Netspeak

Ein Assistent zum Verfassen fremdsprachiger Texte

Martin Trenkmann

Bauhaus Universität Weimar

Web Technology & Information Systems

11. Juli 2008

Gliederung

Motivation: Szenario

- Studenten und Wissenschaftler verfassen oft englische Texte.
- Viele Menschen haben eine andere Muttersprache als Englisch.
- Schwierigkeiten treten auf ...
  - ... beim Finden des richtigen Wortes (z.B. Präpositionen).
  - ... bei der Wahl des gebräuchlichsten Synonyms (z.B. Adjektive).
  - ... bei der Beschreibung konkreter Sachverhalte (z.B. Adverbien).
- Eine Orientierung an phonetisch ähnlichen Worten
  bei der Wahl der Übersetzung ist dabei oft nicht richtig.

Motivation: Beispiel 1

Ich bin ein Student, der ...

- *... **an** Informatik interessiert ist.*
- *... sich **für** Informatik interessiert.*

I am a student, who is interested ...

- × *... **at** computer science.*
- × *... **on** computer science.*
- × *... **for** computer science.*
- √ *... **in** computer science.*

Welche ist die richtige Präposition?

Motivation: Beispiel 2

*Das hängt **hauptsächlich** von ihrem Können ab.*

It depends ...

- √ *... **largely** on your skill.*
- √ *... **heavily** on your skill.*
- √ *... **primarily** on your skill.*
- √ *... **greatly** on your skill.*

Welches ist das gebräuchliste Synonym?

Motivation: Beispiel 3

*Ich parke mein Auto **vor** dem Gebäude.*

× *I park my car **before** the building.*
√ *I park my car **in front of** the building.*

- ***before*** beschreibt einen zeitlichen Vorgang
- ***in front of*** beschreibt eine physikalische Gegebenheit

Motivation: Bisherige Lösungsversuche

- Anfragen an **Online-Wörterbücher** wie *LEO* oder *dict.cc*

  \+ Übersetzungen einzelner Worte (evtl. mit Verwendungsbeispielen)
  – Keine Suche nach Phrasen mit mehreren Worten möglich

- Anfragen an **Internet-Suchmaschinen** wie *Google*

  \+ Suche nach Phrasen möglich (Überprüfung der Richtigkeit)
  \+ Wildcards erlauben die Vervollständigung von Phrasen
  (Suche nach fehlenden Worten, evtl. auch Synonymsuche)
  – Das Suchergebnis ist nach Dokumenten und nicht
  nach den Häufigkeiten der Phrasen gerankt

- Fazit:
  - Manuelle Suche und Bewertung von Formulierungen ist sehr zeitaufwändig.
  - Eine Automatisierung dieses Prozesses könnte viel Zeit sparen und die Qualität der Texte verbessern.

Die Idee zu Netspeak

Netspeak

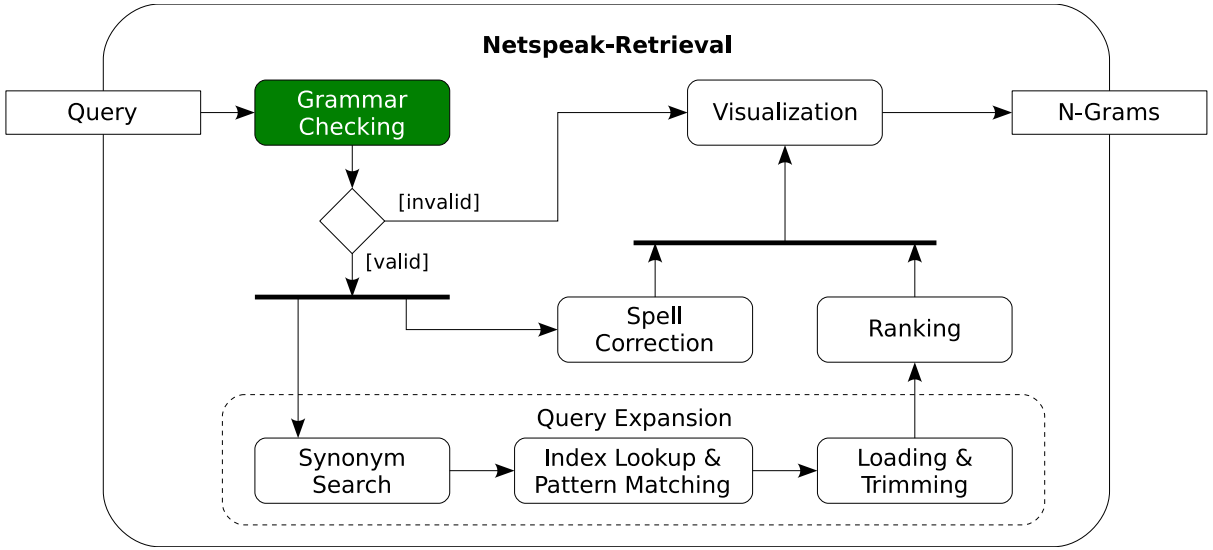
- **Netspeak** ist ein Web-Informationssystem, das die Gebräuchlichkeit von kurzen Textphrasen in der englischen Sprache feststellt.
- Datenbasis: Das **World Wide Web**
- Autoren der meisten Web-Dokumente: **Native-Speaker**
- Annahme:
  - **Häufigkeit** einer Formulierung → **Gebräuchlichkeit** (Richtigkeit)
  - Gilt nicht für grammatikalische Korrektheit (Umgangssprache)

