Verteidigung der Diplomarbeit zur Erlangung des Grades eines
Diplom-Mediensystemwissenschaftlers
an der Fakultät Medien der Bauhaus-Universität Weimar

# Automatische Extraktion von Schlüsselwörtern aus Text

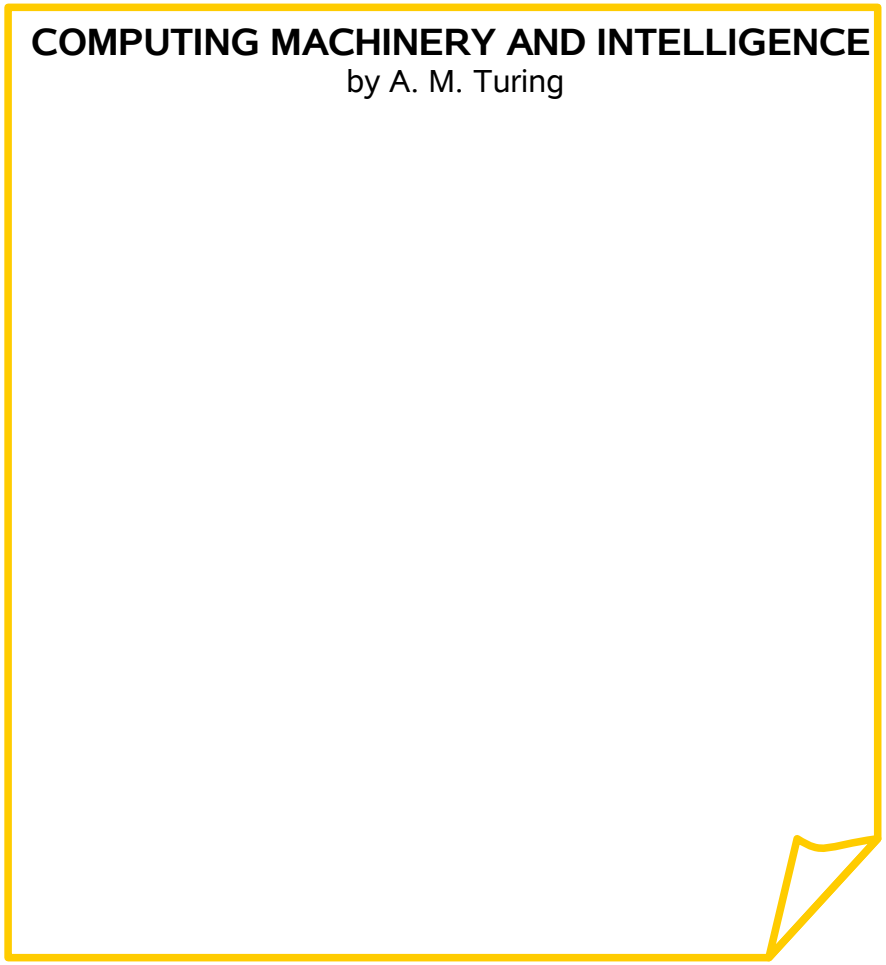
von Karsten Klüger

Weimar, 2006-06-29

# Gliederung

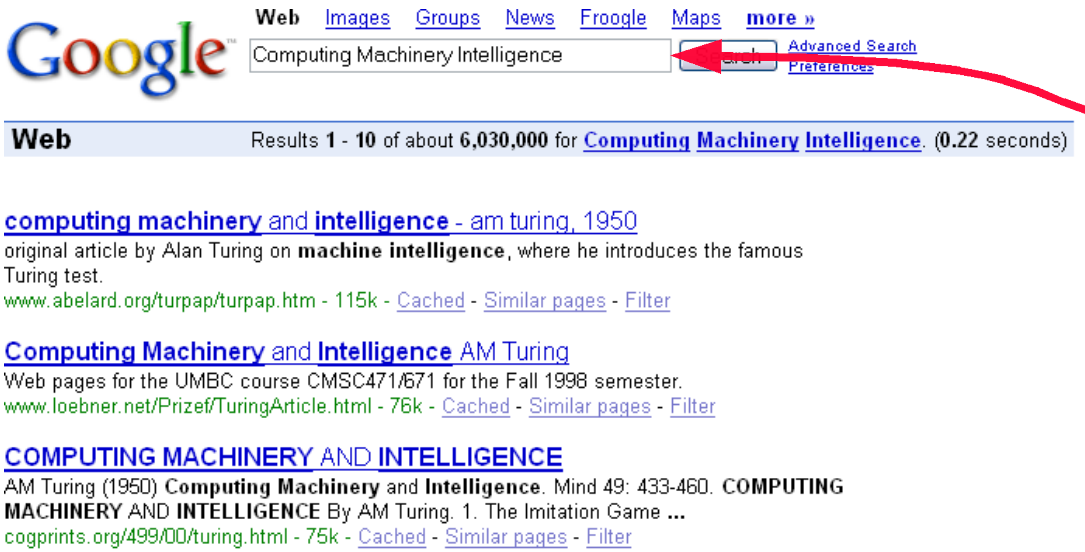
- Motivation
- Algorithmen zur Extraktion von Schlüsselwörtern
- Implementierung
- Evaluierung
- Ausblick
- Zusammenfassung

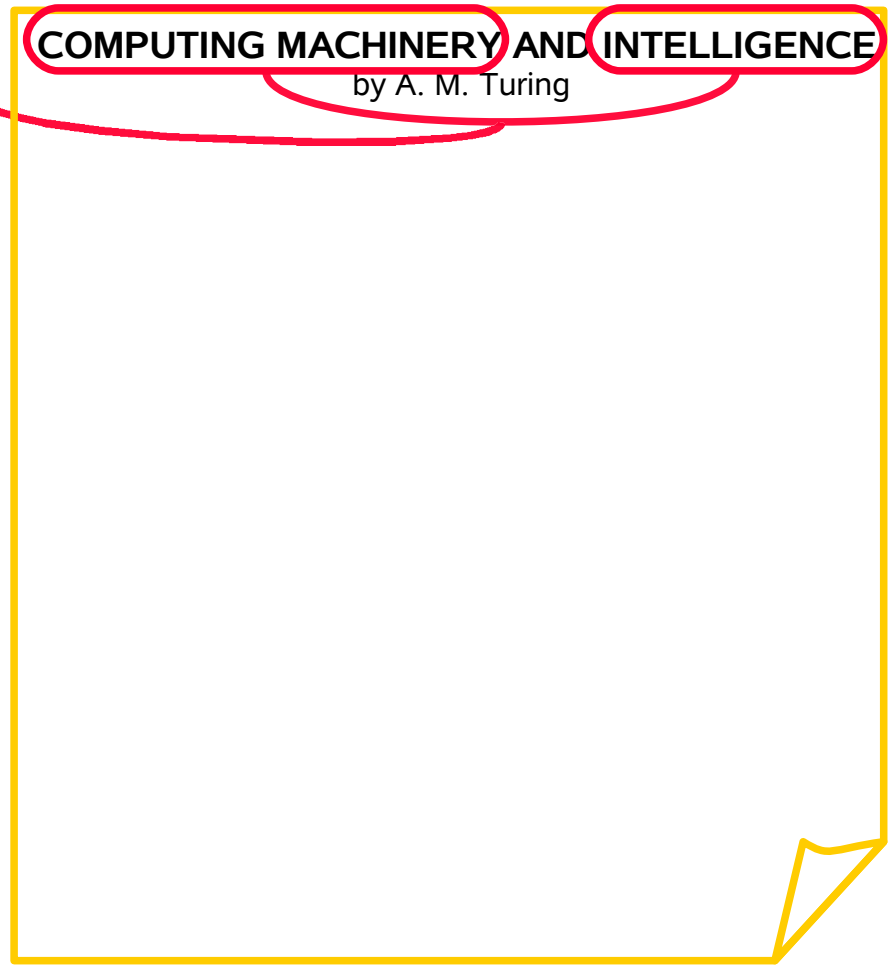
# Motivation




[Turing, 1950]

