Bachelorarbeit

Extraktionsverfahren zur Bestimmung von Eigenschaften einer Produktklasse im Rahmen von „Product Review Mining“

David Terhart (geb. Best)
david-terhart@gmx.de

Betreuer:
Prof. Dr. Heinz Schweppe
Dipl.-Inf. Jürgen Bross

AG Datenbanken und Informationssysteme
Institut für Informatik
Freie Universität Berlin

10. Juni 2008

Zusammenfassung

Diese Arbeit behandelt die Feature-Extraktion, einen Teilschritt im Feature-basierten Product Review Mining. Feature-Extraktion, zu einer Klasse von Produkten Attribute, d. h. Teile, Eigenschaften oder andere Merkmale der Produkte dieser Klasse, aus einem Korpus an Quelldokumenten zu extrahieren. Wir vergleichen Methoden zur Feature-Extraktion und stellen ein neues Verfahren vor, das auf bestehenden Methoden basiert und diese weiterentwickelt. Dieses Verfahren wurde prototypisch als Feature-Extraktionssystem implementiert und anschließend evaluiert.

Danksagung

Ich danke Jürgen Broß sehr für die hervorragende Betreuung dieser Arbeit. Die häufigen Treffen und der rege Austausch haben mir sehr geholfen. Robert und Benjamin Best sowie Felix Reckers danke ich für die Unterstützung bei der Gewinnung von Testdaten zur Evaluation. Ich danke meiner Frau Lena für viele Korrekturen und für die liebevolle Unterstützung während der drei Monate, in denen ich mich dieser Arbeit gewidmet habe.

Erklärung zur Urheberschaft

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Literatur bzw. Hilfsmittel ohne fremde Hilfe angefertigt habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Berlin, den 10. Juni 2008

David Terhart (geb. Best)

Inhaltsverzeichnis

1 Einleitung und Grundlagen

Diese Bachelorarbeit behandelt die sog. *Feature-Extraktion* als Teilaufgabe eines speziellen *Opinion-Mining*-Verfahrens, des *Feature-basierten Product Review Mining*. Mit Feature-Extraktion bezeichnen wir die maschinelle Gewinnung einer Menge von Termini aus einem Quell-Korpus (einer Dokumentensammlung), die als wichtig oder repräsentativ für diesen erachtet werden. Diese Termini tragen im weiteren Verlauf des Feature-basierten Product Review Mining zum Auffinden von Meinungsäußerungen bei und sind außerdem Teil der Ausgabewerte des Prozesses.[1]

Zunächst wird in das Fachgebiet eingeführt und der Zweck des Opinion Mining erläutert. Im folgenden Kapitel widmen wir uns detailliert dem Problembereich der Feature-Extraktion, wobei wir auch verschiedene bestehende Ansätze zur Feature-Extraktion vergleichen. Dabei untersuchen wir die unterschiedlichen Vorgehensweisen ebenso wie die Eignung für das Gebiet der Produktbewertungen, und wenn möglich die Effizienz bzw. Fehlerquoten bei der Extraktion.

Schließlich wird ein eigenes Verfahren zur Feature-Extraktion für Produktbewertungen entwickelt, welches sich auf bestehende Ansätze stützt, etablierte Maße aus dem Gebiet des *Natural Language Processing* verwendet. Dieses wird als Prototyp zur Feature-Extraktion implementiert und evaluiert.

In Kapitel 5 fassen wir die Ergebnisse zusammen und widmen uns einem Ausblick auf die zukünftige Verwendung und Erweiterung dieses Systems, welches Teil eines Projekts zum Product Review Mining sein wird.

1.1 Opinion Mining

Das Opinion Mining, oft auch als *Sentiment Analysis* bezeichnet, ist eine relativ junge Forschungsrichtung in der Informatik, die sich als Aggregat aus Teilbereichen älterer Disziplinen betrachten lässt.[2] Grundlage ist das Bestreben, aus natürlichsprachigen Texten, die Meinungsäußerungen (Opinions, Sentiments) beinhalten, maschinell den Meinungsgehalt zu extrahieren. Dieser kann sich beispielsweise zusammensetzen aus dem Subjekt einer Meinungsäußerung (der Autor), dem Objekt (der Gegenstand) und der Meinung selbst (die Orientierung: positiv/negativ/neutral sowie ihre Stärke). Ziel ist es, große Datenmengen so zusammenfassen zu können, dass ein Überblick über deren Inhalt gegeben ist. Dabei kann die Zusammenfassung unterschiedlich fein sein: Wenn eine einzige Meinungstendenz für ein gesamtes Dokument angegeben wird, ist dies Resultat einer relativ groben Analyse, wogegen die Aufschlüsselung nach einzelnen Aspekten eines betrachteten Gegenstandes feinkörniger ist ([Liu06]: S. 411 f.). Nennenswert ist dabei die Arbeit von [PLV02], die drei verschiedene Methoden zur Klassifizierung ganzer Dokumente vorstellen und dabei gänzlich auf Features verzichten – es handelt sich also um einen Opinion-Mining-Ansatz auf Dokumentebene.

[1] Im Rahmen dieser Arbeit wird davon ausgegangen, dass beim Opinion Mining Meinungsäußerungen mit Autor, Objekt, Tendenz und Stärke der Meinung in zusammengefasster Form ausgegeben werden. Generell sind hier jedoch auch andere Ausgabewerte denkbar. Siehe dazu auch die Einführung zum Opinion Mining von Bing Liu in [Liu06] (Kap. 11).

[2] siehe Kapitel 1.1.1

Auch bei der Extraktion der Meinungen ist der Detailgrad unterschiedlich. So ist die Angabe der Meinungsäußerungen nach einem binären Schema (positiv/negativ bzw. neutral) ein gröberer Ansatz als die Einordnung auf einer Skala, die die Meinung nicht nur klassifiziert, sondern nach Stärke gewichtet, bspw. umgesetzt von [PE05].

1.1.1 Einordnung

Opinion Mining kann man durchaus als Spezialfall der Informationsextraktion bezeichnen. Diese behandelt die Extraktion von Informationen zu einer bestimmten Domäne aus un- oder semistrukturierten Daten. Beispielhaft könnte die Extraktion von Exemplaren einer Klasse oder Domäne auf dem Korpus des World Wide Web genannt werden, wie sie im System KnowItAll[3] umgesetzt ist. Bei solchen Verfahren ist das Ziel allerdings die Extraktion von *Fakten*, wohingegen beim Opinion Mining *Meinungen* extrahiert werden sollen. So gesehen sind die Faktenextraktion und das Opinion Mining verschiedene Anwendungen der Informationsextraktion.

Sowohl die Informationsextraktion als auch das Opinion Mining verwenden etablierte Methoden aus verschiedenen Disziplinen der Informatik. So wird beim Teilbereich der Suche bzw. Datengewinnung auf Erkenntnisse des *Information Retrieval* zurückgegriffen. Bei der Datenverarbeitung werden häufig linguistische Werkzeuge[4] verwendet, die aus dem *Natural Language Processing* bekannt sind, welches der Computerlinguistik bzw. der Künstlichen Intelligenz zugeschrieben werden kann. Im Teilbereich der Datenzusammenführung, bspw. beim Auflösen von Synonymen, werden Methoden der *Datenintegration* angewendet, wo bereits Forschungsergebnisse zu möglichen Problemen bei der Zusammenführung von Daten aus unterschiedlichen Strukturen, Inhalten oder Quellen vorhanden sind.

Je nach Granularität des Mining-Verfahrens werden hier linguistische und statistische Verfahren für die Satzebene oder Text-Klassifikationsalgorithmen für ganze Dokumente eingesetzt ([Liu06]: 11.1.1 und 11.1.2).

1.1.2 Anwendungen

Ein verbreitetes Anwendungsgebiet sind Produktbewertungen im World Wide Web, die von Benutzern verfasst werden und auf Webseiten veröffentlicht sind. Wenn eine maschinelle Auswertung dieser Bewertungen gelingt, kann daraus entweder eine Zusammenfassung für eine Produktklasse mit einer Gewichtung von positiven und negativen Bewertungen erstellt werden, oder sogar eine detaillierte Statistik über den Grad / die Stärke der Meinungen, aufgeschlüsselt nach Eigenschaften des Produktes. Dies wäre z. B. für Menschen nützlich, die sich einen Überblick über Bewertungen zu einem Produkt oder einer Produktklasse verschaffen wollen.

Des Weiteren sind solche Verfahren für den Bereich der *Business Intelligence*[5] interessant: Unternehmen könnten hierüber die Akzeptanz von eigenen Produkten oder Konkurrenzprodukten eruieren. Es ließen sich bspw. Zielgruppen analysieren oder bestimm-

[3] siehe [Bes07b], dort findet sich auch eine detailliertere Beschreibung der Informationsextraktion.

[4] z. B. Part-of-Speech-Tagger zur syntaktischen Annotation von Texten

[5] betriebswirtschaftlicher Begriff: Methoden zur automatischen Analyse von Unternehmensdaten, siehe auch: [Zie06]: S. 106

te Eigenschaften eines Produktes gezielt anpassen, nachdem dazu Meinungen analysiert wurden.

Als weitere Anwendungsgebiete für das Opinion Mining wären bspw. ein sog. „flame detection system“[6] [WWH04] denkbar oder die automatisierte Prüfung von Bewertungen zu Bewerbern auf Stellen in einem Unternehmen („candidate review“, [CNZ05]).

