WikiListSuggest

Automatische Vervollständigung von Wikipedia-Listen

Make your world simpler

Universität Freiburg
Lehrstuhl für Algorithmen und Datenstrukturen
Universität Freiburg
Simon Skilevic, Robin Schirrmeister
26.4.2012

Überblick

- Listenerkennung
- Anfragegenerierung
- Evaluation

Erkennung
Überblick

1. **Kandidatenextraktion**: Mögliche Listenelemente finden
2. **Kandidatenbewertung**: Bewertung berechnen, wie gut sie in Liste passen
3. **Strukturbestimmung**: Struktur bestimmen, in der sich alle Listenelemente befinden

Erkennung
Visualisierung

1. Alle Kandidaten für Listenelemente finden:

Name	Birth	Birthplace	Death	Place of death	Notes
Bruno	1030	Cologne, Germany	1101	Squillace, Italy	
Thiemo (Theodinarus)		Germany	1102	Corozain, HaZafon, Israel	Archbishop of Salzburg
William Firmatus	1026	Tours, France	1103		

Erkennung
Visualisierung

2. Kandidaten bewerten:

Name	Birth	Birthplace	Death	Place of death	Notes
Bruno	1030	Cologne, Germany	1101	Squillace, Italy	
Thiemo (Theodinarus)		Germany	1102	Corozain, HaZafon, Israel	Archbishop of Salzburg
William Firmatus	1026	Tours, France	1103		

Erkennung
Visualisierung

3. Struktur bestimmen, die Listenelemente komplett enthält

Name	Birth	Birthplace	Death	Place of death	Notes
Bruno	1030	Cologne, Germany	1101	Squillace, Italy	
Thiemo (Theodinarus)		Germany	1102	Corozain, HaZafon, Israel	Archbishop of Salzburg
William Firmatus	1026	Tours, France	1103		

Kandidaten-Extraktion

Mögliche Kandidaten: Wikipedia-Links in Tabellen oder HTML-Listen!

Kandidaten-Extraktion mit CSS-Selektoren:

- `.mw-content-ltr td a[href^="/wiki/"]`
- `.mw-content-ltr li a[href^="/wiki/"]`
- `.mw-content-ltr dl a[href^="/wiki/"]`

Kandidaten-Bewertung

Für Kandidaten-Bewertung sind die Kategorien der Kandidaten wichtig:

Bruno ⇨ Saint, Person, ...

Cologne ⇨ City, Location, ...

Meistens gehören die Listenelemente zur gleichen Kategorie!

Kandidaten-Bewertung

Kandidaten-Bewertung berechnet sich aus seinen Kategorien:

1. Alle Kategorien eines Kandidaten mit YAGO bestimmen
2. Bewertungen seiner Kategorien berechnen
3. Summe der Kategorie-Bewertungen ist Bewertung vom Kandidat selbst

Beispiel:

Bruno	23000
Saint -> 20000 Person -> 3000	

Cologne	3100
City -> 3000 Location -> 100	

Kandidaten-Bewertung
Kategorie-Bewertung

Bewertung s(K) einer Kategorie K:

1. s(K) = Häufigkeit in Kandidatenmenge / Häufigkeit in YAGO
2. s(K) = s(K) * 1000, falls Kategorie erstes Nomen im Plural aus dem Titel
3. s(K) = s(K) * 10, falls Kategorie anderes Wort aus dem Titel
4. s(K) = s(K) * 0.1, falls Kategorie zehnmal seltener als die häufigste Kandidaten- Kategorie vorkommt

Multiplikation, damit schon ein sehr wahrscheinliches Listenelement ausreicht, dass eine ganze Gruppe eine hohe Bewertung erhält

Strukturbestimmung

Kandidaten werden in Strukturgruppen eingeteilt:

- Gleicher Typ von HTML-Struktur (Tabelle oder Liste)
- Gleiche Position: Gleiche Spalte in Tabelle, gleicher Link im HTML-Listenelement

Strukturgruppe, bei der Summe der Kandidatenbewertungen am höchsten ist, wird ausgewählt