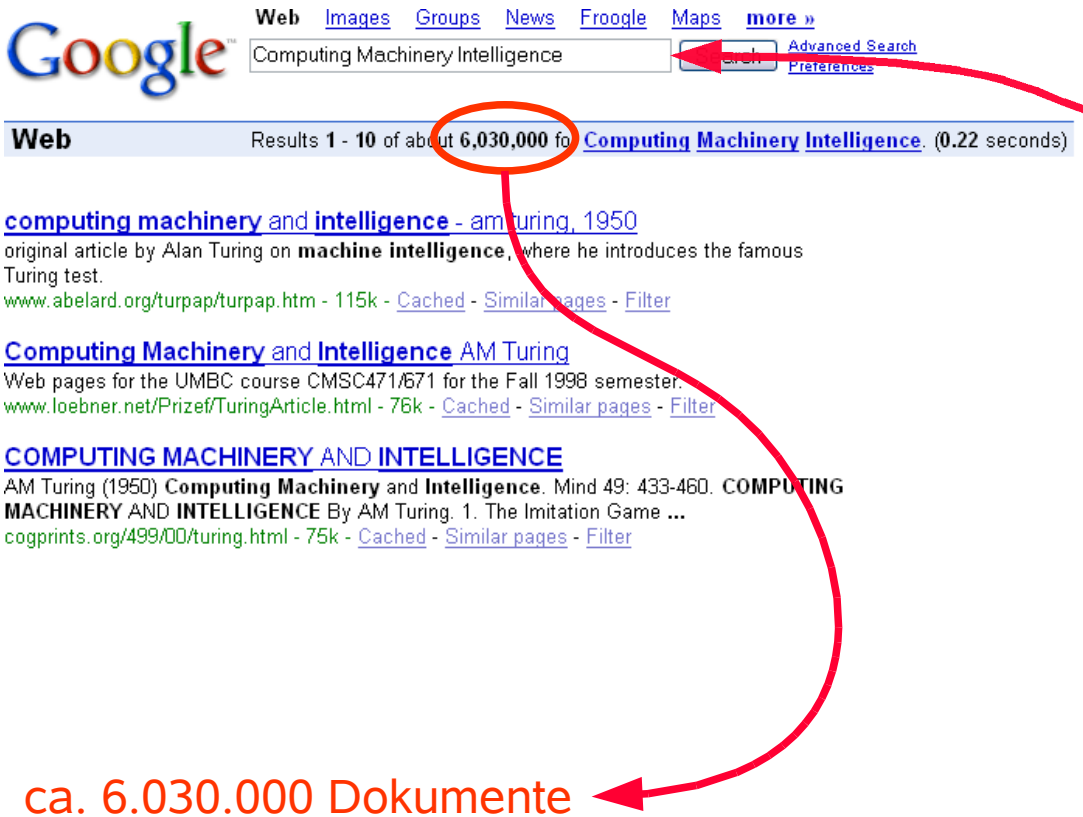
# Motivation



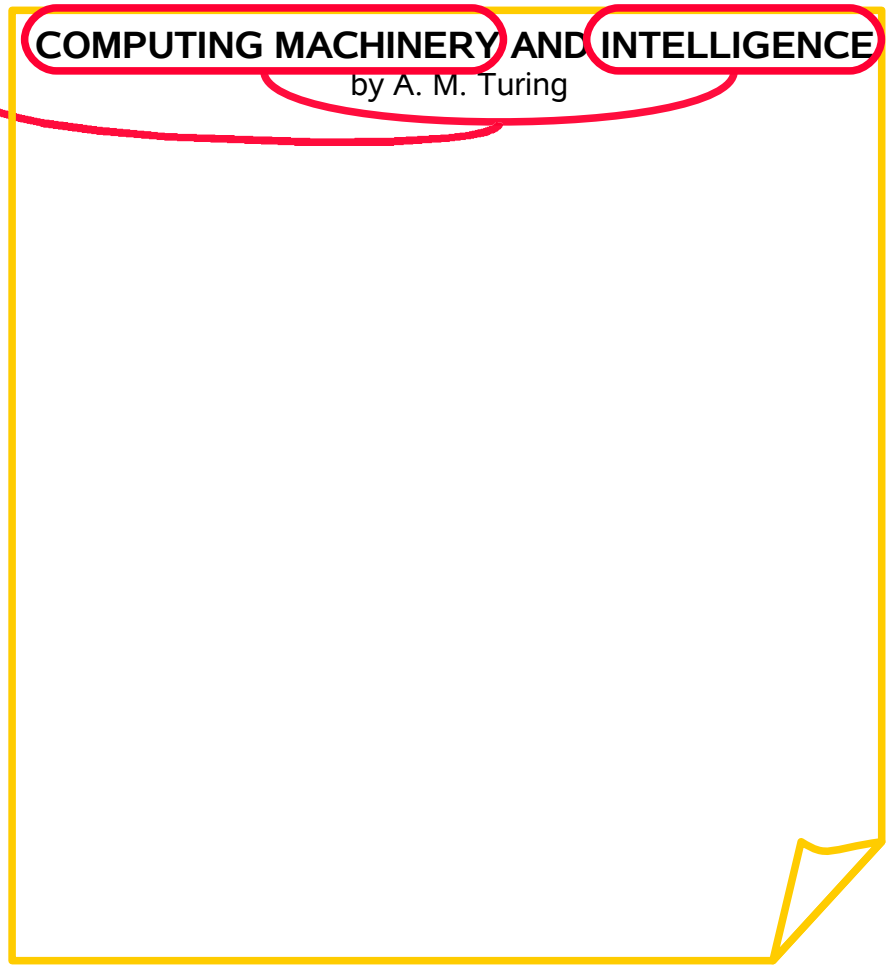




[Turing, 1950]

# Motivation




ca. 6.030.000 Dokumente




[Turing, 1950]

# Motivation


ca. 6.030.000 Dokumente (vorher)

**COMPUTING MACHINERY AND INTELLIGENCE**
by A. M. Turing

...
5. Universality of Digital Computers

The digital computers may be classified amongst the "discrete-state machines". These are the machines which move by sudden jumps or clicks from one quite definite state to another. These states are sufficiently different for the possibility of confusion between them to be ignored. Strictly speaking, there are no such machines. Everything really moves continuously. But there are many kinds of machine which can profitably be thought of as being discrete-state machines. ...

[Turing, 1950]

# Motivation


ca. 6.030.000 Dokumente (vorher)
ca. 228 Dokumente

**COMPUTING MACHINERY AND INTELLIGENCE**
by A. M. Turing

...
5. Universality of Digital Computers

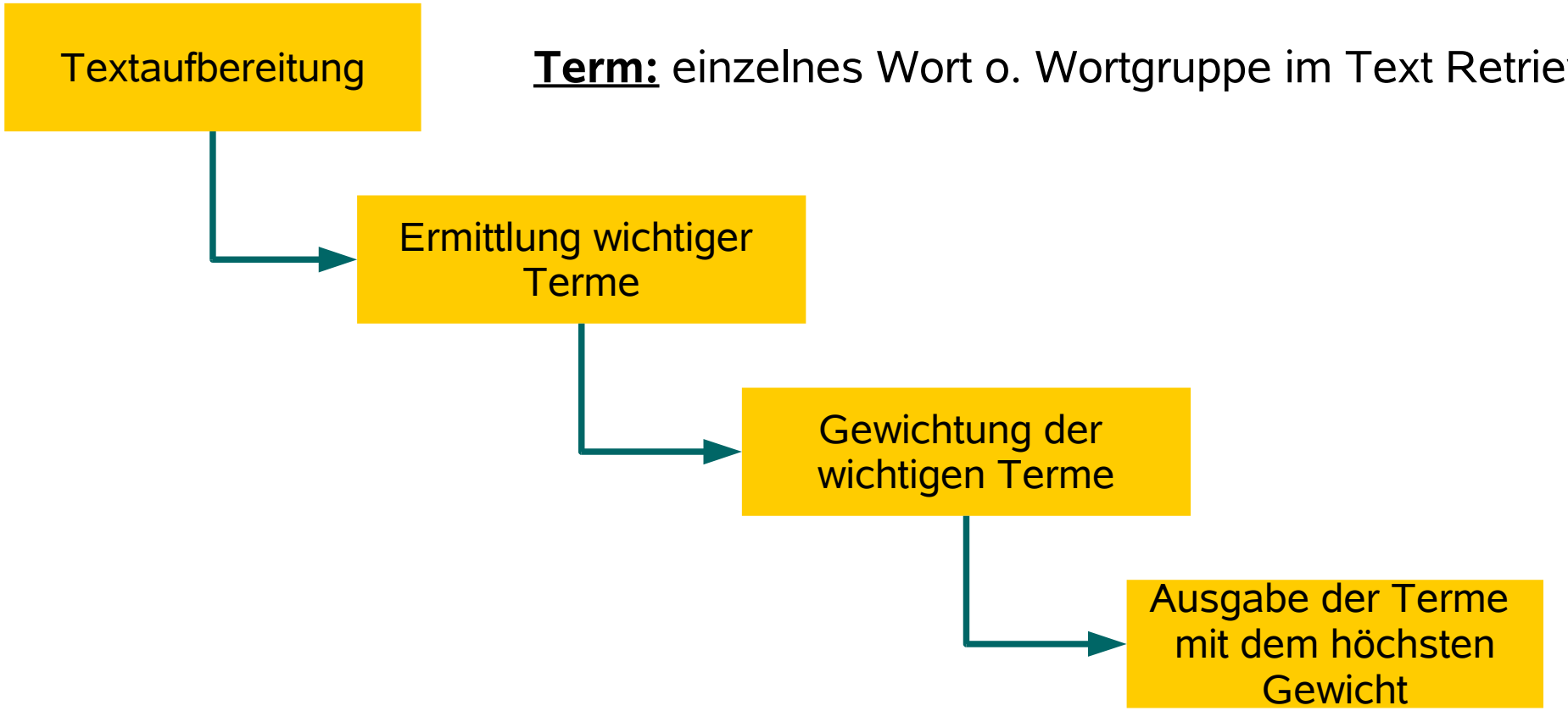
The digital computers may be classified amongst the "discrete-state machines". These are the machines which move by sudden jumps or clicks from one quite definite state to another. These states are sufficiently different for the possibility of confusion between them to be ignored. Strictly speaking, there are no such machines. Everything really moves continuously. But there are many kinds of machine which can profitably be thought of as being discrete-state machines. ...

[Turing, 1950]

# Aufgaben

- Ermittlung geeigneter Algorithmen
- Implementieren
- Erstellen eines Evaluierungskorpus
- Evaluierung der Algorithmen unter Berücksichtigung der Anwendung in fokussierter Suche
- ggf. Aufzeigen von Verbesserungsmöglichkeiten

# Grundlegender Ablauf




**Term:** einzelnes Wort o. Wortgruppe im Text Retrieval

# Textaufbereitung

- Lexikalische Analyse:
  - Überführung eines Textes in einzelne Wörter
  - Entfernung von:
    - Trennzeichen
    - Interpunktionszeichen
    - Steuerzeichen (z.B. Zeilenumbruch)
    - Zahlen
  - Konvertierung in Kleinschreibung

...
5. Universality of Digital Computers↵
↵
The digital computers may be classified amongst the "discrete-state machines". These are the machines which move by sudden jumps or clicks from one quite definite state to another. These states are sufficiently different for the possibility of confusion between them to be ignored. Strictly speaking, there are no such machines. Everything really moves continuously. But there are many kinds of machine which can profitably be thought of as being discrete-state machines. ...

# Textaufbereitung

- Lexikalische Analyse:
  - Überführung eines Textes in einzelne Wörter
  - Entfernung von:
    - Trennzeichen
    - Interpunktionszeichen
    - Steuerzeichen (z.B. Zeilenumbruch)
    - Zahlen
  - Konvertierung in Kleinschreibung

... universality of digital computers the digital computers may be classified amongst the discrete state machines these are the machines which move by sudden jumps or clicks from one quite definite state to another these states are sufficiently different for the possibility of confusion between them to be ignored strictly speaking there are no such machines everything really moves continuously but there are many kinds of machine which can profitably be thought of as being discrete state machines ...

