Bachelorarbeit

Automatische Vorschläge zur Vervollständigung von Wikipedia-Listen - Evaluation, Anfragegenerierung

Simon Skilevic

18. April 2012

Albert-Ludwigs-Universität Freiburg im Breisgau
Technische Fakultät
Institut für Informatik

Bearbeitungszeitraum
18. 01. 2012 – 18. 04. 2012

Gutachter
Prof. Dr. Hannah Bast

Erklärung

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen/Hilfsmittel verwendet habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung angefertigt wurde.

Inhaltsverzeichnis

Zusammenfassung

Wikipedia-Listen sind Wikipedia-Artikel, die Listen zu einem Thema enthalten. Zum Beispiel enthält die Wikipedia-Liste mit dem Thema „essbare Pilze“ als Listenelemente unterschiedliche Arten von Pilzen, die man ohne Risiko für die eigene Gesundheit verzehren kann. Da solche Listen manuell erstellt werden, können sie unvollständig sein. In dieser Arbeit beschreiben wir ein Programm, WikiListSuggest, das Vorschläge zur Vervollständigung solcher Listen liefert. Dafür extrahiert das Programm entsprechende Listenelemente einer Wikipedia-Liste und generiert automatisch eine Anfrage an die Semantische Suchmaschine “Broccoli”. Durch diese Anfrage soll WikiListSuggest möglichst die vollständige Liste zum Thema der Wikipedia-Liste erhalten. Diese Liste kann dann zur Vervollständigung der Wikipedia-Liste verwendet werden. Schwerpunkte dieser Arbeit sind die Anfragegenerierung sowie die Evaluation.

1 Einleitung

Die Online-Enzyklopädie „Wikipedia“ enthält Millionen von Artikeln zu allen möglichen Begriffen. Einige Fragen können jedoch nicht durch Artikel beantwortet werden, die nur einen bestimmten Begriff beschreiben. Sagen wir zum Beispiel, man sucht eine Liste essbarer Pilze. Hierfür würde es sehr lange dauern, alle Wikipedia Artikel, die Pilze beschreiben, durchzulesen und herauszufinden, welche gegebenenfalls essbar sind. Dieser Aufwand lässt sich vermeiden, wenn die Antwort als fertig recherchierte Liste von Artikeln essbarer Pilze bereits in Wikipedia existieren würde.

1.1 Wikipedia-Listen

List of actors who played President of the United States

Actor	President	Movie	Year
Duke R. Lee	George Washington	*In the Days of Daniel Boone*	1923
Frank Windsor	George Washington	*Revolution*	1985
Jeff Daniels	George Washington	*The Crossing*	2000
Terry Layman	George Washington	*The Patriot*	2000
William Daniels	John Adams	*1776*	1972
Pat Hingle	John Adams	*Independence*	1976
Ken Howard	Thomas Jefferson	*1776*	1972
Ken Howard	Thomas Jefferson	*Independence*	1976
Nick Nolte	Thomas Jefferson	*Jefferson in Paris*	1995

Abbildung 1.1: Beispiel einer Wikipedia-Liste

Eine *Wikipedia-Liste* ist ein Wikipedia-Artikel, der im Gegensatz zu üblichen Artikeln, nicht nur einen bestimmten Begriff beschreibt. Stattdessen enthält er eine Liste von Begriffen zu einem bestimmten Thema. Eine Wikipedia-Liste kann als Liste, Tabelle oder auch als andere Struktur, die sich für eine Auflistung eignet, dargestellt werden. Um Verwirrungen zu vermeiden, werden wir unabhängig von der Struktur immer von der Liste und deren *Listenelementen* sprechen.

Wikipedia-Listen werden aufwändig manuell erstellt und existieren natürlich nicht für alle möglichen Themen. Dafür bekommt man aber eine hohe Sicherheit, dass

die in einer Wikipedia-Liste enthaltenen Listenelemente auch wirklich in diese Liste hinein gehören.

1.2 Ziel der Arbeit

Das Ziel unserer Arbeit ist es ein Programm zu entwickeln, um Erstellung und Erweiterung der englischsprachigen Wikipedia-Listen dadurch zu erleichtern, dass es Vorschläge für die neuen Listenelemente einer Wikipedia-Liste automatisch generiert.

Um dies zu erreichen, werden folgende Schritte von WikiListSuggest durchgeführt:

- Ermittlung der Listenelemente und des Titels einer Wikipedia-Liste, um das Thema der Liste herauszufinden und die schon existierenden Elemente zu kennen.
- Generierung einer Anfrage an die semantische Suchmaschine „Broccoli“, um die Vorschläge für weitere Listenelemente zu bekommen.
- Bereitstellung eines bequemen User-Interfaces, um aus den Vorschlägen tatsächliche neue Listenelemente zu ermitteln.

Im Folgenden werden diese Herausforderungen genauer erklärt:

1.2.1 Ermittlung der Listenelemente (Listenerkennung).

Wikipedia-Listen haben keine einheitliche Form und unterscheiden sich in der Struktur voneinander. Um darin Listenelemente überhaupt zu erkennen, wird die Tatsache ausgenutzt, dass Listenelemente oft Verweise auf ihre Artikel enthalten. Diese Erkenntnis reicht allein leider nicht aus. Denn ein Wikipedia-Listen-Artikel kann viele andere Elemente besitzen, die zwar Verweise haben, aber keine Listenelemente sind. Auf der ***Abbildung 1.1*** sehen wir zum Beispiel eine Liste von Schauspielern, die die Rolle eines US-Präsidenten gespielt haben. Für das menschliche Auge ist es leicht zu erkennen, dass nur die erste Spalte Listenelemente enthält. Für ein Programm dagegen muss erstmals definiert werden, was überhaupt eine Spalte ist.

Um Listenelemente automatisch zu identifizieren, betrachtet WikiListSuggest daher die HTML-Struktur der Wikipedia-Liste. Es untersucht dabei alle Verweise, die als Kandidaten in Frage kommen (Verweise, die z.B. aus dem Inhaltsverzeichnis kommen, werden nicht betrachtet). Für alle Kandidaten werden die letzten Elemente ihres DOM-Pfades untersucht, um die Kandidaten anhand ihrer HTML-Strukturen zu gruppieren. Dadurch bekommt das Programm unterschiedliche Gruppen von Kandidaten, die zum Beispiel als Spalten einer Tabelle interpretiert werden können.

Als nächstes werden allen Kandidaten ihre Relevanz-Werte vergeben (je höher, desto größer die Wahrscheinlichkeit, dass es ein entsprechendes Listenelement ist).

Dabei betrachtet man die Ontologie-Kategorien, in denen die Kandidaten vorkommen. *Ontologie-Kategorien* sind Kategorien, die verschiedene Begriffe unter einem Oberbegriff in sich vereinen (z.B. enthält die Kategorie „Pilz“ alle Begriffe, im semantischen Wortfeld „Pilz“ vorkommen können). Dabei können Begriffe bei mehreren Kategorien vorkommen. Zum Beispiel kommt „Birkenpilz“ in den Kategorien „Organismus“, „Lebewesen“, „Pilz“ usw. vor.

Für jede Kategorie, die mindestens eines der Listenelemente enthält, wird ein Wert ausgerechnet, der sich aus den Häufigkeiten der Vorkommnisse auf der Seite (Wikipedia-Liste), in der Ontologie und im Seitentitel ergibt. Die Summe der Kategorie-Werte eines Kandidaten ist dann sein *Relevanz-Wert*.

Am Ende werden die Relevanz-Werte für jede Gruppe zusammenaddiert. Die Gruppe mit der größten resultierenden Summe wird von WikiListSuggest als die gesuchte Menge der Listenelemente behandelt. Näher wird dieses Teilproblem in der Arbeit von Robin Schirrmeister erläutert[Sch12].

1.2.2 Generierung einer Anfrage an die semantische Suchmaschine 'Broccoli'

Nach der erfolgreichen Ermittlung der Listenelemente generiert WikiListSuggest automatisch eine Anfrage an 'Broccoli', eine semantische Suchmaschine, die in Wikipedia nach den Vorschlägen für neue Listenelemente suchen soll. Broccoli erlaubt es dem Benutzer, seine Suchanfrage genauer als bei klassischen Volltext-Suchmaschinen zu definieren. So kann der Nutzer Broccoli zum Beispiel nach Begriffen aus der Kategorie „Pilz“ (die alle als Pilze bekannten Begriffe enthält) suchen lassen, die im Zusammenhang mit dem Wort „essbar“ auftauchen. Und der Benutzer kann sicher sein, dass die gefundenen Begriffe Pilze und nicht was anderes sind.

Die Voraussetzung für eine derartige Suche ist eine möglichst umfangreiche und genaue *Ontologie* (Wissensdatenbank), die Fakten über möglichst viele *Entitäten* (Begriffe) in der Form „Birkenpilz ist ein Pilz“, „Ein Pilz ist ein Organismus“ enthält.

Für die Erstellung einer Anfrage an Broccoli versucht WikiListSuggest möglichst viele Informationen der Wikipedia-Liste zu entnehmen. So steht zum Beispiel im Titel der Seite der Name der Liste, aus dem man eventuell schließen kann, um welche Art von Begriffen es sich bei den Listenelementen handelt. Weitere Informationen lassen sich aus den Listenelementen selbst extrahieren. Diese Informationen werden dazu genutzt, um zwei Mengen zu generieren.

Die erste Menge, nennen wir sie *Kategorie-Menge*, enthält Kandidaten für die Ontologie-Kategorie der Wikipedia-Liste. Für die Beispiel-Anfrage „Essbare Pilze aus Deutschland“ sollte dann in dieser Menge die Kategorie „Pilze“ enthalten sein.

Die zweite Menge, nennen wir sie *Eigenschaftsmenge*, enthält Wörter, mit denen ein Suchtext gebildet werden soll. Dabei enthält der *Suchtext* eine Menge von Eigen-

schaften, die Listenelemente möglichst gut beschreiben sollen. Für das obige Beispiel wäre der Suchtext „essbar aus Deutschland“ gewesen.

Durch Kombinieren der Kandidaten aus der zweiten Menge bildet man mehrere Suchtexte. Aus diesen Suchtexten baut WikiListSuggest durch Kombinationen mit jeder der Kategorien aus der Kategorie-Menge mehrere Anfragen der folgenden Form:

Suche nach dem **Suchtext** und beschränke das Ergebnis auf Begriffe aus **Kategorie**.

Jede dieser Anfragen wird dann getestet, indem man sie an Broccoli stellt und die Übereinstimmung der Ergebnisliste mit der Wikipedia-Liste betrachtet. Die Anfrage, die zur besten Übereinstimmung geführt hat, wird schließlich an das User-Interface von WikiListSuggest weitergeleitet und dem Benutzer präsentiert (Auf *der* ***Abbildung 1.2*** kann man die generierte Anfrage in Broccoli-Box sehen). Genauer wird das Vorgehen in ***Kapitel 3*** erläutert.