Google-N-Gramm-Kollektion

- Google stellt das Web in Form einer Kollektion von N-Grammen zur Verfügung.
- Ein **N-Gramm** ist in diesem Zusammenhang eine Folge von N Worten.
- Die Google-N-Gramm-Kollektion umfaßt die Menge der 1-, 2-, 3-, 4- und 5-Gramme aller indizierten englischsprachigen Web-Dokumente.
- Für Netspeak wurde der 5-Gramm-Korpus mit einer speziellen Implementierung eines **invertierten Index** indiziert.

	Anzahl	Dateien	komprimierte Größe	unkomprimierte Größe
Sätze	95.119.665.584			
1-Gramme	13.588.391	1	70,2 Megabyte	177,00 Megabyte
2-Gramme	314.843.401	32	1,6 Gigabyte	5,0 Gigabyte
3-Gramme	977.069.902	98	5,5 Gigabyte	19,0 Gigabyte
4-Gramme	1.313.818.354	132	8,4 Gigabyte	30,5 Gigabyte
5-Gramme	**1.176.470.663**	**118**	**8,8 Gigabyte**	**32,1 Gigabyte**

Netspeak-Retrieval: Anfragesprache



Beispiel Query: `it depends * ~skill`

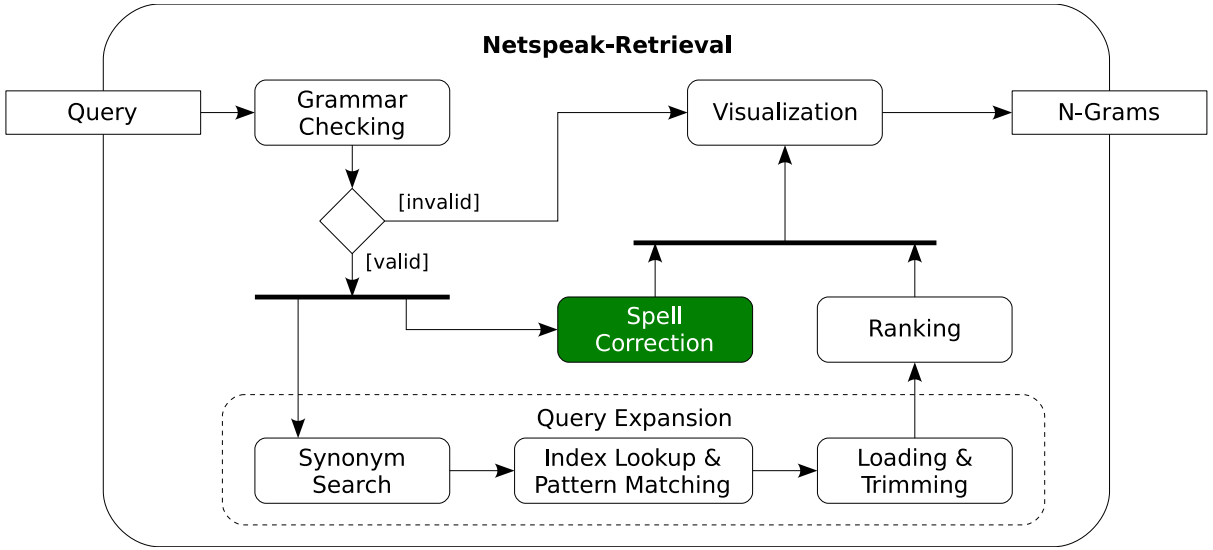
Drei spezielle Wildcards werden unterstützt:

- `*` steht für null bis beliebig viele Worte
- `?` steht für genau ein einzufügendes Wort
- `~` markiert ein Wort für eine Synonymsuche

vereinfachte Query Grammatik:

```
QUERY = { WORD | SYNWORD | '?' | '*' };
SYNWORD = '~' WORD;
WORD = LETTER { LETTER };
LETTER = 'a' .. 'z' | 'A' .. 'Z';
```

Netspeak-Retrieval: Rechtschreibkorrektur



Vorschläge zur Rechtschreibkorrektur
werden auf der Antwortseite eingeblendet.