Das kommerzielle Potential des Opinion Mining lässt sich schon an der Vielzahl der Lösungen ablesen, die junge Unternehmen hier anbieten. Beispielhaft wären zu nennen:

Bazaarvoice: *Bazaarvoice offers technology and services for your web site that help you capture and amplify customer conversations.*[7] [Baz08]

PowerReviews: *a fully-managed service for retailers that captures, moderates and displays customer reviews & recommendations* [Pow07]

Jodange: *Jodange provides the ability to isolate people's opinion and sentiment about key topics over time.* [Jod08]

SentiMetrix: *SentiMetrix, Inc. offers an innovative technology framework to measure sentiments or opinions expressed in the electronic media (such as news, blogs, newsgroups) worldwide.* [SM08]

Opinmind: *Opinmind's state-of-the-art classification algorithms utilize natural language processing and machine learning to ensure that your message is delivered to the right person at the right time.* [Opi08]

1.2 Product Review Mining

Das Anwendungsgebiet des Product Review Mining[8] zeigt nicht nur auf unmittelbare Weise den direkten Nutzen des Opinion Mining, sondern ist auch das Objekt diverser Forschungsarbeiten auf diesem Gebiet, wie [HL04], [PE05], [CNZ05]. Dabei lassen sich bei den jeweiligen Verfahren trotz einiger Unterschiede doch auffallende Gemeinsamkeiten festhalten:

1. Ziel ist die Überführung von un- oder semistrukturiertem Text in strukturierte Form,[9] die eine Zusammenfassung der Quellinhalte bieten soll.

2. Eine Vorgabe für den *Opinion-Mining*-Prozess ist die Angabe einer Produktklasse, zu der Reviews analysiert werden sollen.

[6] Erkennungssystem für Beschimpfungen oder andere starke verbale Auseinandersetzungen, einsetzbar z. B. in unmoderierten Webforen

[7] Bazaarvoice operiert nach eigenen Angaben profitabel [Baz07] und hatte laut Pressebericht im Februar 2008 190 Mitarbeiter [Hip08]

[8] s. o.: Maschinelle Analyse von Produktbewertungen. Im Folgenden verwenden wir den englischen Begriff *Review* für Bewertungen im Sinne einer Bewertung eines Autors zu einem bestimmten Produkt; diese Bewertung kann z. B. eine einzelne Aussage sein („Produkt XY ist schlecht“) oder ein größeres Dokument mit einer ausführlichen Rezension.

[9] z. B. eine Datenbank oder XML-Dateien

3. Der initiale Schritt ist das Erlangen eines Korpus von Review-Dokumenten, auf dem die Analyse durchgeführt wird.

4. Ein weiterer initialer Schritt ist häufig das Identifizieren und Analysieren der Objekte der Meinungsäußerungen. Beispiele hierzu finden sich im Kapitel 2.

5. Ein zentraler Punkt ist das Identifizieren und Analysieren der Textstellen, die eine Meinungsäußerung darstellen und darauffolgend die Analyse der Meinungsäußerung; Details hierzu in den folgenen Kapiteln.

Oft sind diese Schritte stark miteinander verzahnt und nicht eindeutig voneinander abgrenzbar: So findet bspw. bei [HL04] und [PE05] das Identifizieren und Analysieren der Objekte in zwei Stufen statt, nämlich vor und nach dem Auffinden der Meinungen.

1.2.1 Feature-basiertes Product Review Mining

Wenn beim Product Review Mining die Meinungsanalyse auf Feature-Ebene geschieht, wird es als *Feature-basiertes Product Review Mining* bezeichnet ([Liu06]: S. 417 ff.). Voraussetzung ist hierfür, dass eine Menge an Features erlangt wird, über welche Meinungen geäußert werden. Daher steht vor der Extraktion von Meinungen das Auffinden von Features (Schritt 4 in der obigen Aufzählung). Dazu gibt es unterschiedliche Ansätze: Bei [CNZ05] werden sog. *benutzerdefinierte Taxonomien* hinzugezogen; das sind dort Listen von Produktattributen, die aus externen Quellen stammen.[10] [HL04] verwenden einen Trainingskorpus, um syntaktische Muster festzulegen, auf deren Grundlage Features extrahiert werden können. Eine weitere Möglichkeit wäre, ausschließlich eine vorher definierten Menge an Features für auf eine vorgegebenen Produktklasse zu verwenden, also die Review-Dokumente überhaupt nicht auf Features hin zu analysieren. Eine große Zahl der bestehenden Ansätze setzt jedoch auf die Extraktion der Features aus den Review-Dokumenten, wofür auch mehrere Gründe sprechen:

1. Produkt-Features aus bestehenden Taxonomien oder Listen könnten von den in den Review-Dokumenten genannten abweichen, und zwar in Bezug auf den Umfang der Features als auch deren Bezeichnung.[11]

2. Meinungsäußerungen treten innerhalb eines Satzes meist in der Nähe von Produkt-Features auf (siehe [HL04, PE05]). Dadurch kann der Feature-Extraktionsschritt auch gleich zur Meinungsextraktion verwendet bzw. eng mit diesem verwoben werden.

3. Wir vermuten, dass sich Änderungen bei Features (Umfang oder Bezeichnungen) verlässlicher aus (ständig neu erscheinenden) Review-Dokumenten extrahieren lassen als aus starren Taxonomien, die möglicherweise nicht aktualisiert werden.

Für Product-Review-Mining-Systeme, die eine Feature-Extraktion beinhalten, lässt sich die in Abb. 1 aufgezeigte generelle Systemarchitektur festhalten (basierend auf den in 1.2 genannten Schritten):

[10] Bei [CNZ05] konkret: `http://www.activebuyersguide.com`

[11] Synonyme, ähnliche/abweichende Bezeichnungen, alternative Schreibweisen, fehlende/zusätzliche Features

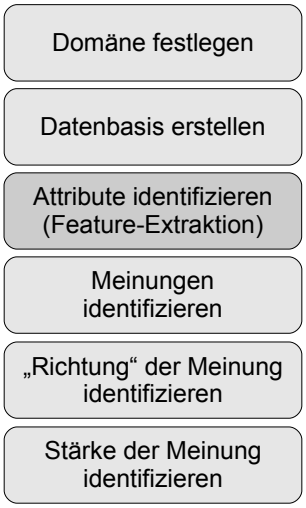

Abbildung 1: Feature-basiertes Product Review Mining

Dem Feature-basierten Product Review Mining stehen Systeme gegenüber, bei denen die Review-Dokumente nicht auf Feature-Ebene, sondern auf Satz- oder Dokumentebene analysiert und klassifiziert werden, also die Granularität der Textzusammenfassung gröber ist (vgl. Kap. 1.1).

1.3 Problemstellung

In dieser Arbeit wird die Feature-Extraktion als Teilbereich eines *Feature-basierten Product Review Mining* behandelt.

Dazu müssen wir zunächst die Herausforderungen und mögliche existierende Ansätze bei der Feature-Extraktion untersuchen und prüfen. Um selbst ein System zu entwickeln, müssen für die wesentlichen Punkte dieser Herausforderungen eigene Lösungen entwickelt und im Implementierungsprozess umgesetzt werden. Die Implementierung muss Qualitätsansprüchen an die Feature-Extraktion genügen, welche noch festzulegen sind. Dies ist zu evaluieren. Zudem soll das entstandene System im Rahmen eines im Aufbau befindlichen Projekts zum Product Review Mining verwendbar sein.

Im nächsten Kapitel werden nun verschiedene existierende Verfahren zur Feature-Extraktion untersucht. In Kap. 3 wird ein eigenes Verfahren vorgestellt und in einer prototypischen Implementierung umgesetzt. Dieses wird in Kap. 4 evaluiert.

2 Feature-Extraktion

2.1 Begriffsklärungen

2.1.1 Feature

Ein *Feature* ist ein Attribut zu Produkten einer Produktklasse. Dies kann ein physisches Teil eines Produktes sein, aber auch eine Eigenschaft. Die Bezeichnung *Feature* verwenden wir abkürzend für die eigentlich genauere Bezeichnung *Produktfeature*, auch weil der Begriff schon durch vorhandene Product-Review-Mining-Ansätzen etabliert ist ([PE05, Liu06, HL04, CNZ05] und andere). Denkbar wären allerdings auch alternative Bezeichnungen wie *Attribut*, *Merkmal* oder *Aspekt*. Features können wir in (mindestens) vier Kategorien unterteilen, hier mit Beispielen wie bei [PE05]:

Featurekategorie	**Beispiel**
das Produkt selbst	Scanner
Teil des Produkts	Deckel
Eigenschaft des Produkts	Größe
Eigenschaft eines Teils des Produkts	Batterie-Lebensdauer

Tabelle 1: Featurekategorien

2.1.2 Feature-Extraktion

Mit *Feature-Extraktion* (als Teilaufgabe des Product Review Mining, siehe Kap. 1.2) bezeichnen wir Verfahren, mit denen Produktfeatures zu den Produkten einer bestimmten Produktklasse ermittelt werden, und zwar aus den Review-Dokumenten selbst. Diese Attribute finden Verwendung bei den weiteren Schritten im Product Review Mining und können Teil dessen Resultats darstellen.

