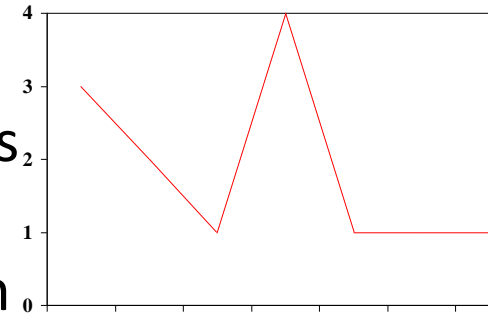
Testwort: READABLE

Präfix	Nachfolger-Varietät	Zeichen
R	3	E, I, O
RE	2	A, D
REA	1	D
READ	4	A, I, S, blank
READA	1	B
READAB	1	L
READABL	1	E
READABLE	1	blank

Stemming mit Successor Variety

Auf Grund der Nachfolger-Varietäten werden die Wörter segmentiert.
Verfahren für Auswahl des Zeichens, hinter dem segmentiert wird
(einzeln oder kombiniert):

- **Cutoff:** Varietät erreicht einen bestimmten Wert.
- **Peak & Plateau:** Varietäten des vorherigen und des nachfolgenden Zeichen sind deutlich niedriger.
- **Complete Word:** Das Präfix kommt im Korpus auch als selbstständiges Wort vor.



Nach Segmentierung muss das geeignete Segment als Stamm ausgewählt werden.

Typische Regel:

Kommt erstes Segment in maximal 12 Wörtern des Korpus vor, dann wähle erstes Segment als Stamm, sonst zweites Segment.

Stemming mit „affix removal“

Regelbasierte Entfernung von Präfixen und/oder Suffixen und bedarfsweise Transformation des entstehenden Stammes.

Beispiel (Pluralentfernung):

CASE word

(ends in "ies" but not "eies" or "aies"): ("ies" → "y"; STOP);

(ends in "es" but not "aes", "ees" or "oes"): ("es" → "e"; STOP);

(ends in "s" but not "us" or "ss"): ("s" → "NULL"; STOP);

(ends otherwise): STOP;

Porter-Algorithmus (affix removal)

Der **Porter-Algorithmus** ist der am weitesten verbreitete Stemming-Algorithmus für Englisch. Seine Effektivität wurde empirisch nachgewiesen.

- Die Reduktion erfolgt in **fünf sequentiellen Phasen**.
- Für jede Phase existieren verschiedene Konventionen zur Auswahl von **Reduktionsregeln**.

Beispiel: Selektiere die Regel, die zu dem längsten Wort-Suffix passt, das durch Reduktion weggelassen werden könnte.

Beispiel

- Delete suffix ***ement*** if what remains is longer than 1 character.
- **replacement** → **replac**
- **cement** → **cement**

Beispiel: Stemming

Stemming allows a little leeway when matching query terms to index terms so that, for example compressed and compression are both accepted as equivalent to compress. Stemming involves stripping one or more suffixes off a word to reduce it to root form, converting it to a neutral term that is devoid of tense and plurality.



stem allow a littl leewa when match queri term to index term so that for exampl compress and compress are both accept as equival to compres stem involv strip one or more suffix off a word to reduc it to root form convert it to a neutral term that is devoid of tens and plural

Bewertung von Lemmatisierung und Stemming

- Lemmatisierung führt höchstens zu **sehr kleinen Vorteilen** beim Retrieval.
- Stemming **erhöht die Ausbeute**, aber **verschlechtert** in der Regel **die Präzision**.