# Textaufbereitung

- Stoppworteliminierung
  - Häufige und gleich verteilt vorkommende Wörter, z.B. Artikel, Konjunktionen und Präpositionen
  - Tragen keine Information
    → Entfernung = Rauschreduktion
  - etwa 20-30% der Wörter in einem Text
  - Anhand sprachspezifischer Listen

... universality of digital computers the digital computers may be classified amongst the discrete state machines these are the machines which move by sudden jumps or clicks from one quite definite state to another these states are sufficiently different for the possibility of confusion between them to be ignored strictly speaking there are no such machines everything really moves continuously but there are many kinds of machine which can profitably be thought of as being discrete state machines ...

# Textaufbereitung

- Stoppworteliminierung
  - Häufige und gleich verteilt vorkommende Wörter, z.B. Artikel, Konjunktionen und Präpositionen
  - Tragen keine Information
    → Entfernung = Rauschreduktion
  - etwa 20-30% der Wörter in einem Text
  - Anhand sprachspezifischer Listen

... universality digital computers
digital computers classified
discrete state machines
machines
sudden jumps clicks
definite state states
sufficiently different possibility
confusion ignored
strictly speaking
machines moves
continuously kinds
machine profitably thought
discrete state machines ...

# Textaufbereitung

- Stammformreduktion (stemming)
  - Wörter in verschiedenen Formen (z.B. Einzahl / Mehrzahl)
  - Rückführung der Wörter auf Wortstamm durch Entfernung der flexivischen Formveränderungen
  - Regelbasierte Ansätze und statistische Verfahren (z.B. Porter-Stemmer)

... universality digital computers
digital computers classified
discrete state machines
machines
sudden jumps clicks
definite state states
sufficiently different possibility
confusion ignored
strictly speaking
machines moves
continuously kinds
machine profitably thought
discrete state machines ...

# Textaufbereitung

- Stammformreduktion (stemming)
  - Wörter in verschiedenen Formen (z.B. Einzahl / Mehrzahl)
  - Rückführung der Wörter auf Wortstamm durch Entfernung der flexivischen Formveränderungen
  - Regelbasierte Ansätze und statistische Verfahren (z.B. Porter-Stemmer)

... univers digit comput
digit comput classifi
discret state machin
machin
sudden jump click
definit state state
suffici differ possibl
confus ignor
strict speak
machin move
continu kind
machin profit thought
discret state machin ...

# Algorithmen: KEA - [Witten et al., 1999]

- Referenzalgorithmus
- Basis: Maschineller Lernalgorithmus zur Klassifikation
  - 1. statistisches Termgewicht (*tf.idf* )
  - *2.* erste Auftreten des Terms im Dokument (*first occurrence* )
- Feature-Vektor für jeden Term
- Klassifikation durch naiven Bayes Klassifizierer
- Training anhand einer Menge von Trainingsdokumenten mit bereits vom Autor vergebenen Schlüsselwörtern
  - Kea – MyColl: auf der Testkollektion
  - Kea – CSTR: auf in Distribution gelieferten Trainingsdokumenten

# Algorithmen: RSP - [Tseng, 1998]

- Basis: häufig auftretende Wortgruppen (n-Gramme)
- Gewichtung der Schlüsselwörter:
  - Größte Häufigkeit (TF)
    1) „digital computer“
    2) „discrete state machine“
  - Erstes Auftreten (FO)
    1) „digital computer“
    2) „discrete state machine“

... universality of digital computers the digital computers may be classified amongst the discrete state machines these are the machines which move by sudden jumps or clicks from one quite definite state to another these states are sufficiently different for the possibility of confusion between them to be ignored strictly speaking there are no such machines everything really moves continuously but there are many kinds of machine which can profitably be thought of as being discrete state machines ...

# Algorithmen: B&C - [Barker & Cornacchia, 2000]

- Basis: Nominalphrasen mit häufig auftretendem nominalen Kern (Substantiv)
- Gewichtung der Schlüsselwörter:
  - Größte Häufigkeit (TF)
    1) „discrete state machine“
    2) „digital computer“
  - Erstes Auftreten (FO)
    1) „digital computer“
    2) „discrete state machine“

... universality of digital computers the digital computers may be classified amongst the discrete state machines these are the machines which move by sudden jumps or clicks from one quite definite state to another these states are sufficiently different for the possibility of confusion between them to be ignored strictly speaking there are no such machines everything really moves continuously but there are many kinds of machine which can profitably be thought of as being discrete state machines ...

# Algorithmen: Cooccurrence - [Matsuo & Ishizuka, 2003]

- Basis: Kookkurrenzen und deren Verteilungseigenschaften im Text
- Kookkurrenz: gemeinsames Auftreten zweier Terme
- Annahme: kookkurrente Terme wichtig, wenn häufiges Auftreten im Dokument
- Untersuchung der Verteilungs-eigenschaften der kookkurrenten Terme durch Chi-Quadrat-Verteilungstest

...
universality of digital computers
the digital computers may be classified amongst the discrete state machines
these are the machines which move by sudden jumps or clicks from one quite definite state to another
these states are sufficiently different for the possibility of confusion between them to be ignored
strictly speaking there are no such machines
everything really moves continuously
but there are many kinds of machine which can profitably be thought of as being discrete state machines
...

# Algorithmen: Cooccurrence

- Kookkurrenzmatrix (N×N-Matrix)

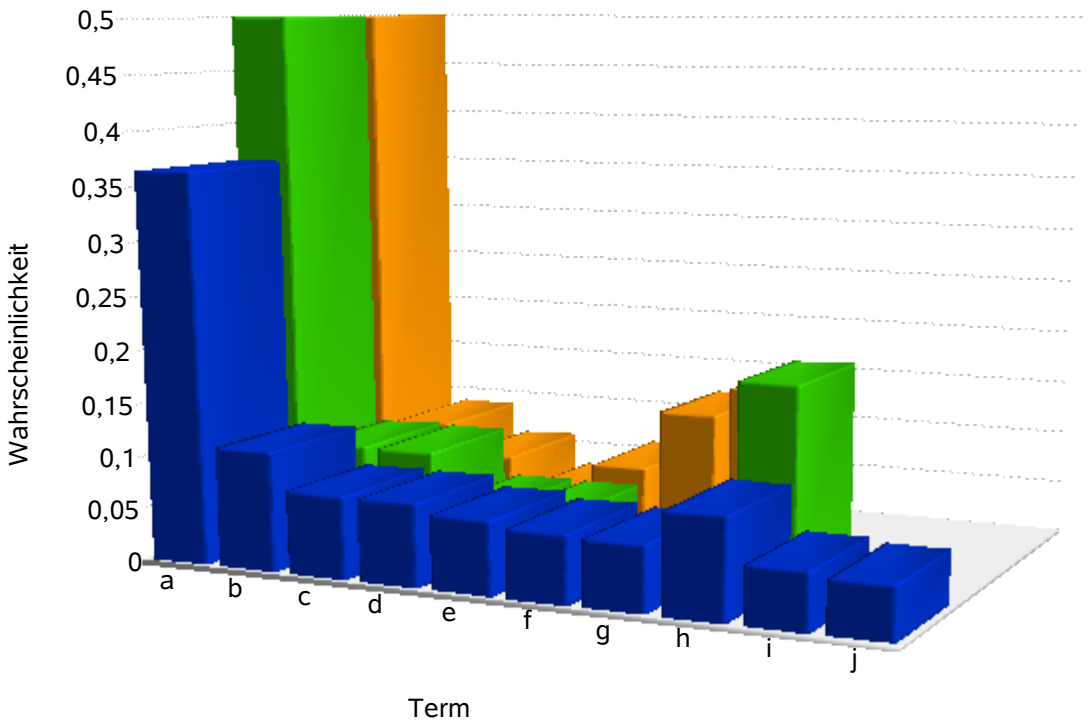
| | machine | computer | question | digital | answer | game | argument | make | state | number | ... | imitation | digital computer | kind | make |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **machine** | - | 30 | 26 | 19 | 18 | 12 | 12 | 17 | 22 | 9 | ... | 3 | 13 | 11 | 17 |
| **computer** | 30 | 0 | 5 | 50 | 6 | 11 | 1 | 3 | 2 | 3 | ... | 5 | 40 | 2 | 3 |
| **question** | 26 | 5 | - | 4 | 23 | 7 | 0 | 2 | 0 | 0 | ... | 5 | 4 | 2 | 2 |
| **digital** | 19 | 50 | 4 | - | 3 | 7 | 1 | 1 | 0 | 4 | ... | 3 | 35 | 1 | 1 |
| **answer** | 18 | 6 | 23 | 3 | - | 7 | 1 | 2 | 1 | 0 | ... | 3 | 3 | 1 | 2 |
| **game** | 12 | 11 | 7 | 7 | 7 | - | 2 | 4 | 0 | 0 | ... | 18 | 6 | 0 | 4 |
| **argument** | 12 | 1 | 0 | 1 | 1 | 2 | - | 5 | 1 | 0 | ... | 2 | 1 | 1 | 5 |
| **make** | 17 | 3 | 2 | 1 | 2 | 4 | 5 | - | 0 | 0 | ... | 2 | 0 | 4 | 0 |
| **state** | 22 | 2 | 0 | 0 | 1 | 0 | 1 | 0 | - | 7 | ... | 1 | 0 | 0 | 0 |
| **number** | 9 | 3 | 0 | 4 | 0 | 0 | 0 | 0 | 7 | - | ... | 0 | 2 | 0 | 0 |

*Tabelle 2: Kookkurrenzmatrix*

# Algorithmen: Cooccurrence

- Berechnung der $\chi^2$-Werte
  - **1.)** Ist Term $t$ unabhängig von der Menge der häufigsten Terme $t'$, dann ist die Verteilung der Kookkurrenz von $t$ und $t'$ ähnlich der Verteilung der Auftrittswahrscheinlichkeit $t'$.