1.2.3 User-Interface

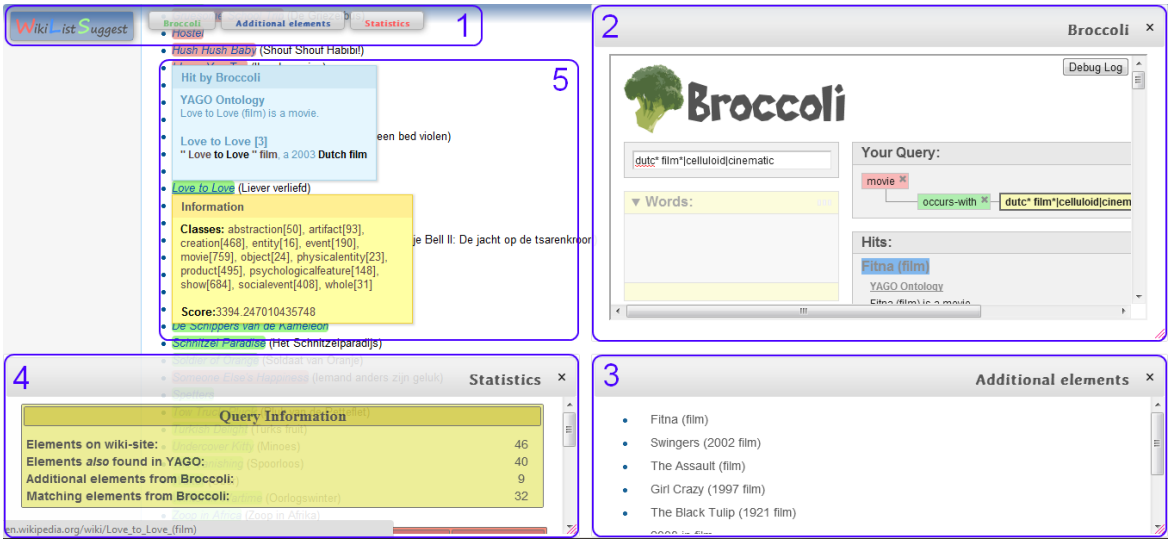

Abbildung 1.2: Das User-Interface von WikiListSuggest

Das User-Interface von WikiListSuggest wird in die Wikipedia-Liste, auf der sich Benutzer gerade befindet, eingebaut. Dabei analysiert das Programm den HTML-Code der Seite und verändert es teilweise mit Hilfe von Javascript. Wie auf der ***Abbildung 1.2*** zu sehen ist, besteht das Interface aus folgenden Teilen:

1. **Menü**

   Das Menü steuert den Programmablauf. Von hier aus startet man die Bearbeitung der Wikipedia-Liste durch Klicken des Start-Knopfes. Nach dem

Start kann man von hier aus die Sichtbarkeit der Boxen durch Betätigen von den entsprechenden Knöpfen steuern. Außerdem lässt sich das Menü auf- und zuklappen durch Klicken auf das Logo.

2. **Broccoli-Box**

   Die Broccoli-Box öffnet sich automatisch nach der Analyse der Wikipedia-Liste und enthält bereits eine Trefferliste für die von WikiListSuggest generierte Anfrage. Von hier aus lässt sich diese Anfrage manuell ändern. Genauer wird das in ***Kap. 2.2*** erläutert.

3. **Vorschlag-Box**

   Hier befinden sich Vorschläge für neue Listenelemente, die evtl. in die Wikipedia-Liste hineingehören. Dabei erscheint beim Darüberfahren mit der Maus über den entsprechenden Vorschlag ein Informationsfenster. In diesem stehen Informationen und Verweise darauf, warum das Element bei Broccoli gefunden wurde. Dies kann dem Nutzer helfen zu verstehen, warum das Element in die Liste passt.

4. **Statistik-Box**

   In diesem Fenster befinden sich einige Statistiken, die durch die Ausführung von WikiListSuggest entstanden sind. Zum Beispiel: Wie viele Listenelemente wurden auf der Seite erkannt und wie viele wurden davon durch Broccoli gefunden.

5. **Listenelemente**

   Durch WikiListSuggest erkannte Listenelemente werden auf der Seite mit Farben hervorgehoben. Mit grün werden diejenigen markiert, die auch durch die Anfrage an Broccoli gefunden wurden und mit rot werden die restlichen Listenelemente markiert. Dabei erscheinen beim Darüberfahren über die grün markierten Elemente zwei Fenster mit Informationen und Verweisen darauf, warum das Element bei Broccoli gefunden wurde und in welchen Ontologie-Kategorien das Listenelement auftaucht. Bei den rot markierten Elementen gibt es nur das Fenster mit den Ontologie-Kategorien.

Der übliche Bedienungsablauf verläuft recht einfach. Man geht auf die entsprechende Wikipedia-Liste, ruft WikiListSuggest auf und klickt auf den Start-Knopf. Dabei erscheint automatisch die Broccoli-Box mit der generierten Anfrage und der daraus resultierenden Liste. Diese Anfrage kann der Benutzer jetzt manuell verändern. Dabei aktualisiert sich die Liste in der Vorschlag-Box automatisch. Für weitere Information lesen Sie [Sch12]

2 Vorhandene Systeme

2.1 YAGO

Bei der Ermittlung der Listenelemente, der Generierung der Anfragen und vielen anderen Schritten braucht WikiListSuggest eine umfangreiche Ontologie. In unserem Projekt verwenden wir YAGO, eine Wissensdatenbank, die am Max-Planck-Institut für Informatik in Saarbrücken entwickelt wird. YAGO enthält Informationen über Millionen von Entitäten in der Form:

[Entität] [Relation] [Kategorie]

Viele Entitäten sind gleichzeitig Kategorien für untergeordnete Entitäten.(Siehe ***Abb. 2.1***).[Inf]

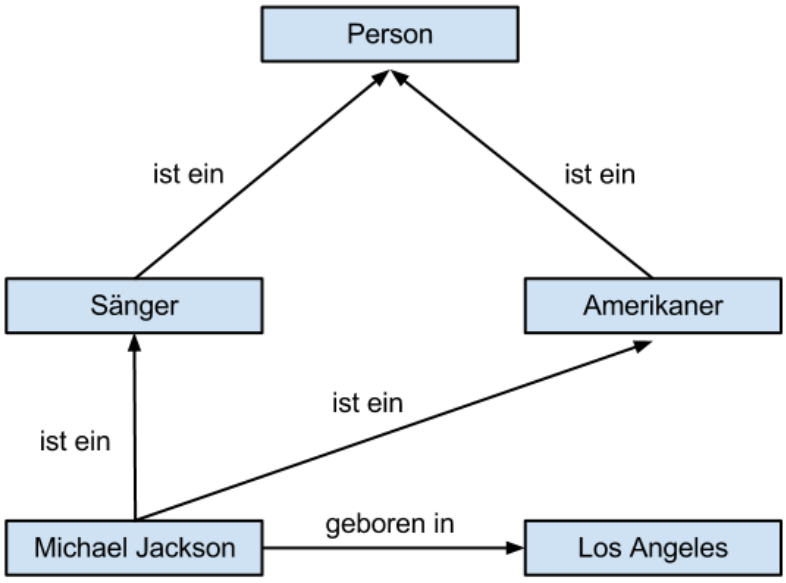

Abbildung 2.1: Beispiel für Beziehungen zwischen Entitäten und Kategorien in YAGO.

2.2 Broccoli

Broccoli ist eine semantische Suchmaschine, die an der technischen Fakultät der Universität Freiburg entwickelt wird[Buc10]. Sie betreibt Volltextsuche in Kombination mit Wissen aus einer Wissensdatenbank. In unserem Programm benutzen wir eine Broccoli-Version, die als Wissensdatenbank YAGO verwendet.

2.2.1 Broccoli-User-Interface

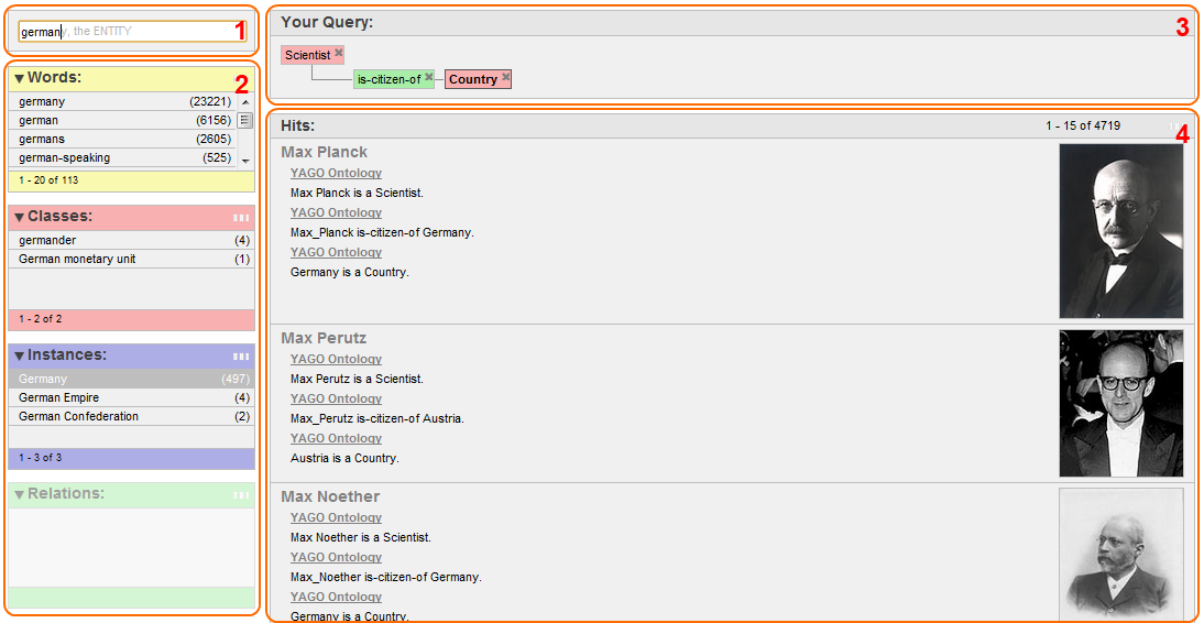

Abbildung 2.2: User-Interface von Broccoli

Nach der automatischen Generierung einer Anfrage schickt WikiListSuggest die Anfrage an Broccoli. Dabei wird ein Fenster mit dem Broccoli-User-Interface[Bäu11] geöffnet, um die generierte Anfrage und ihre Ergebnisliste darzustellen. Leider sind die von WikiListSuggest generierten Anfragen nicht immer optimal, aus diesem Grund hat ein Benutzer die Möglichkeit, die Anfragen manuell über das Broccoli-User-Interface zu verbessern. Aus diesem Grund werden wir die Bedienung von Broccoli kurz vorstellen.

Aufbau

Wie auf der ***Abb. 2.2*** zu sehen ist, besteht das User-Interface aus folgenden Teilen:

1. **Eingabe-Feld**

   Hier gibt der Benutzer seine Anfrage ein.

2. **Vorschlagkästchen**

   Während der Benutzer eine Anfrage in das Eingabefeld eintippt, werden ständig Vorschlagkästchen aktualisiert, die dann Vorschläge von unterschiedlichen Typen enthalten. So enthält das erste gelbe Kästchen Wörter, die der Benutzer gemeint haben könnte. Das rote Kästchen enthält Ontologie-Kategorien. Im Violetten befinden sich Ontologie-Entitäten und im grünen Kästchen sind mögliche Ontologie-Relationen aufgelistet.

3. **Anfrage-Editor**

   Hier wird die aktuelle Anfrage dargestellt. Es existieren auch ein Paar Werkzeuge, um die Anfrage vom Editor aus zu ändern. So kann man zum Beispiel manche Teile der Anfrage löschen, in dem man sie einfach weg klickt.

4. **Treffer-Liste**

   Die Treffer-Liste, wie der Name schon verrät, enthält eine Liste der gefundenen Artikel. Sie wird nach jeder Änderung der Anfrage aktualisiert.

2.2.2 Beispiel einer Ontologie-Anfrage

Das Wissen aus YAGO erlaubt es Broccoli, unterschiedliche Sucharten zu verwenden. Eine davon ist die reine *Ontologie-Anfrage*. Bei dieser Anfrage werden für das Suchen Entitäten, Kategorien und Beziehungen zwischen ihnen in der Ontologie betrachtet. Um die gesuchten Begriffe zu finden, wählt der Benutzer aus den Vorschlagkästchen einfach die nötigen Kategorien, Relation und Entitäten. Im Folgenden zeigen wir an Hand eines Beispieles, wie das funktioniert.

Um die Liste der deutschen Wissenschaftler zu erhalten, gibt man als erstes „Scientist“ in das Eingabefeld ein, danach kann man aus dem roten Kästchen die Kategorie „Scientist“ wählen. Nun erhalten wir die Liste von allen Wissenschaftlern. Um davon die Deutschen zu erhalten, wählen wir Relation „is-citizen-of“ aus und geben schließlich in das Eingabefeld „Germany“ ein. Von Broccoli bekommen wir alle Entitäten im violetten Kästchen vorgeschlagen, die durch Relation „is-citizen-of“ mit Entitäten aus der Kategorie „Scientist“ in Ontologie auftauchen und deren Namen in Ontologie „germany“ enthalten sind. Nun wählen wir einfach die Entität „Germany“ aus und erhalten die gesuchten Artikel.