Der Porter Stemmer reduziert die Wörter

**operate operating operates operation operative operatives
operational**

alle zu

oper

Am Beispiel der folgenden Anfragen wird klar, warum die Präzision durch Stemming herabgesetzt werden kann:

- operational AND research
- operating AND system
- operative AND dentistry

Thesauri

Ein **Thesaurus** (oder **Wortnetz**) beschreibt Äquivalenzklassen (**Synsets**) von Wörtern bzw. Phrasen (Sequenzen von Wörtern) gleicher Bedeutung, sogenannter **Synonyme**. Er verzeichnet in der Regel auch **Homonyme** und **Polyseme**, d.h. Wörter, die verschiedene Bedeutungen haben können (z.B. Bank).

Ein Thesaurus kann beispielsweise zu einem Wort W eine Kollektion von Wörtern enthalten, die in der gleichen Äquivalenzklasse wie W liegen.

Thesauri enthalten oft **Konzept-Hierarchien** (s.a. Ontologien), stellen also neben der Äquivalenzklassenbildung eine hierarchische Anordnung von Konzepten und Subkonzepten (bzw. **Hypernymen** und **Hyponymen**) zur Verfügung.

WordNet

WordNet ist ein großer Thesaurus für die englische Sprache mit mehr als 115.000 Wortbedeutungen.

Noun

- [S: \(n\)](#) **bank** (sloping land (especially the slope beside a body of water)) *"they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents"*
 - [direct hyponym](#) / [full hyponym](#)
 - [S: \(n\)](#) [riverbank](#), [riverside](#) (the bank of a river)
 - [S: \(n\)](#) [waterside](#) (land bordering a body of water)
 - [direct hypernym](#) / [inherited hypernym](#) / [sister term](#)
 - [S: \(n\)](#) [slope](#), [incline](#), [side](#) (an elevated geological formation) *"he climbed the steep slope"; "the house was built on the side of a mountain"*
 - [S: \(n\)](#) [geological formation](#), [formation](#) ((geology) the geological features of the earth)
 - [S: \(n\)](#) [object](#), [physical object](#) (a tangible and visible entity; an entity that can cast a shadow) *"it was full of rackets, balls and other objects"*
 - [S: \(n\)](#) [physical entity](#) (an entity that has physical existence)
 - [S: \(n\)](#) [entity](#) (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))
 - [derivationally related form](#)
- [S: \(n\)](#) [depository financial institution](#), **bank**, [banking concern](#), [banking company](#) (a financial institution that accepts deposits and channels the money into lending activities) *"he cashed a check at the bank"; "that bank holds the mortgage on my home"*

Preprocessing: Thesaurus-Bildung

Ziele:

- **Kontrolliertes Vokabular**
(Feste Menge von Suchbegriffen mit klarer Semantik)
- **Konzeptualisierung**
(Erweiterung der Suche auf Synonyme)
- Grundlage für eine korrekte Stammbildung

Stoppwörter

Die 10 am häufigsten verwendeten englischen Wörter machen etwa 20% bis 30% der meisten englischsprachlichen Texte aus.

Wörter, die praktisch in jedem Dokument vorkommen (z.B. „und“), tragen nicht zur Auffindung bestimmter Texte bei und werden daher oft nicht indexiert.

Exkurs: Zipfs Law (nach George Kingsley Zipf)

Die **Kollektionsfrequenz** cf_i des i -häufigsten Terms ist **invers proportional zum Rang i**

$$cf_i \propto \frac{1}{i}$$

Für die relative Kollektionsfrequenz erhalten wir (mit einer sprachabhängigen Konstanten c , **für Englisch $c \approx 0.1$**)