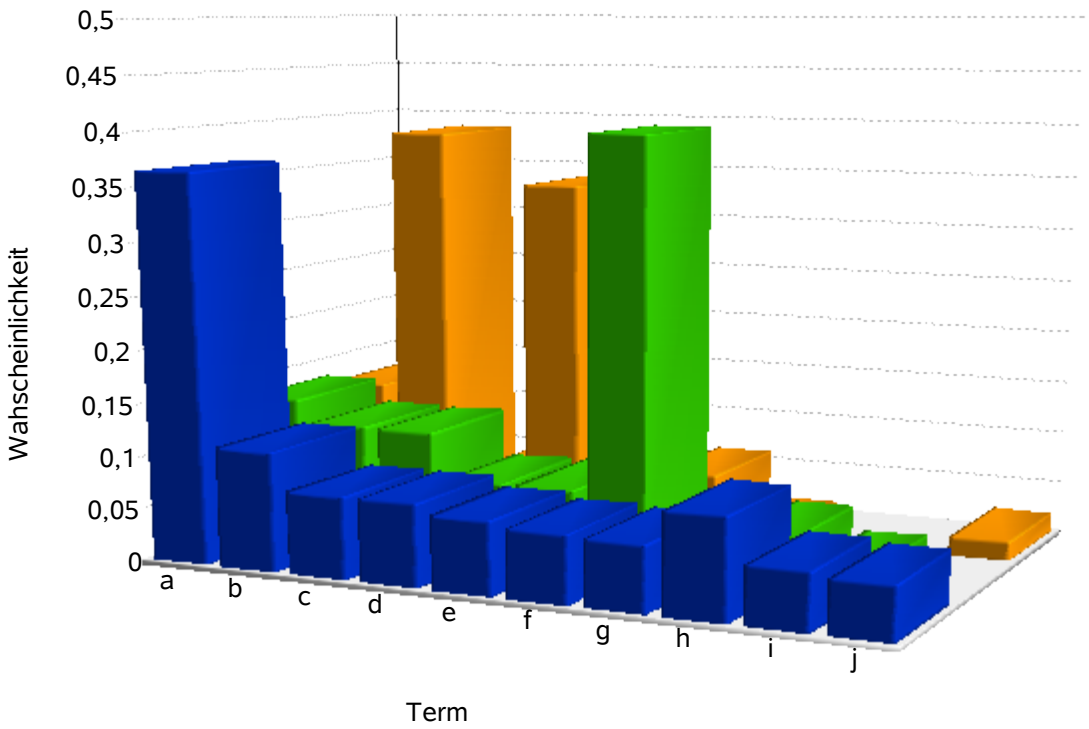
häufige Terme
„kind“
„make“



# Algorithmen: Cooccurrence

- Berechnung der $\chi^2$-Werte
  - **2.)** Steht ein Term $t$ in einer semantischen Beziehung zu $t'$, dann ist das gemeinsame Auftreten von $t$ und $t'$ höher als erwartet, d.h. die Verteilung ist abweichend.

häufige Terme
„imitation“
„digital computer“



# Algorithmen: Cooccurrence

- Berechnung der $\chi^2$-Werte
  - $n_t$ : absolute Anzahl der Kookkurrenzen von $t$ mit den häufigen Termen $T'$
  - $p_{t'}$ : Wahrscheinlichkeit des Auftretens eines häufigen Terms $t' \in T'$

$$\chi^2(t) = \sum_{t' \in T'} \frac{(f(t,t') - n_t p_{t'})^2}{n_t p_{t'}}$$

  - $n_t p_{t'}$ : erwartete Häufigkeit der Kookkurrenzen von $t$ und $t'$
  - $f(t,t')$ : beobachte Häufigkeit der Kookkurrenzen von $t$ und $t'$

**hoher Wert für $\chi^2(t)$ → starke Abweichung → hohe Relevanz**

# Algorithmen: Cooccurrence

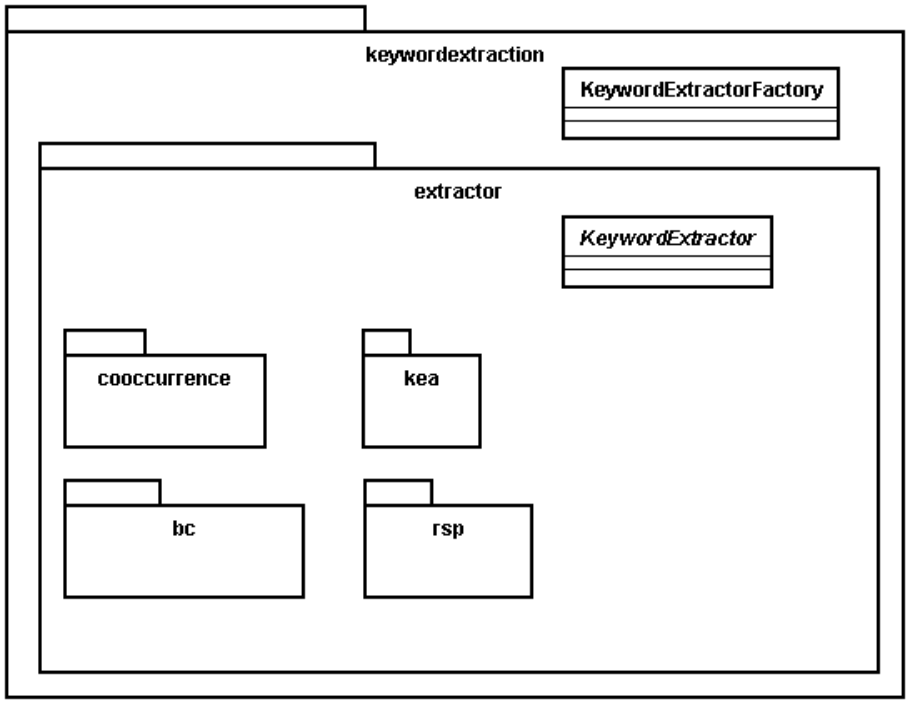
- 4. Ausgabe

| Rang | $\chi^2$ | Schlüsselwort | Häufigkeit |
|---|---|---|---|
| 1 | 380,4 | digital computer | 63 |
| 2 | 259,7 | storage capacity | 11 |
| 3 | 202,5 | imitation game | 16 |
| 4 | 174,4 | discrete-state machine | 203 |
| 5 | 132,2 | human mind | 2 |
| 6 | 94,1 | universality | 6 |
| 7 | 93,7 | logic | 10 |
| 8 | 82,0 | property | 11 |

*Tabelle 3: die ersten acht Schlüsselwörter*

# Implementierung

- In Java implementiert
- Paket im Aitools-Framework
- Fabriklasse für einfachen Zugriff



# Evaluierung

- Evaluierungskorpus
  - 250 wiss. Dokumente (PDF)
  - Mit Schlüsselwörtern ausgezeichnet
  - Automatisch in XML konvertiert
  - 170 Testdokumente
  - 80 Trainingsdokumente
  - ∅ Schlüsselwortanzahl: 10,5

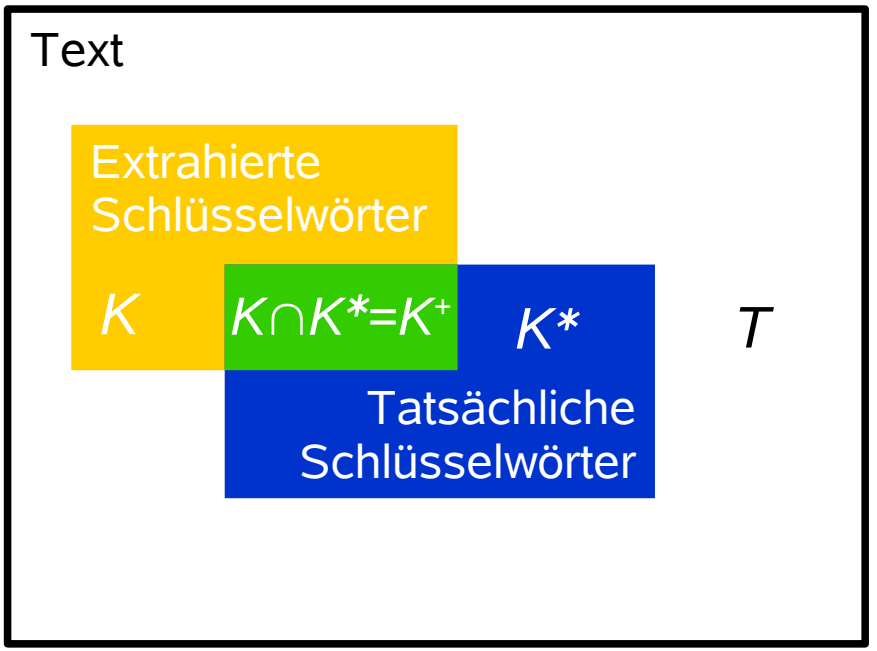
```
<?xml version='1.0' encoding='iso-8859-1'?>
<document>
 <keywords>
  dynamic load-balancing, ecological simulations,
  reaction-migration systems, parallelism, remapping
 </keywords>
 <text>
  dynamic load-balancing strategies for parallel implementations
  of reaction-evolution-migration systems
  mark smith
  edinburgh parallel computing centre, university of edinburgh
  kings buildings, edinburgh, 3jz, uk
  31st july
  <abstract>
  we introduce reaction-evolution-migration systems and explain
  their importance in the scientific field. details are given of
  parallel implementations of such systems, and how a naive
  ...
  </abstract>
  introduction
  reaction-evolution-migration systems are an adaption of
  standard spatial reaction-migration models used in many
  fields of science. In pioneering paper from alan turing studied
  these systems in the generalised case of linear interactions
  ...
 </text>
</document>
```