Wir verwenden *Feature-Extraktion* abkürzend für die eigentlich korrekte, speziellere Form *Produktfeature-Extraktion*.

2.2 Einordnung

Betrachtet man das Problem der Feature-Extraktion isoliert vom Rest der Herausforderungen beim Product Review Mining, handelt es sich um die Anwendung von Methoden der *(automatischen) Terminologieextraktion*, einem Teilgebiet der *Informationsextraktion*, deren Anfänge auf eine Arbeit von Luhn (1957) zurückdatieren.[12] Dementsprechend wurden hier standardisierte Verfahren entwickelt, auf die hier zurückgegriffen werden kann. Dazu gehören linguistische ebenso wie statistische Methoden, um natürlichsprachige Texte zu analysieren und Informationen aus ihnen zu extrahieren. Einige davon werden in den Kapiteln 2.4.1 und 2.4.2 vorgestellt.

Im Folgenden werden wir die Herausforderungen bei der maschinellen Extraktion von Features untersuchen. Im Kapitel zu existierenden Verfahren (2.4) gehen wir detailliert

[12] nach: [SB87]: Kap. 1, [KU96]: Kap. 1

auf Unterschiede zwischen der Terminologieextraktion im Allgemeinen und der Feature-Extraktion ein. Außerdem stellen wir verschiedene mögliche Ansätze zur Lösung unserer speziellen Herausforderungen vor.

2.3 Herausforderungen bei der Produkt-Feature-Extraktion

Um die Herausforderungen systematisch betrachten zu können, bedarf es der genauen Betrachtung von exemplarischen Review-Dokumenten. Deshalb wählen wir die Produktklasse **Digitalkameras**, englisch *digital cameras*, als Beispiel und Anwendungsfall für diese Untersuchung.[13] Auf diese Produktklasse werden sich die Beispiele und Analysen in dieser Arbeit beziehen. Wenn bei Textfragmenten eine Amazon-Produktnummer (z. B. B00005B6TI) angegeben ist, so kann das Produkt über eine Webseite zur Produktnummer[14] abgerufen werden.

2.3.1 Nominalphrasen

In [KU96] wird ein umfassender Überblick über Verfahren zur Terminologieextraktion[15] in der Computerlinguistik bzw. als Teil des Information Retrieval gegeben. In einer Zusammenfassung verschiedener linguistischer Untersuchungen zur Terminologieextraktion (S. 15) stellen sie fest, dass die meisten Begriffe in die linguistische Kategorie der *Nominalphrasen*[16] fallen. Auch laut [Liu06] gehören knapp 90 % aller direkt genannten Features zu dieser syntaktischen Kategorie, und diese Annahme liegt auch den Systemen von Popescu/Etzioni, Hu/Liu und Daille zugrunde.[17]

Das Identifizieren von Nominalphrasen (und im Folgenden die Beurteilung der Relevanz jeder Nominalphrase für die Feature-Extraktion) ist also eine erste Herausforderung.

2.3.2 Koreferenzprobleme

Features werden nicht immer direkt genannt. Sie können (z. B. über Personalpronomen) referenziert auftreten. Daher stammt die Bezeichnung Koreferenz: Es gibt für ein semantisches Konzept mehrere Referenzen. Im folgenden Beispiel beziehen sich die Personalpronomen **mine** und **it** beide auf dieselbe Kamera:

> (1) First, I have had **mine** for about 2 years, and **it** is still going strong, even after buying **it** as a factory refurb. (B00005B6TI)

Eine weitere Art der Koreferenz sind Variationen beim Auftreten von Features. Dazu gehören z. B. alternative oder orthographisch abweichende Schreibweisen, lexikalische Alternativen (Synonyme), aber auch das Auftrennen von Phrasen. Bei Variationen stellt sich zudem die Frage, welche der Alternativen jeweils als „Normalform“ gewählt werden

[13] Gründe sind die eindeutige Identifizierbarkeit von Features für Digitalkameras sowie die hohe Anzahl an vorhandenen Reviews im World Wide Web. Alle Auszüge aus Review-Dokumenten stammen, sofern nicht anders angegeben, von der US-Webseite des Online-Versandhaus *Amazon.com*.

[14] in der Form `http://www.amazon.com/gp/product/<Produktnummer>`

[15] dort: *ATR (Automatic Term Recognition)*

[16] „Syntaktische Kategorie, die normalerweise ein Nomen [...] oder Pronomen [...] als Kern enthält, der in verschiedener Weise erweitert sein kann“ [Bm02]

[17] [PE05]: Kap. 3.1, [HL04]: Kap. 3.1, [Dai96]: S. 29

soll; dazu lassen sich unterschiedliche Kriterien heranziehen (Details hierzu in Kapitel 2.4.1). Hier Beispiele für die synonyme Verwendung des Features **picture**:

(2) The **picture** quality is great, [...] (B0000C4E4P)

(3) The **photos** were amazing, [...] (B0000C4E4P)

(4) **Images** were blurred in parts. (B0001659AE)

2.3.3 Implizite Features

Features können auch implizit[18] auftreten, d. h. sie können aus den entsprechenden Textstellen abgeleitet werden. Ein Beispiel für ein implizites Auftreten der Features **size** und **weight**:

(5) It's **small**, it's compact, it's versatile, and it's **lightweight**. (B0000C4E4P)

Ein weiteres Beispiel für das implizit genannte Feature **usability**:

(6) The camera is flat out **easy to use** [...] (B00000JYLO)

2.3.4 Komposita

Für extrahierte Begriffe, die sich aus mehreren Wörtern zusammensetzen, muss festgestellt werden, ob es sich um ein tatsächliches Kompositum (Englisch: *compound word*) handelt. Hier ist beispielsweise der Begriff **shooting photos** kein Kompositum:

(7) **Shooting photos** on a diskette was easier than [...] (B00000JYLO)

Der Begriff **carrying case** hingegen ist ein Kompositum, obwohl er dasselbe syntaktische Muster (Gerundium plus Substantiv) besitzt:

(8) [...] with an extra battery and a **carrying case**. (B00000JYLO)

2.3.5 Typen von Features

Ein weiteres Problem stellt die Kategorisierung von Features dar. Wie aus Tabelle 1 (S. 6) ersichtlich, ist häufig eine hierarchische Struktur feststellbar: Teile eines Produkts stehen zum Produkt in einer *Meronymie*-Relation.[19] Eigenschaften eines Produkts sind semantisch ebenfalls dem Produkt untergeordnet.

Eine solche Einteilung in verschiedene Feature-Typen kann für potentielle Nutzer hilfreich sein, um einen geordneten Überblick über Meinungen zu einem Produkt zu erhalten. Ein Beispiel ist die aus [CNZ05] entlehnte, in Abb. 2 wiedergegebene hierarchische Gruppierung.

[18] *implicit features* bei [Liu06] und [PE05]

[19] „Semantische Relation zwischen sprachlichen Ausdrücken zur Bezeichnung der Beziehung des Teils zum Ganzen bzw. zur Bezeichnung von Besitzverhältnissen“ [Bm02]

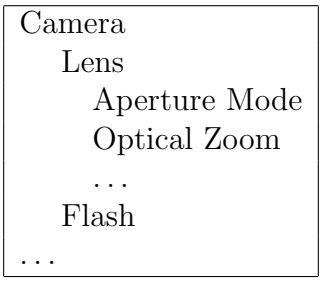

Abbildung 2: Ausschnitt aus einer Taxonomie für Digitalkameras

2.4 Existierende Verfahren

Die existierenden Verfahren zur Feature-Extraktion, wie bei [PE05, CNZ05, HL04], entsprechen einer *automatischen Indizierung*, wie es von [KU96] im Kontext der *Terminologieextraktion* erläutert wird. Hierbei werden zunächst sog. *Index-Begriffe* extrahiert und anschließend gewichtet.[20] Index-Begriffe sind solche, die als repräsentativ oder relevant im Zusammenhang der Terminologieextraktion gelten. Im einfachsten Fall sind dies bspw. alle Substantive eines Dokuments. Bei der Feature-Extraktion als Spezialfall der Terminologieextraktion sind die Index-Begriffe die späteren Features, sozusagen die *Featurekandidaten*.[21] Die Gewichtung der Index-Begriffe entspricht dann der *Bewertung* dieser Featurekandidaten. Durch die Bewertung gewinnt man eine Rangliste, bei der ggf. ein Grenzwert festgesetzt ist, anhand welchem Featurekandidaten in Features und Nicht-Features klassifiziert werden können. Dies ist eine Standard-Vorgehensweise der automatischen Indizierung (Details: [KU96]: Kap. 2.1).

In den folgenden Unterkapiteln gehen wir detaillierter auf die Methoden der automatischen Indizierung ein, die linguistischer und statistischer Art sind: die Extraktion der Begriffe, die unterschiedlichen Methoden zum Lösen der Herausforderungen, die wir in Kap. 2.3 beobachtet haben, sowie die Bewertung (Gewichtung) als Grundlage für eine Klassifizierung, die zu Kandidatenbegriffen *Features* liefert.

2.4.1 Linguistische Methoden: Extraktion

Eine linguistische Analyse der Quelldokumente dient bei der Feature-Extraktion als Vorbereitung zur Extraktion von Features auf der Grundlage vorgegebener syntaktischer Muster. Der Einsatz von linguistischen Filtern ist ein Standardverfahren für das Gebiet der automatischen Terminologieextraktion[22] und hat auch Einzug in das Opinion Mining gehalten.