$$\frac{cf_i}{\sum_j cf_j} \propto \frac{c}{i}$$

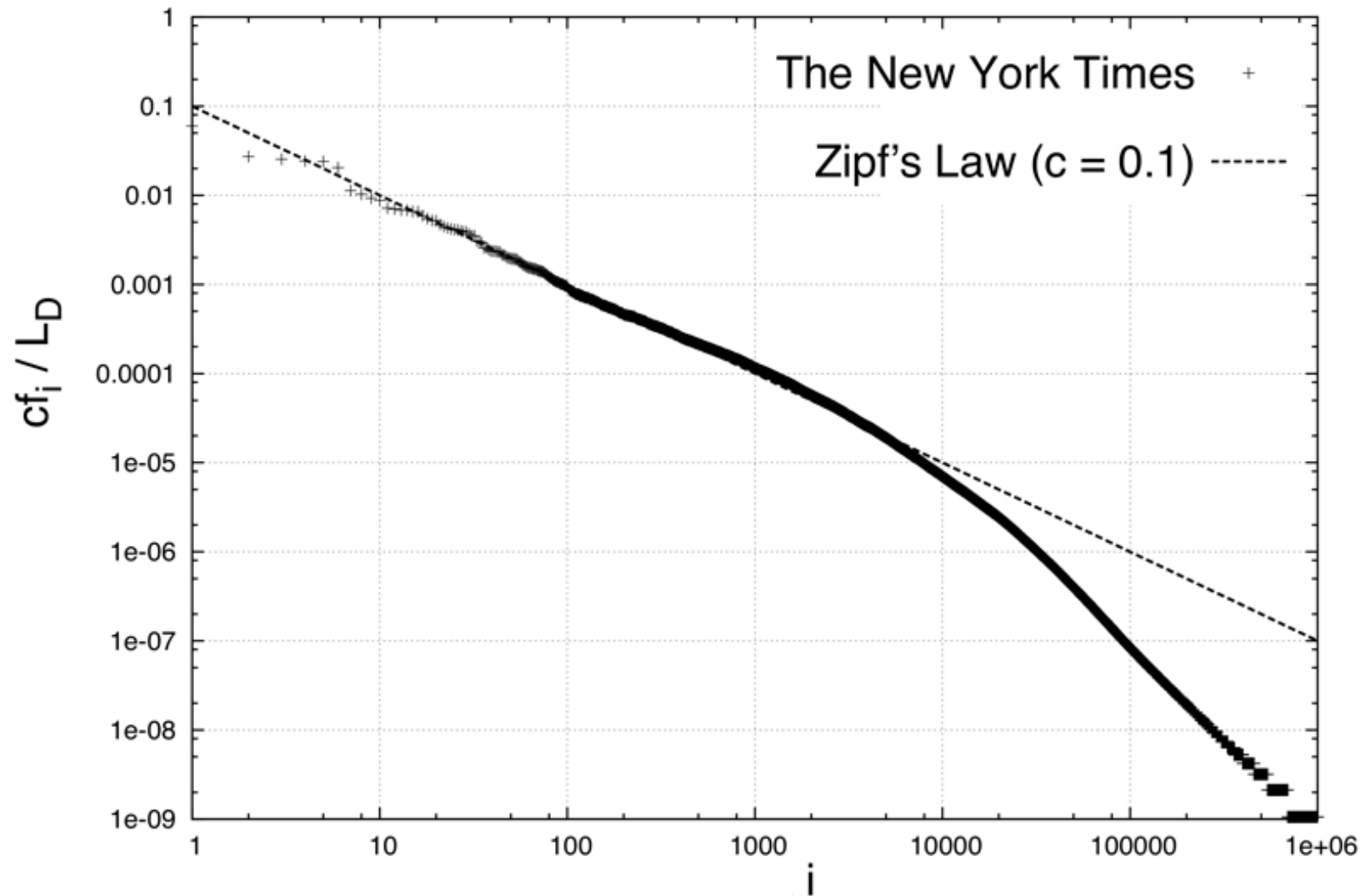
In einer **englischen Dokumentkollektion** können wir daher erwarten, dass der häufigste Term 10% aller Vorkommen umfasst.