- **∅ Anteil der Schlüsselwörter im Text: 97%**

# Evaluierung

- Bewertung von Retrieval-Ergebnissen
  - Klassifikationsmöglichkeiten

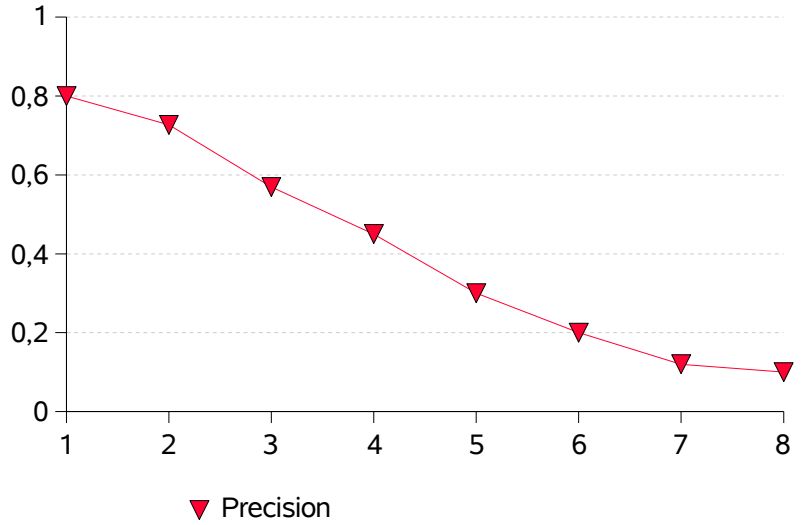
| | **Schlüsselwörter** | **keine Schlüsselwörter** |
|---|---|---|
| **Extrahierte Terme** | $\lvert K^{+}\rvert = a$ | $\lvert K \backslash K^{*}\rvert = b$ |
| **Nicht extrahierte Terme** | $\lvert K^{*} \backslash K\rvert = c$ | $\lvert T \backslash ( K \cup K^{*}) \rvert = d$ |



# Evaluierung

- Bewertung von Retrieval-Ergebnissen
  - Bewertungsgrößen

| | Schlüsselwörter | keine Schlüsselwörter |
|---|---|---|
| Extrahierte Terme | $\lvert K^{+}\rvert = a$ | $\lvert K \backslash K^{*}\rvert = b$ |
| Nicht extrahierte Terme | $\lvert K^{*} \backslash K\rvert = c$ | $\lvert T \backslash (K \cup K^{*})\rvert = d$ |

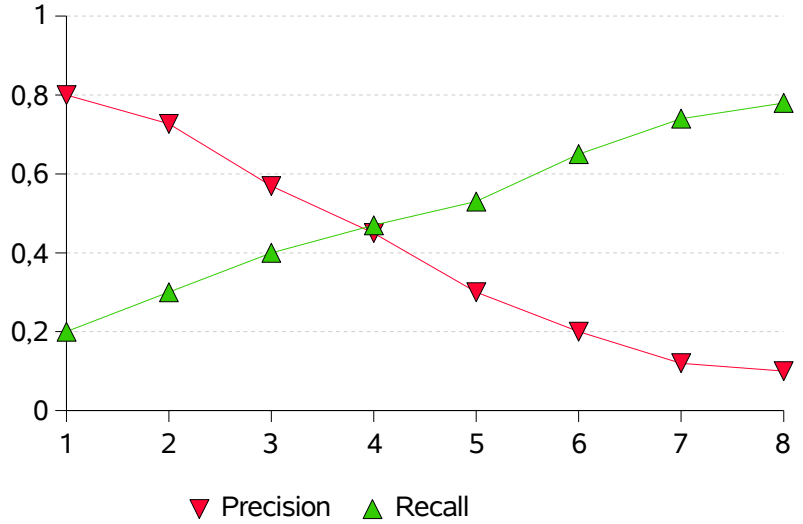



$$Precision = \frac{a}{a + b} \quad \in [0,1]$$

# Evaluierung

- Bewertung von Retrieval-Ergebnissen
  - Bewertungsgrößen

| | **Schlüsselwörter** | **keine Schlüsselwörter** |
|---|---|---|
| **Extrahierte Terme** | $\lvert K^{+}\rvert = a$ | $\lvert K \backslash K^{*}\rvert = b$ |
| **Nicht extrahierte Terme** | $\lvert K^{*} \backslash K\rvert = c$ | $\lvert T \backslash (K \cup K^{*})\rvert = d$ |




$$Precision = \frac{a}{a + b} \quad \in [0,1]$$

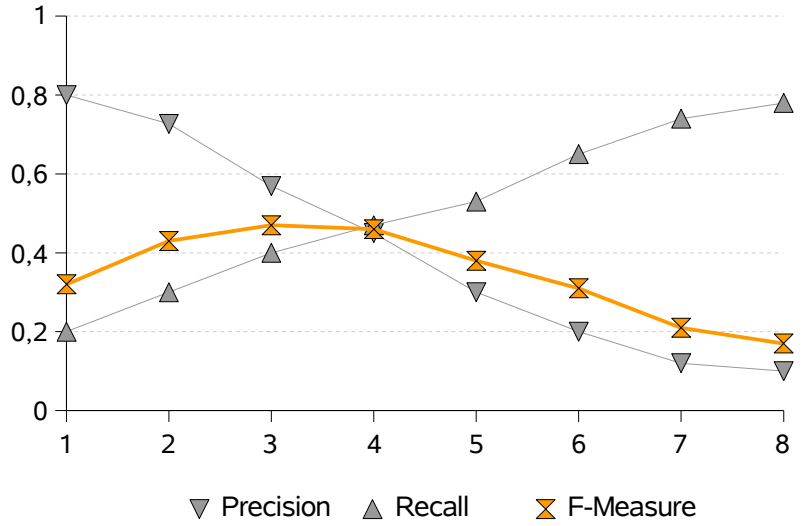
$$Recall = \frac{a}{a + c} \quad \in [0,1]$$

# Evaluierung

- Bewertung von Retrieval-Ergebnissen
  - Bewertungsgrößen

$$F\text{-}Measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad \in [0,1]$$

  - Harmonisches Mittel
  - Ziel: Maximierung des F-Measures

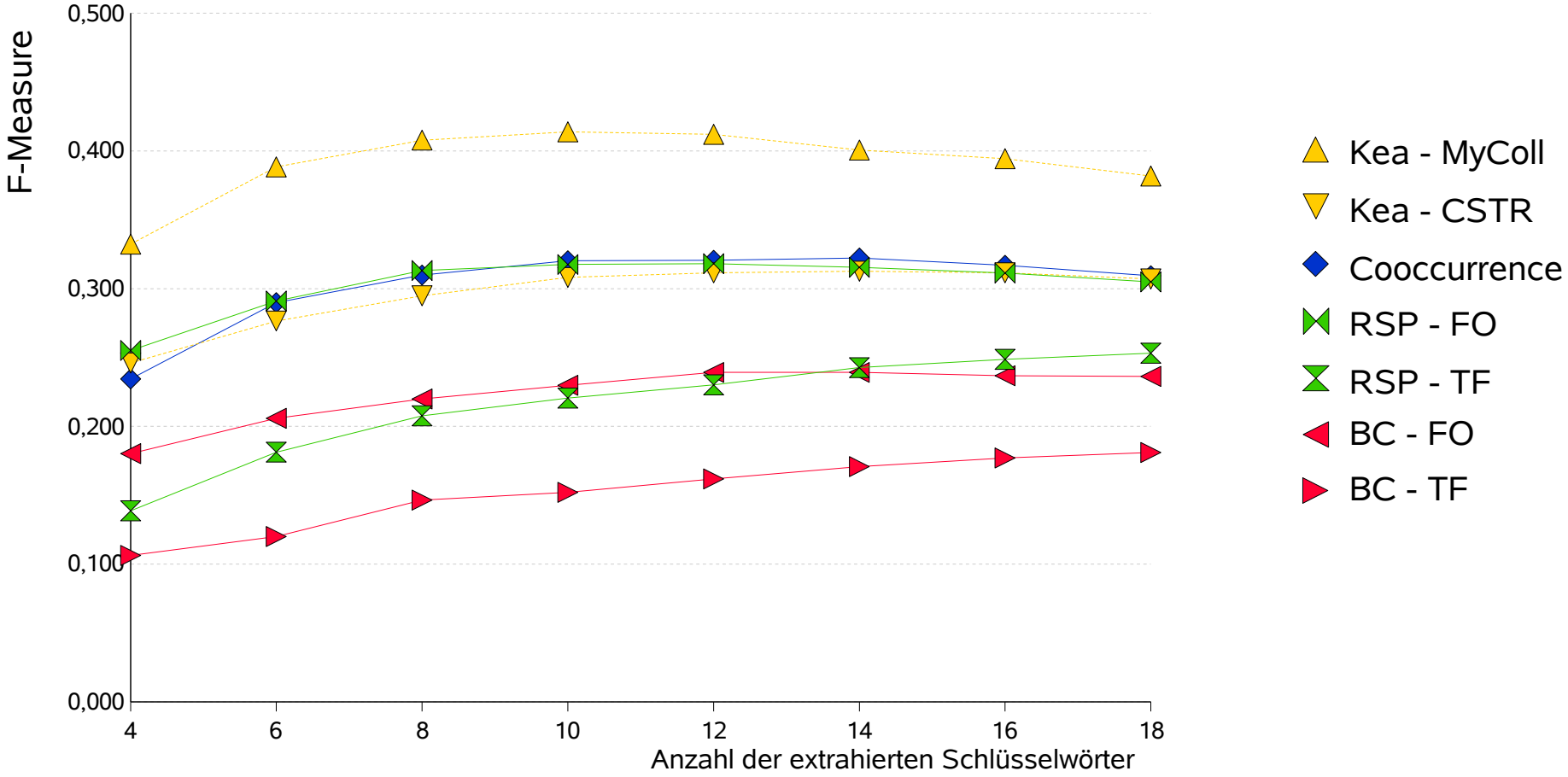


# Evaluierung

- Ergebnisse
  - Vier Experimente
    - → Variable Extraktion aus langem Text
    - → Variable Extraktion aus kurzem Text
    - → Klassifikationsrate für langen Text
    - → Klassifikationsrate für kurzen Text