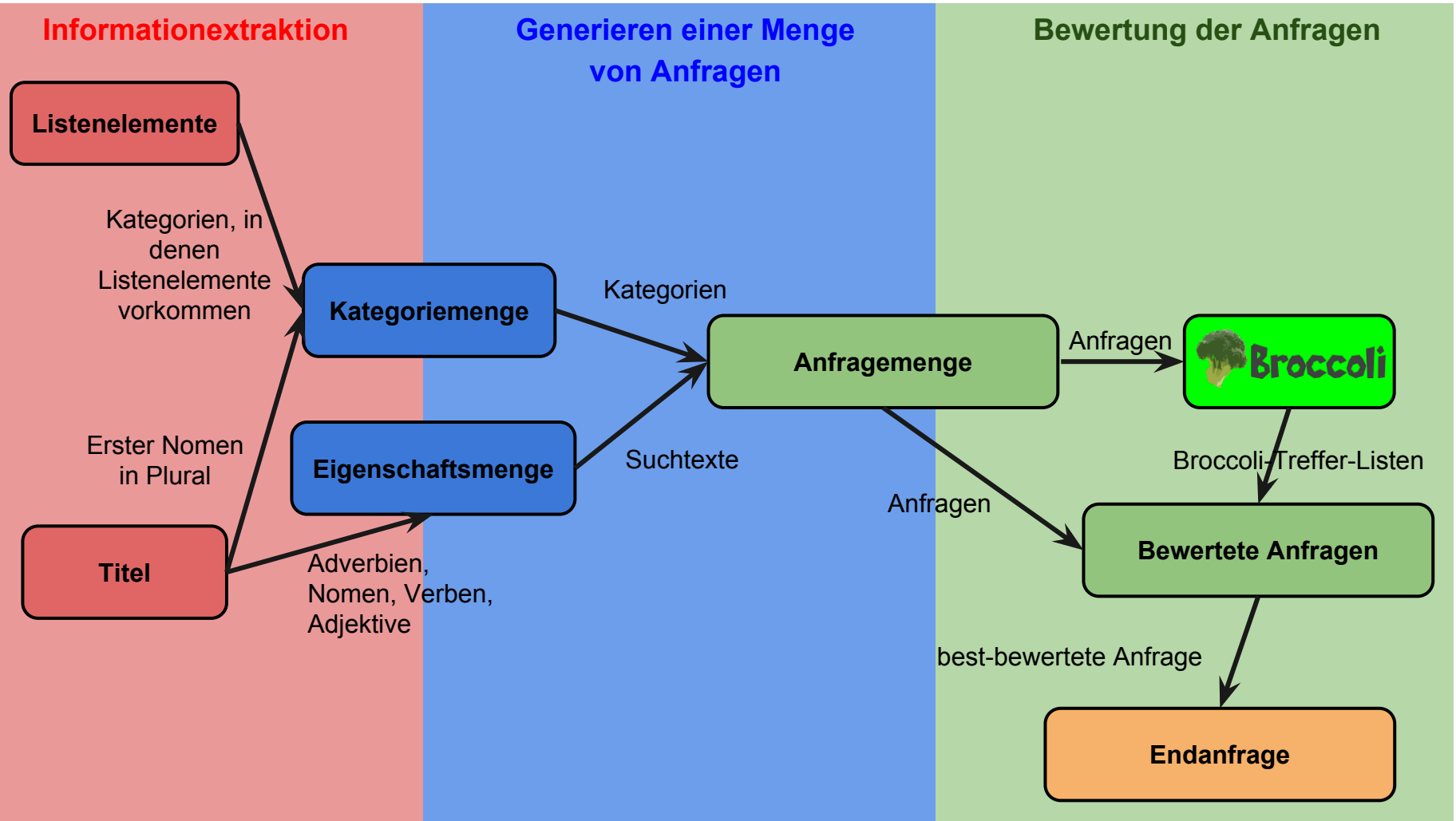
Anfrageform

[Kategorie] **occurs-with** **[Suchtext]**

Suche in einem Kontext nach