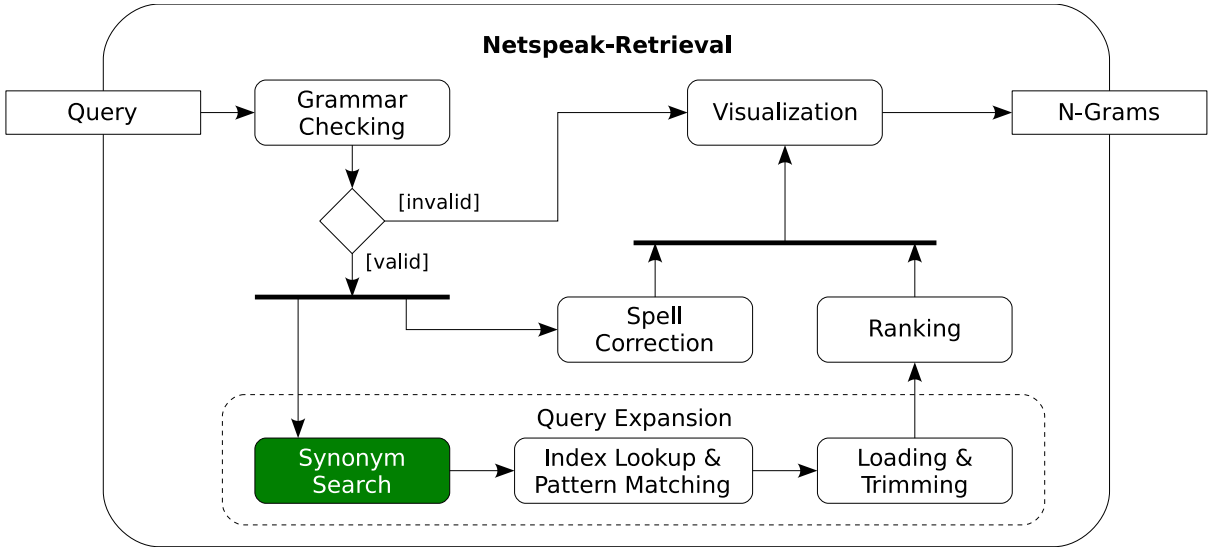
- Fehlerhafte Query:

  `intristed ? compjuter scienz`

- Korrekturvorschlag:

  `interested ? computer science`

Netspeak-Retrieval: Synonymsuche



Vorschläge zur Rechtschreibkorrektur werden auf der Antwortseite eingeblendet.

- Fehlerhafte Query:

  `intristed ? compjuter scienz`

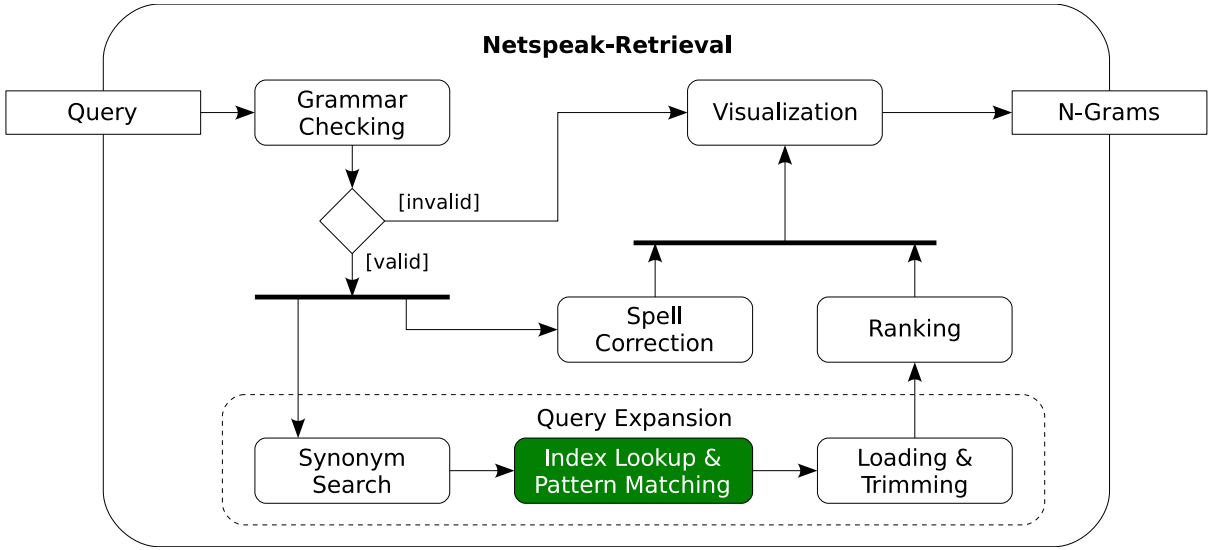
- Korrekturvorschlag:

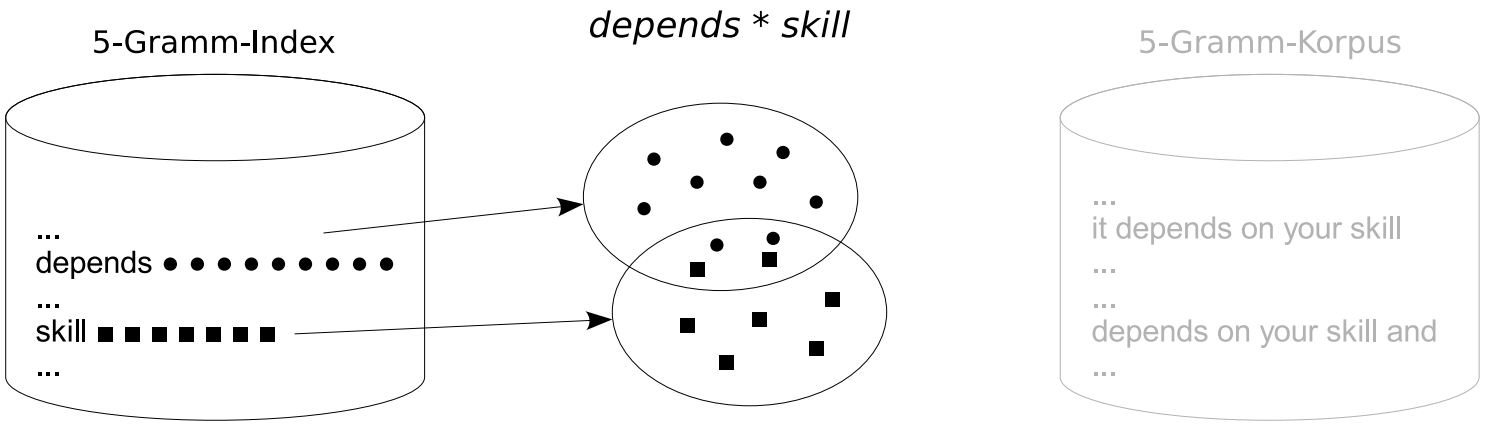
  `interested ? computer science`

Für entsprechend gekennzeichnete Worte werden Synonyme gesucht und weitere Queries generiert.

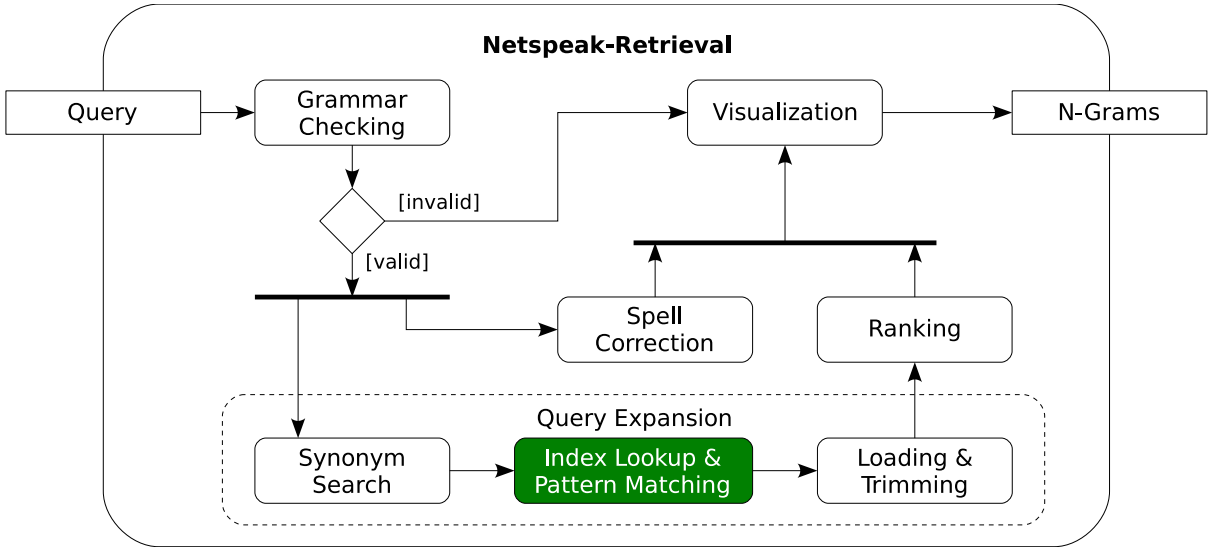
- Query: `it depends * ~skill`
- Generierte Queries:
  - `it depends * skill`
  - `it depends * accomplishment`
  - `it depends * acquirement`
  - `it depends * acquisition`
  - `it depends * attainment`

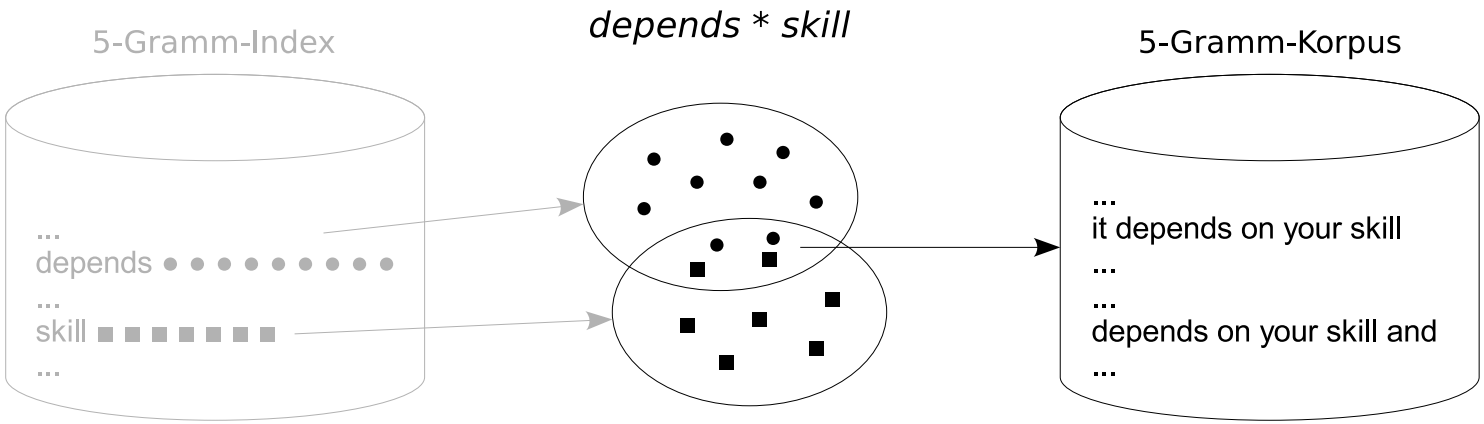
Netspeak-Retrieval: Mustersuche im 5-Gramm-Index