Kollektionsfrequenz = Gesamtzahl von Vorkommen
eines Terms in der Kollektion

[George Kingsley Zipf](#)

Zipfs Law



$$L_D = \sum_j cf_j$$

Exkurs: Heaps Law (nach Harold S. Heaps)

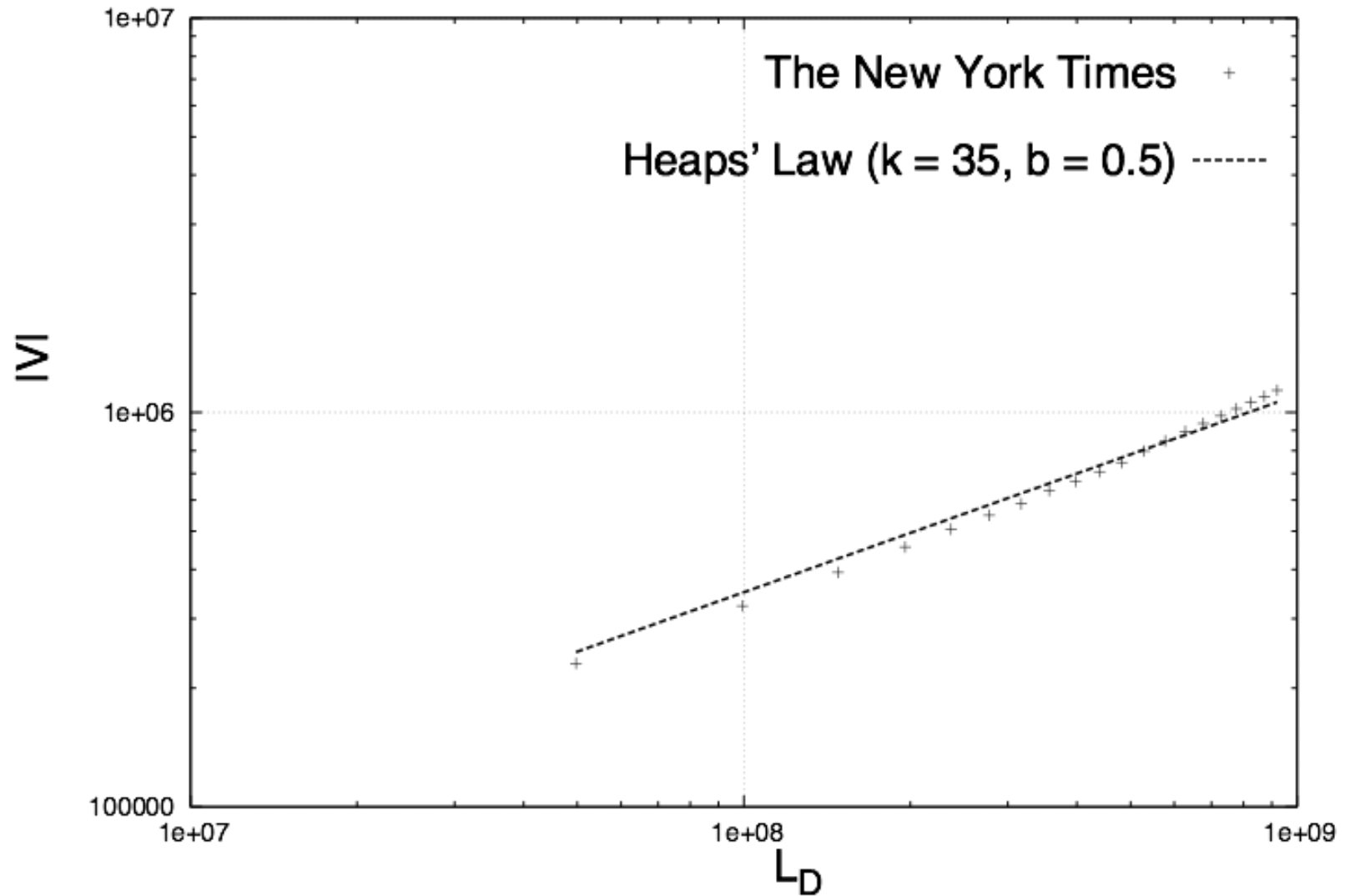
Die **Anzahl der verschiedenen Terme** $|V|$ in einer Dokumentkollektion (d.h. die Größe des Vokabulars) hängt wie folgt von der **Gesamtzahl der Termvorkommen**

$$|V| = k \left(\sum_{v \in V} cf(v) \right)^b$$

ab, mit kollektionsspezifischen Konstanten k und b .