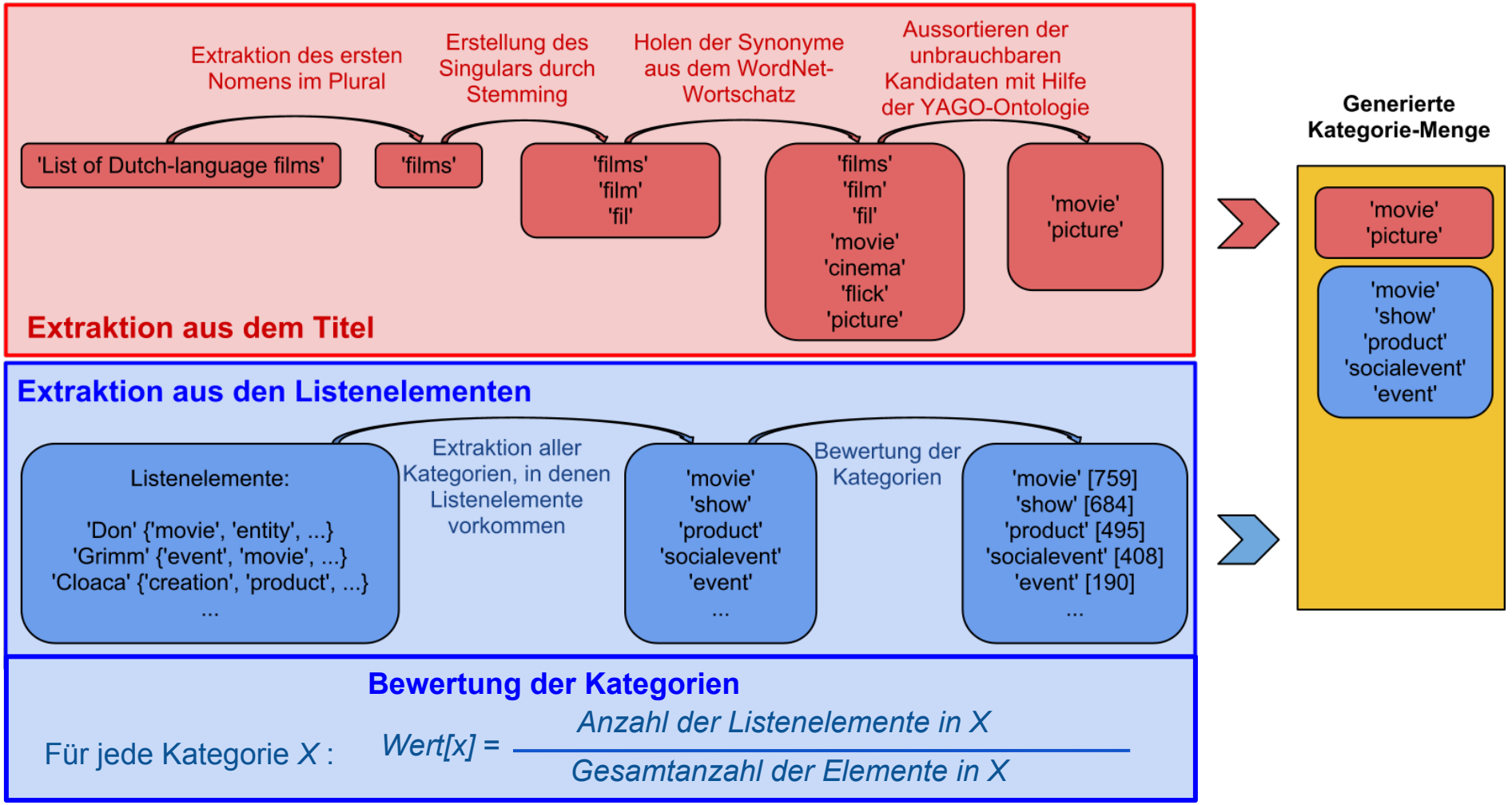
- den Begriffen aus der **[Kategorie]**
- zusammen mit dem **[Suchtext]**