2.2.3 Beispiel einer kombinierten Anfrage

Durch das Anwenden der Relationen „occurs-with“ kann man Ontologie-Suche mit Volltextsuche kombinieren. Dabei wird die Volltextsuche auf die Menge der Entitäten angewendet, die zuvor durch eine Ontologie-Anfrage ermittelt wurde. Zum Beispiel wollen wir Wissenschaftler finden, die politisch aktiv waren. Um das zu tun, wählen

wir nach der Wahl der Kategorie „Scientist“, die Relation „occurs-with“. Nun haben wir die Möglichkeit, den Suchtext „politically active“ einzugeben. Somit würde Broccoli nach allen Entitäten der Kategorie „Scientist“ suchen, die in einem Satz zusammen mit dem Suchtext „politically active“ vorkommen.

Anfragen dieser Art nennen wir *kombinierte Anfragen.*

Suchtext

Der *Suchtex*t soll aus den Wörtern bestehen, die die gesuchten Begriffe so gut wie möglich beschreiben. Um bei einer Suche durch den Suchtext erfasst zu werden, muss ein Begriff zusammen mit diesem Suchtext innerhalb ein- und desselben Satzes vorkommen. Dabei werden die Wörter im Suchtext miteinander mit Leerzeichen oder vertikalen Strichen verbunden. Leerzeichen zwischen zwei Wörtern bedeuten, dass nach Vorkommnissen des ersten UND zweiten Wortes zusammen mit dem Begriff in einem Satz gesucht wird. Vertikale Striche zwischen zwei Wörtern bedeuten, dass nach Vorkommnissen des ersten ODER zweiten Wortes zusammen mit dem Begriff in einem Satz gesucht wird (siehe das Besipel in ***Abb. 2.3***). Durch das Anhängen eines Sterns („*“) an das Wort aktiviert man die Präfix-Suche für das entsprechende Wort.

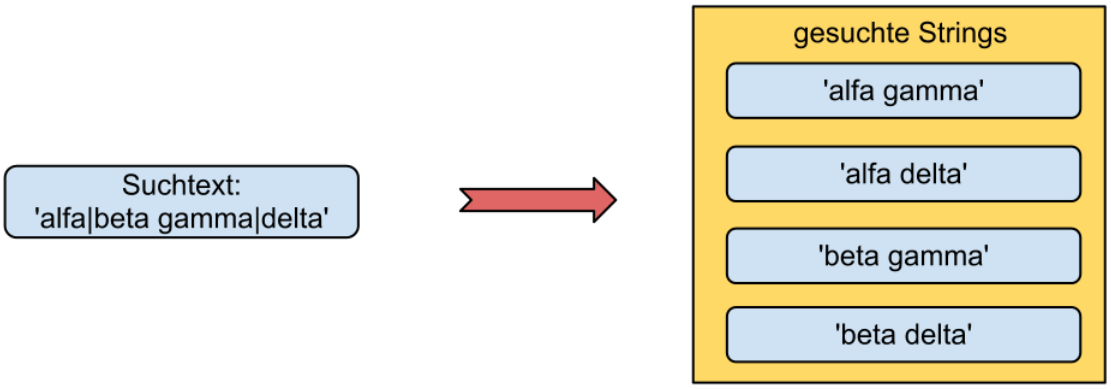

Abbildung 2.3: Der Suchtext (links auf dem Bild) führt dazu, dass nach Vorkommnissen der Strings (rechts auf dem Bild) in ein- und demselben Satz gesucht wird.

2.3 Illinois POS-Tagger

Der Titel einer Wikipedia-Liste ist eine der wichtigsten Informationsquellen für WikiListSuggest. Ein Teil dieser Informationen sind die Wortarten der Wörter, die sich im Titel befinden. Anhand dieser Informationen kann WikiListSuggest für sich brauchbare Wörter (z.B Nomen) von unbrauchbaren Wörtern (z.B. Präpositionen) unterscheiden.

Die Wortarten der Titelwörter erhält WikiListSuggest durch den Illinois POS-Tagger.[CCG]

2.4 WordNet-Wortschatz

WikiListSuggest entnimmt dem Titel einer Wikipedia-Liste die *Eigenschaftswörter*, die die Listenelemente gut beschreiben könnten. Um die Listenelemente noch besser beschreiben zu können, benutzt WikiListSuggest auch die Synonyme dieser Eigenschaftswörter.

Die Synonyme bekommt WikiListSuggest aus dem WordNet-Wortschatz.[uni]

3 Anfragegenerierung

3.1 Problembeschreibung

Wenn wir eine Wikipedia-Liste vervollständigen wollen, brauchen wir eine Liste von weiteren Begriffen, die in der Wikipedia-Liste fehlen, die aber thematisch in sie hinein gehören. Diese Liste versucht WikiListSuggest von Broccoli zu erhalten. Dafür generiert das Programm automatisch eine Anfrage, die eine Liste mit möglichst guter Übereinstimmung zur entsprechenden Wikipedia-Liste liefern soll. In ***Kap. 2.2*** haben wir einige Beispiele für Anfragen an Broccoli gesehen. Für einen menschlichen Benutzer ist es recht unkompliziert, derart einfache Anfragen zu generieren. Man muss nur drei Dinge wissen:

1. Die Art der Begriffe, die man sucht (die Kategorie der gesuchten Begriffe, z.B. „Scientist“).

2. Die Besonderheit der Begriffe (Objekte oder Eigenschaften, die die gesuchten Begriffe beschreiben, z.B. „Germany“ für die Ontologie-Suche oder „politically active“ für die Volltextsuche).

3. Die Verbindung zwischen den gesuchten Begriffen und ihren Besonderheiten (z.B. die Relation. „is-citizen-of“ oder die Relation „occurs-with“ für die Volltextsuche).

Für WikiListSuggest dagegen ist es jedoch eine sehr anspruchsvolle Aufgabe die oben genannten Informationen herauszufinden.

Die optimale Lösung dieser Aufgabe bedarf eines sehr hohen Aufwandes. Um das zu verstehen, betrachten wir die Wikipedia-Liste „List of drug-related deaths“ auf der ***Abb. 3.1***. Durch einen kurzen Blick auf die Liste würde ein Mensch sofort erkennen, dass sie als Listenelemente Namen von Menschen enthält, die durch Drogenkonsum ums Leben kamen. Für eine Maschine dagegen ist es nicht mal klar, ob es sich dabei um Menschen oder Dinge handelt. Aus dem Namen der Liste kann man das schon mal nicht erfahren. Selbst wenn die Kategorie automatisch richtig erkannt wird, ist nicht klar, welche Relation und welche Eigenschaften gewählt werden sollten, damit nur infolge des Drogenkonsums verstorbene Menschen als Ergebnis von Broccoli gefunden werden.

List of drug-related deaths

Name	Life	Profession	Cause of death
Herb Abrams	1954–1996	Professional wrestling promoter	Cocaine related heart attack.
Janet Achurch	1864–1916	Actor	Morphine overdose.
Brian Adams	1963–2007	Professional wrestler	Painkiller overdose. Accidental.
Nick Adams	1931–1968	Actor	Barbituate overdose. Death officially certified as "accidental-suicidal and undetermined". High levels of sedatives were found in his blood.
Stuart Adamson	1958–2001	Musician (Big Country)	Hanged himself under the influence of alcohol.
Michael Adkisson, aka Mike Von Erich	1964–1987	Professional wrestler	Tranquilizer overdose. Suicide.
George Albert, aka George V	1865–1936	King	Cocaine and morphine overdose. Euthanasia.
Asa A. Allen	1911–1970	Evangelist	Liver failure due to acute alcoholism.
Dennis Allen	1951–1987	Drug dealer	Heart failure. "Pieces of his heart actually broke off after decades of heavy drug abuse".
Ryūnosuke Akutagawa	1892–1927	Writer	Veronal overdose. Suicide.
William Barnsley Allen	1892–1933	Army officer	Unspecified narcotic overdose. Accidental.

Abbildung 3.1: Liste von Menschen, die an Drogenkonsum gestorben sind

Dennoch ist es möglich, in vielen Fällen relativ gute Anfragen automatisch zu generieren. Dafür beschränkt sich WikiListSuggest auf kombinierte Anfragen einer bestimmten Form und generiert für jede Wikipedia-Liste gleich mehrere Anfragen. Aus diesen ermittelt es dann die beste Anfrage durch das Vergleichen der Ergebnislisten mit der Wikipedia-Liste.

Man kann die Bearbeitungsschritte der Anfragegenerierung bei WikiListSuggest in drei Hauptschritte einteilen: Informationsextraktion, Generierung der Menge von Anfragen, Auswertung der Anfragen. Bevor wir aber diese Schritte genau einzeln untersuchen, schauen wir uns die Form der Anfragen an, die WikiListSuggest an Broccoli stellt.

3.2 Anfrageform

Reine Ontologie-Anfragen zu generieren, die mindestens eine Relation (außer „occurs-with“) enthalten, scheint schwer und vor allem nicht so sinnvoll zu sein. Die Anzahl der Wikipedia-Listen, für die es möglich wäre, eine entsprechende Ontologie-Anfrage zu generieren, ist begrenzt. Dies liegt daran, dass die Anzahl der Relationen und Kategorien begrenzt ist. So existiert zum Beispiel für die Wikipedia-Liste „List of drug-related deaths“ weder eine Kategorie in der Ontologie, die alle an Drogenkonsum gestorbene Menschen enthalten würde, noch existiert eine Relation wie z.B. „died-from“, die man dann als Beziehung zwischen „Person“ und „Drug“ für die Generierung einer Anfrage benutzen könnte.

Um alle möglichen Listen finden zu können, muss man kombinierte Anfragen verwenden, da man sie flexibler gestalten kann. So kann man zum Beispiel nach Begriffen suchen, die in einem Satz zusammen mit dem Suchtext „died from drugs“ vorkommen und das Ergebnis auf die Kategorie „Person“ begrenzen. Dadurch gefundene

Begriffe werden eventuell nicht alle zu der Wikipedia-Liste „List of drug-related deaths“ passen. Dafür kann man aber mit Anfragen dieser Art für jede Liste eine Anfrage erstellen.

WikiListSuggest generiert kombinierte Anfragen einer bestimmten Form:

[**Ontologie-Kategorie**] [**Relation: „occurs with“**] [**Suchtext**]

- [**Ontologie-Kategorie**]: Durch die Anfrage gefundene Begriffe werden auf Begriffe dieser Kategorie beschränkt.
- [**Relation: „occurs with“**]: Die Relation „occurs-with“ lässt Broccoli Volltextsuche anwenden. Dabei wird nach Vorkommnissen der Begriffe aus der [Ontologie-Kategorie] zusammen mit dem [Suchtext] in einem Satz gesucht.
- [**Suchtext**]: Wörter, die zusammen mit Begriffen aus der [Ontologie-Kategorie] in einem Satz auftauchen sollen.

3.3 Informationsextraktion

Im vorherigen Kapitel haben wir die Form der Anfragen kennengelernt, die WikiListSuggest generiert. Dadurch, dass die Form der Anfrage festgelegt ist, bleiben noch zwei Unbekannte, die WikiListSuggest herausfinden muss, um eine Anfrage zu generieren: Die Ontologie-Kategorie der Wikipedia-Liste und der Suchtext mit den Wörtern, die die Listenelemente beschreiben.

Im ersten Bearbeitungsschritt der Anfragegenerierung beschäftigt sich WikiListSuggest damit, möglichst viele Informationen für diese Unbekannten zu sammeln. Dabei verwendet WikiListSuggest zwei Quellen: Den Titel der Wikipedia-Liste und die Listenelemente, die von WikiListSuggest bereits ermittelt wurden. Im Titel steht der Name der Wikipedia-Liste. Die Listenelemente dienen WikiListSuggest als Beispiele für Begriffe, die er finden muss. An sich geben aber der Titel und die Listenelemente nicht viele Informationen her. Sie müssen von WikiListSuggest noch bearbeitet werden, um darin enthaltene Informationen zu extrahieren und zu erweitern. Dafür verwendet es folgende Werkzeuge:

- YAGO-Ontologie

  Mit Hilfe der YAGO-Ontologie erstellt WikiListSuggest für jedes Listenelement eine Menge von Kategorien, in denen dieses Element vorkommt.