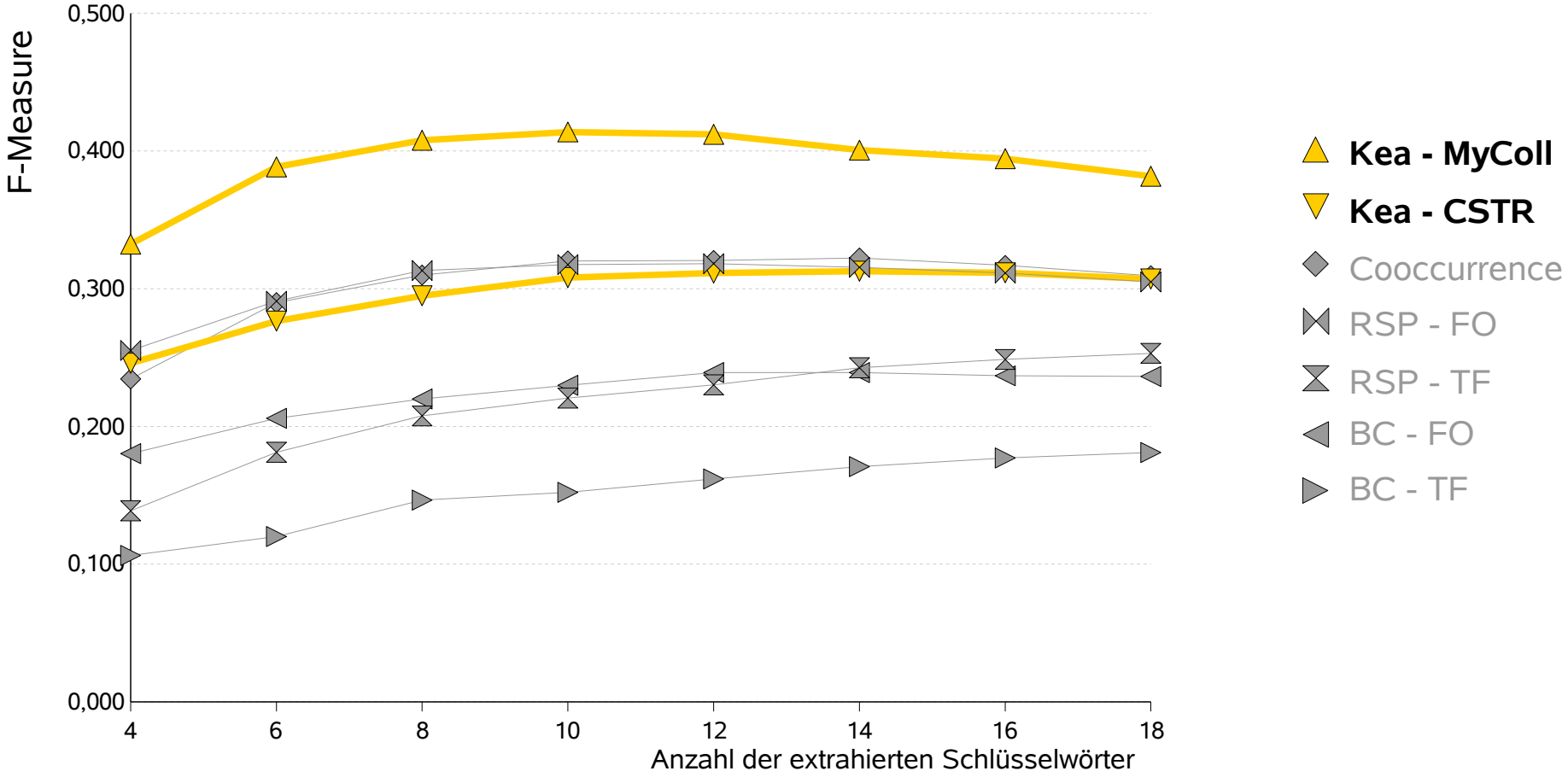
# Evaluierung

- Ergebnisse: Variable Extraktion aus langem Text



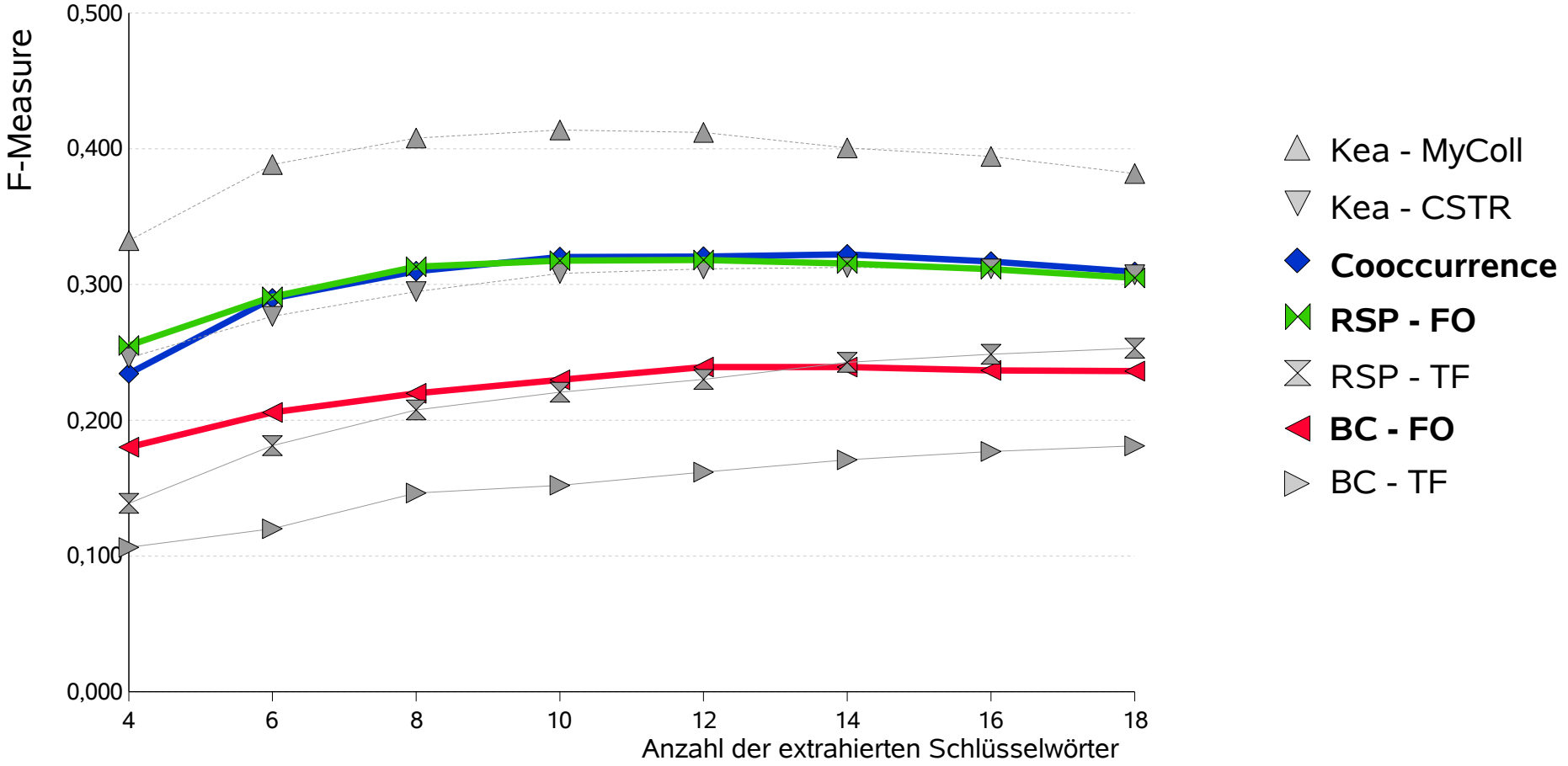
# Evaluierung

- Ergebnisse: Variable Extraktion aus langem Text



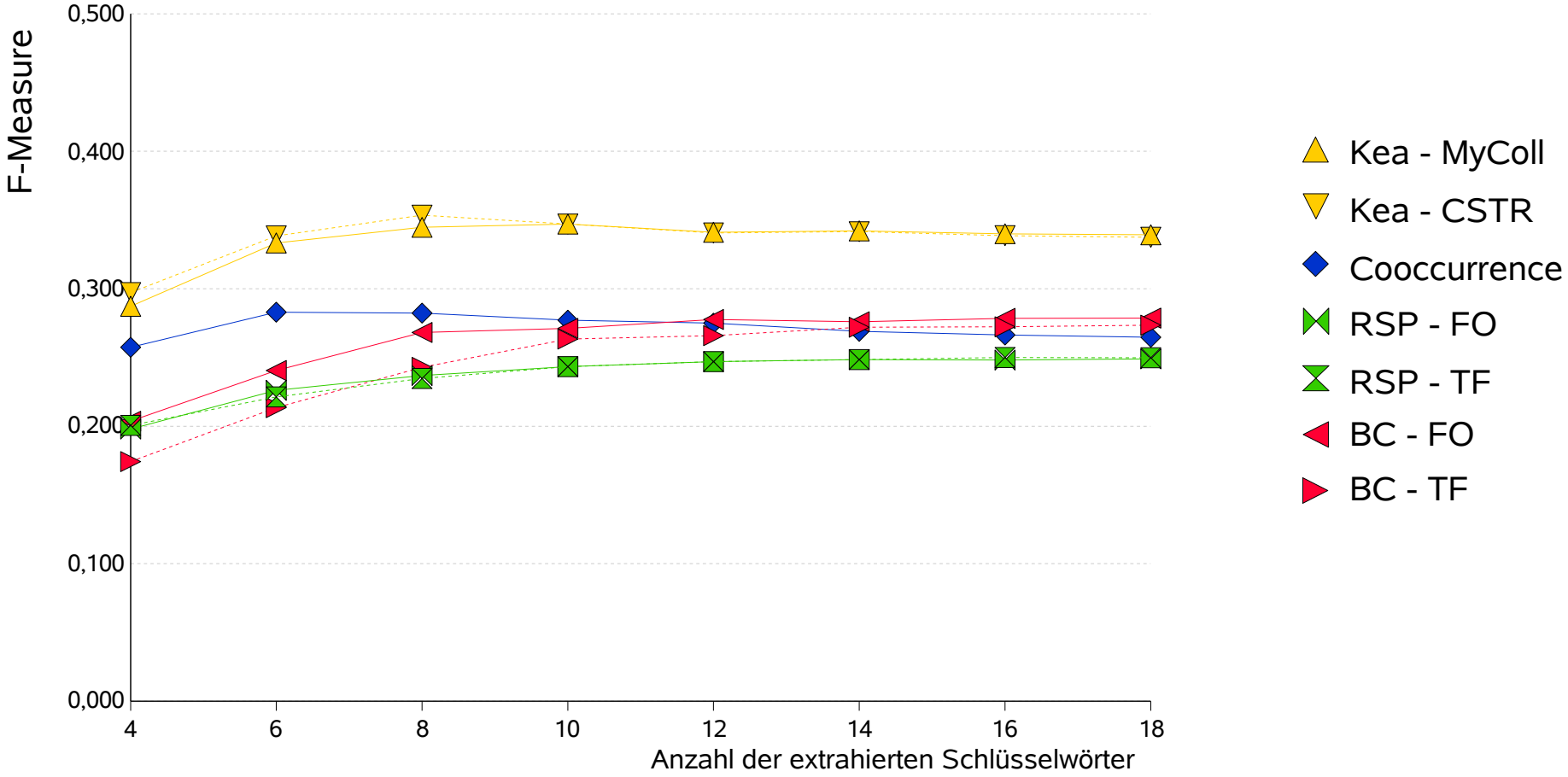
# Evaluierung

- Ergebnisse: Variable Extraktion aus langem Text



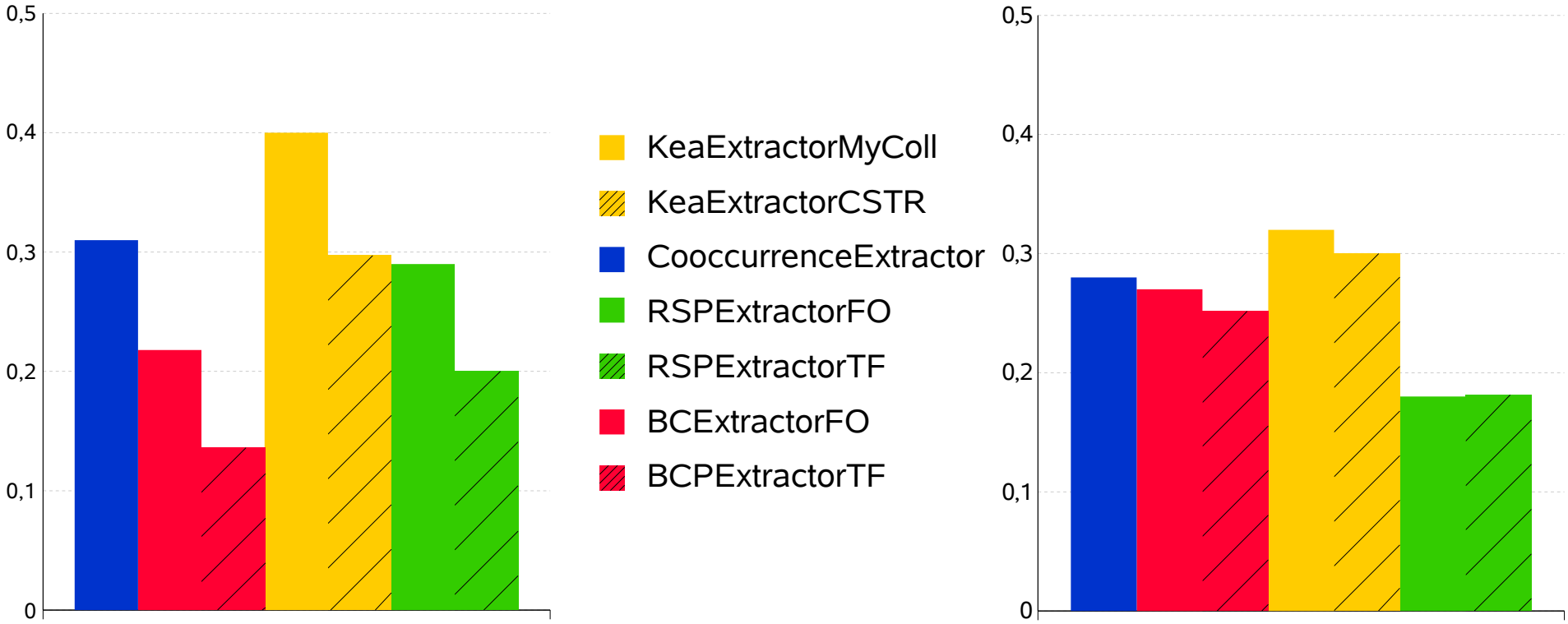
# Evaluierung

- Ergebnisse: Variable Extraktion aus kurzem Text



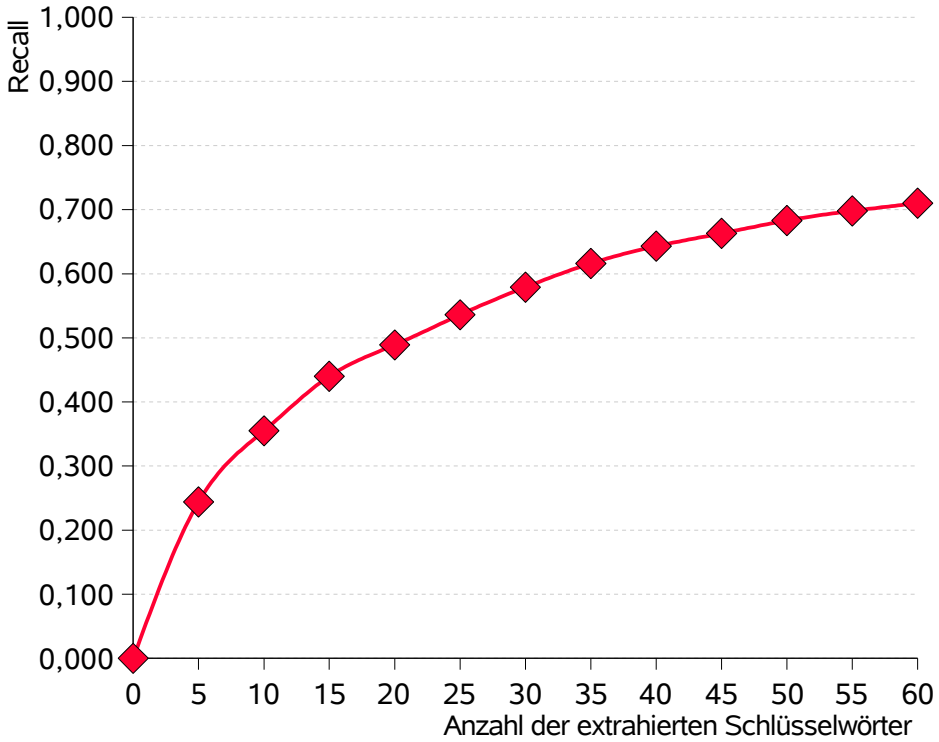
# Evaluierung

- Ergebnisse: Klassifikationsrate
  - Langer Text
  - Kurzer Text