Anfragegenerierung : Algorithmus



Informationextraktion

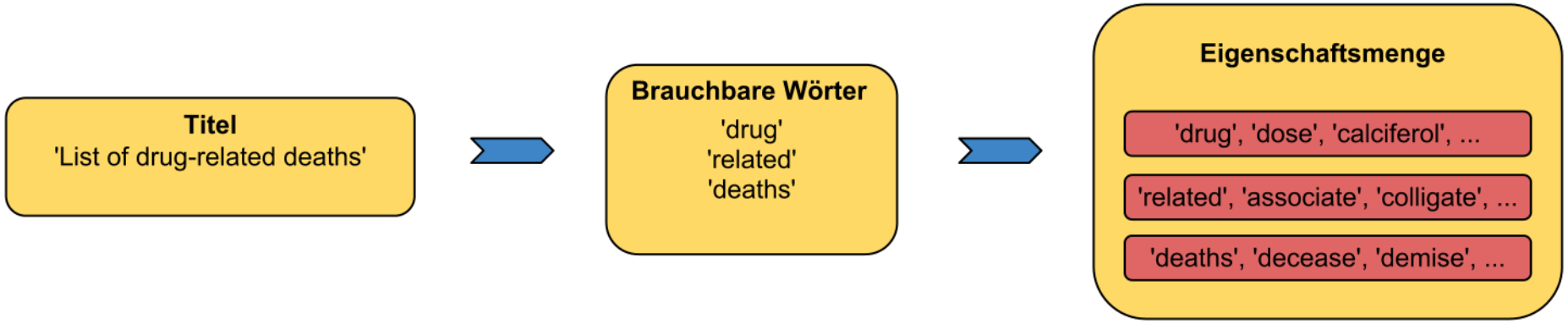
Generierung der Kategorie-Menge



Informationextraktion

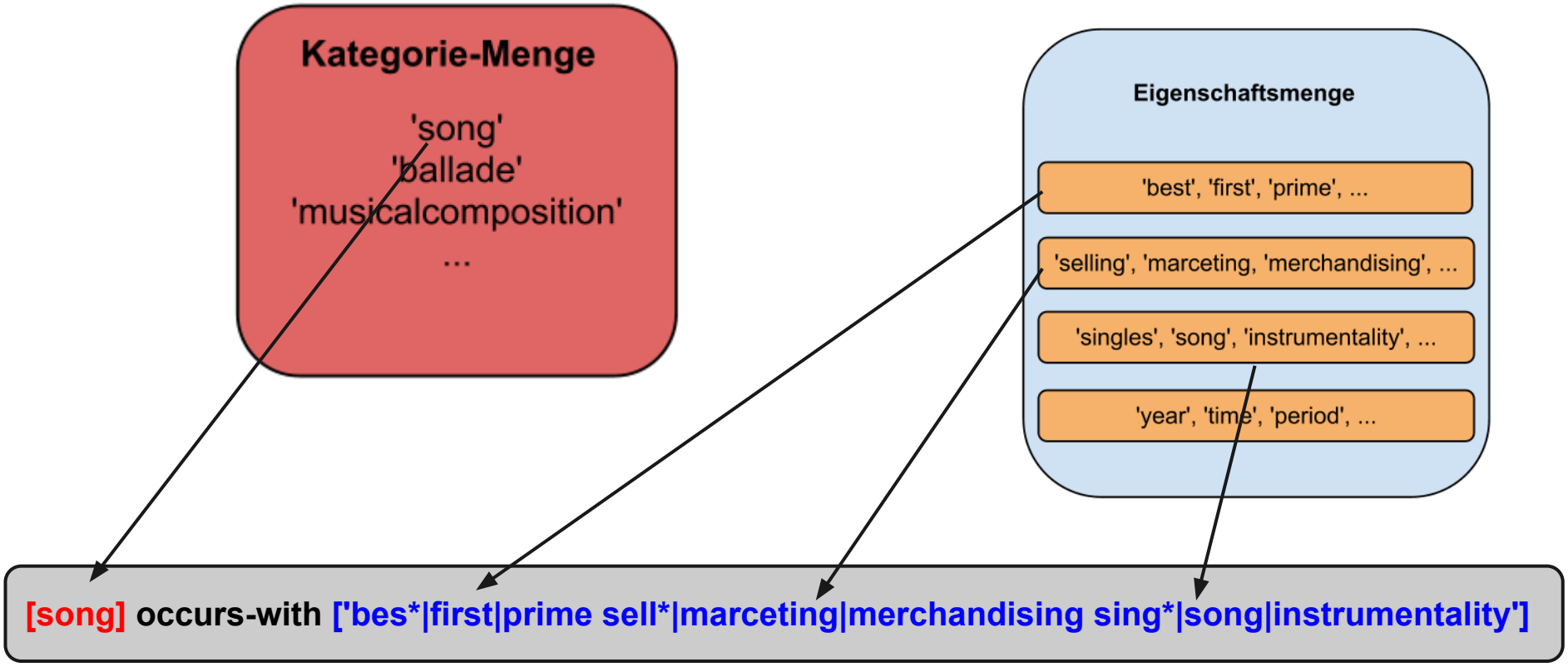
Generierung der Eigenschaftsmenge

Extraktion der brauchbaren Titelwörter und ihrer Synonyme



Generieren einer Menge von Anfragen

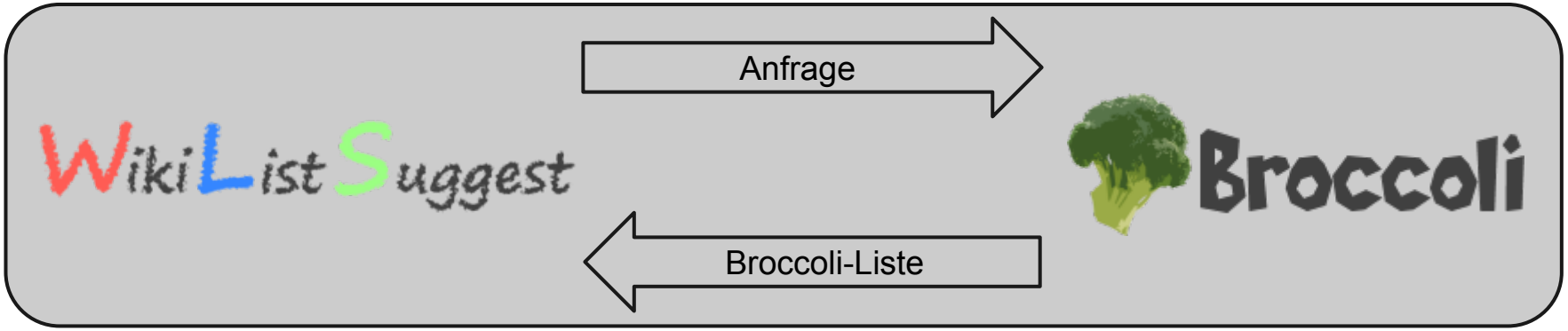
Generierung einer Anfrage



Vertikaler Strich - suche nach dem ersten **ODER** zweiten Wort in einem Kontext
Leerzeichen - suche nach dem ersten **UND** zweiten Wort in einem Kontext

Bewertung der Anfragen

Ablauf



Jede Broccoli-Liste wird auf die Übereinstimmung mit der Wikipedia-Liste anhand von zwei Faktoren getestet:

- Anzahl der gleichen Elemente
- Größendifferenz

Bewertung der Anfragen

Bewertung

$$QualitätWert = hitsNumber - \frac{bSize - wSize}{diffFactor}$$

hitsNumber: Anzahl der Elemente der Wikipedia-Liste, die von Broccoli gefunden wurden.

bSize: Größe der Broccoli-Liste.

wSize: Größe der Wikipedia-Liste.

diffFactor: Durch den diffFactor kann man einstellen, wie stark der Einfluss der Größendifferenzen zwischen den beiden Listen auf den QualitätWert sein soll (aktuell auf 13 gesetzt).

Evaluation

- **Evaluation der Listenerkennung**
  - Wie viele Listenelemente wurden auf der Seite erkannt?

- **Evaluation der Anfrage-Generierung**
  - Wie gut ist die Übereinstimmung zwischen der von Broccoli ermittelten Liste und der Wikipedia-Liste?
  - Wie gut sind die Vorschlagslisten zum Vervollständigen der Wikipedia-Listen?

- **Laufzeit von WikiListSuggest**

Evaluation

Listenerkennung

Nummer	Name
WL1	List of current A&M Records artists
WL2	List of banks in Albania
WL3	List of aerospace engineers
WL4	List of films based on English-language comics
WL5	List of Royal Australian Air Force aircraft squadrons
WL6	List of female philosophers
