Produktvergleiche linguistische Stemmer

Getestest wurden vier Tools:

- SMOR von Centrum für Informations- und Sprachverarbeitung in München (SMOR)
- Linguistic Engine von der IAI Linguistic Content AG (iailc)
- Find-It von der Canoo Engineering AG (canoo)
- Search Plugin von IntraFind (intrafind)

Zusammenfassung

	SMOR	Iailc	Canoo	Intrafind	Bemerkung	Komplexitätsfaktor (ca.)
Analysequalität	Mittel	Sehr gut	Mittel	Gut (keine Stammformbildung)	Kompositazerlegung, Lemmatisierung, Stammformbildung, Disambiguierung	7
Erweiterungsbedarf bei der Indexierung	Hoch	Mittel	Hoch	Sehr niedrig	Anbindung des Tools in die Indexierung	5
Änderungsbedarf in der bestehenden Anwendung	Mittel	Niedrig	Mittel	Hoch	Einbindung des Tools in die bestehende Suche	1
Änderungsbedarf für verbessertes Ranking (nur anhand des Suchwortes)	Hoch	Hoch	Hoch	Mittel – (Niedrig + Risiko für Schwierigkeiten bei eigenem Ranking)	Treffer zu „Stellen“ in einem Feld werden gewichtet: „Stellen (Nomen)“ vor „stellen (Verb)“ vor „Stelle“ vor „Ausbildungsstelle“ vor „Stellenbeschreibung“ vor „Ausbildungsstellenangebot“ vor	5

					„vorange-stellt“	
Erweiter-barkeit (Synonyme, Boosting einzelner Begriffe, etc.)	Sehr gut	Gut	Mittel	Nicht mög-lich	Bedarf für Erweiter-barkeit noch unklar	
Erwarteter Pflegeauf-wand	Hoch	Mittel	Mittel	Niedrig	Aktualisie-ren des zu-grundelie-genden Le-xikons und Aktualise-ren der Ein-bindung bei neuer Ver-sion von Elasticse-arch	

Besonderheiten

SMOR

+ Quellcode verfügbar

+ Erweiterbar auf Named Enitity Recognition (für Verschlagwortung und z.B. Ortsnamen Extraktion)

- nicht direkt für die Textzerlegung bei der Indexierung einsetzbar (Extraktion Sonderzeichen etc).

- kein kommerzieller Support (aber Ansprechpartner vorhanden und Pflegevertrag möglich)

- Bei Kompositazerlegung ist die Unterscheidung nach relevanten und irrelavanten Zerlegungen schwierig z.B. wird „Verkäuferinnen“ auch nach „verkauf#rinne“ zerlegt

- Anbindung über jni muss selbst programmiert werden

Iailc

+ Zerlegung in Stammformen meist eindeutig

+ Verbesserung für alle Suchen der Elasticsearch Query DSL erwartbar

+ Vorschläge für relevanten Zerlegungen bei längeren Wörtern

- keine reine Java-Anwendung (Anbindung über jni)

Canoo

+ Java API

- Zerlegungen schwer nach relevanten und irrelevanten Möglichkeiten unterscheidbar z.B. wird „Sanktionszeitraum“ auch nach „sankt#ionium#zeitraum“ zerlegt und „Ökonomie“ wird nach „Ök#o#nom#ie“ zerlegt.

Intrafind

+ Produkt extra für die Suche entwickelt und als Elasticsearch Plugin verfügbar

+ Ranking anhand des Vorkommens des Teilwortes in der Kompositazerlegung

+ Mehrsprachigkeit innerhalb eines Dokumentes

- Festlegung auf den Lucene Query Parser (z.B. Nutzung von Common Terms Query nicht mehr möglich)

- kein Synonym-Lexikon einbindbar (aktuelle Version, vermutlich mit nächster Version verfügbar)

- keine Stammwortbildung: arbeitslos kann nicht Arbeitslosigkeit finden; krank kann nicht Krankheit finden

Fachliche Anforderungen

Die Suche ist ein zentraler Zugangskanal zu den Informationsangeboten der Homepage. Dabei ist zu beachten, dass das Publikum sehr heterogen ist, in Sprachfähigkeit wie in der Perspektive auf das vorgebrachte Anliegen. Entsprechend vielfältig formuliert sind die Suchanfragen, die zum jeweils passsenden Inhalten geleitet werden müssen. Bei mehreren zehntausend aktiven Inhalten ist von entscheidender Bedeutung, dass die verwendeten Such- und Stemmertechnologien schon in ihrer Grundkonfiguration sehr gute Treffermengen liefern.