Wir können daher erwarten, dass das Vokabular mit der Größe der Dokumentkollektion wächst.

Heaps Law



Stoppwörter

Ein **Stoppwort** für eine Dokumentenmenge D ist ein Wort, das als nicht signifikant für das Retrieval von Dokumenten aus D angesehen wird.

Beispiele für Stoppwörter im Englischen sind

the, a, be, may, like, on, should, . . .

Beispiele für Stoppwörter  im Deutschen sind

der, die, das, und, nicht, in, auf, . . .

Für beide Sprachen (und viele andere) gibt es fertige Stoppwortlisten.

Aber: Keine Suche nach Phrasen ohne vollständige Indexierung.

Stoppwort-Listen für Englisch

Stoppwortliste mit 7 Termen:

and an by from of the with

Stoppwortliste mit 425 Termen (Ausschnitt):

a about above across after again against
all almost alone along already also although
always among an and another any anybody
anyone anything anywhere are area areas around
as ask asked asking asks at away
b back backed backing backs be because
became become becomes been before began
behind ...

Stoppwörter

Die Stoppwort-Eigenschaft ist i.a. **kontextabhängig**, d.h. sie variiert von Dokumentenmenge zu Dokumentenmenge bzw. von Anwendungsbereich zu Anwendungsbereich.

Beispiel:

Bei der Suche nach

to be or not to be

sollten Stoppwörter besser nicht eliminiert werden, da sonst die leere Anfrage zurückbleiben würde.

Beispiel: „to be or not to be“ 2008

The screenshot shows a Google search interface from 2008. At the top left is the Google logo. To its right are navigation links: Web, Bilder, Groups, Verzeichnis, and News. Below these is a search bar containing the text 'to be or not to be' and a 'Suche' button. To the right of the search bar are links for 'Erweiterte Suche' and 'Einstellungen'. Below the search bar, there are radio buttons for search scope: 'Das Web' (selected), 'Seiten auf Deutsch', and 'Seiten aus Deutschland'. A message states: 'Die folgenden Wörter kommen sehr häufig vor und wurden daher in Ihrer Suchanfrage ignoriert: **to be to be**. [Einzelheiten]'. Another message says: 'Der kleingeschriebene Operator "or" wurde ignoriert. Benutzen Sie "OR", um nach entweder dem ersten oder dem zweiten der beiden Begriffe zu suchen. [Einzelheiten]'. A blue header bar displays 'Web Ergebnisse 1 - 10 von ungefähr 6.630.000 Seiten auf Deutsch für to be or not to be . (0,36 Sekunden)'. The first result is titled 'Mamis in Not' with a snippet: '... "Mamis in Not" hatte ich von Anfang an in meiner Linkliste, da die Seiten thematisch sehr gut zu einander passen. Birgitts Seiten werde ich nicht verändern. ...'. It includes the URL 'www.mamis-in-not.de/' and links for '33k', 'Im Cache', and 'Ähnliche Seiten'. The second result is titled 'Education is not for sale!' with a snippet: 'Education is not for sale! http://education-is-not-for-sale.org/'. It includes the URL 'www.education-is-not-for-sale.org/' and links for '1k', 'Im Cache', and 'Ähnliche Seiten'. At the bottom left, the status 'Fertig' is shown.