  Beispiel: *„Birkenpilz“* $\rightarrow$ *{„organismus“, „Pilz“, „Entität“, . . . }*

- Illinois POS-Tagger

  Illinois POS-Tagger erlaubt WikiListSuggest zu erkennen, welche Wortart die Wörter im Titel haben.

  Beispiel: „Liste von essbaren Pilzen“:

„Liste“ → *„SingularNomen“*

„von“ → *„Präposition“*

„essbaren“ → *„Adjektive“*

„Pilzen“ → *„Plural Nomen“*

- WordNet-Wortschatz

  WikiListSuggest hat die Möglichkeit, mit Hilfe des WordNet-Wortschatzes für jedes Wort eine Menge seiner Synonyme zu generieren.

  Beispiel: *„Sieger“* → *{„Gewinner“, „Champion“, „Erster“ . . . }*

Diese Informationen nutzt WikiListSuggest, um die Mengen der möglichen Kandidaten für die Ontologie-Kategorie der Wikipedia-Liste und die möglichen Kandidaten der Eigenschaftswörter, die die Listenelemente beschreiben, zu ermitteln.

3.3.1 Erstellung der Kategorie-Menge.

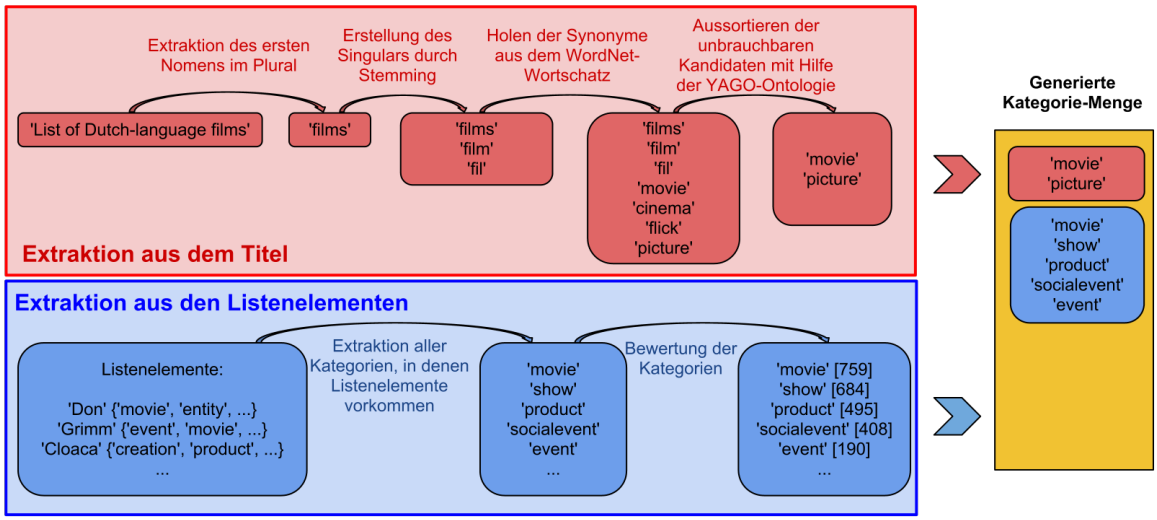

Abbildung 3.2: Prozess der Generierung eine Kategorie-Menge am Beispiel der Wikipedia-Liste „List of Dutch-language films“

Beide Quellen, sowohl der Titel, als auch die Listenelemente, enthalten Hinweise auf die Ontologie-Kategorie der Wikipedia-Liste. In den meisten Fällen steht im Titel, um welche Begriffe es in dieser Liste geht. Zum Beispiel ist im Titel der Wikipedia-Liste „List of Dutch-language films“ das Wort „films“ zu finden , das uns darauf schließen lässt, dass es um Filme geht. Auch mit Hilfe der Listenelemente kann man an wichtige Informationen herankommen, in dem WikiListSuggest die Ontologie nach den Kategorien fragt, in welche die Listenelemente gehören. Eine davon sollte dann die gesuchte Kategorie aller Listenelemente sein.

Man kann aber nicht mit hundertprozentiger Sicherheit die richtige Kategorie feststellen. Aus diesem Grund generiert WikiListSuggest eine Menge von Kategorien. Diese werden generiert aus

- den aus dem Titel gewonnenen Informationen
- den Kategorien, die aus den Listenelementen extrahiert wurden

3.3.1.1 Kandidaten von dem Titel

Viele Titel von Wikipedia-Listen fangen mit den Wörtern „List of ...“ an. Basierend darauf haben wir uns überlegt, dass das erste Nomen im Plural, das nach diesen Worten folgt, ein Kandidat für die Ontologie-Kategorie sein kann. Zum Beispiel: „List of Dutch-language **films**“ oder „List of healthcare reform advocacy **groups** in the United States“. WikiListSuggest findet dieses Wort mit Hilfe des Illinois POS-Taggers.

Nun ist das Problem, dass unser Kandidat im Plural steht und die Namen der Kategorien in der YAGO-Ontologie normalerweise im Singular angegeben werden. Also imitiert WikiListSuggest einfaches Stemming (ermitteln des Stammes eines Wortes), in dem es eine bzw. zwei der letzten Buchstaben löscht. Dabei kommen zwei weitere Kandidaten heraus. Zum Beispiel bekommen wir aus „films“ „film“ und „fil“. Die Überlegung dahinter ist, dass der Plural im Englischen meistens durch Anhängen der Endungen „-s“ bzw. „-es“ gebildet wird.

Es kann sein, dass der Name der gesuchten Kategorie in der Ontologie nicht einem von WikiListSuggest erstellten Kandidaten entspricht. aber einem Synonym eines von WikiListSuggest erstellten Kandidaten. Zum Beispiel kann die Kategorie in der Ontologie den Namen „movie“ statt „film“ tragen. Aus diesem Grund erstellt WikiListSuggest mit Hilfe des WordNet-Wortschatzes Synonyme der schon erstellten Kandidaten und fügt sie in die Kategorie-Menge hinzu.

Jetzt besitzt WikiListSuggest mehrere Kandidaten, wobei manche von ihnen komplett unbrauchbar sind, da sie als Namen in der YAGO-Ontologie nicht vorkommen. Um diese Kandidaten herauszufiltern, wird bei jedem Kandidat von WikiListSuggest überprüft, ob er als Name einer Kategorie in der Ontologie vorkommt.

3.3.1.2 Kandidaten von den Listenelementen

An dieser Stelle ist WikiListSuggest mit der Bearbeitung des Titels fertig und versucht weitere Kandidaten durch die Analyse der Listenelemente zu bekommen.

Dafür erstellt WikiListSuggest mit Hilfe der YAGO-Ontologie für jedes Listenelement eine Menge der Kategorien, in denen es vorkommt. Dann vereinigt WikiListSuggest alle diese Mengen von Kategorien und bekommt dadurch eine *Vereinigungsmenge* von allen Kategorien, in denen Listenelemente dieser Wikipedia-Liste vor-

kommen. Nun weist WikiListSuggest jeder Kategorie dieser Menge einen Wert zu. Dabei wird folgende Formel verwendet:

$$Wert[x] = \frac{\text{Anzahl der Listenelemente in Kategorie[x]}}{\text{Größe der Kategorie[x]}}$$

falls Anzahl der Listenelemente in Kategorie[x] >= 1/3 der Größe von Wikipedia-Liste

$$Wert[x] = 0$$

falls Anzahl der Listenelemente in Kategorie [x] < 1/3 der Anzahl der Listenelemente

X: eine der Kategorien der Vereinigungsmenge

Anzahl der Listenelemente in Kategorie: Anzahl der Listenelemente einer Wikipedia-Liste, die in Kategorie x vorkommen

Größe der Kategorie: Gesamtanzahl der Begriffe, die in der YAGO-Ontologie in Kategorie x vorkommen

Die Kategorien mit den höchsten Werten werden als Kandidaten der Kategorie-Menge hinzugefügt. Die Überlegungen hinter der Formel sind folgende: eine Kategorie, die viele Listenelemente einer Wikipedia-Liste enthält und dabei aber nur wenige Begriffe innerhalb der gesamten Ontologie enthält, kann mit hoher Wahrscheinlichkeit die gesuchte Ontologie-Kategorie der Wikipedia-Liste sein. Eine Kategorie hingegen, die weniger als 1/3 der Listenelemente enthält, kann schlecht die gesuchte Ontologie-Kategorie sein.

Zum Beispiel kommen viele Listenelemente der Wikipedia-Liste „List of Dutch-language films“ in Kategorien wie „movie“ und „entity“ vor. Dabei enthält die Kategorie „movie“ in der YAGO-Ontologie relativ wenige Begriffe, „entity“ enthält dagegen jeden Begriff der Ontologie. Die Kategorie „comedy“ ist in der Ontologie noch seltener als „movie“, besitzt aber weniger als 1/3 der Listenelemente.

3.3.2 Erstellung der Eigenschaftsmenge

Aus den Kandidaten, die in der Eigenschaftsmenge enthalten sind, werden Suchtexte für die Anfragen an Broccoli generiert. Aus diesem Grund sollten die Wörter dieser Menge Listenelemente einer Wikipedia-Liste so gut wie möglich charakterisieren. Ihr Auftauchen in Wikipedia zusammen mit einem Begriff soll darauf hinweisen, dass dieser Begriff in die Wikipedia-Liste hinein gehört. Zum Beispiel werden durch einen

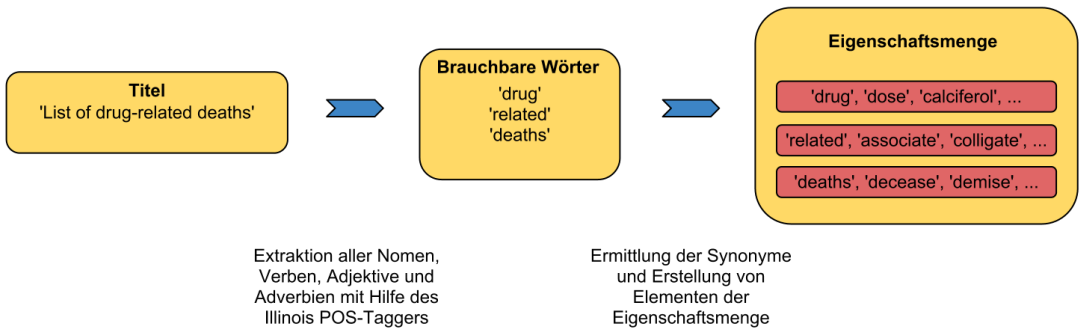

Abbildung 3.3: Prozess der Generierung einer Eigenschaftsmenge am Beispiel der Wikipedia-Liste „List of drug-related deaths“

Suchtext, der die Wörter „died“ und „overdosis“ enthält, tatsächlich viele Artikel (Beschreibungen) von Menschen gefunden, die an einer Überdosis gestorben sind.

WikiListSuggest verwendet den Titel einer Wikipedia-Liste um auf Eigenschafts-Wörter der Listenelemente zu kommen. Als erstes nimmt WikiListSuggest alle brauchbaren Wörter aus dem Titel, in dem es die Wortarten der Titelwörter mit Hilfe vom Illinois POS-Tagger betrachtet. Brauchbare Wörter sind dann alle Nomen außer „List“ sowie Verben, Adjektive und Adverbien. Der Rest, wie Präpositionen oder Artikel werden als unbrauchbar betrachtet. Die Überlegung dahinter ist, dass Artikel oder Präpositionen wie „a“, „an“ oder „of“ keine Hinweise auf passende Begriffe sein können. Ebenfalls das Nomen „List“, da es in den Namen der meisten Wikipedia-Listen vorkommt.

Um die Anzahl der Eigenschaftswörter zu erweitern, erstellt WikiListSuggest mit Hilfe des WordNet-Wortschatzes die Synonyme von jedem Wort, das es dem Titel entnommen hat. Nun erstellt WikiListSuggest die Eigenschaftsmenge auf folgende Weise. Jedes Wort, das dem Titel entnommen wurde, wird mit seinen Synonymen von WikiListSuggest zu einer Menge vereinigt. Diese Menge wird dann als Menge in die Eigenschaftsmenge hinzugefügt. Die Eigenschaftsmenge enthält dann als Elemente wiederum Mengen der Form:

{Titel-Wort, sein 1. Synonym, sein 2. Synonym, ... }

So ein Aufbau der Eigenschaftsmenge als Menge von Wörtern und ihren Synonymen bringt Vorteile bei der späteren Generierung der Anfragen.