Zusätzlich ist erforderlich, dass bei den zentralen Anliegen Ranking und Einstiegspunkte flexibel beeinflusst und korrigiert werden können, vor allem in Bezug auf

- Finden einzelner wichtiger Dokumente (boost einzelner Dokumente und direkte Zuordnung zu bestimmten Suchwörtern und Phrasen)
- Anpassung und Optimierung der Ergebnissreihenfolge (Ranking) auf Grundlage von Analysen typischen Nutzerverhaltens in der Vergangenheit
- Schnelle Einschränkung bzw. Umgewichtung der Suchergebnisse aus bestimmten Themenkomplexen/Anliegen auf Basis von Segment- und Profilinformationen
- Verhindern von Null-Treffer Suchergebnissen durch zu scharfe Trefferbegrenzungen

Vorgehen des Produktvergleiches

Für den Vergleich der linguistischen Tools wurde die Auswirkung auf die Suchergebnisse simuliert.

Dabei wurde das Vorgehen für den Vergleich der heuristischen Stemmer angepasst:

Beim Vergleich der heuristischen Stemmer wurde zu einer Wortliste je Wort verglichen, welche anderen Wörter damit gefunden werden.

Dafür wurde kein wirklicher Lucene Index aufgebaut sondern je Wort durch den Stemmer ein Token erzeugt. Ein Wort findet ein anderes Wort, falls die beiden erzeugten Token übereinstimmen.

Im Unterschied zu heuristischen Stemmern erzeugen die linguistischen Stemmer bei Mehrdeutigkeiten (Sucht als „er sucht“ und als „Alkohol-Sucht“) und bei Kompositazerlegung („Pilotenjacke“ enthält „Jacke“ und „Pilot“) mehrere Token je Wort.

Die Query DSL von Elasticsearch erwartet für die meisten Klauseln, dass jedem Wort maximal ein Token zugeordnet wird.

Erst seit Version 5.2 ändert sich dieses schrittweise

https://www.elastic.co/blog/multitoken-synonyms-and-graph-queries-in-elasticsearch

https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-synonym-tokenfilter.html

Für den Vergleich der Tools wurde daher für drei der Tools je Wort ein Token ausgewählt, um es bei der Suche zu präsentieren.

Je Wort gibt es also beliebig viele Token für die Indexierung aber nur einen Token für die Suche. Zu diesem „Query"-Token wurden dann alle Wörter ausgewählt, für die das Tool diesen Token selbst auch für die Indexierung erzeugt (dann als einen von vielen).

Insgesamt ergibt dies je Wort eine Liste von anderen Wörtern, die über ersters Wort gefunden werden.

Als Basis für die Wörter wurden alle Wörter gewählt, die auf der Homepage Mitte Juni 2017 gefunden wurden (nur html).

Getestest wurden vier Tools: Canoo(Find-it), iailc(Linguistic Engine), Intrafind(Linguistik Plugin) und SMOR

Intrafind ist als einziges getestetes Tool direkt als Plugin für Elasticsearch verfügbar und bringt eine Erweiterung der Elasticsearch Query DSL mit, die mit mehreren Token je Wort suchen kann. Daher wurden für Intrafind die Wortlisten anders erzeugt:

Je Wort wurde ein Dokument in ES angelegt und dann je Wort in ES gesucht. Die Treffer bilden dann die Liste je Wort analog zu den gerade beschriebenen Listen.

Damit liefert Intrafind je Wort normalerweise die längsten Wortlisten. Intrafind rankt dabei die Ergebnisse so, dass Wörter oben stehen, wenn zu dem Suchwort eine größere Ähnlichkeit besteht (Übereinstimmungen in der Komposita-Zerlegung, siehe „ES Query mit Gewichtung" in der Funktionsmatrix).

Im Ergebnis gibt es je Wort eine Liste von gefundenen Wörtern.