Bei [PE05, HL04, DLP03, YNBN03] und anderen werden die Quelldokumente durch externe Werkzeuge zur Verarbeitung natürlicher Sprache[23] wie Part-of-Speech-Tagger[24] annotiert. Resultat sind Wörter und Phrasen, die nach syntaktischen Kategorien an-

[20] sog. *term weighting* [KU96]
[21] Bezeichnung für Begriffe, die mögliche Features sind
[22] siehe dazu u. a.: [Dai96]: S. 29 und [FAM00]: Kap. 2
[23] Natural Language Processing (NLP)
[24] leistet eine Annotation von Wortarten, vgl. Kap. 3.4.1

notiert sind. Von diesen werden ausschließlich Nominalphrasen berücksichtigt, die als primärer Indikator für Features angesehen werden (vgl. 2.3.1).

Variationen Um die unter 2.3.2 beschriebenen Variationen von Features zu erkennen, gibt es mehrere linguistische Ansätze, die meist auf syntaktischen Mustern basieren. So können unterschiedliche Formen von synonymen Begriffen bspw. über Regeln auf eine einheitliche Form geführt werden (folgendes Beispiel aus [NAM04]):

clones of human → *human clone*

Bei [CNZ05] wird eine Ähnlichkeitsanalyse durchgeführt, um Variationen bei Begriffen zu erkennen und sie zusammenzuführen.[25] Diese basiert auf mehreren „Distanzmaßen" lingustischer Art zwischen zwei Begriffen. So werden bspw. die einzelnen Wörter einen zusammengesetzen Begriffs auf Gleichheit, ähnliche Schreibung und Zugehörigkeit zur selben Synonymgruppe geprüft. Anhand einer gewichteten Kombination dieser Maße können mögliche Variationen auf eine gemeinsame einheitliche Form geführt werden.

Normalisierung Ein weiterer linguistischer Schritt bei der Feature-Extraktion ist die Rückführung auf „Grundformen" bezüglich Numerus und Kasus, d. h. dass Plural in Singular und Kasusdeklinationen in den Nominativ geändert werden müssen. Letzteres ist für die englische Sprache jedoch nicht relevant, da es bis auf das Genitiv-S keine syntaktischen Abweichungen bei Kasus gibt.[26]

Im Rahmen des Feature-basierten Product Review Mining werden ebenfalls Methoden verwendet, die das Finden einer „lemmatisierten" Form zum Ziel haben, allerdings nicht bei allen betrachteten Ansätzen: Von den o. g. Verfahren, die Werkzeuge zur Verarbeitung natürlicher Sprache verwenden, erwähnen nur [HL04][27] und [DLP03][28] explizit eine Nachbearbeitung der Nominalphrasen. Allerdings Auch werden auch bei den o. g. Verfahren aus [CNZ05] zur Ähnlichkeitsanalyse Normalisierungen von Begriffen durchgeführt.

Typen von Features Um das Problem der unterschiedlichen Typen von Features (Teile, Eigenschaften usw., vgl. Kap. 2.3.5) zu lösen, betrachten [PE05] u. a. lexikalische Hierarchien unter Wörtern. So kann die Hyponymie-/Hyperonymie-Beziehung[29] ein Hinweis für das Auftreten von Teilen oder Eigenschaften eines Produkts sein. Bei [CNZ05] wird dieses Problem hingegen auf manuelle Weise angegangen (durch Interaktion mit dem Benutzer), und viele Product-Review-Mining-Systeme verzichten auf einen Hierarchisierung der Features anhand ihres Typs.

2.4.2 Statistische Methoden: Gewichtung

Das simpelste statistische Maß zur Gewichtung von Featurekandidaten ist die absolute Häufigkeit des Auftretens. Dabei werden jedoch Begriffe, die sehr häufig in einer Sprache

[25] englisch: „Similarity Matching"

[26] „The noun only has a common case for subject, object, and after prepositions, and a possessive case (also called s-genitive) [...]" [UMS94]

[27] „stemming, stopwords and fuzzy matching"

[28] Experimente mit Stemming, die nicht erfolgreich waren

[29] „lexikalische Unter-/Überordnung, auch: Teil-Mengen-Beziehung" [Bm02]

auftreten, bevorzugt, unabhängig davon, ob sie für die Domäne (hier: Produktklasse) relevant sind. Dementsprechend werden bei diesem Maß selten genannte aber relevante Begriffe benachteiligt. Zur expliziten Extraktion dieser selten auftretenden Features analysieren [HL04] in Review-Dokumenten, zu denen kein Feature gefunden wurde, die Textstellen mit wertenden Begriffen und extrahieren dort vorkommende Nominalphrasen als *infrequent features*.

Kollokationen Zum Auffinden von für eine Produktklasse relevanten Begriffen (und Ausschluss der anderen) unabhängig von Häufigkeiten bietet sich an, die Stärke der *Verbindung* (englisch: *association*) zwischen zwei Begriffen zu untersuchen. Dabei wird davon ausgegangen, dass relevante Begriffe im Kontext einer Domäne häufiger auftreten als im Kontext anderer Domänen, oder im Kontext einer Domäne häufiger als generell (d. h. ohne einen spezifischen Kontext) (vgl. [KU96]: S. 16). In diesem Fall ist die linguistische *Kollokation* zwischen den beiden Begriffen hoch, was auf eine semantische Nähe zwischen ihnen hindeutet.

Ein etabliertes Maß der Informationstheorie zur Berechnung der Kollokation ist der *PMI-Wert*[30] ([MS99]: S. 178 ff.). Darin fließen die Häufigkeiten des alleinigen und gemeinsamen Auftretens zweier Zufallsvariablen ein. Dieses Maß wird bspw. bei der Initialisierung des Terminologieextraktionsprozesses in [Etz04] zum Finden geeigneter Extraktionsmuster verwendet.

$$\mathrm{PMI}(A,B) = \frac{\text{Wahrscheinlichkeit des gemeinsamen Auftretens von } A \text{ und } B}{\text{W-keit des Auftretens von } A \cdot \text{W-keit des Auftretens von } B}$$

Hier sind A und B zwei Zeichenketten, zu denen es in einem Korpus Häufigkeiten des Auftretens gibt.

Komposita Für das in 2.3.4 beschriebene Problem, herauszufinden, ob ein zusammengesetzter Begriff A ein tatsächliches Kompositum ist, schlagen [FAM00] die Berechnung des *C-Value* für solche Begriffe vor.

Positiv ins Gewicht fällt dabei die Gesamthäufigkeit des Begriffs im Korpus ($f(A)$). Negativ hingegen wirkt sich aus, wenn er als als Teil eines oder mehrerer Begriffe mit mehr Einzelwörtern ($B_1 \dots B_n$) vorkommt und diese dabei durchschnittlich auch noch relativ häufig auftreten. Der Logarithmus dient hier dem Abschwächen des positiven Effekts, der entsteht, wenn ein Begriff aus vielen Einzelwörtern besteht (nach: [FAM00], S. 4).

$$\text{C-value}(A) = log_2|A| \cdot \left(f(A) - \frac{\sum_{i=1}^{n} B_i}{\text{Anzahl der } B_i} \right)$$

Dabei fällt der Subtrahend weg, wenn es keine längeren Begriffe gibt (also auch immer, wenn $|A|$ maximal ist).

Obwohl dieses Maß auf heuristischen Annahmen basiert, hat es bereits Einzug in diverse Ansätze zur Terminologieextraktion gefunden und ist für dieses Fachgebiet als etabliert zu bezeichnen.

[30] *Pointwise Mutual Information*

Nicht direkt genannte Features Zur Lösung des Problems der impliziten Features (Kap. 2.3.3) gibt es bisher kaum Ansätze [Liu06]. Dort wird das manuelle Erstellen von Listen vorgeschlagen, welche Zuordnungen von impliziten Features zu ihrer expliziten Form enthalten;[31] eine Idee, die großen manuellen Aufwand erfordert. Das System aus [PE05] leistet laut eigenen Angaben eine Erkennung impliziter Features; es fehlt aber eine Erläuterung der Vorgehensweise dazu.

2.5 Gütemaße für Extraktionsverfahren

2.5.1 Precision, Recall, F-Maß

Ein Verfahren zur Extraktion von Features hat als Eingabe eine Menge von Dokumenten und gibt eine Menge von Features aus. Es sollte zwei funktionale Kriterien erfüllen:

1. Es sollen möglichst viele tatsächliche Features gefunden werden.
2. Es sollen möglichst wenige Begriffe gefunden werden, die keine Features sind.

Diese Kriterien entsprechen denen an ein beliebiges System zur Informationsextraktion, und sie lassen sich, jeweils für einen Extraktionslauf, anhand von *Recall* und *Precision* messen, zwei Standardmaßen für die Qualität von Extraktionen im *Information Retrieval*.[32] Der *Recall*-Wert ist ein Maß für relevante, der *Precision*-Wert für irrelevante Extraktionen:[33]

$$\text{Recall: } \frac{\text{Anzahl gefundener tatsächlicher Features}}{\text{Anzahl aller tatsächlichen Features}}$$

$$\text{Precision: } \frac{\text{Anzahl gefundener tatsächlicher Features}}{\text{Anzahl aller gefundenen Features}}$$

Zum Erreichen eines hohen *Recall*-Werts müssen also möglichst viele der tatsächlich vorhandenen Features gefunden werden. Dazu müssen die (o. g.) Formen berücksichtigt werden, in denen Features auftreten können. Der *Precision*-Wert ist dann hoch, wenn nicht allzu viele Begriffe fälschlicherweise als Feature ausgegeben werden. Also sollten solche Begriffe in einem Feature-Extraktionsverfahren ausgeschlossen werden.