3.4 Generierung einer Menge von Anfragen

WikiListSuggest besitzt nun alle nötigen Informationen um eine Anfrage an Broccoli zu generieren. Es kennt mögliche Kategorien, die die Liste erfassen sowie möglicherweise die Liste beschreibende Eigenschaftswörter. Da WikiListSuggest aber nicht

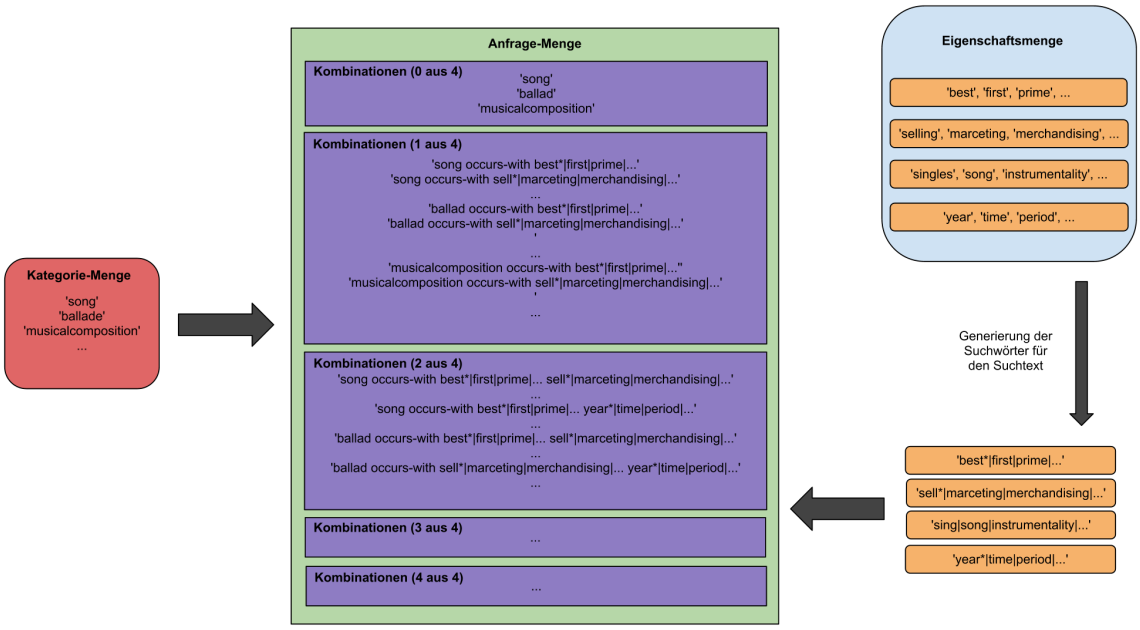

Abbildung 3.4: Prozess der Generierung einer Anfrage-Menge am Beispiel der Wikipedia-Liste „List of best-selling singles by year“

exakt weiß, welche Kategorie und welche Eigenschaftswörter am besten für die Anfrage geeignet sind, generiert es durch das Kombinieren gleich mehrere Anfragen und fasst sie zu einer *Anfrage-Menge* zusammen.

Für unsere Form der Anfrage gibt es zwei Unbekannte, die WikiListSuggest auswählen soll: die Kategorie und den Suchtext.

Die Bestimmung der Kategorie ist einfach, da wir jede Kategorie aus der Kategorie-Menge nacheinander einsetzen können.

Mit dem Suchtext ist es etwas komplizierter. Die erste Möglichkeit wäre, alle Wörter, die wir dem Titel entnommen haben, zu nehmen und sie durch Leerzeichen zu verbinden. Dies würde bedeuten, dass alle Wörter zusammen auftauchen müssen. Zum Beispiel wäre für die Wikipedia-Liste „List of best-selling singles by year“ der generierte Suchtext „best selling singles year“. Somit würden die Lieder gefunden, die in einem Satz zusammen mit den Wörtern „ best“, „selling“, „singles“ und „year“ vorkommen.

Dabei gibt es ein Paar Probleme: Die Verwendung von manchen Wörtern aus dem Titel kann dazu führen, dass beim Suchen einige richtige Begriffe nicht gefunden werden. So macht zum Beispiel die Verwendung von „year“ in dem Suchtext „best selling singles year“ wenig Sinn, da es dem Benutzer einfach sagen soll, wie die Liste sortiert ist. Aus diesem Grund ist es sinnvoll, mehrere Suchtexte mit unterschiedlichen Kombinationen der Titelwörter auszuprobieren.

Ein weiteres Problem könnte sein, dass bei den Sätzen von den gesuchten Begriffen

nicht die Titelwörter vorkommen, sondern ihre Synonyme oder dieselben Wörter nur in anderer Form. Zum Beispiel kann in einem Satz nicht „singles“, sondern „single“ oder „song“ vorkommen. Das Problem kann man zum Teil lösen, in dem man für jedes Titelwort im Suchtext durch das Ersetzen der letzten Buchstaben des Wortes mit einem Stern („*“) die Präfix-Suche aktiviert. Außerdem kann jedes dieser Präfix-Titelwörter durch vertikale Striche mit seinen Synonymen verbunden werden. So reicht ein Auftreten von mindestens einem dieser Wörter im Text zusammen mit einem möglichen Treffer, um diesen in das Ergebnis aufzunehmen.

So zum Beispiel wird „singles“ im Suchtext durch „sing*|song|solo“ ersetzt. Nun reicht der Text „'I will always love you' is a **song** by Whitney Houston“ oder „... Whitney Houston was **sing**ing 'I will always love you'“, um das Lied „I will always love you“ in der Ergebnisliste auftauchen zu lassen.

Um zu sehen, wie WikiListSuggest diese Lösungsansätze im Detail umsetzt, betrachten wir den Ablauf der Generierung.

3.4.1 Ablauf

Für jede Kategorie *„X“* der Kategorie-Menge erstellt WikiListSuggest mehrere Suchtexte auf folgende Weise:

1. Für jedes Element der Eigenschaftsmenge (Menge {Titelwort, sein 1. Synonym, sein 2. Synonym, ...}) erstellt WikiListSuggest einen String, in dem man das Titel-Wort und seine Synonyme durch vertikale Striche verbindet und dabei beim Titelwort die letzten Buchstaben (höchstens 3, abhängig von der Länge des Wortes) durch „*“ ersetzt. Der entstandene String ist der Suchtext.
   Beispiel: ***Menge**: {„singles“, „song“, „solo“}*
   → ***Suchtext**: „sing*|song|solo“*
2. Für jeden von je zwei Elementen der Eigenschaftsmenge erstellt WikiListSuggest einen String wie eben beschrieben. Diese verbindet es mit Leerzeichen zu einem Suchtext.
   Beispiel: ***Menge 1** : {„singles“, „song“, „solo“} und **Menge 2**: {„selling“, „commerce“, „marketing“}*
   → ***Suchtext:** „sing*|song|solo sell*|commerce|marketing“*
3. Dasselbe wie bei 2) nur für drei Elemente, dann für vier usw., bis die Suchtexte für alle Kombinationen erstellt wurden.
4. Mit allen Suchtexten und Kategorie *„X“* erstellt WikiListSuggest Anfragen und fügt sie der Anfrage-Menge hinzu.

3.4.2 Anzahl der Anfragen

Jede der erstellten Anfragen wird an Broccoli geschickt. Dabei kann es bei manchen besonders komplizierten Anfragen (mit vielen Wörtern im Suchtext) auch länger

als 1 Sekunde dauern, bis man eine Antwort darauf bekommt. Aus diesem Grund darf die Anzahl der Anfragen nicht beliebig groß sein. Sei nun M die Größe der Kategorie-Menge und N die Anzahl der Elemente in der Eigenschaftsmenge.

So lässt sich die Anzahl der erstellten Anfragen mit folgender Formel errechnen:

$$
\begin{aligned}
\textbf{\textit{Anzahl der Anfragen}} \quad = \quad & M * \quad [\quad (0 \text{ aus } N) \\
& \qquad + \quad (1 \text{ aus } N) \\
& \qquad + \quad (2 \text{ aus } N) \\
& \qquad + \quad \ldots \\
& \qquad + \quad (N\text{-}1 \text{ aus } N) \\
& \qquad + \quad (N \text{ aus } N) \\
& \quad] \\
= \quad & M * 2^N
\end{aligned}
$$

Dabei entsprechen *(1 aus N)*, *(2 aus N)* usw. den Mengen der Kombinationen, wenn man ein Element aus N Elementen der Eigenschaftsmenge wählt, zwei Elemente aus N Elementen der Eigenschaftsmenge wählt usw. Die Anzahl der Kombinationen, die in diesen Mengen jeweils enthalten sind, entspricht dem Binomialkoeffizient *(N k)*, wo *k = 1..N.*

So ist zum Beispiel für die Wikipedia-Liste „List of healthcare reform advocacy groups in the United States“ *N = 6* und die Anzahl der Anfragen gleich *M * 64*. Dabei hängt M davon ab, wie groß die erstellte Menge der möglichen Kategorien ist. Schon für 10 Kategorien haben wir 640 Anfragen, was nicht mehr in akzeptabler Zeit bearbeitet werden kann.

Um die Anzahl der Anfragen so klein wie möglich zu halten, haben wir folgende Optimierungen implementiert: Es ergab sich durch Ausprobieren, dass es in den meisten Fällen ausreicht, wenn man nur die 5 am besten bewerteten Kategorien der Listenelemente in die Kategorie-Menge rein nimmt. Dazu kommen noch maximal drei dem Titel entnommene Kategorien, also insgesamt höchstens 8 Kategorien. Außerdem erstellen wir Kombinationen nur für jeweils *0, 1, 2, N – 2, N – 1* und N Elemente der Eigenschaftsmenge. Die Idee dahinter ist, dass die Kombinations-Mengen zwischen *(2 aus N)* und *(N-1 aus N)* besonders viele Kombinationen enthalten und es in vielen Fällen ausreicht, wenn wir nur die Kombinationen aus den restlichen Mengen betrachten.

Neue Formel:

$$
\begin{aligned}
\textbf{\textit{Anzahl der Anfragen}} \quad = \quad & Min(8,\ M) * \quad [\quad (0 \text{ aus } N) \\
& \qquad + \quad (1 \text{ aus } N) \\
& \qquad + \quad (2 \text{ aus } N) \\
& \qquad + \quad (N\text{-}2 \text{ aus } N) \\
& \qquad + \quad (N\text{-}1 \text{ aus } N) \\
& \qquad + \quad (N \text{ aus } N) \\
& \quad] \\
= \quad & Min(8,\ M) \quad * \quad (2 + 2 * N + N * (N\text{-}1))
\end{aligned}
$$

3.5 Bewertung der Anfragen

WikiListSuggest hat nun viele mögliche Anfragen generiert. Jetzt müssen wir sie alle auf irgendeine Art und Weise bewerten können, um die sinnvollste darunter zu finden. Dafür brauchen wir die Listen der Begriffe, die durch diese Anfragen gefunden werden. Diese Listen können wir von Broccoli erhalten und dann schauen, inwiefern sie mit der Wikipedia-Liste übereinstimmen.

Im Folgenden schauen wir genauer, wie WikiListSuggest die Bewertung einer Anfrage aus der Anfrage-Menge durchführt. Als erstes schickt WikiListSuggest diese Anfrage an Broccoli. Anschließend bekommt es eine Liste von gefundenen Begriffen zurück, nennen wir sie die *Broccoli-Liste*. Basierend auf dieser Liste ermittelt WikiListSuggest den Qualitätswert der Anfrage. Dieser Wert soll zwei Ziele widerspiegeln:

- Die Anzahl der gefundenen Elemente der Wikipedia-Liste muss so groß wie möglich sein.
- Die Differenz zwischen der Anzahl der von Broccoli gefundenen Elemente und der Anzahl der Listenelemente der Wikipedia-Liste muss so klein wie möglich sein.