Je zwei Stemmer lassen sich dann dadurch vergleichen, dass pro Wort die zugehörigen Listen je Wort verglichen werden.

Pro Wort wird dabei festgestellt, welche Wörter von beiden Stemmern (A und B) gefunden werden, welche nur von A und welche nur von B.

Die Tools

- „smor",
- „canoo" (Find-It) und
- „iailc"

liefern nicht direkt Token sondern Wortzerlegungen mit zugehörigen Informationen über z.B. Wortart und Grammatik.

Für die drei Produkte wurde jeweils ein Programm geschrieben um die Token zu erzeugen. Dabei wurde versucht die sich dadurch ergebenen Wortlisten möglichst ähnlich zu den Ergebnissen von IntraFind zu erzeugen.

Danach wurden die Wortlisten nur für die hundert Top-Suchbegriffe (Stand 2016) manuell verglichen, mit dem Ziel festzustellen, ob die Tools jeweils alle Informationen liefern könnten, um theoretisch gleiche Token wie Intrafind zu erzeugen.

Die Erfahrungen mit den Tools und die Unterschiede in den Tool lieferten die Grundlage der Bewertungsmatrix.

Für jedes Tool wurden Wörter gefunden, die nicht korrekt zerlegt wurden. Wenn das Tool hierzu eine Fehlermeldung ausgibt, wurden diese Wörter nicht verglichen. Ohne Fehlermeldung wird das Wort meistens nur von sich selbst gefunden (weil nur in Token erzeugt wurde, und dieser das originale Wort enthält. Alle Anbieter erklärten sich bereit, ihre Lexika zukünftig so zu erweitern, dass alle deutschen (echten, korrekt geschriebenen) Wörter der jetzigen Homepage auch zerlegt bzw. lemmatisiert werden können.

Besonderheiten

Canoo

Auf der Homepage von Canoo gibt es eine Anwendung, ob die sich das Tool für einzelne Wörter testen läßt:

http://www.canoo.net/services/Controller?input=Teststellungsabschluss

Wie dort zu sehen, ist der Fokus von Canoo darauf, alle Möglichkeiten der Bedeutung und Zerlegung anzuzeigen.

Andere Tools sind in der Zerlegung sparsamer und eindeutiger.

So findet Canoo in Ladenöffnungszeiten als einziges Tool die „Lade“ wie „Bundeslade“ als Wortbestandteil.

Canoo ist auch das einzige Tool, dass „Ökonomie“ weiter zerlegt(Ök#o#nom#ie):

http://www.canoo.net/wordformation/%C3%B6konomie:N:F

und das einzige, das „Brücke“ in „überbrücken“ findet, das „eins“ in „Einführungszeitraum“ findet und „Ionium“ in Sanktionszeitraum

http://www.canoo.net/services/Controller?input=Sanktionszeitraum&service=canooNet

Bei der Suche mit nur einem Token (Standard-ES-Verhalten) ist es daher sehr schwer diesen so zu wählen, dass die Treffermengen nicht zu eingeschränkt sind.

Laut Hersteller werden bei den Zerlegungen standardmäßig nur drei Wortbestandteile erwartet. Diese Zahl ließe sich auf vier erhöhen, was zu deutliche längeren Analysezeiten führen soll (wahrscheinlich trotzdem akzeptable Zeiten).

http://www.canoo.net/services/Controller?input=Teststellungsabschluss-party&service=canooNet

vs.

http://www.canoo.net/services/Controller?input=Teststellungsabschlusspar-tygast&service=canooNet

(letzteres kann nicht zerlegt werden).

Canoo bietet keine Möglichkeit alle Wortbestandteile in einem Schritt zu erzeugen. Anderseits ist die vorhandene Zerlegung in Bestandteile oft nicht ausreichend bzw. irreführend.

So enthält eine der Zerlegungen zu Vollzeitstudiengang „Zeitstudie“ ober keine Zerlegung „Vollzeit“ als einen Bestandteil.

Für einen Analyzer in Elasticsearch müsste daher jeder Bestandteil der Zerlegung selbst nochmal zerlegt werden.

http://www.canoo.net/services/Controller?input=Vollzeitstudiengang&service=canooNet

Dies kann zu unnötigen Mehrdeutigkeiten führen:

http://www.canoo.net/wordformation/zeitstudie:N:F

Zeitstudie liefert nicht mehr „Studium“ als Grundform (sondern „Studie“) aber

http://www.canoo.net/wordformation/studiengang:N:M

liefert (nur) „Studium“.