# Ausblick

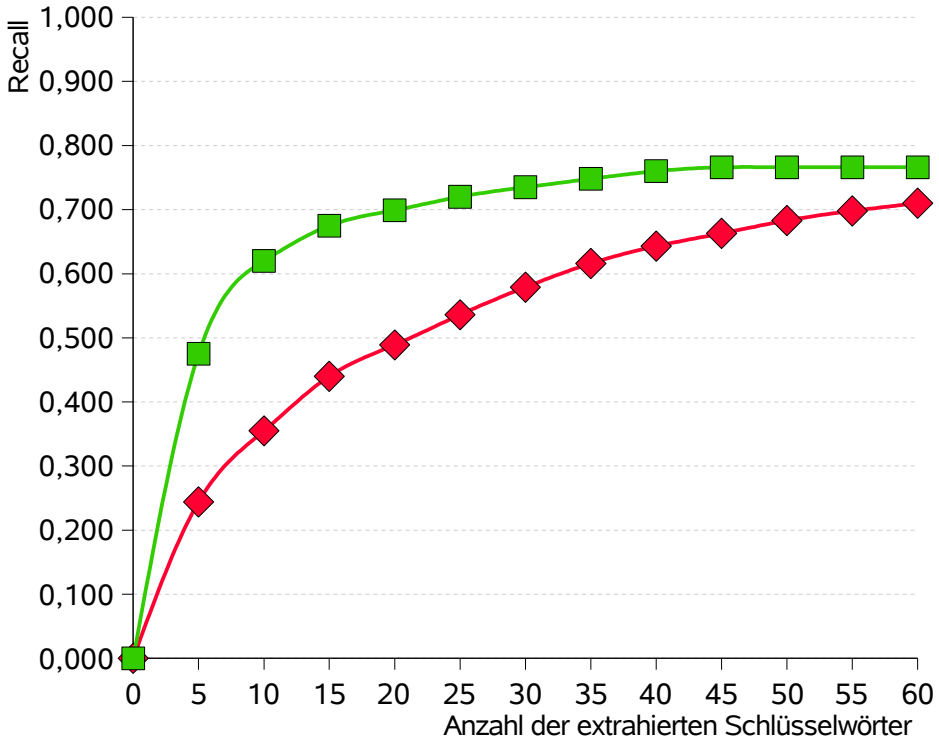
- Verbesserung der Ergebnisse
  - Ziel: Recall-Optimierung
    - aktueller Recall



# Ausblick

- Verbesserung der Ergebnisse
  - Ziel: Recall-Optimierung
    - aktueller Recall
    - optimierter Recall
  - z.B. durch externes Wissen

→ aktuelle Forschung an der BUW



# Zusammenfassung

- Java-Bibliothek implementiert
- Evaluierungskorpus erstellt
- Algorithmen unter Berücksichtigung der Anwendung in fokussierter Suche evaluiert
- Verbesserungsmöglichkeiten aufgezeigt

# Ende

Vielen Dank für Ihre Aufmerksamkeit.