WL7	List of atheist activists and educators
WL8	List of Albania international footballers born outside Albania
WL9	African Americans in the United States Congress
WL10	List of EN standards
WL11	List of the world's 100 worst invasive species
WL12	Chronological list of saints and blesseds in the 12th century
WL13	List of Disney theme park attractions
WL14	List of class action lawsuits
WL15	List of constellations
WL16	List of best-selling albums
WL17	List of awards and nominations received by John Abraham
WL18	List of archaeological sites by country
WL19	List of alternate history fiction
WL20	List of hospitals in Australia
WL21	List of accidents and disasters by death toll
WL22	List of European birds
WL23	List of astronauts by name
WL24	Abydos King List
WL25	List of Roman amphitheatres
WL26	List of The Amazing Race Asia contestants
WL27	List of firsts in aviation
WL28	List of unsolved problems in biology
WL29	List of Alpha Kappa Alpha Boulés
WL30	ISO 3166-2:CA
WL31	Dolch word list
WL32	List of ABS–CBN Corporation slogans
WL33	List of Berlin Wall segments
WL34	List of words in English containing all the vowels

- Listenelemente von **16** der **34 (47%)** Listen komplett richtig erkannt
- Bei **5** von **34 (15%)** Listen wurden zu viele Elemente erkannt
- Bei **3** von **34 (9 %)** Listen wurden nicht alle Listenelemente erkannt
- Bei **4** von **34 (12%)** der Wikipedia-Listen wurden die Listenelemente gar nicht erkannt.
- Bei **6** von **34 (17%)** der Wikipedia-Listen war die Erkennung der Listenelemente nicht möglich.

Listenelemente von 21 der 28 (75%) Listen wurden vollständig erkannt

Evaluation

Anfragegenerierung

21 Listen wurden getestet

Im Schnitt

- enthält jede Broccoli-Liste **36%** aller Listenelemente
- ist die Größe der Broccoli-Liste um **5%** größer als der Wikipedia-Liste.