Google

Web Bilder Groups Verzeichnis News

to be or not to be Suche

[Erweiterte Suche](#)
[Einstellungen](#)

Suche: ☐ Das Web ☒ Seiten auf Deutsch ☐ Seiten aus Deutschland

Die folgenden Wörter kommen sehr häufig vor und wurden daher in Ihrer Suchanfrage ignoriert: **to be to be**. [[Einzelheiten](#)]

Der kleingeschriebene Operator "or" wurde ignoriert. Benutzen Sie "OR", um nach entweder dem ersten oder dem zweiten der beiden Begriffe zu suchen. [[Einzelheiten](#)]

Web Ergebnisse 1 - 10 von ungefähr 6.630.000 Seiten auf Deutsch für to be or not to be . (0,36 Sekunden)

[Mamis in Not](#) :

... "Mamis in **Not**" hatte ich von Anfang an in meiner Linkliste, da die Seiten thematisch sehr gut zu einander passen. Birgitts Seiten werde ich nicht verändern. ...

www.mamis-in-not.de/ - 33k - [Im Cache](#) - [Ähnliche Seiten](#)

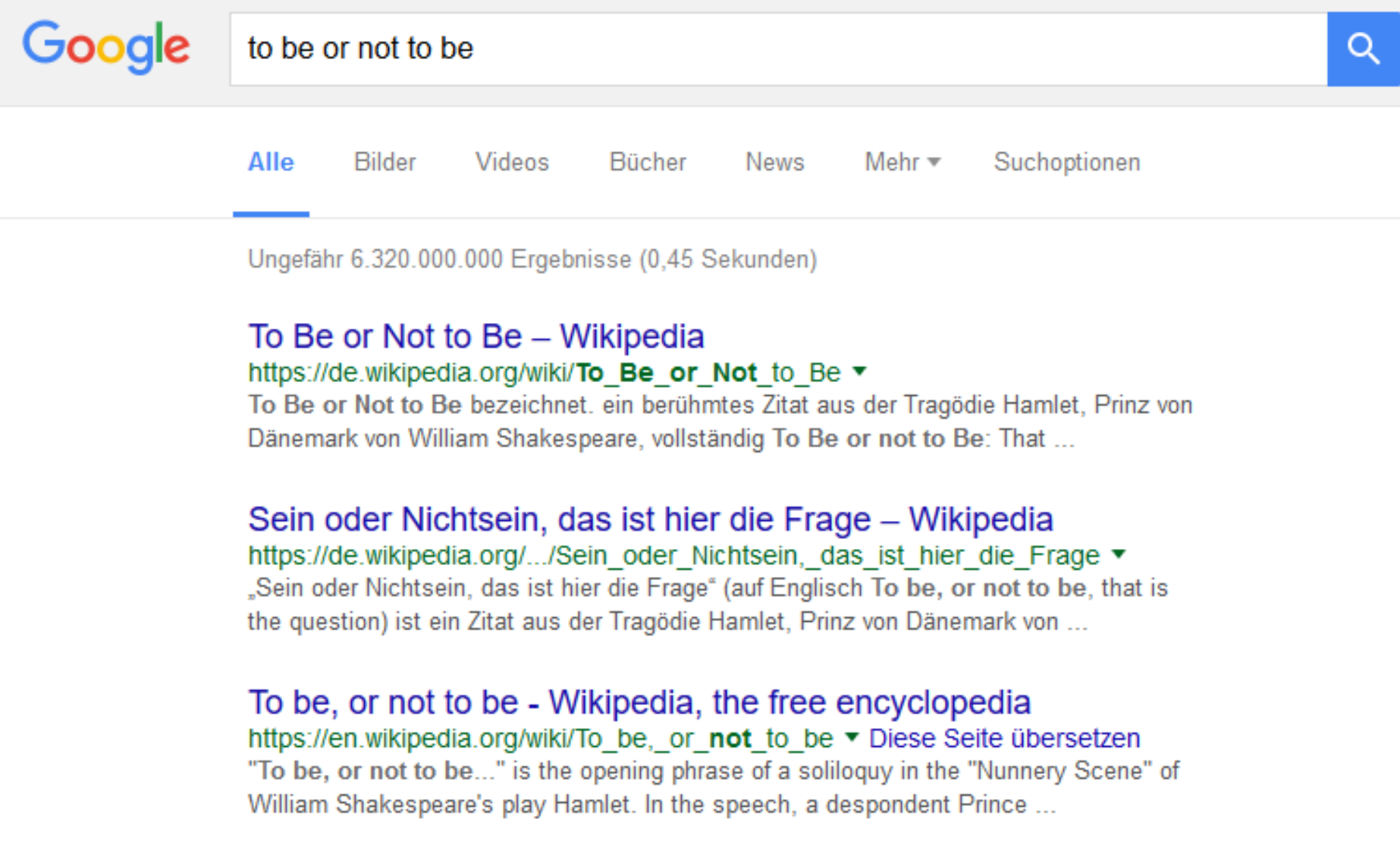
[Education is not for sale!](#) - [[Diese Seite übersetzen](#)]