SMOR

Smor ist ein Open-Source-Tool. Kostenpflichtig ist das Stammwortlexikon. Smor erlaubt bei der Zerlegung und Stammwort-Bildung mit einem Parameter die Ausgabe auf die simpelste Analyse einzuschränken (disambiguate symbolically - print the simplest analyses).

Damit wird dann z.B. Köchin nicht mehr auf Koch zurück geführt und „saisonüblich“ wird nicht mehr nach „saison#üblich“ aufgeteilt und „Verkäuferinnen“ wird nach „verkauf#rinne“ aufgeteilt.

„Jobsuche“ wird aufgeteilt, aber nur noch nach job#suche nicht auch nach job#suchen während „Jobsuchmaschine“ immer auf „job#suchen#maschine“ zurück geführt wird (d.h. Jobsuche kann nicht Jobsuchmaschine finden).

Auch bei simpler Analyse ist die Zerlegung nicht immer eindeutig („Freizeitraum“ -> „frei#zeitraum“ und „freizeit#raum“).

Die Tests erfolgten mit der simpleren Analyse. Für die hundert Top-Suchbegriffe wurde manuell überprüft, ob SMOR die Wortlisten von intrafind auch erzeugen könnte. Dem ist so, wenn SMOR ebenfalls mehrere Token je Wort in der Suchanfrage erzeugen dürfte.

Ohne Einschränkung auf simple Analyse werden meistens mehr Zerlegungen gefunden als bei intrafind oder iailc, damit auch viele ohne Relevanz bzw. Zerlegungen die erschweren Prefix und Suffix aus der Wortbildung zu erkennen.

Beispiele:

„saisontypischer“ wird zerlegt zu

Saison#typisch, Saison#Type#isch, Saison#Type#isch

„Unfallversicherung“ wird zerlegt zu

Unfall#ver#sichern#ung, Unfall#versichern#ung, Unfall#Versicherung

Unfallversicherungsmodernisierungsgesetz wird in sechs Varianten zerlegt.

Durch Analyse eines ganzen Textes statt nur eines Wortes lassen sich diese Varianten einschränken. Dafür ist ein weiteres Open-Source-Tool nötig (TreeTagger).

Da hinter SMOR keine kommerzielle Firma steht, ist die automatische Verbesserung/zeitnahe Pflege des Stammwortlexikons nicht sicher. Dafür liegt das Stammwortlexikon in einem lesbaren und damit theoretisch auch pflegbarem Format vor.

iailc

Das Tool vom IAI bietet als einziges eine direkte Zerlegung eines zusammengesetzten Wortes in die kleinsten Bestandteile in einem Schritt.

Unfallversicherung → unfall#versichern~ung

Unfallversicherungsmodernisierungsgesetz → unfall#versichern~ung#modernisieren~ung#gesetz

saisontypischer → saison#typ~isch

Freizeitraum → frei#zeit#raum

Da diese Zerlegung meistens eindeutig ist, lässt sich damit sinnvoll ein Elasticsearch Query Analyzer bauen, mit dem die Elasticsearch Query DSL ohne Änderungen genutzt werden kann. Aber auch hier gilt, dass mehrere Token je Query die Ergebnisse verbesserten (sucht findet suchen und Sucht); insbesondere wenn der Nutzer nach einem Imperativ wie „Suche!“ oder „Schule!“ suchen will.

Das Tool bittet für viele Wörter auch empfohlene Zerlegungen an wie

Vollzeitstudiengang → vollzeitstudium;vollzeit;zeitstudium;studiengang

Berufsunfähigkeitsversicherung → berufsunfähigkeit

mit dem ein höheres Ranking für sinnvolle Zerlegungen möglich wäre.

Intrafind

Intrafind unterstützt aus der Gesamtheit der Elasticsearch Query DSL nur den Lucene Parser an („query_string“), welcher absehbar alle Anforderungen der Suche erfüllt.

Die Suchanfragen gegen den IF-Analyser sind in der (erweiterten) Lucene-Query-Syntax.