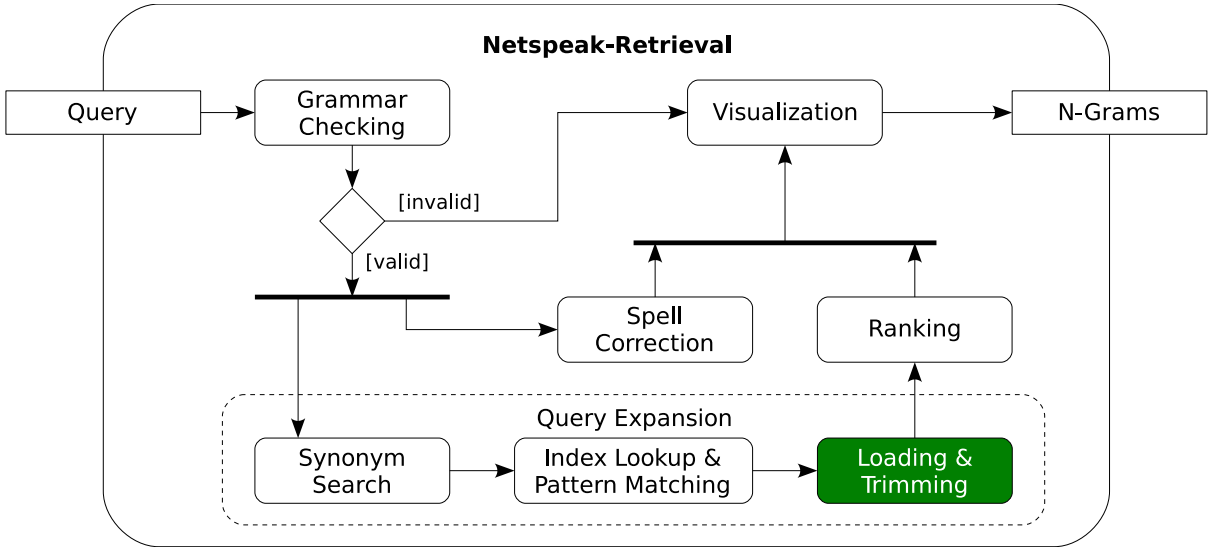


Netspeak-Retrieval: Mustersuche im 5-Gramm-Index

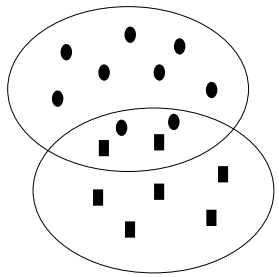




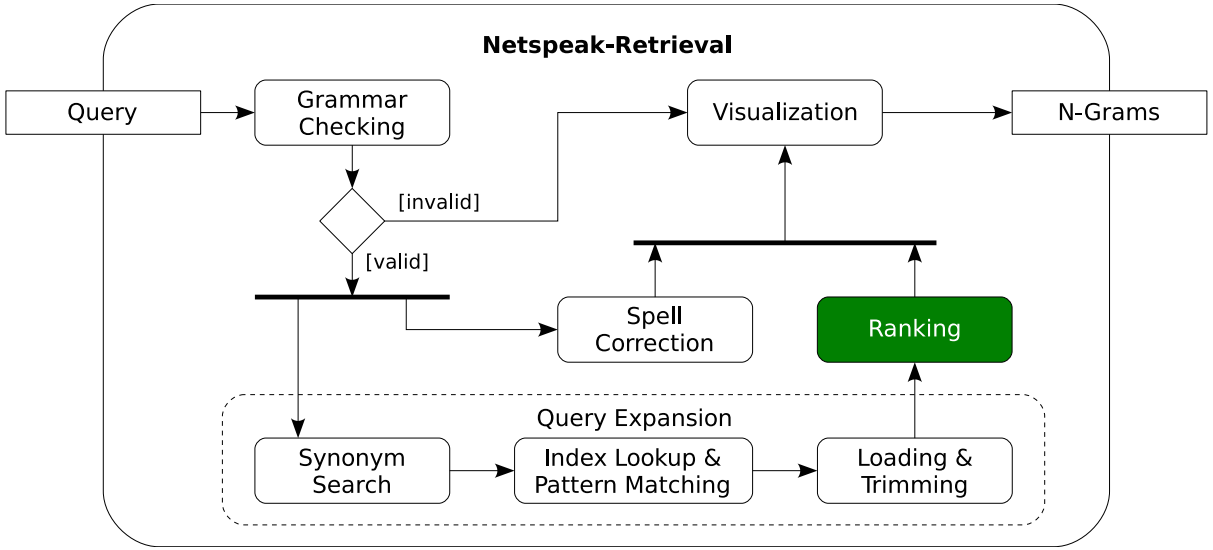
Netspeak-Retrieval: Zusammenfassen von Duplikaten



- Alle 5-Gramme aus der Mustersuche erfüllen die eingegebene Query.
- Allerdings befinden sich darunter viele gleiche Übereinstimmungen:
  - `it depends on your skill`
  - `depends on your skill and`
- Diese Duplikate müssen zusammengefaßt werden.
- Die Häufigkeitswerte der N-Gramme werden summiert.