Bei 16 unvollständigen Listen wurden 160 vorgeschlagene Elemente untersucht

Insgesamt konnten wir **63 neue Listenelemente** finden.

- Im Schnitt sind **fast 4 von 10** geprüften Elementen neue Listenelemente. Das ergibt eine Trefferquote von **39%**.

Evaluation

Anfragegenerierung

hitsNumber Anzahl der gefundenen Listenelemente

wSize Größe der Wikipedia-Liste.

bSize Größe der Broccoli-Liste (Je näher zu wSize, desto besser).

Num	Query	hitsNum	wSize	bSize
WL1	Singer occurs-with curr*|flow|stream arti*	1	28	318
WL2	Bank occurs-with bank*|camber|cant alba*	6	13	12
WL3	Creator occurs-with aerosp* engine*|direct|mastermind	11	131	43
WL4	Movie occurs-with base*|alkali|bag comi*|amusing|comedian	80	224	358
WL5	Squadron occurs-with roya*|imperial|majestic austral*|aussie	83	85	125
WL6	Writer occurs-with fema*|distaff philosoph*	2	99	16
WL7	Militant occurs-with athe*|atheistic|atheistical	35	72	272
WL8	football_player occurs-with alba* internatio*|external|outside	5	13	36
WL9	Congressman occurs-with afri*	72	133	113
WL10	Standard occurs-with en|nut|east standa*|banner|criterion	14	31	36
WL11	Organism occurs-with invas*|encroaching|incursive spec*|coinage|mintage	32	100	422
WL12	Saint occurs-with 12th cent*|c|hundred	25	165	158

Evaluation

Anfragegenerierung

Gründe für schlechte Anfragen:

- Der Suchtext zu speziell → zu wenige Listenelemente gefunden
- Der Suchtext zu allgemein → zu große Broccoli-Liste

Synonyme sind öfters keine gute Eigenschaftswörter

Evaluation

Anfragegenerierung

Mögliche Verbesserung:

Statt Synonymen Wörter verwenden, die

- oft auf der Seite
- selten in der ganzen Wikipedia

vorkommen.

Evaluation

Performance

Einflüsse auf die Laufzeit

Listenerkennung

- Anzahl der Listenelemente

Anfragegenerierung

- Anzahl der Anfragen.
- Komplexität der Anfragen.

Zusammenfassung

- Bei **21** von **28** Listen, bei denen die Erkennung möglich war, hat WikiListSuggest alle Listenelemente erkannt.
  - Das ist eine Erkennungsquote von **75%**.

- **63** von **160** vorgeschlagenen Elementen waren neue Listenelemente
  - Das ist eine Trefferquote von etwa **39%.**