Aus Elastisearch Sicht arbeitet "intrafind_query_string“ analog zu der Query "query_string“ und kann entsprechend allgemein eingebunden werden.

Die Suche in mehreren Feldern wird unterstützt.

Die wurde auch um eine dis_max Ranking-Funktion erweitert. "intrafind_query_string“ kann –mit entsprechendem Parameter- wie https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-query-phrase.html eingesetzt werden.

Arbeitslosigkeit wird nur durch Arbeitslosigkeit gefunden (nicht durch Arbeit und nicht durch arbeitslos), weil keine Stammformbildung gemacht wird, sondern „nur“ Lemmatisierung und Kompositaverlegung.

Andererseits werden zum Teil die Vorsilben von Verben abgetrennt, so dass „arbeiten“ auch „nacharbeiten“ (aber nicht umgekehrt, da hier nach beiden Wörtern („nach“ und „arbeiten“) gesucht wird.

An eine Erweiterung auf Stammformbildung wird gedacht, es ist jedoch nicht sicher, ob dies kommt.

„Abijahrgang“ ist ein Beispiel, wo das Lexikon angepasst werden sollte (kann nicht von Jahrgang gefunden werden).

Das Lexikon wird nur von intrafind angepasst, grundsätzlich ist dies laut Herstellerangabe nicht aufwändig.

Bisher ist keine Möglichkeit vorgesehen, das Verhalten des Analysers zu überschreiben (z.B. weil bestimmte Teilungen erwünscht sind oder das Wort bisher nicht bekannt ist).

Eine solche Whiteliste einzuführen wäre möglich, vermutlich auch für andere Kunden sinnvoll und nicht allzu schwer umzusetzen, aber ist bisher nicht in der Planung.

Das Wort Förderhöchstgrenze wird von Höchstfördergrenze gefunden, weil zu den Komposita auch ein Phrasen Suche (genauer Span-Query) gestartet wird (da durch findet „Risikoversicherung“ auch „Versicherung des Risikos“), allerdings findet deshalb „Hausfrau“ auch „Frauenhaus“.

Lucene-Attribute/Payloads werden für die Suche nicht benötigt, d.h. die meisten Codecs funktionieren. Für Sonderfunktionen wie die Gewichtung werden zu den Token unterschiedliche Prefixe vergeben. Alle wesentlichen Informationen sind also über

http://localhost:9200/infrafind/_analyze?analyzer=default_index&text=Arbeitslosigkeit

abfragbar.

Für Schlagwörter wäre vorstellbar bei der Indexierung keine Kompositazerlegung vor zu nehmen. Bisher ist dieses nur für alle Felder gleichzeitig konfigurierbar.

Synonyme könne bisher nicht über den Analyser eingebunden werden. Mit der neuen Lucene-Phrasensuche wäre dies möglich und es wird laut Herstellerangabe wahrscheinlich nächstes Jahr auch kommen, ohne dass dafür eine Zusage gemacht wird.

Funktionsmatrix

	SMOR	Iai	Canoo	Intrafind	Beispiel / Erläuterung
Komposita-Zerlegung	Ja Konfigurierbar zwischen „wenige“ und „viele“	Ja Mit Empfehlung der „besten“ Zerlegungen	Ja	Ja	Vollzeitstudium →Vollzeit, Studium Vorjahreszeitraum → Vorjahreszeit, Zeitraum
Stammformbildung	Ja – in einem Aufruf Schritt Zwei Aufrufe bei ignorieren der Groß/Kleinschreibung	Ja – in einem Aufruf Schritt	Ja – teilweise mehreren Aufrufe nötig	Nein Suche nur über Komposita-Zerlegung und Lemmatisierung	Vorjahreszeitraum → vor#jahr#zeit#raum Vormerkliste → vor#merken#liste
Disambiguierung durch Satzzusammenhang	Ja – eigenes kostenloses Produkt	Ja	Nein – zumindest nicht über die API	Zum Teil	Sucht → suchen (Verb, „Sucht nicht weiter“) Sucht → sucht (Nomen)
Nutzbar mit ES Query DSL	Ja, eingeschränkt – bei Wahl der wahrscheinlichsten Zerlegung	Ja, eingeschränkt – bei Verzicht auf unwahrscheinliche Zerlegung	Ja, eingeschränkt – bei Wahl der richtigen Zerlegung	Nein – nur "query_string" mit Lucene Syntax	ES Query DS erwartet meistens nur eine Token je Wort
ES Query mit Gewichtung nach Art der Zerlegung	Programmierbar	Programmierbar	Programmierbar	Ja	Treffer zu „Stellen“ in einem Feld werden gewichtet: „Stellen (Nomen)“ vor „stellen (Verb)“ vor „Stelle“ vor „Ausbildungsstelle“ vor „Stellenbeschreibung“ vor „Ausbildungsstellenangebot“ vor „vorangestellt“
Umlaute ausschreibbar (oe statt ö)	ja	Ja	ja	Ja	Suche nach „loeschen“ findet „löschen“
Umlaute weglassbar (o statt ö)	Nein	Nein	Nein	Nein	Suche nach „loschen“ findet „löschen“
Fugen-S weglassbar	ja	Zum Teil, Erweiterung vorgesehen	Nein	Zum Teil, Erweiterung vorgesehen	Rentenantragstellung wird wir Rentenantragsstellung behandelt.