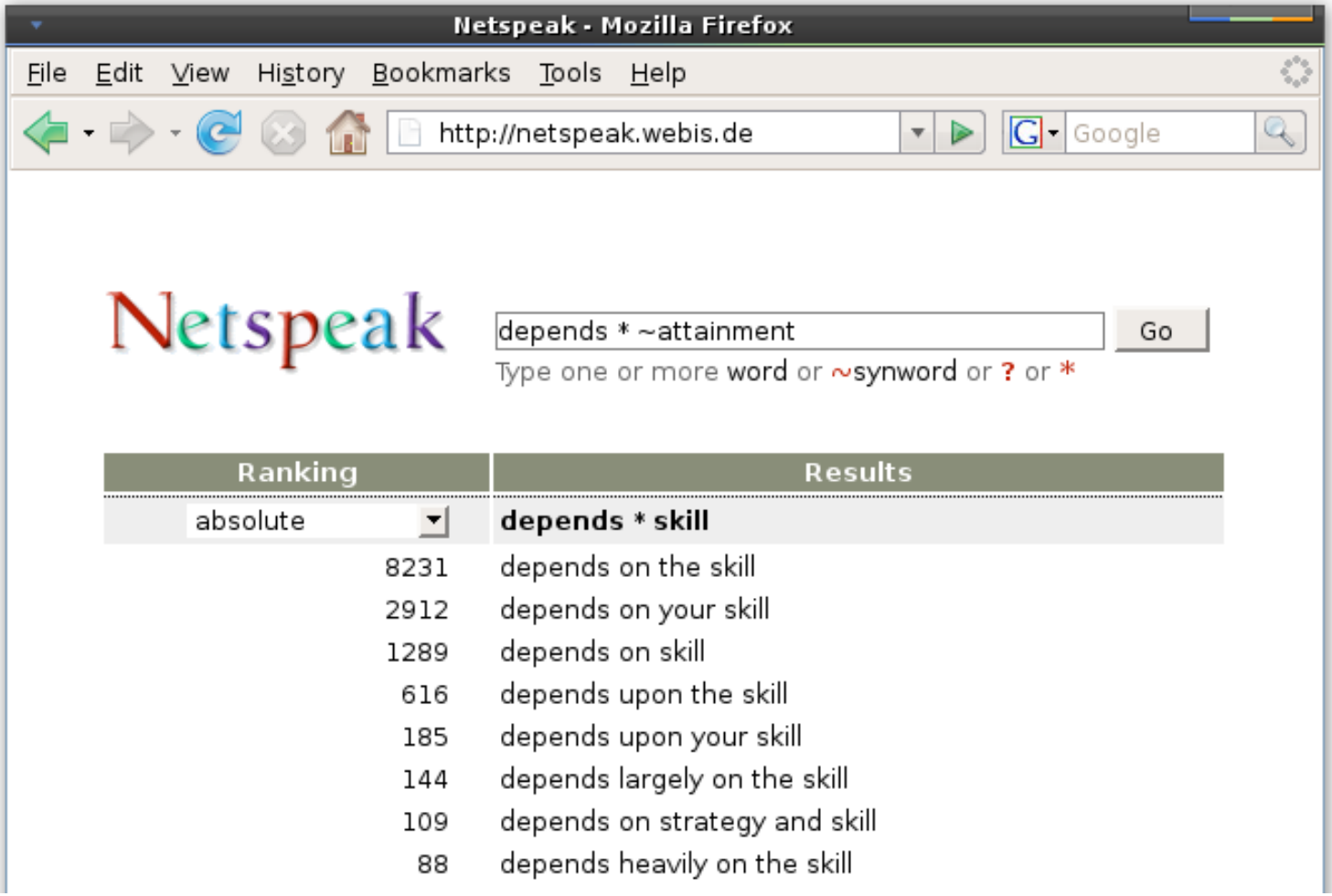
Netspeak-Retrieval: Ranking



- Die ermittelten N-Gramme werden nach ihren Häufigkeiten gerankt.
- D.h., die Liste der N-Gramme wird absteigend sortiert.
- Es werden zwei Rankings unterschieden:

1. Ein **absolute** Ranking auf Grundlage der absoluten N-Gramm-Häufigkeiten.
2. Meherere **relative** Rankings der N-Gramme bezüglich der absoluten Häufigkeit eines bestimmten Wortes aus der Query.

Netspeak-Retrieval: Visualisierung



Indizierung: Prinzip eines Index

Ein Index ...

- ist eine (verteilte) Datenstruktur.
- ermöglicht effizienten Zugriff auf eine große Menge von Daten.
- enthält nicht die eigentlichen Nutzdaten sondern Metadaten.
- ist eine Abbildung von Schlüsseln auf Metadaten (Referenzen).