Um ein Verfahren über ein einziges Maß beurteilen zu können, wurde das *F-Maß* als Kombination der Werte für Precision und Recall über das harmonische Mittel vorgeschlagen [MKSW99]:

$$F = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

[31] Beispiel: *heavy* → *weight*

[32] Siehe [KU96], S. 2: „In IR, index terms are evaluated by their retrieval performance, namely recall and precision.“ sowie: [MS99]: S. 277 ff.

[33] Eine Beispielrechnung zu Precision und Recall findet sich bei [Bes07a], Folie 24.

2.5.2 Spezifische Maße

Die speziellen Herausforderungen bei der Feature-Extraktion (vgl. Kap. 2.3) können bisher nur stichprobenartig gemessen werden. Beispielsweise wäre ein Maß für die Güte der Lösung der Koreferenzprobleme:

> *Wurden Features als unterschiedlich gekennzeichnet, obwohl sie dasselbe semantische Konzept bezeichnen?*

Für implizite Features:

> *Wurden implizite Features übersehen? Wurden implizite Features fälschlicherweise als solche identifiziert?*

Für die Typen von Features:

> *Wurden Features falschen Kategorien zugeordnet?* (Beispiel: Ein *Teil* eines Produkts wird als *Eigenschaft* gekennzeichnet.)

Für Komposita:

> *Wurden Begriffe, die Teile von Komposita sind, fälschlicherweise als Feature identifiziert?*

Anschließend können Statistiken erstellt werden, die die jeweiligen Fehlerquoten wiedergeben und dadurch eine genauere Bewertung des Extraktionssystems ermöglichen.

3 Ein Feature-Extraktionssystem

Aus den im vorhergehenden Teil vorgestellten möglichen unterschiedlichen Ansätzen wählen wir einige Methoden aus, die wir als relevant für eine erfolgreiche Feature-Extraktion erachten, und machen uns diese für ein eigenes, neues Verfahren zunutze, welches prototypisch implementiert wird. Resultat dieser Implementierung ist ein System zur Feature-Extraktion für eine vorgegebene Produktklasse und aus englischsprachigen Review-Dokumenten. Dieses wird im Folgenden vorgestellt und evaluiert.

Dieses Feature-Extraktionssystem ist Teil eines Projekts zum Product Review Mining. Auch aus diesem Grund wurde Wert darauf gelegt, dass die Daten weiterverwendbar und die Schnittstellen wohldefiniert sind. Ein Eckpfeiler ist hierbei die Datenbankstrategie, über die sichergestellt ist, dass gefundene Features gespeichert werden und die Review-Dokumente entsprechend annotiert vorgehalten bleiben. Außerdem werden durch die Verwendung des *UIMA*-Frameworks eine einfache Weiterverwendbarkeit, wohldefinierte Schnittstellen und eine große Modularität gewährleistet; Details dazu in Kap. 3.2.

3.1 Überblick

Das implementierte System deckt einen Teilbereich eines *Product-Review-Mining*-Prozesses ab und wurde so konzipiert, dass es Daten bereitstellt, die im weiteren Verlauf des Product Review Mining verwendet werden können. Die Datenbasis von Review-Dokumenten, also der **Korpus**, wird in einer relationalen Datenbank gespeichert. Diese Dokumente werden durch das System mit Annotationen versehen, d. h. Markierungen einzelner Segmente, so auch der *Features* als Endergebnis des Prozesses. Sowohl die annotierten Dokumente als auch eine Liste der Features bleiben als Datensätze in der Datenbank verfügbar.

Die **Extraktion** der Features wird auf Basis von syntaktischen Mustern durchgeführt. Außerdem werden Filterregeln zur Identifizierung von möglichen Features (Featurekandidaten) eingesetzt.

In einem **Normalisierungsschritt** werden diese Featurekandidaten auf Grundformen zurückgeführt sowie mehreren domänenunabhängigen Plausibilitätsprüfungen unterzogen.

Schließlich findet über statistische Verfahren eine Bewertung jedes Featurekandidaten statt. Dadurch kann für sie anhand eines festzulegenden Grenzwertes eine **Klassifizierung** in zwei Gruppen (Features und keine Features) durchgeführt werden. Dabei werden nur explizite Features ermittelt; implizite Features (siehe Kap. 2.3.3) werden im Rahmen dieser Arbeit nicht berücksichtigt.

Die Unterschritte Korpusaufbau, Extraktion, Normalisierung und Klassifizierung sind in einzelne Systemkomponenten aufgeteilt. Diese werden in den folgenden Kapiteln erläutert.

Das System wurde in Java implementiert. Es wurde auf Grundlage der Java-Version 1.5 erstellt und getestet. Der gesamte Quellcode wurde im Javadoc-Format dokumentiert und wird zusammen mit dieser Arbeit bereitgestellt. Im Anhang dieser Arbeit (Kap. 6.1) befinden sich Angaben zur erforderlichen Datenbank, Beschreibungen zum Aufruf der

einzelnen Komponenten, wie bspw. UIMA-Schnittstellen und -Konfigurationsdateien, sowie eine Liste der technischen Voraussetzungen.

Abb. 3 zeigt die Systemarchitektur; grau hervorgehoben die logischen Systemkomponenten.

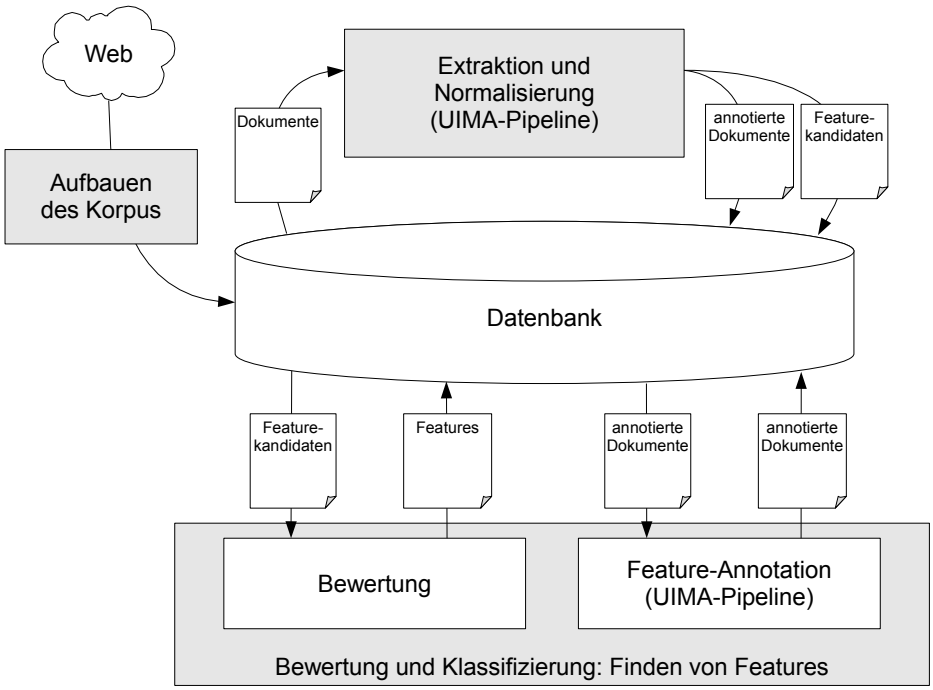

Abbildung 3: Komponenten des Feature-Extraktionssystems

3.2 UIMA-Framework

Wesentliche Teile des Systems wurden basierend auf dem *UIMA*-Framework umgesetzt. UIMA ist ein von *IBM* initiiertes, mittlerweile bei *Apache* als Open-Source-Projekt frei verfügbares Framework für die Verwendung sowie die Entwicklung von Anwendungen, die unstrukturierte Informationen wie Text, Audio oder Video verarbeiten.[34] Durch den Einsatz von UIMA-Frameworks sind wohldefinierte Abläufe, Schnittstellen und Fehlerbehandlungen für die erstellten Java-Klassen gewährleistet. Ein weiterer Grund für die Entscheidung, UIMA-basiert zu programmieren, ist die einfache Weiterverwendbarkeit und Einbindung der einzelnen Komponenten in neue Systeme, bzw. eine mögliche Erweiterung dieses Systems mit nur relativ geringem Einarbeitungsaufwand dank der wohldefinierten Schnittstellen.

Ein herausragendes Merkmal von UIMA ist die Aufteilung eines größeren Prozesses in einzelne Komponenten. Jede dieser Komponenten dient zur Verarbeitung eines Doku-

[34] „UIMA is a component framework for analysing unstructured content such as text, audio and video. It comprises an SDK and tooling for composing and running analytic components [...]“ [Apa08b]

ments[35] und hat wohldefinierte Schnittstellen zum Dokument-Ein- und -Ausgang.