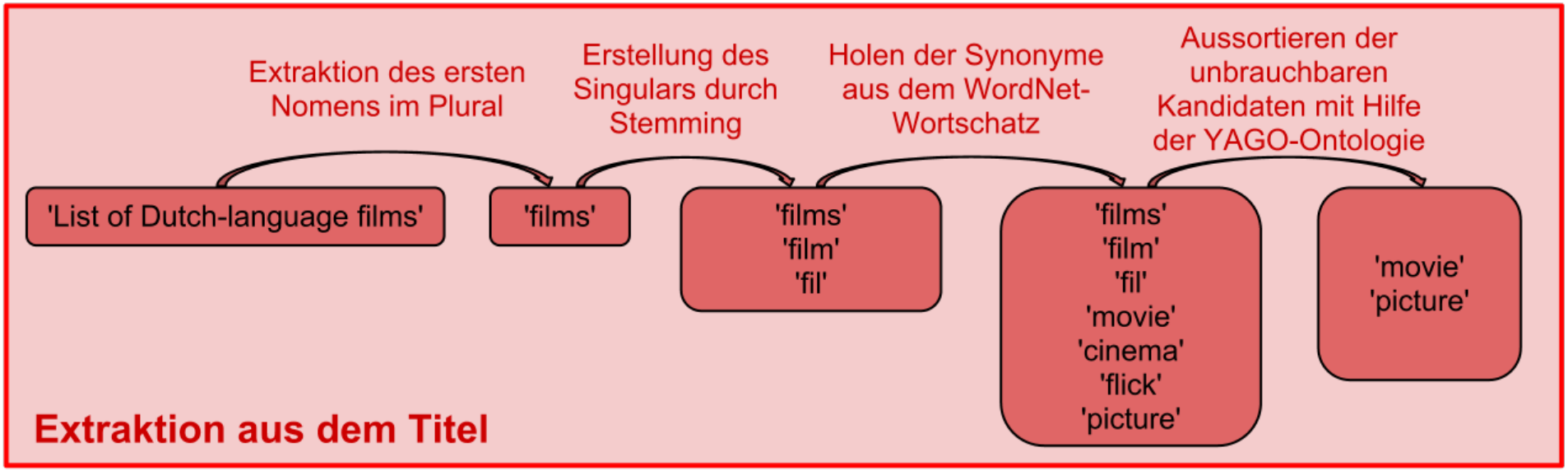
Ausblick

- Kontextrelevante Wörter
- Performance-Optimierungen
- Usability-Verbesserungen

Informationextraktion

Generierung der Kategorie-Menge

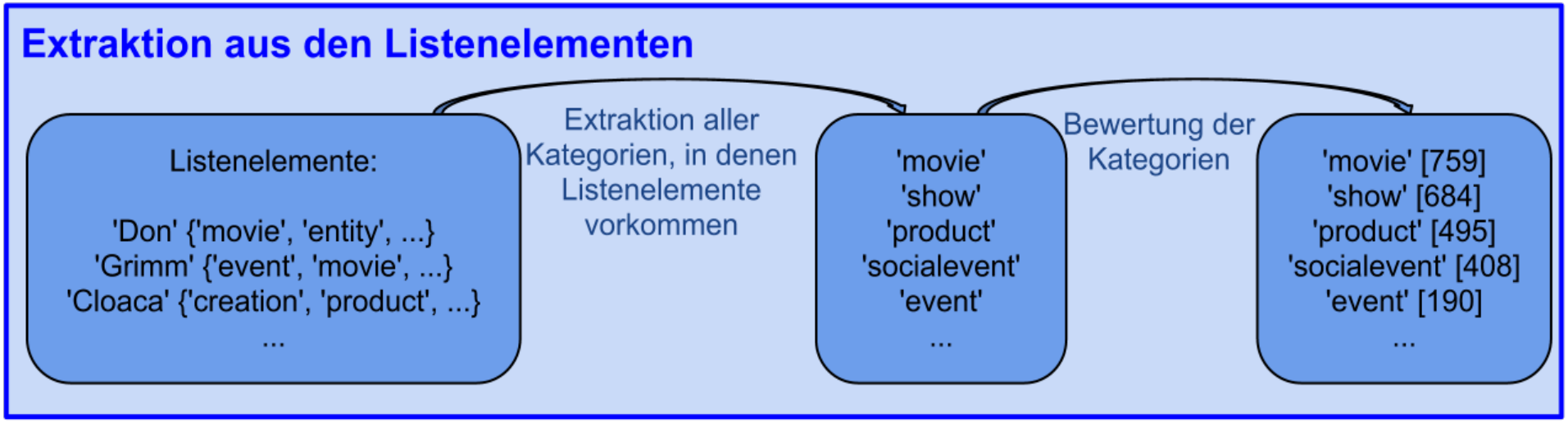
Extraktion der Kategorie-Kandidaten aus dem Titel:



Informationextraktion

Generierung der Kategorie-Menge

Extraktion der Kategorie-Kandidaten anhand der Listenelemente:



Bewertung der Kategorien

Für jede Kategorie *X* :

$$Wert[x] = \frac{Anzahl\ der\ Listenelemente\ in\ X}{Gesamtanzahl\ der\ Elemente\ in\ X}$$

Generieren einer Menge von Anfragen

Anzahl der Anfragen

Probleme:

- Broccoli kann länger als 1 sec für eine Anfrage brauchen
- Zu viele Anfragen werden generiert