Abbildung von Schlagworten auf ...

Seitenzahlen (Texte)

Internetadressen (Webseiten)

Google™

GNGramPointer (N-Gramme)

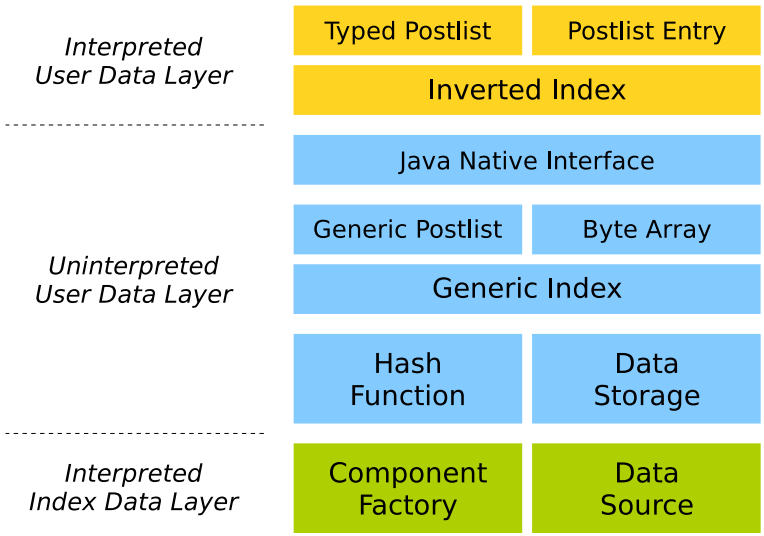
Netspeak

Indizierung: Invertierte Liste

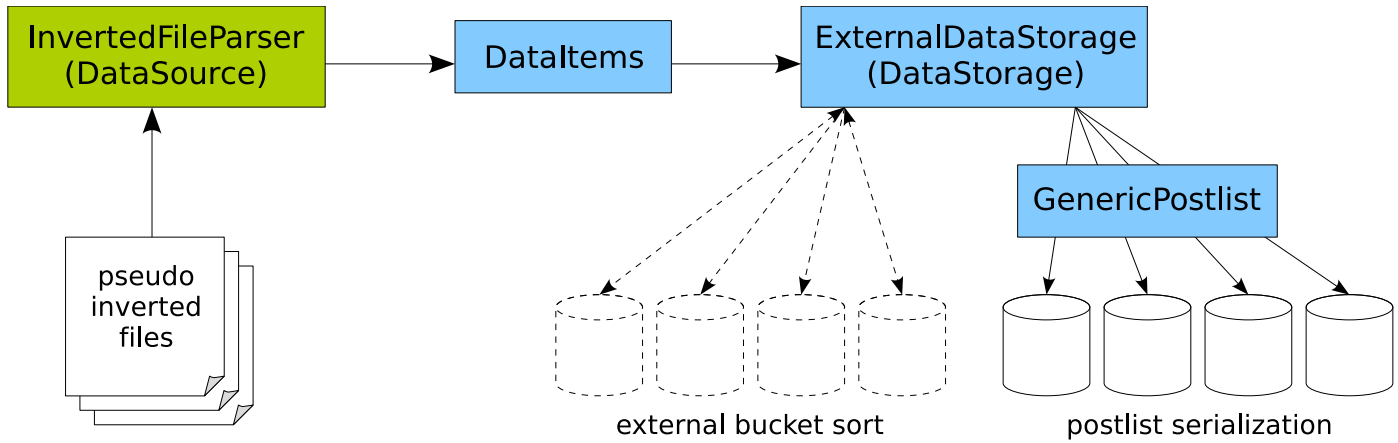
Vokabular	**Postlisten**
$\text{Wort}_{\mathbf{1}}$	$\text{Referenz}_{\mathbf{11}}$, $\text{Referenz}_{\mathbf{12}}$, $\text{Referenz}_{\mathbf{13}}$, ...
$\text{Wort}_{\mathbf{2}}$	$\text{Referenz}_{\mathbf{21}}$, $\text{Referenz}_{\mathbf{22}}$, ...
$\text{Wort}_{\mathbf{3}}$	$\text{Referenz}_{\mathbf{31}}$, $\text{Referenz}_{\mathbf{32}}$, ...
...	...
$\text{Wort}_{\mathbf{n}}$	$\text{Referenz}_{\mathbf{n1}}$, $\text{Referenz}_{\mathbf{n2}}$, $\text{Referenz}_{\mathbf{n3}}$, ...

- **Vokabular** (Indexterme):
  - Enthält alle Worte für die der Index Daten indiziert hat
  - Evtl. Unterscheidung von Groß-/Kleinschreibung
  - Evtl. Entfernung von Stoppworten (Artikel, Präpositionen)
  - Evtl. Reduktion auf den Wortstamm (Entfernung von Affixen)
- **Postlisten**: Jedem Wort ist eine Liste mit Referenzen auf Nutzdaten zugewiesen.
- **Herausforderung**: Implementierung einer Datenstruktur, die bei wenig Speicherverbrauch einen schnellen Zugriff auf eine invertierte Liste gewährt.