Die Idee dahinter ist: Je besser wir diese zwei Ziele umsetzen, desto genauer entspricht die Broccoli-Liste der Wikipedia-Liste. Damit erhöht sich auch die Wahrscheinlichkeit, dass die Elemente der Broccoli-Liste, die nicht in der Wikipedia-Liste vorkommen, weitere Listenelemente sein könnten.

$$\textbf{QualitätWert} = hitsNumber - \frac{bSize - wSize}{difFactor}$$

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

bSize: Größe der Broccoli-Liste.

wSize: Größe der Wikipedia-Liste.

difFactor: Durch den *difFactor* kann man einstellen, wie stark der Einfluss der Größendifferenzen zwischen den beiden Listen auf den *QualitätWert* sein soll (aktuell durch die Ergebnisse aus systematischen Experimenten auf 13 gesetzt). Je größer der *difFactor* ist, desto bedeutender ist die Anzahl der von Broccoli gefundenen Listenelemente bei der Bewertung der Anfrage und desto unbedeutender ist die Größendifferenz zwischen der Broccoli- und Wikipedia-Liste.

Nachdem alle Anfragen auf diese Weise bewertet wurden, wählt WikiListSuggest die mit dem größten Qualitätswert aus und schickt sie an das User-Interface.

4 Zukünftige Verbesserung

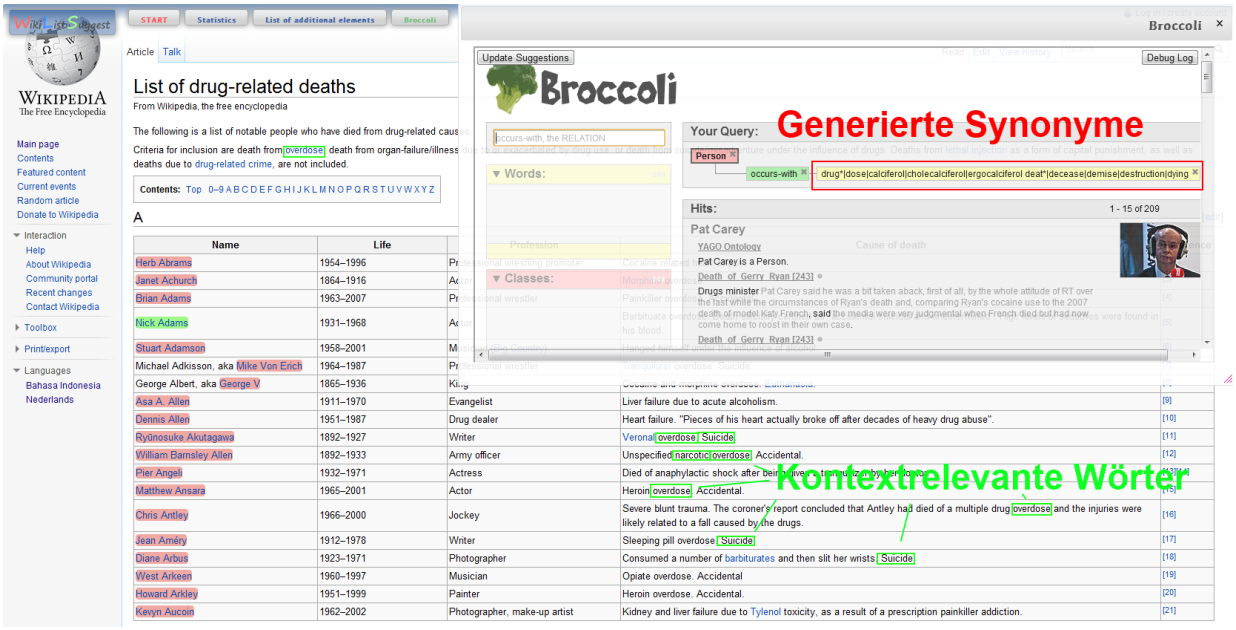

Abbildung 4.1: Durch Synonyme wurden zu wenige Listenelemente gefunden. Es wäre sinnvoller, stattdessen kontextrelevante Wörter zu verwenden.

Im ***Kap. 3.3.2„Erstellung der Eigenschaftsmenge“*** haben wir darüber gesprochen, wie man die Anzahl der gefundenen Begriffe vergrößern kann, in dem man Synonyme der Titelwörter erstellt und sie in den Suchtext einsetzt. In der Praxis ist es leider so, dass dieses Vorgehen nicht nur Vorteile, sondern auch Nachteile mit sich bringen kann.

Betrachten wir zum Beispiel die Wikipedia-Liste „List of drug-related deaths“ ***Abb. 4.1*** Für das Titelwort „death“ liefert der WordNet-Wortschatz folgende Synonyme: “decease“, „demise“, „destruction“, “dying“, „end“, „expiry“, „last“. Das Verwenden von allen diesen Synonymen im Suchtext führt dazu, dass durch die gestellte Anfrage falsche Begriffe gefunden werden. So landete „Mario Villanueva“ durch den folgenden Satz in der Broccoli-Liste: „Accused of **drug** trafficking at the **end** of his gubernatorial period“.

Das Problem, das dahinter steckt, ist, dass zwei Wörter, die in einem Kontext die gleiche Bedeutung haben, in einem anderen Kontext verschiedene Bedeutungen haben können. So wären zum Beispiel „death“ und „destruction“ gute Synonyme für

eine Wikipedia-Liste von im Kampf gestorbenen Terminatoren. Im Kontext der „List of drug-related deaths“-Liste dagegen machen sie wenig Sinn.

Aus diesem Grund scheint es sinnvoll zu sein nach anderen Quellen für die Eigenschaftswörter zu suchen. Wenn man die Wikipedia-Liste „List of drug-related deaths“ genauer betrachtet, erkennt man, dass es auf der Seite dieser Liste viele Wörter wie „suicide“, „overdose“ oder „narcotic“ gibt, die sehr gut zum Thema der Liste passen. Die Frage ist aber, wie man von allen Wörtern, die es auf der Seite gibt, genau diese *kontextrelevanten Wörter* findet.

Eine der Möglichkeiten wäre es, die Häufigkeiten der Vorkommnisse von allen Wörtern der Wikipedia-Liste auf der Seite und in der ganzen Wikipedia zu betrachten. Die Überlegung dahinter ist die, dass kontextrelevante Wörter auf der Seite oft und in der ganzen Wikipedia eher selten vorkommen werden. Zum Beispiel erscheint das Wort „overdose“, das auf der Seite sehr oft vorkommt, in der ganzen Wikipedia bestimmt seltener als etwa „big“ oder „different“. Diese kontextrelevanten Wörter können sehr gute Suchwörter sein.

5 Evaluation

In diesem Kapitel untersuchen wir, welche Ergebnisse WikiListSuggest erzielt. Dabei interessieren wir uns vor allem dafür, wie gut WikiListSuggest die Listenelemente einer Wikipedia-Liste erkennt und wie gut die Anfragen sind, die WikiListSuggest generiert. Basierend darauf schauen wir dann, ob es uns tatsächlich gelingt Wikipedia-Listen mit Hilfe von WikiListSuggest zu vervollständigen.

Außerdem werden wir die Performanz von WikiListSuggest untersuchen, um zu sehen, beim Lösen welcher Aufgaben am meisten Zeit verbraucht wird.

5.1 Qualität

Um über die Qualität vom WikiListSuggest eine Aussage treffen zu können, werden wir als erstes eine Test-Menge von Wikipedia-Listen definieren. Dann testen wir WikiListSuggest auf dieser Menge und werten die entstehenden Ergebnisse aus. Dadurch wollen wir in erster Linie Antworten auf folgende Fragen finden:

- Wie viele Listenelemente wurden auf der Seite erkannt?
- Wie gut ist die von WikiListSuggest generierte Anfrage und daraus folgend wie gut sind die dadurch entstehenden Vorschlagslisten zum Vervollständigen der Wikipedia-Listen?

5.1.1 Testdaten

Als Quelle für unsere Test-Listen haben wir eine Wikipedia-Seite[1] genommen, wo alle möglichen Wikipedia-Listen in unterschiedlichen Kategorien zusammen gefasst wurden. Jeder Kategorie haben wir durch folgendes Vorgehen eine Liste entnommen: Wir öffnen die Kategorie und wenn keine Liste vorhanden ist, dann öffnen wir die darin enthaltene, als erstes aufgelistete Unterkategorie und so weiter und so weiter, bis wir zum ersten Mal auf eine Wikipedia-Liste treffen („unten in Kategorie-Hierarchie ankommen“). Diese fügen wir dann unserer Testmenge hinzu. Dieses Verfahren soll uns die Zufälligkeit der gewählten Listen garantieren.

Da die von uns gewählte Seite 34 Kategorien enthält, bestehen unsere Testdaten aus 34 Wikipedia-Listen aus unterschiedlichen Themenbereichen (siehe ***Tab. 5.1***).

[1] http://en.wikipedia.org/wiki/Category:Lists

5.1.2 Evaluation der Listenerkennung

Zum Testen der Listenerkennung haben wir für jede Wikipedia-Liste aus unserem Testdatenbestand WikiListSuggest gestartet. Nach dem die Listenelemente erkannt sind, werden sie auf der Seite markiert. Durch manuelles Überprüfen der erkannten Listenelemente (einfach durch Anschauen, ob die richtigen Elemente markiert wurden) haben wir folgendes herausbekommen:

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

Tabelle 5.1: Zum Testen gewählte Wikipedia-Listen. Die Bedeutungen der Farben werden im folgenden beschrieben.

[grün]

Bei **16** von **34** (**47%**) der Wikipedia-Listen wurden die Listenelemente komplett richtig erkannt.

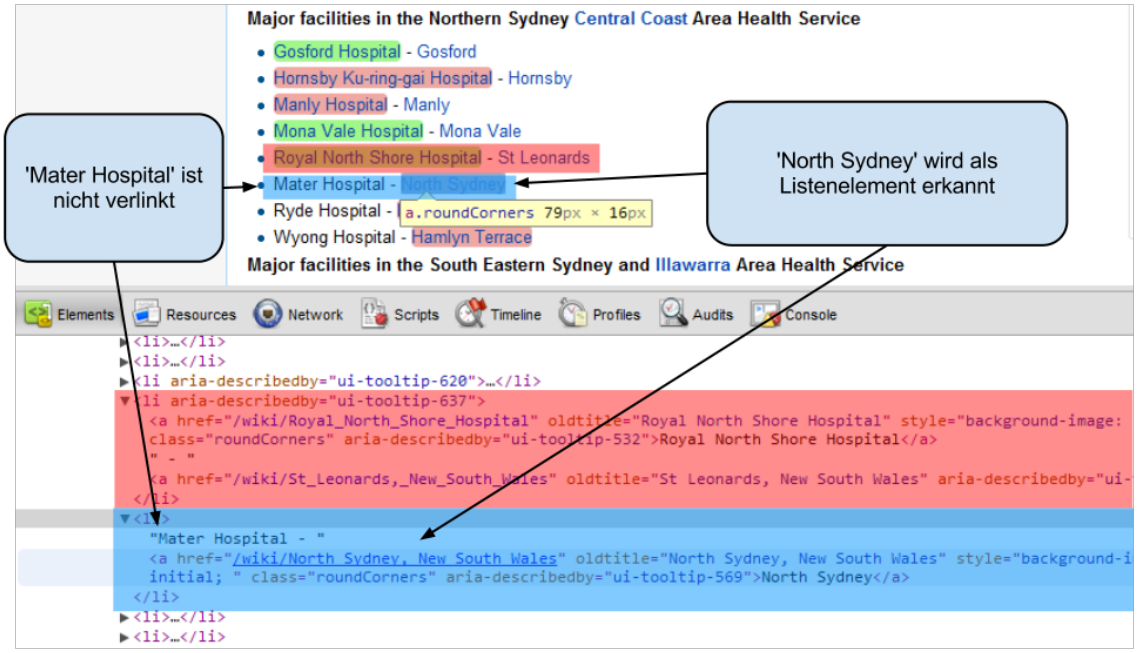

Abbildung 5.1: Auszug aus der Wikipedia-Liste „List of hospitals in Australia“. Unten auf dem Bild ist die HTML-Struktur der oben markierten Zeilen abgebildet. Daran kann man erkennen, dass „Mater Hospital“ nicht verlinkt ist. Aus diesem Grund wird „North Sydney“ fälschlicherweise als Listenelement erkannt.