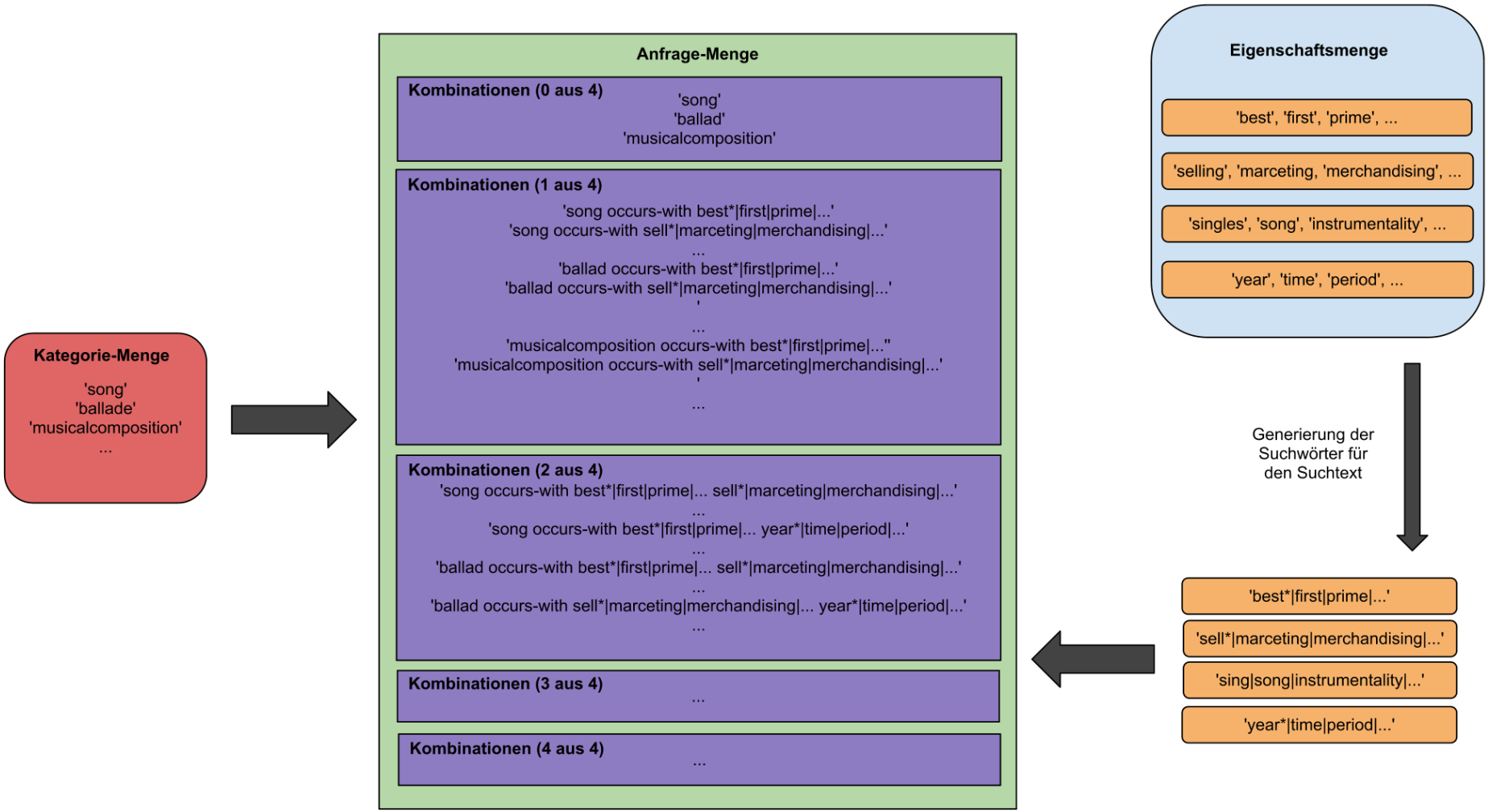
⇨ Laufzeit ist nicht mehr akzeptabel

Optimierung:

- Anfragen, die durch Teilmengen der Größe *3 - N-3* generiert wurden, werden ausgelassen, da es zu viele sind.
- Anzahl der Kategorie-Kandidaten wird auf 8 beschränkt
- Anzahl der Synonyme wird für jedes Titel-Wort auf 3 beschränkt

Generieren einer Menge von Anfragen

Generierung einer Anfragemenge



Kandidaten-Bewertung
Kategorie-Bewertung

Bewertung s(K) einer Kategorie K:

$$s(K) = \frac{\text{Häufigkeit bei Kandidaten}}{\text{Häufigkeit in YAGO}} \cdot Gewicht(K)$$

Kandidaten-Bewertung
Kategorie-Bewertung

$$Gewicht(K) = 1000^{x_1} \cdot 10^{x_2} \cdot 0.1^{x_3}$$

$x_1 = 1$, falls Kategorie gleich erstem Nomen im Plural aus dem Titel

$x_2 = 1$, falls Kategorie anderes Wort aus dem Titel

$x_3 = 1$, falls Kategorie zehnmal seltener als häufigste Kategorie der Kandidaten

sonst $x_1 = 0, x_2 = 0, x_3 = 0$

Multiplikation, damit schon ein sehr wahrscheinliches Listenelement ausreicht, dass eine ganze Gruppe eine hohe Bewertung erhält