„selbst-ständig“ wie „selb-ständig“	Nein, pro-grammierbar	Ja	Ja	Ja	
Erweiter-bar um Sy-nonyme	Ja – program-mierbar	Ja – pro-grammier-bar	Ja - pro-grammier-bar	Nein, mögli-cherweise ab nächstem Jahr	
Andere Sprachen: Englisch, Franzö-sisch	Ja	Ja	Ja	Ja	
Eigene Pflege der Wortzerle-gungen möglich	Ja	Ja – als Program-mierung in den Analy-sern	Ja – als Program-mierung	Nein, mögli-cherweise als Er-weiterung (Whitelist)	
Vorschläge bei Recht-schreibfeh-lern	Nein, pro-grammierbar	Nein, be-grenzt program-mierbar	Ja, teil-weise und program-mierbar	Nein, aber Spracherken-nung (Fremd-worte nicht wie Recht-schreibfehler)	`detailierte statt detaillierte Anlagenmechinaker statt Anlagenme-chaniker`

Integrationsaufwände

	SMOR	Iai	Canoo	Intrafind
Aufruf aus ES	– mittel – Aufruf von C-Libary	– mittel – Aufruf von C-Li-bary	– Niedrig - Aufruf von Java API	– Sehr niedrig – Eigenes ES Plugin
Programmie-rung Analyzer für die Indexie-rung	– mittel – Keine Unter-stützung bei Auswahl der sinnvolleren Zerlegungen	Niedrig – Möglichkeiten der Zerlegung meistens ein-deutig vorge-geben	– Hoch – Wechsel von Stammform-bildung und Kompositazer-legung nötig. Keine Zerle-gung ganzer Sätze.	Entfällt – Ana-lyzer vorhan-den
Notwendige Erweiterung des Lexikons	Mittel – Sehr ehrliche API: Ausgabe „no results“	Niedrig –einige Beispiele ge-funden. auskunftsfreu-dige API, er-kennt auch Na-men	Mittel – einige hundert Beispiele gefunden	Sehr niedrig – ein Beispiel und mehrere Beispiele mit fehlendem Fu-gen-S
Programmie-rung Analyzer für die Stan-dard-ES-Suche	Mittel – Risiko, dass Empfeh-lung für beste Zerlegung	Niedrig – Kleinstmögli-che Zerlegung in Stammworte vorgegeben	Hoch – Risiko, dass es nicht möglich ist, sinnvoll auszu-wählen. Tool hat keine Empfehlung für die beste Zerlegung. Seltene Zerle-gung schwer erkennbar	Nicht möglich (bzw. nicht nö-tig)

	selbst programmiert werden muss.	und häufig eindeutig. Empfehlung für Komposita häufig gegeben.	(Sankt#Ion#Zeitraum für Sanktion http://www.canoo.net/services/Controller?input=Sanktionszeitraum&service=canooNet)	
Programmierung Analyzer und Query Parser für Suche mit Gewichtung	Hoch	Hoch	Hoch	Entfällt – Analyzer vorhanden für Lucene DSL Syntax