[gelb]

Bei **5** von **34** (**15%**) der Wikipedia-Listen wurden zwar alle Listenelemente richtig erkannt, zusätzlich wurden aber noch andere wenige Elemente auf der Seiten von WikiListSuggest als Listenelemente gedeutet.

Die Hauptfunktionalität von WikiListSuggest, das Vorschlagen von weiteren Elementen für die Wikipedia-Liste, wird dadurch nicht negativ beeinflusst. Damit diese Funktionalität gewährleistet wird, müssen alle Listenelemente erkannt werden und die Anzahl der zusätzlich falsch erkannten Listenelemente gering gehalten werden.

Der Grund für die zusätzlichen Elemente ist bei allen diesen Artikeln gleich: In ***Kap. 1.2.1*** haben wir darüber gesprochen, dass WikiListSuggest bei der Listenerkennung die HTML-Struktur der Wikipedia-Listen untersucht und für alle Verweise die letzten Elemente ihres DOM-Pfades anschaut und sie danach

gruppiert. Nun ist das Problem bei den oben genannten Listen, dass manche von den Listenelementen nicht verlinkt sind. Dadurch haben ihre rechten Nachbarn manchmal den gleichen DOM-Pfad wie die tatsächlichen Listenelemente. So sehen wir auf der ***Abb. 5.1*** eine der Test-Listen „List of hospitals in Australia“. Jede HTML-Listen-Zeile (*<li>-Tag*) enthält den Namen eines Hospitals (1. *<a>-Tag*) und den Ort, in dem es sich befindet (2.*<a>-Tag*). Wenn der Name des Hospitals nicht verlinkt ist (das Hospital keine Referenz hat), dann erkennt WikiListSuggest die Ortschaft (wenn sie verlinkt ist) als Listenelement, da sie nun der 1. <a>-*Tag* in der Zeile ist.

[rosa]

Bei **3** von **34** **(9 %)** der Wikipedia-Listen wurden die Listenelemente zum größten Teil erkannt.

Die Tatsache, dass nicht alle Listenelemente erkannt wurden, beeinträchtigt die Funktionalität von WikiListSuggest, da man bei den Elementen aus der Vorschlags-Liste nicht mehr sicher sein kann, ob sie tatsächlich in die Wikipedia-Liste hineinpassen oder ob sie von WikiListSuggest einfach übersehen wurden.

Der Grund für die nicht erkannten Listenelemente ist in der HTML-Struktur zu suchen. Hier haben wir das Problem, dass manche Listenelemente als linken Nachbarn ein anderes Element besitzen, das verlinkt ist. Da dieser Nachbar nur bei wenigen Listenelementen existiert und dadurch den gleichen DOM-Pfad hat wie die restlichen Listenelemente, wird er von WikiListSuggest fälschlicherweise für ein Listenelement gehalten.

Ein weiteres Problem besteht darin, dass manche Listenelemente selbst als Nachbarn von den anderen Listenelementen vorkommen. Dies ist jedoch nicht immer der Fall und dadurch nicht einfach zu erkennen. (Siehe ***Abb. 5.2***)

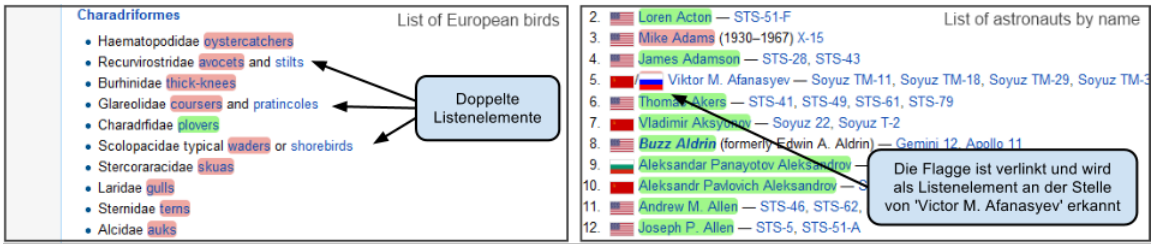

Abbildung 5.2: Auf dem Bild werden Auszüge von zwei Wikipedia-Listen abgebildet, bei denen auf Grund der HTML-Struktur Listenelemente nicht erkannt werden.

[rot]

Bei **4** von **34** **(12%)** der Wikipedia-Listen wurden die Listenelemente nicht erkannt.

Der Grund dafür ist, dass WikiListSuggest nicht richtig bestimmen konnte, worum es sich in der Liste handelt. Das kann vorkommen, wenn es zu wenige Listenelemente im Vergleich zu den restlichen Elementen der Seite gibt bzw. zu wenige davon mit Wikipedia-Artikeln verlinkt sind und wenn die Ontologie-Kategorie der Liste aus dem Titel falsch ermittelt wurde. (siehe ***Abbildung 5.3***)

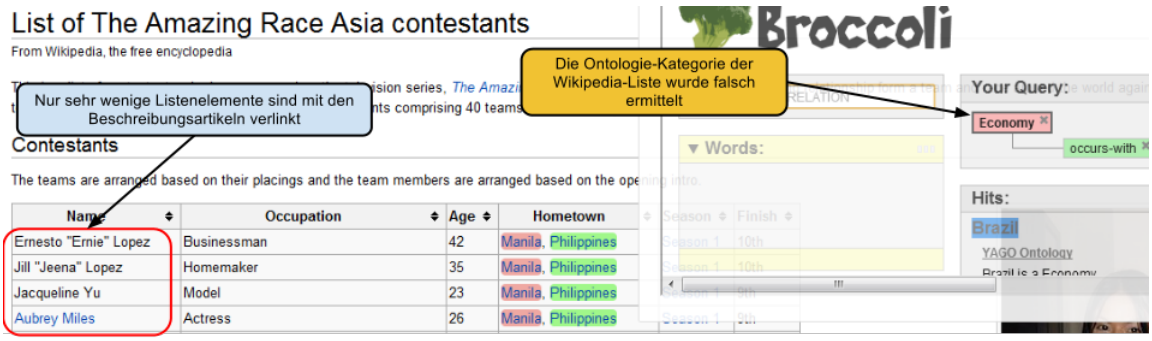

Abbildung 5.3: Die Wikipedia-Liste „List of The Amazing Asia contest“, deren Listenelemente durch WikiListSuggest nicht erkannt wurden.

[grau]

Bei **6** von **34 (17%)** der Wikipedia-Listen war die Erkennung der Listenelemente nicht möglich.

Die Gründe dafür liegen darin, dass es keine Listenelemente mit Links auf Wikipedia-Artikel gab und sie somit sie von WikiListSuggest nicht erfasst werden konnten (Wikipedia-Listen: WL29-WL33). Dies kann auch passieren, wenn die Listenelemente mit Artikeln verlinkt sind, die nicht auf Wikipedia liegen (z.B. WL34: Die Listenelemente sind hier mit Artikeln von Wiktionary verlinkt).

Insgesamt sind in der Testmenge 21 von 34 (62%) Listen, deren Listenelemente von WikiListSuggest richtig erkannt wurden. Wir zählen auch die gelb markierten Listen dazu, da bei ihnen die Funktionalität von WikiListSuggest nicht beeinträchtigt wird. Es scheint auch sinnvoll zu sein, die grau markierten Listen aus der Wertung herauszunehmen, da sie die Qualität der Listenerkennung nicht widerspiegeln können. Somit erreichen wir eine Erkennungsquote von 75%.

5.1.3 Evaluation der Anfrage-Generierung

Um die Funktionalität der Anfragegenerierung objektiv beurteilen zu können, brauchen wir Wikipedia-Listen, in denen alle Listenelemente (bzw. die verlinkten Listenelemente) durch WikiListSuggest erkannt wurden. Aus diesem Grund können wir

die grauen, roten und rosa Listen nicht mehr verwenden. Somit schrumpft unsere Test-Menge auf 21 Listen.

Die Qualität einer Anfrage spiegelt sich im Übereinstimmungsgrad der durch sie generierten Broccoli-Liste und der Wikipedia-Liste wider. Der Übereinstimmungsgrad lässt sich durch folgende Zahlen ausdrücken:

- **hitsNumber** Anzahl der Listenelemente, die auch in der Broccoli-Liste vorkommen (Je größer, desto besser)
- **wSize** Größe der Wikipedia-Liste.
- **bSize** Größe der Broccoli-Liste (Je näher zu wSize, desto besser).

In der ***Tab. 5.2*** sehen wir diese Zahlen zusammen mit

- **Query** Von WikiListSuggest generierte Anfrage
- **NLE** Neue passende Listenelemente, die durch diese Anfrage ermittelt werden konnten. Dabei wurden die ersten 10 von WikiListSuggest vorgeschlagenen Elemente untersucht (siehe ***Kap. 5.1.3.2***)

Num	Query	hitsNum	wSize	bSize	NLE
WL1	Singer occurs-with curr*|flow|stream arti*	1	28	318	0
WL2	Bank occurs-with bank*|camber|cant alba*	6	13	12	0
WL3	Creator occurs-with aerosp* engine*|direct|mastermind	11	131	43	8
WL4	Movie occurs-with base*|alkali|bag comi*|amusing|comedian	80	224	358	7
WL5	Squadron occurs-with roya*|imperial|majestic austral*|aussie	83	85	125	0
WL6	Writer occurs-with fema*|distaff philosoph*	2	99	16	4
WL7	Militant occurs-with athe*|atheistic|atheistical	35	72	272	8
WL8	football_player occurs-with alba* internatio*|external|outside	5	13	36	1
WL9	Congressman occurs-with afri*	72	133	113	0
WL10	Standard occurs-with en|nut|east standa*|banner|criterion	14	31	36	2
WL11	Organism occurs-with invas*|encroaching|incursive spec*|coinage|mintage	32	100	422	0
WL12	Saint occurs-with 12th cent*|c|hundred	25	165	158	2
WL13	Drive occurs-with disn*|orcus	126	267	204	1
WL14	Case occurs-with clas*|assort|category acti*|accomplish|activeness	2	25	7	2
WL15	Configuration	83	89	277	0
WL16	Album occurs-with best|better|outdo sell*|marketing|merchandising	36	76	603	0
WL17	Award occurs-with john|bathroom|can abra*|ibrahim	6	15	16	0
WL18	Site occurs-with archaeologi*|archaeologic|archeologic	477	1439	2314	7
WL19	Fiction occurs-with altern*|alternating|alternative hist*|account|chronicle	127	338	272	5
WL20	Hospita occurs-with hospit*|infirmary austra*	56	165	96	7
WL21	Accident	294	1130	1543	9

Tabelle 5.2: Testergebnisse der Anfragegenerierung

5.1.3.1 Auswertung des Übereinstimmungsgrades der Mengen

Im Schnitt wurden bei jeder Liste 36% aller Listenelemente erkannt. Dabei ist die Größe der Broccoli-Liste im Schnitt um 5% größer als die Größe der entsprechenden Wikipedia-Liste.

Die Wahl der Ontologie-Kategorien sieht bei allen Anfragen vernünftig aus. Das bei manchen Listen zu wenige Listenelemente erkannt wurden liegt am Suchtext. Schauen wir ein Paar Beispiele an:

- **WL1 „List of current A&M Records artists“**

  Der Hauptgrund für die schlechte Anfrage liegt im Titel dieser Liste oder um genauer zu sein in der Zeichenkombination „A&M“. Der Illinois POS-Tagger konnte die Wortart von „A&M“ nicht bestimmen (was vermutlich am „&“ liegt) und somit wurde dieses Wort nicht in die Eigenschaftsmenge aufgenommen. Die restlichen Wörter konnten dann die Menge der Listenelemente nicht gut genug beschreiben. Das hat zu Anfragen geführt, durch welche Broccoli-Listen entstanden sind, die kaum mit der Wikipedia-Liste übereinstimmen.