Jede Komponente besteht aus einem *Descriptor* und einer implementierten Umsetzung der Komponente. Der Descriptor ist eine XML-Datei, in der u. a. Parameter zum Starten der Komponente und die erforderlichen UIMA-Datentypen angegeben werden. Diese Datentypen werden in einem oder mehreren *Typsystemen* spezifiziert, einer Art Objektschema, welches von der/den Komponente(n) verwendet wird. Diese Wohldefiniertheit der Komponentenschnittstellen erlaubt es, sie in einer linearen Abfolge zu kombinieren und so für zu analysierende Dokumente eine *Pipeline* zu erstellen.

Auf einzelne Dokumente wird über ein Konzept namens *Common Analysis Structure* (CAS) zugegriffen, welches eine Struktur für Objekte, Eigenschaften und Werte bildet.

Innerhalb einer UIMA-Pipeline können Komponenten unterschiedlicher Formen kombiniert werden:

Analysis Engine: Eine Analysis Engine besteht typischerweise aus einem *Annotator* und einem Descriptor. In ihr wird die Analyse eines Dokuments durchgeführt. So gibt es einzelne Analysis Engines für verschiedene NLP-Aufgaben, wie das Auffinden von Satzgrenzen oder Part-of-Speech-Tagging (Erläuterungen hierzu in Kap. 3.4.1).

Collection Reader: Diese Komponente dient zum Einlesen von Quelldokumenten, z. B. aus einer Datenbank oder vom Dateisystem, und Initialisieren der CASes (Common Analysis Structure, s. o.), die von folgenden Komponenten verarbeitet werden können.

CAS Consumer: Dies ist eine spezielle Analysis Engine, die am Ende der Pipeline steht und die Analyseergebnisse aus- bzw. weitergibt. Ein CAS Consumer kann bspw. nach bestimmten Annotationen filtern, bestimmte Werte in Datenbanken schreiben oder komplette Dokumente als Dateien auf Datenträger schreiben.

Dabei können einzelne Analysis Engines zu *Aggregate Analysis Engines* gruppiert werden. Eine gesamte Pipeline wird als *Collection Processing Engine* bezeichnet. Wie die einzelnen Komponenten innerhalb der Pipeline erfordern auch Aggregate Analysis Engines und Collection Processing Engines XML-Descriptor-Dateien, wobei darin gesetzte Parameter die Parameter der hierarchisch untergeordneten Komponenten-Descriptor-Dateien überschreiben.

In Abbildung 4 ist die Anordnung von UIMA-Komponenten innerhalb einer Collection Processing Engine zu sehen.[36]

Die UIMA-Komponenten im vorliegenden Feature-Extraktionssystem annotieren jedes Review-Dokument bezüglich syntaktischen Elementen sowie Featurekandidaten. Details zur jeweiligen UIMA-Implementierung folgen in den Kapiteln 3.4 und 3.5. Im Anhang (Kap. 6.1) sind Beschreibungen zum Aufruf der UIMA-Komponenten zu finden.

[35] Der Einfachheit halber setzen wir voraus, dass es sich um ein Textdokument handelt, obwohl UIMA auch für andere Medienformen wie z. B. Tondokumente entworfen wurde.

[36] Originaltitel: *High-Level UIMA Component Architecture from Source to Sink*, entnommen aus: [Apa08a], Copyright 2004, 2006 International Business Machines Corporation; Copyright 2006, 2008 The Apache Software Foundation. Hier wiedergegeben auf Grundlage der *Apache License*, Punkt 4. Redistribution, `http://incubator.apache.org/uima/license.html`, zuletzt geprüft: 2008-06-06

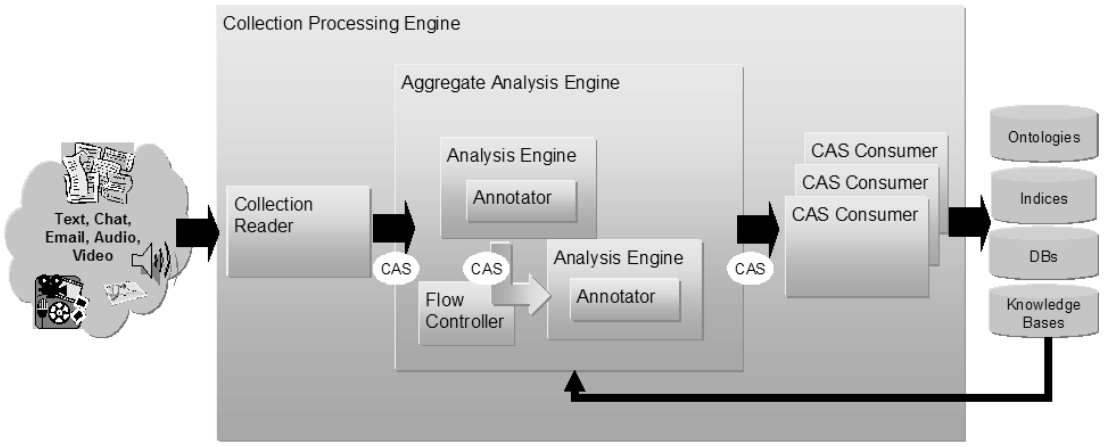

Abbildung 4: UIMA-Collection-Processing-Engine

3.3 Komponente: Aufbauen des Korpus

Review-Dokumente zu Produkten finden sich im World Wide Web an vielen unterschiedlichen Stellen. Einerseits bieten viele Online-Händler ihren Kunden die Möglichkeit, Texte mit Bewertungen zu den angebotenen Produkten zu verfassen, wie das Online-Versandhaus *Amazon.com* oder der Reisevertrieb *opodo*. Des Weiteren gibt es mehrere auf Produktbewertungen spezialisierte Webseiten, wie *Epinions*[37] oder *CNET Reviews*.[38] Zudem werden Produkte in Weblogs[39] oder auf Homepages bewertet. Um das *Opinion Mining* für eine bestimmte Produktklasse möglichst umfassend durchzuführen, ist es wünschenswert, von möglichst vielen dieser Stellen die Texte mit den entsprechenden Meinungsäußerungen zu erfassen. Dazu wäre es nötig, diese sehr heterogen aufgebauten Webseiten auf eine generische Art und Weise zu durchsuchen und die Review-Texte von dort zu extrahieren – über einen *generischen Web-Crawler für Produkt-Reviews*.

Das hier vorgestellte System stellt hingegen eine vorerst prototypische Implementierung dar, sodass von einer einfacher zugänglichen und trotzdem umfangreichen Datenquelle Gebrauch gemacht wird – dem *Amazon Associates Web Service*. Das ist ein *Webservice*,[40] der Zugriff auf den Katalog der Produkte (inklusive Review-Dokumente) von Amazon ermöglicht.

Es wurde ein Programm geschrieben, welches über den Amazon Associates Web Service für eine vorgegebene Produktklasse alle Review-Dokumente[41] aus dem Katalog der US-Webseite von Amazon abruft und in einer *MySQL*-Datenbank speichert. Das Programm ist für beliebige Produktklassen und Suchanfragen konfigurierbar. Details zum parametrisierten Aufruf und zur Datenbank finden sich im Kap. 6.1. Das Resultat dieser Komponente ist eine Datenbasis, die den Korpus für die folgenden Schritte darstellt. Diese beinhaltet Informationen zu Produkten einer Produktklasse, zugehörigen

[37] `http://www.epinions.com/`
[38] `http://reviews.cnet.com/`
[39] Beispiel: `http://dpreview.com/`
[40] Online-Dienst zur maschinellen Interaktion zwischen zwei Agenten über Daten in XML-Form
[41] momentan begrenzt auf maximal 4000 Produkte, da eine Abfrage von mehr als 800 Ergebnisseiten (mit je 50 Produkten) durch Amazon nicht zugelassen ist

Review-Texten sowie einer Liste von Produktattributen, die der Webservice ebenfalls liefert und die Hinweise für die Klassifizierung der Features liefern können.

3.4 Komponente: Extraktion und Normalisierung

Die Komponente zur Extraktion der Featurekandidaten und deren Normalisierung wurde, wie unter 3.2 beschrieben, als UIMA-Anwendung umgesetzt. Dabei durchlaufen die Dokumente (dies sind die einzelnen Reviews) eine Pipeline von Verarbeitungsschritten. An deren Anfang steht ein *CollectionReader*, welcher die in der Datenbank vorhandenen Dokumente einliest und jedes Dokument an die folgenden UIMA-Komponenten übergibt.

Die Weiterverarbeitung geschieht mit vier Annotatoren, welche jedes Dokument mit Annotationen zu verschiedenen syntaktischen Kategorien bereichern. Dazu mehr im folgenden Kapitel, ebenso wie zur Extraktion und Normalisierung der Featurekandidaten.

Abb. 5 zeigt die Systemkomponente *Extraktion und Normalisierung*, implementiert als UIMA-Pipeline.