# Quellen


- [Barker & Cornacchia, 2000] Ken Barker, Nadia Cornacchia: "Using Head Noun Phrases to Extract Document Keyphrases" in Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence, S. 40-52, 2000
- [Matsuo & Ishizuka, 2003] Y. Matsuo, M. Ishizuka: "Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information" in International Journal on Artificial Intelligence Tools, Vol.13, No.1, S. 157-169, 2003


- [Matsuo & Ishizuka, 2003] Y. Matsuo, M. Ishizuka: "Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information" in International Journal on Artificial Intelligence Tools, Vol.13, No.1, S. 157-169, 2003

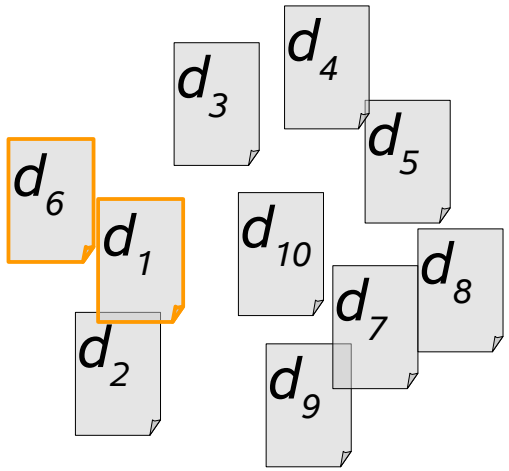

- [Tseng, 1998] Yuen-Hsien Tseng: "Multilingual Keyword Extraction for Term Suggestion" in Proceedings of the 21st annual international ACM SIGIR conference on Research and Development in Information Retrieval, Melbourne, Australia, S. 377-378, 1998
- [Turing, 1950] Alan Mathison Turing: "Computing machinery and intelligence" in VOL. LIX. No.236, S. 433, 1950

# Repräsentation von Textdokumenten im IR

- Thesen für die Wichtigkeit eines Terms

**(1) Terme, die in einem Dokument häufig vorkommen sind für dessen Inhalt wichtig.**

→ Termhäufigkeit $tf(t_i, d_j)$: absolute Anzahl des Auftretens des Terms $t_i$ im Dokument $d_j$



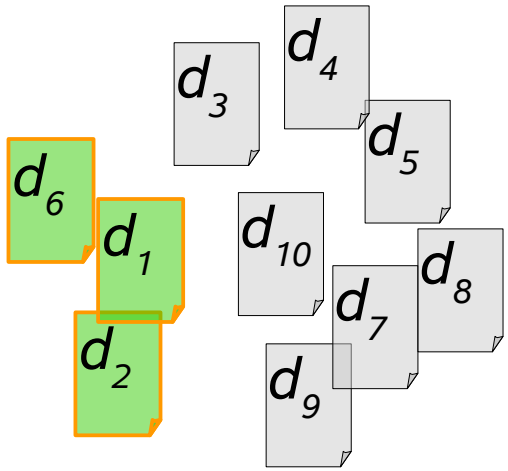

| $t$ | $tf(t,d_1)$ | $tf(t,d_2)$ | $tf(t,d_3)$ | $tf(t,d_4)$ | $tf(t,d_5)$ | $tf(t,d_6)$ | $tf(t,d_7)$ | $tf(t,d_8)$ | $tf(t,d_9)$ | $tf(t,d_{10})$ |
|---|---|---|---|---|---|---|---|---|---|---|
| **„digital computer“** | 63 | 15 | 0 | 0 | 0 | 42 | 0 | 0 | 0 | 0 |
| „make“ | 33 | 24 | 26 | 18 | 20 | 27 | 19 | 15 | 12 | 8 |

# Repräsentation von Textdokumenten im IR

- Thesen für die Wichtigkeit eines Terms

**(2) Terme, die innerhalb einer Kollektion in nur wenigen Dokumenten vorkommen sind stark diskriminierend und damit wichtig.**

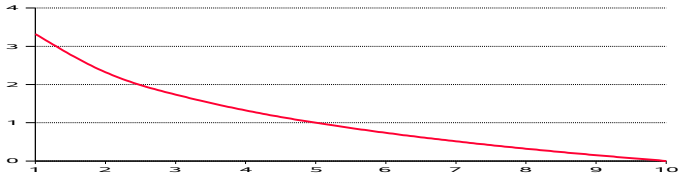
→ Dokumenthäufigkeit $df(t_i, D)$: absolute Anzahl der Dokumente der Kollektion $D$, in denen der der Term $t_i$ auftritt

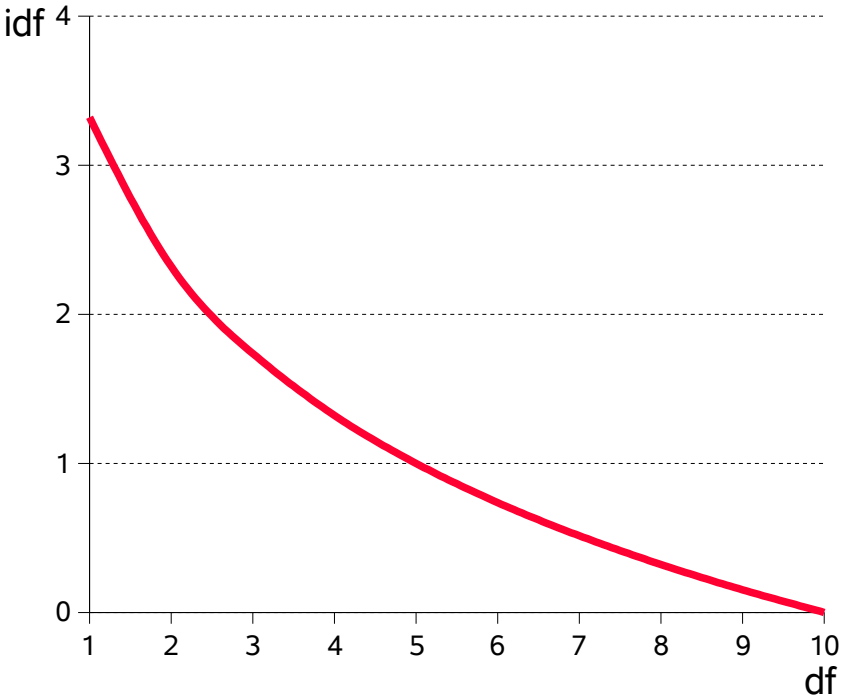



| $t$ | $tf(t,d_1)$ | $tf(t,d_2)$ | $tf(t,d_3)$ | $tf(t,d_4)$ | $tf(t,d_5)$ | $tf(t,d_6)$ | $tf(t,d_7)$ | $tf(t,d_8)$ | $tf(t,d_9)$ | $tf(t,d_{10})$ | $df(t,D)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **„digital computer“** | 63 | 15 | 0 | 0 | 0 | 42 | 0 | 0 | 0 | 0 | 3 |
| „make“ | 33 | 24 | 26 | 18 | 20 | 27 | 19 | 15 | 12 | 8 | 10 |

# Repräsentation von Textdokumenten im IR

- Modifikation $df(t_i, D)$
  - Ziel: Terme mit geringer Dokument-häufigkeit bekommen hohes Gewicht
  - → inverse Dokumenthäufigkeit






| $t$ | $tf(t,d_1)$ | $tf(t,d_2)$ | $tf(t,d_3)$ | $tf(t,d_4)$ | $tf(t,d_5)$ | $tf(t,d_6)$ | $tf(t,d_7)$ | $tf(t,d_8)$ | $tf(t,d_9)$ | $tf(t,d_{10})$ | $df(t,D)$ | $idf(t,D)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **„digital computer“** | 63 | 15 | 0 | 0 | 0 | 42 | 0 | 0 | 0 | 0 | 3 | 1,74 |
| „make“ | 33 | 24 | 26 | 18 | 20 | 27 | 19 | 15 | 12 | 8 | 10 | 0 |

# Repräsentation von Textdokumenten im IR

- Repräsentation als Vektor
- Termhäufigkeit *tf* als Merkmal:

Termvektor $\mathbf{t} =$

| |
|---|
| *classifi* |
| *click* |
| *comput* |
| *confus* |
| *continu* |
| *definit* |
| *differ* |
| *digit* |
| *discret* |
| *ignor* |
| *jump* |
| *kind* |
| *machin* |
| *move* |
| *possibl* |
| *profit* |
| *speak* |
| *state* |
| *strict* |
| *sudden* |
| *suffici* |
| *thought* |
| *univers* |

Merkmalsvektor $\mathbf{d} =$

| |
|---|
| 1 |
| 1 |
| 2 |
| 1 |
| 1 |
| 1 |
| 1 |
| 2 |
| 2 |
| 1 |
| 1 |
| 1 |
| 5 |
| 1 |
| 1 |
| 1 |
| 1 |
| 4 |
| 1 |
| 1 |
| 1 |
| 1 |
| 1 |

# Algorithmen zur Extraktion von Schlüsselwörtern

- Implementation und Evaluation von vier unterschiedlichen Verfahren
- Klassifikation der Algorithmen:

| | | |
|---|---|---|
| Dokumentbasis | korpusbasiert | dokumentbasiert |
| Training | überwacht | unüberwacht |
| Domainabhängigkeit | domainabhängig | domainunabhängig |
| Sprachabhängigkeit | sprachabhängig | sprachunabhängig |
| Ranking der Schlüsselwortkandidaten | statistisch | durch externes Wissen |

# Algorithmen: KEA - [Witten et al., 1999]

- Referenzalgorithmus, „New Zealand Digital Library“
- Basis: Maschineller Lernalgorithmus mit zwei Klassifikationsmerkmalen
  - 1. Termgewicht *tf.idf* $tf.idf(t) = wtf(t) \cdot idf(t)$ $wtf(t) = \frac{tf(t)}{n_t}$
  - *2. first occurrence* $(fo)$ $fo(t) = \frac{pre(t)}{n_t}$
- Klassifikation durch naiven Bayes Klassifizierer
- Training anhand einer Menge von Trainingsdokumenten mit bereits vom Autor vergebenen Schlüsselwörtern

# Algorithmen: Cooccurrence

- Extraktion häufiger Terme

| Häufige Terme | Auftrittshäufigkeit | Wahrscheinlichkeit |
|---|---|---|
| machine (a) | 203 | 0,366 |
| computer (b) | 63 | 0,114 |
| question (c) | 44 | 0,079 |
| digital (d) | 44 | 0,079 |
| answer (e) | 39 | 0,070 |
| game (f) | 36 | 0,065 |
| argument (g) | 35 | 0,063 |
| make (h) | 33 | 0,059 |
| state (i) | 30 | 0,054 |
| number (j) | 28 | 0,050 |

*Tabelle 1: Die zehn häufigstenTerme*