- **WL14 „List of class action lawsuits“**

  Hier ist das Problem eher ein allgemeines Problem, dass bei manchen Listen mehr und bei anderen weniger offensichtlich ist. Die Suchtexte, die aus den Titelwörtern generiert wurden, können die Listenelemente entweder zu allgemein (was zu viel zu großen Broccoli-Listen führt) oder zu speziell (was zu viel zu wenigen erkannten Listenelementen führt) beschreiben. Bei dieser Anfrage trat der zweite Fall auf. Außerdem waren die Synonyme, die mit Hilfe von dem WordNet-Wortschatz generiert wurden, für den Kontext der Wikipedia-Liste nicht sinnvoll und führten zu keinen weiteren Listenelementen.

  Ein weiteres Problem ist: in der Ontologie-Kategorie „Case“, die für die Anfrage gewählt wurde, kamen nur vier Listenelemente der Wikipedia-Liste vor. Alle anderen Kategorien enthielten entweder noch weniger Listenelemente oder waren zu groß, wie z.B. „Entity“ (enthält jeden Begriff in der YAGO-Ontologie), so dass die generierten Suchtexte zu den zu großen Broccoli-Listen führten.

5.1.3.2 Auswertung der vorgeschlagenen Elemente

Wir haben für jede Wikipedia-Liste die ersten 10 von WikiListSuggest vorgeschlagenen Elemente angeschaut. Dabei haben wir die Begriffe gesucht, die tatsächlich in die entsprechende Wikipedia-Liste passen könnten.

Insgesamt konnten wir 63 neue Listenelemente finden. Im Schnitt sind 3 von 10 geprüften Elementen neue Listenelemente. Das ergibt eine Trefferquote von 30%.

Durch das genaue Betrachten der Test-Wikipedia-Listen kann man erkennen, dass die meisten Listen (5 von 8), für die keine neuen Elemente gefunden wurden, ent-

weder vollständig oder nur schwer erweiterbar sind, wobei die Grenze zwischen den Definitionen von vollständig und schwer erweiterbar eher verschwommen ist.

- Mit **vollständigen Listen** sind diejenigen gemeint, in denen die Anzahl der möglichen Begriffe sehr überschaubar ist und alle existierenden Begriffe schon vorhanden sind. Zum Beispiel WL15 „List of constellations“, wo alle 88 von der Internationalen Astronomischen Union (IAU) verbindlich festgelegten Sternbilder aufgeführt sind. So welche kleinen, klar definierten Listen sind meistens bereits vollständig.

- Mit **schwer erweiterbaren Listen** sind die Listen gemeint, die zur Zeit vollständig sind und deren Anzahl von Listenelementen durch bestimmte Bedingungen begrenzt ist und dadurch überschaubar ist. Zum Beispiel WL16 „List of best-selling albums“. Auf der Seite der Liste steht, dass nur die Alben berücksichtigt werden, die über 20 Millionen Mal verkauft wurden. Das macht die Liste überschaubar klein und entsprechend auch die Wahrscheinlichkeit klein, dass irgendein Album, das mindestens so oft verkauft wurde, übersehen wurde. Außerdem kommt es auch sehr selten vor, dass ein Album mehr als 20 Millionen Mal verkauft wird und für längere Zeit nicht in die Liste eingefügt wird.

Wenn wir diese Listen aus der Wertung nehmen, dann bekommen wir eine Trefferquote von 39 %.

5.1.3.3 Manuelle Verbesserung der Anfragen.

Für alle oben genannten Wikipedia-Listen (WL1-WL21) haben wir durch manuelle Verbesserung der generierten Anfragen versucht, die Anzahl der neuen Listenelemente zu vergrößern. Dabei haben wir uns bemüht, den Fall zu imitieren, dass ein Benutzer nur anhand seines allgemeines Wissens und der vorliegenden Wikipedia-Liste eine Anfrage zu verbessern versucht. Wir haben z.B. die gefundenen Ontologie-Kategorien nicht detailliert untersucht oder lange mögliche uns bekannte Relationen in Betracht bezogen. Aus diesem Grund ist es durchaus möglich, dass man für Broccoli bessere Anfragen generieren kann. Die auf Wikipedia selbst vorhandene Information haben wir allerdings benutzt.

Als Beispiel betrachten wir WL1 „List of current A&M Records artists“ Auf der Seite von der Wikipedia-Liste haben wir entdeckt, dass „A&M Records“ früher „Octone Records“ hieß. Darauf basierend haben wir die Anfrage „singer occurs-with octon*|A&M* Record*“ zusammengestellt, die uns immerhin ein neues Listenelement gebracht hat.

Durch dieses Vorgehen konnten wir 9 weitere neue Elemente finden.

5.2 Performance

Durch das Performance-Test wollen wir in erster Linie herausfinden, mit welchen Aufgaben WikiListSuggest die meiste Zeit verbringt und welcher Zusammenhang zwischen den Eigenschaften der geprüfter Wikipedia-Liste (z.B: Anzahl der Listenelementen und Größe des Titels) und der Laufzeit von WikiListSuggest besteht.

Als Test-Daten haben wir die gleichen Wikipedia-Listen genommen, die wir bei Evaluation der Anfragegenerierung verwendet haben (21 Listen). Dabei haben wir für jede Test-Liste die Laufzeit der Listenerkennung und der Anfragegenerierung getestet.

Num	Listenerkennung	Anfrage-Generierung
WL1	1.77	2.64
WL2	0.08	0.32
WL3	0.08	0.69
WL4	0.25	6.62
WL5	0.04	5.41
WL6	0.12	0.89
WL7	1.01	1.13
WL8	0.01	4.34
WL9	0.18	18.83
WL10	0.02	1.13
WL11	0.08	1.82
WL12	0.67	6.52
WL13	0.12	5.11
WL14	0.08	2.83
WL15	0.19	0.09
WL16	0.25	2.77
WL17	0.01	6.20
WL18	9.64	7.74
WL19	0.26	1.91
WL20	0.22	0.39
WL21	8.55	6.16
Schnitt	**1.13**	**3.98**

Tabelle 5.3: Testergebnisse der Performance

5.2.1 Listenerkennung

Die Bearbeitungszeiten der Listenerkennung variieren von 0.01 bis 9.64 Sekunden. Die Laufzeiten stehen im direkten Zusammenhang mit der Anzahl der Listenelemente, die erkannt werden müssen (wSize).

Es gibt viele Stellen bei Listenerkennung, die optimiert werden können. Zum Beispiel ist die Identifizierung der möglichen Listenelemente am Beginn der Erkennung sehr ineffizient: Zunächst werden in mehreren Suchläufen für die verschiedenen Typen von möglichen Kandidaten (Element in Tabellenzelle, Element in HTML-Liste) alle Kandidaten gesucht. Nachdem alle möglichen Kandidaten gesammelt wurden, werden Elemente gesammelt, die keine Kandidaten sein können: Zum Beispiel Elemente des Inhaltsverzeichnisses oder des Verweise-Abschnitts des Wikipedia-Artikels. Jede dieser Listen von nicht möglichen Kandidaten wird einzeln mit den am Anfang gefundenen Kandidaten verglichen, um unmögliche Kandidaten herauszufiltern.

Dies ließe sich z.B. durch entsprechende Änderungen beschleunigen, indem schon bei der ersten Identifizierung der Kandidaten gleich geprüft wird, ob sie überhaupt Listenelemente sein können. Ob allerdings diese oder eine andere Stelle im Ablauf der Listenerkennung die größte Zeit benötigt, müsste man noch genauer untersuchen.

5.2.2 Anfragegenerierung

Die Bearbeitungszeiten der Anfragegenerierung variieren ebenfalls stark und reichen von 0.9 bis 18.83 Sekunden. Dabei entfällt der größte Anteil der Laufzeit auf das Testen der Anfragen, da das Generieren der Anfragen nur Millisekunden dauert. Die Anfragen werden zum Testen an Broccoli geschickt, dabei spielen für die Laufzeit zwei Aspekte eine Rolle:

- **Anzahl der Anfragen.** Diese hängt von den Größen der Eigenschaftsmenge und der Kategorie-Menge ab.
- **Komplexität der Anfragen.** Die Bearbeitungszeiten einer Anfrage durch Broccoli sind nicht immer gleich und hängen von der Komplexität dieser Anfrage ab. Für die Komplexität einer Anfrage sind unter Anderem die Anzahl der Suchwörter und die Größe der Kategorie von Bedeutung.

6 Zusammenfassung und Ausblick

In dieser Arbeit wurde WikiListSuggest vorgestellt, ein Programm, das Vorschläge zur Vervollständigung Wikipedia-Listen generiert.

Dabei wurden folgende Schritte erläutert:

- Für die Informationsextraktion werden der Titel und die Listenelemente untersucht.
- Aus der gewonnenen Information werden die Eigenschaftsmenge und die Kategorie-Menge erstellt. Dabei besteht die Kategorie-Menge aus Kandidaten der Ontologie-Kategorien, die die Listenelemente am besten definieren sollen und die Eigenschaftsmenge aus den Wörtern, die die Listenelemente am besten beschreiben sollen.
- Durch Kombinationen der Elemente der Eigenschaftsmenge und der Kategorie-Menge werden mehrere Anfragen generiert.
- Alle generierten Anfragen werden an Broccoli geschickt, woraufhin Broccoli die Treffer-Listen zurückschickt.
- Alle Treffer-Listen werden auf die Übereinstimmung mit der Wikipedia-Liste getestet. Die Anfrage, die zur der Liste mit der besten Übereinstimmung geführt hat, wird an das User-Interface übergeben.

Die Evaluation von WikiListSuggest hat vielversprechende Ergebnisse hervorgebracht:

- Bei 21 von 28 Listen, bei denen die Erkennung möglich war, hat WikiListSuggest alle Listenelemente erkannt. Das ist eine Erkennungsquote von 75%.
- Bei 16 unvollständigen Listen wurden 160 Vorschläge für neue Listenelemente untersucht. Dabei konnten wir 63 neue Listenelemente entdecken, die in diesen Wikipedia-Listen fehlten. Das ist eine Trefferquote von etwa 39%.

Trotz der guten erzielten Ergebnisse ist es möglich WikiListSuggest weiter zu verbessern, indem man zum Beispiel bessere Eigenschaftswörter für die Generierung der Anfragen findet. Zur Zeit verwenden wir für die Anfragegenerierung neben den Wörtern aus dem Titel noch die Synonyme, die wir von WordNet-Wordschatz für diese Titelwörter erhalten. Diese scheinen nicht immer sinnvoll zu sein. Aus diesem Grund könnte man statt Synonyme z.B. die Wörter verwenden, die auf der Seite von der Wikipedia-Liste besonders oft vorkommen. Genauer wurde dies in ***Kapitel 4*** erläutert.

Danksagung

Ich möchte mich bei einigen Personen bedanken, die mich bei dieser Arbeit unterstützt haben.

Zunächst möchte ich mich bei Robin Schirrmeister bedanken, mit dem ich an diesem Projekt gearbeitet habe und der ebenfalls darüber seine Bachelorarbeit geschrieben hat. Man könnte sich keinen anderen Teamkameraden wünschen.

Großer Dank gilt meiner Betreuerin und Gutachterin Prof. Dr. Hannah Bast, für ihre ständige Hilfsbereitschaft und ihr großes Verständnis.

Ich möchte auch Björn Buchold danken für seine Einführung in die geheimnisvolle Welt von Broccoli.

Außerdem vielen Dank auch an Vadim Landhäußer für seine Rechtschreibkorrekturen, was sicher kein einfaches Anliegen war.

Literaturverzeichnis

[Bäu11] Bäurle, Florian. *A User Interface for Semantic Full Text Search.* 2011

[Buc10] Buchhold, Björn. *SUSI: Wikipedia Search Using Semantic Index Annotations.* 2010

[CCG] Cognitive Computation Group, University of Illinois at Urbana-Champaign. *Illinois Part of Speech Tagger.* `http://cogcomp.cs.illinois.edu/page/software_view/3`

[Inf] mpi für Informatik. *YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia.* `http://www.mpi-inf.mpg.de/yago-naga/yago/`

[Sch12] Schirrmeister, R. T. *Automatische Vorschläge zur Vervollständigung von Wikipedia Listen - Listenerkennung, Benutzeroberfläche.* 2012

[uni] university, Princeton. *WordNet: A lexical database for English.* `http://wordnet.princeton.edu/`