Education is **not** for sale! <http://education-is-not-for-sale.org/>.

www.education-is-not-for-sale.org/ - 1k - [Im Cache](#) - [Ähnliche Seiten](#)

Fertig

Beispiel: „to be or not to be“ 2016



The image is a screenshot of a Google search interface. At the top left is the Google logo. To its right is a search bar containing the text "to be or not to be". A blue search button with a magnifying glass icon is on the far right of the search bar. Below the search bar, there are navigation tabs: "Alle" (which is underlined), "Bilder", "Videos", "Bücher", "News", "Mehr ▾", and "Suchoptionen". Below the tabs, it says "Ungefähr 6.320.000.000 Ergebnisse (0,45 Sekunden)". The first search result is titled "To Be or Not to Be – Wikipedia" in blue. Below the title is the URL "https://de.wikipedia.org/wiki/To_Be_or_Not_to_Be ▾" in green. The snippet below the URL reads: "To Be or Not to Be bezeichnet. ein berühmtes Zitat aus der Tragödie Hamlet, Prinz von Dänemark von William Shakespeare, vollständig To Be or not to Be: That ...". The second search result is titled "Sein oder Nichtsein, das ist hier die Frage – Wikipedia" in blue. Below the title is the URL "https://de.wikipedia.org/.../Sein_oder_Nichtsein,_das_ist_hier_die_Frage ▾" in green. The snippet below the URL reads: "„Sein oder Nichtsein, das ist hier die Frage“ (auf Englisch To be, or not to be, that is the question) ist ein Zitat aus der Tragödie Hamlet, Prinz von Dänemark von ...". The third search result is titled "To be, or not to be - Wikipedia, the free encyclopedia" in blue. Below the title is the URL "https://en.wikipedia.org/wiki/To_be,_or_not_to_be ▾" in green, followed by the text "Diese Seite übersetzen" in blue. The snippet below the URL reads: "„To be, or not to be...“ is the opening phrase of a soliloquy in the "Nunnery Scene" of William Shakespeare's play Hamlet. In the speech, a despondent Prince ...".

Stoppwörter

Die **Kontrolle des Vokabulars**, d.h. der Menge der Wörter, die als Index-Terme in Frage kommen, kann zu einer **Steigerung der Präzision** führen.

Andererseits kann ein reduziertes Vokabular eine **Reduktion des Recall** bewirken und aus Anwendersicht nicht immer leicht nachvollziehbare Anfrageergebnisse verursachen.

Aufgrund verbesserter Techniken zur Indexkompression und Anfrageoptimierung wird zunehmend auf die Elimination von Stoppwörtern verzichtet, zumal sie gelegentlich zur Anfragebearbeitung benötigt werden (z.B. in 'King of Denmark', 'Flüge nach Berlin').

Web-Suchmaschinen verwenden heute keine Stoppwort-Elimination.