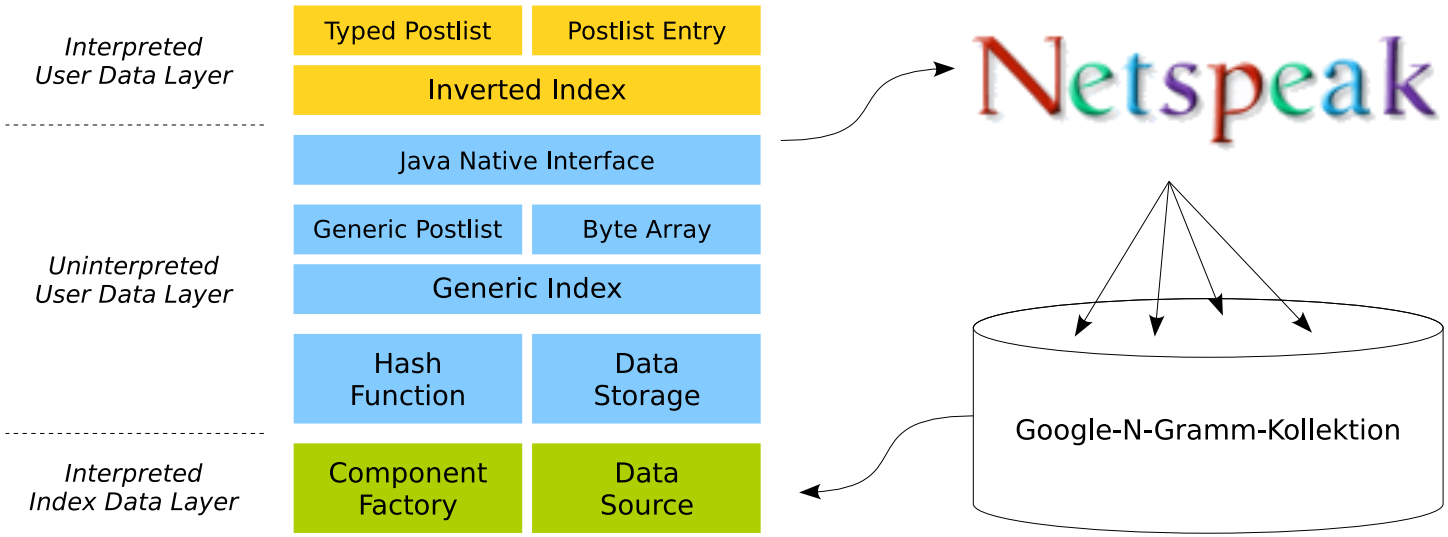
C++ Implementierung: Komponenten



- DataSource parst eine zu indizierende Kollektion oder (pseudo-) invertierte Textdateien und erzeugt daraus DataItems.
- HashFunction dient der Zuweisung von DataItems auf GenericPostlists.
- DataStorage implementiert eine Strategie zur Vorhaltung bzw. Bereitstellung von GenericPostlists (interner oder externer Speicher oder Cache).



C++ Implementierung: Java Anbindung



- Aufbau in 3 Schichten nach Repräsentation der Daten:
  - Interpreted Index Data Layer: Daten werden zum Parsen exakt interpretiert (Strings, Integer).
  - Uninterpreted User Data Layer: Postlisten enthalten (neutrale) Byte-Arrays
  - Interpreted User Data Layer: Byte-Arrays werden wieder exakt interpretiert (Strings, Integer).
- Generische Schnittstelle zu Java (JNI): Postlisten werden als Byte-Arrays übertragen.

C++ Implementierung: Minimale Perfekte Hashfunktion

Der Index verwendet eine minimale perfekte Hashfunktion (MPHF)
um jeden Indexterm aus dem Vokabular auf einen Ganzzahlenwert abzubilden.

Anwendung:

- Indizierung: Einsortierung der zu indizierenden Daten in Postlisten
- Suche im Index: Bereitstellung der Postliste eines gesuchtes Wort

Eigenschaften:

- Eine Hashfunktion ist *perfekt*, wenn sie keine Kollisionen erzeugt.
- Eine Hashfunktion ist *minimal perfekt*, wenn sie *n* Indexterme
  auf das halboffene Intervall [0,n) ohne Kollisionen abbildet.

Vorteile einer MPHF:

- Optimaler Speicherverbrauch der Hashtabelle, da keine leeren Slots
- Keine Kollisionsbehandlung notwendig

Der Index kann eine MPHF einsetzen, da das Vokabular im Vorhinein bekannt ist.

C++ Implementierung: Fakten

Limitierungen:

- Größe einer Postliste: Maximale Dateigröße des Dateisystems (FAT32: 4 GB, NTFS/Ext3/ReiserFS: Festplattengröße)
- Größe eines Index: Festplattengröße
- Größe des Vokabulars: Maximaler signed Integer (2.147.483.647)

Netspeak Index:

- Größte Postliste: Wort "the" mit 1,7 GB und 156 Mio. Einträgen
- Gesamte Indexgröße: rund 30 GB
- Speicherverbrauch zur Laufzeit: 200 MB
- Größe des Vokabulars: rund 3 Mio. Worte

Demonstration

http://Netspeak.webis.de

Ende

Danke für Ihre Aufmerksamkeit

Fragen ?