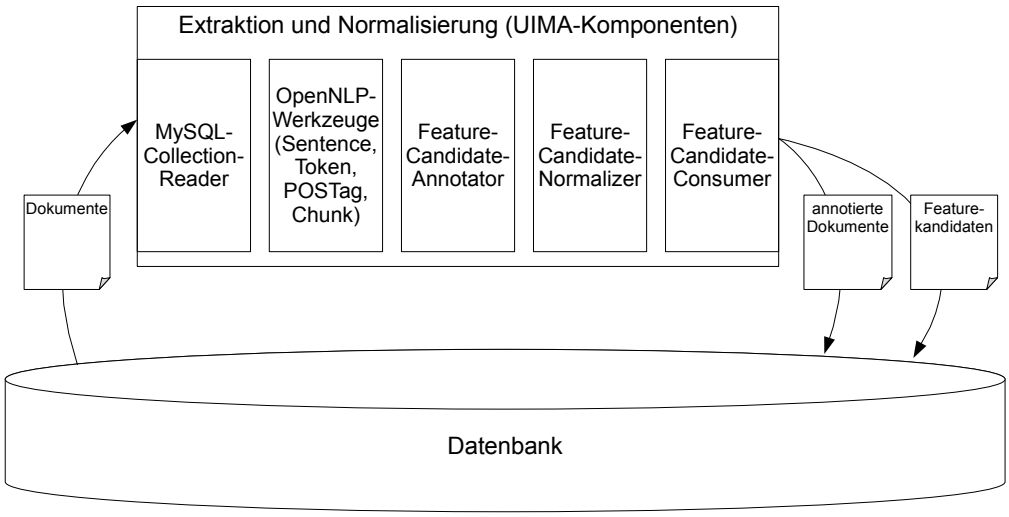

Abbildung 5: Systemkomponente Extraktion und Normalisierung

3.4.1 Syntaktische Annotation

Vier UIMA-Annotatoren dienen der syntaktischen Annotation der Review-Dokumente. Ein *SentenceAnnotator* unterteilt das Dokument anhand von Satzgrenzen. Ein *TokenAnnotator* markiert jedes Token (Wort oder Zeichen) im Dokument. Die Wortart (Englisch: part-of-speech) jedes Tokens wird durch einen *PosTagAnnotator* annotiert.[42] Die unmittelbaren Konstituenten eines Satzes[43] markiert ein *ChunkAnnotator*.

[42]Dieser verwendet die Syntax für POS-Tags des *Penn-Treebank-Projekts* der University of Pennsylvania; siehe Kap. 6.5

[43]in der strukturellen Satzanalyse hierarchisch direkt unterhalb eines Satzes stehende sprachliche Einheiten; nach: [Bm02]

Diese Annotatoren sind frei verfügbar und als UIMA-Wrapper[44] für bestehende Werkzeuge der Open-Source-Initiative *OpenNLP*[45] implementiert. Dabei werden Modelldateien (beinhalten Wortschatzlisten zu bspw. POS-Tags, Akronymen und Vornamen) eingesetzt, die ebenfalls auf die Arbeit von OpenNLP-Aktiven zurückgehen. Zur einfacheren Konfiguration und Einbindung in diese Systemkomponente sind die vier Annotatoren über das UIMA-Konzept *Aggregate Analysis Engine* (vgl. Kap. 3.2) zusammengefasst.

Die dementsprechend syntaktisch annotatierten Dokumente dienen als Eingabe für die folgenden beiden UIMA-Komponenten, die das Herzstück der Anwendung bilden.

3.4.2 Annotation von Featurekandidaten

Wie in Kap. 2.3.1 begründet, werden Substantive und Nominalphrasen als hauptsächliche grammatikalische Kategorie für Features angesehen. Die UIMA-Analysis-Engine *FeatureCandidateAnnotator* betrachtet von den vorangegangenen syntaktischen Annotationen daher ausschließlich Segmente mit Markierungen dieser beiden Kategorien, also sämtliche Substantive und Nominalphrasen eines Dokuments.[46] Als Featurekandidat qualifiziert sich ein Segment dieser Kategorien, wenn es weitere Prüfungen besteht:

- Das Segment wird gegen eine Liste von Stoppwörtern geprüft. Dies sind allgemeinsprachliche Wörter, die keine domänenspezifische Semantik besitzen. Beispiele (Substantive): **anything**, **example**, **thanks**
- Das Segment muss mindestens zwei aufeinanderfolgende alphanumerische Zeichen beinhalten. Beispiele für Wörter oder Phrasen ohne diese Eigenschaft: **S1000** (Produktnamen), **(** (fälschlicherweise erkanntes Klammerzeichen). Die einzigen beiden Wörter in der englischen Sprache, die aus nur einem Buchstaben bestehen, sind *a* und *I*, die als Feature nicht infrage kommen.
- Nominalphrasen müssen mindestens ein Substantiv enthalten (POS-Tag `NN` oder `NNS`), welches der Kopf der Nominalphrase ist. Dadurch werden fälschlicherweise als Nominalphrasen erkannte `Chunks` (Konstituenten) gefunden. Beispiel: **more 'normal** wurde mit den POS-Tags `RBR JJ` (Adverb, comparative; adjective), aber als `ChunkNP` (Nominalphrase) annotiert.
- Das Segment darf nicht (einem Teil der) Produktklasse entsprechen. Beispiel: **camera** für die Produktklasse **Digital Cameras**

3.4.3 Normalisierung der Featurekandidaten

Diese Komponente namens *FeatureCandidateNormalizer* dient dazu, Koreferenzen aufzulösen, d. h. Featurekandidaten, die dasselbe semantische Konzept beschreiben, aber

[44] implementiert und öffentlich bereitgestellt vom *JulieLab* (Jena University Language & Information Engineering Lab) der Friedrich-Schiller-Universität Jena, `http://www.julielab.de/content/view/117/174/`, zuletzt geprüft: 2008-06-02

[45] *Open Natural Language Processing*, stellen verschiedene Werkzeuge zur Verarbeitung natürlicher Sprache bereit, `http://opennlp.sourceforge.net/`, zuletzt geprüft: 2008-06-02

[46] annotiert als `ChunkNP` (Noun Phrase Chunk), `NN` (Noun (Singular/Mass)) oder `NNS` (Noun (Plural))

unterschiedliche syntaktische Formen aufweisen, auf eine gemeinsame Grundform zu führen (daher der Name „Normalizer“). Dabei werden heuristische linguistische Methoden angewendet, die sich teilweise auch in anderen Verfahren zur Feature-Extraktion wiederfinden. So werden Variationen bezüglich Numerus (Singular/Plural), Schreibweise, Groß- und Kleinschreibung sowie dem Auftreten mit Artikel o. ä. aufgelöst. Des Weiteren werden Synonyme wenn möglich auf eine gemeinsame Form abgebildet, sowie stark wertende Adjektive von Nominalphrasen entfernt.

Werkzeuge Der FeatureCandidateNormalizer setzt zwei externe Werkzeuge linguistischer Art ein. Das frei verfügbare Softwarepaket *WordNet* ist eine lexikalische Datenbank, die einen Wortschatz der englischen Sprache mit Verknüpfungen zwischen Begriffen enthält [Fel98]. Diese Verknüpfungen sind sowohl syntaktischer als semantischer Art. Der Zugriff auf die WordNet-Datenbasis wird über die Java-API *JWI* durchgeführt.[47] Dabei müssen die POS-Tags, die bei der Annotation Verwendung fanden, auf die WordNet-Äquivalente abgebildet werden. Dies sind einerseits die Penn-Treebank-POS-Tags wie in Kap. 6.5 angegeben und andererseits die WordNet-POS-Tags `NOUN`, `ADJECTIVE`, `VERB` und `ADVERB`.

Die WordNet-Erweiterung *SentiWordNet* ist eine Liste von WordNet-Einträgen mit Annotationen, ob ein Begriff eine wertende Konnotation beinhaltet, und wenn ja, wie stark diese ist [ES06]. Dies wird ausgedrückt auf Skalen zu *Positivity*, *Negativity* und *Objectivity*. Diese Liste ist als tabulatorgetrennte Textdatei verfügbar und kann über eine Hilfsklasse eingelesen und verwendet werden. Ein Beispiel ist das Wort **fresh** in der ersten von 13 Bedeutungen mit Positivity = 0.5, Negativity = 0.375 und Objectivity = 0.125 zu nennen.

Variationen Über WordNet-Wortstämme kann für Substantive[48] die Singularform eines Begriffs gefunden werden. Wenn dies gelingt, wird auch das POS-Tag dementsprechend geändert (Beispiel: `NNS` → `NN`).

Um mögliche Features zusammenzuführen, die mit modifizierenden Elementen eingeleitet auftreten können, wird das jeweils erste Token eines Featurekandidaten je nach Wortart verworfen.[49] Beispiele hierfür:

- Artikel: **The battery**
- Präposition: **of the resolution**

Des Weiteren werden alle Token an erster Position verworfen, die in der Liste der Stoppwörter vorkommen (Beispiel: **much power**).[50] Die Schreibungen **&** und **+** für das Wort **and** werden auf **and** abgebildet.

[47] *MIT Java Wordnet Interface*, `http://www.mit.edu/~markaf/projects/wordnet/`, zuletzt geprüft: 2008-06-02

[48] POS-Tags: `FW`, `NN`, `NNS`, `NNP`, `NNPS`, vgl. Referenz in Kap. 6.5

[49] Die zu verwerfenden Wortarten sind v. a. solche, die keine Bedeutung für die Feature-Extraktion tragen können, wie Präpositionen / Konjunktionen (CC, IN), Artikel (DT, PDT, WDT), Symbole (SYM) oder vergleichende Adjektive und Adverben (JJR, JJS, RBR, RBS).

[50] Dabei wird dieselbe Liste wie in Kap. 3.4.2 verwendet. Die Prüfung hier bezieht sich nur auf einzelne Token an erster Position, nicht auf gesamte Featurekandidaten.

Synonyme Sofern es zu einem Token des Featurekandidaten in WordNet eine Grundform („Lemma“) gibt, wird diese verwendet. Dabei wird stets der erste Eintrag eines WordNet-*synsets* (Synonymgruppe) verwendet, sodass exakte Synonyme dieselbe Lemmaform erhalten. Wörter mit nahen, aber nicht synonymen Bedeutungen wie **picture** und **image** bleiben daher unterschiedlich.

Wertende Adjektive In *SentiWordNet* ist ein *objektives* Wort eines, das weder positive noch negative Charakterisitika aufweist.[51] Dies lässt sich aus den Werten für *Positivity* und *Negativity* ableiten: Die *Objectivity* ist die Differenz der Summe von Positivity und Negativity zu 1, also desto größer, je weniger positiv und negativ konnotiert ein Wort ist. Da ein Wort viele Bedeutungen haben kann (s. o.: **fresh** hat 13 Bedutungen) und wir nicht wissen, welche der Bedeutungen im jeweiligen Kontext relevant ist, wählen wir den Durchschnittswert der Objectivity aller Bedeutungen eines Wortes als Maß. Beispielswerte für Adjektive sind:

Wort	**Objectivity-Wert**
stupid	0.75
old	$0.7\overline{2}$
amateur	0.6875
great	$0.680\overline{5}$
bright	$0.602\overline{72}$

Tabelle 2: SentiWordNet-Beispiel für Adjektive

Beim Betrachten dieser Tabelle fällt auf, dass die Objectivity eine Worts nicht immer den Wert hat, den man vielleicht intuitiv für die Feature-Extraktion vermuten würde. So wird **stupid** im Vergleich zu **bright** als ein relativ objektives Adjektiv eingestuft, was daran liegt, dass zwei der drei Bedeutungen von **stupid** neutralen Charakter haben, **bright** aber viele Bedeutungen mit positiver Konnotation hat. Dies ist ein ohne manuelle Kontrolle des Wortschatzes nicht generell lösbares Problem, sodass vorerst nur mit Heuristiken gearbeitet werden kann.

So legen wir einen Grenzwert, ab welchem wir Adjektive aus Featurekandidaten verwerfen wollen, rein willkürlich nach dem Betrachten einiger Beispiele fest. Für diese Komponente haben wir uns für einen Objectivtiy-Wert von 0.685 entschieden, der bspw. **amateur** als *objektiv*, aber **great** schon als *subjektiv* (d. h. mit wertender Konnotation) klassifiziert. Dementsprechend werden Featurekandidaten mit Adjektiven an erster Position folgendermaßen normalisiert:

great look and feel → *look and feel*

aber:

optical zoom → *optical zoom*

[51] [...] an objective term can be defined as a term that does not have either positive or negative characteristics [...] [ES06]

Abschluss der UIMA-Pipeline

Eine letzte UIMA-Komponente, der *FeatureCandidateConsumer*, schreibt für jedes Dokument die vorgenommenen Annotationen zurück in die Datenbank. Das sind sämtliche Featurekandidaten mit ihrer normalisierten Form sowie Häufigkeiten für deren Auftreten über alle Dokumente hinweg. Schließlich wird jedes Review-Dokument in der annotierten Form (in einer UIMA-spezifischen XML-Syntax) in der Datenbank gespeichert, auch im Hinblick auf folgende Schritte im Product-Review-Mining-Prozess, bei denen Review-Dokumente weiterverarbeitet werden sollen.

3.5 Komponente: Bewertung und Klassifizierung: Finden von Features

In dieser Systemkomponente wird die Bewertung der Featurekandidaten vorgenommen. Diese findet auf Basis verschiedener Maße statt, die wir nach der Analyse bestehender Verfahren und dem Identifizieren der spezifischen Herausforderungen auf Grundlage der Struktur der Review-Dokumente (vgl. Kap. 2.3) auswählen. Jeder Featurekandidat ordnet sich für jedes Maß auf einer Rangliste ein. Diese einzelnen Ranglistenposition sind Parameter der Bewertungsfunktion, welche letztendlich für jeden Featurekandidaten einen Wert zwischen 0 und 1 berechnet.

Über diesen Wert können wir eine Rangliste der Kandidatenbegriffe erstellen. Je höher ein Begriff dort steht, desto eher erachten wir ihn als *Feature*. Mit einem Grenzwert auf dieser Liste ist eine Klassifizierung in Features und Nicht-Features möglich. Dieser Klassifizierung entsprechend können die Annotierungen der Review-Dokumente (beim Abschluss des vorherigen Schrittes, s. o.) erweitert werden. Dazu muss eine weitere Systemkomponente zur Annotation von Features gestartet werden; siehe dazu den Abschnitt zu **FeatureAnnotation** in Kap. 6.1.

In die Beurteilung, ob es sich bei einem Featurekandidaten tatsächlich um ein Produktfeature handelt, fließen mit ein:

Die absolute Häufigkeit im Korpus

Die absolute Häufigkeit gilt als genereller Indikator für die Relevanz eines Begriffs (vgl. [KU96]: Kap. 2.2.2). Um diese hier zur Verfügung zu haben, wurden im Extraktions- und Normalisierungsschritt die Häufigkeiten der Featurekandidaten mitgezählt und in der Datenbank gespeichert.

Die 20 am häufigsten auftretenden Featurekandidaten für die Beispiel-Produktklasse **Digital Cameras** bei einem Testlauf mit 5000 Review-Dokumenten sind in Tabelle 3 dargestellt.

Der PMI-Wert für Featurekandidat und Produktklasse

Über den PMI-Wert können Begriffe mit enger semantischer Beziehung (Kollokation) zur Produktklasse von allgemeineren oder irrelevanten Begriffen unterschieden werden.

Featurekandidat	**Häufigkeit**	**Featurekandidat**	**Häufigkeit**
picture	5121	mode	1455
quality	2341	price	1352
lens	2099	screen	1300
canon	2041	flash	1286
feature	2005	video	1158
image	1965	zoom	1126
battery	1894	setting	1113
photo	1845	size	1044
shot	1839	point	1034
time	1480	card	960

Tabelle 3: Die 20 häufigsten Featurekandidaten für die Produktklasse **Digital Cameras**

Dadurch soll eine höhere Bewertung möglicher Features gegenüber anderen häufig auftretenden, aber irrelevanten Begriffen sichergestellt werden.

Zur Berechnung des PMI-Wertes verwenden wir die logarithmisch wiegende Version aus [GCHH91]:[52]

$$\mathrm{PMI}(A, B) = log_2 \cdot \frac{P(A, B)}{P(A) \cdot P(B)}$$

Dabei sind A und B Featurekandidat und Produktklasse; $P(\dots)$ die Wahrscheinlichkeiten für das Auftreten der Begriffe einzeln bzw. zusammen. Der größte sich anbietende Korpus für die Berechnung der einzelnen und gemeinsamen Auftreten bzw. Wahrscheinlichkeiten ist das World Wide Web. Deshalb verwenden wir einen Webservice der Suchmaschine *Yahoo!*, um zu einem Begriff Trefferzahlen zu erhalten.[53][54] Zur Information geben wir hier die verwendete Syntax für Suchanfragen an:

- Ein Suchbegriff:
  `+"Suchbegriff"`
- Zwei Komposita als Suchbegriffe:
  `+"Suchbegriff Eins" +"Suchbegriff Zwei"`

Zur Berechnung der Wahrscheinlichkeit des Auftretens benötigen wir die Größe des Suchindexes der Yahoo!-Websuche, bezogen auf englischsprachige Dokumente. Dazu

[52] [GCHH91] geben eine Übersicht über statistische Maße zum Einsatz bei lexikalischen Analysen. Die hier verwendete Version der PMI-Berechnung wird als geeignet zur Berechnung der „genuinen Verbindung“ zwischen zwei Begriffen angegeben.

[53] Offizielle Seite des Webservice: `http://developer.yahoo.com/search/web/V1/webSearch.html`, zuletzt geprüft: 2008-06-06

[54] In seltenen Fällen ist die Trefferzahl größer als der größtmögliche Wert des Datentyps `Integer`. In diesem Fall wird der PMI-Wert auf -1 gesetzt, was bedeutet, dass der Featurekandidat als Feature verworfen wird, da wir davon ausgehen, dass es sich um ein sehr allgemeines Wort der englischen Sprache handelt, wie bspw. **more**. Weiteres zu diesem Problem in Kap. 32.

gibt es keine offiziellen Informationen; allerdings war im August 2005 in einem offiziellen Yahoo!-Blog zu lesen, dass der Index mehr als 20 Mia. Dokumente umfasse.[55] Da aber am 09.06.2008 eine Suche nach `A` schon 29,5 Mia. Ergebnisse an englischsprachigen Dokumenten lieferte, schätzen wir den momentanen Suchindex auf eine Größe von ca. 100 Mia. englischsprachigen Dokumenten. Demzufolge berechnen wir die Wahrscheinlichkeit als die Anzahl der Treffer dividiert durch 10^{11}.

Die Wirkung der PMI-Berechnung lässt sich beispielhaft an den häufig auftretenden Featurekandidaten **mode** und **zoom** erkennen:

Featurekandidat	**Häufigkeit**	**PMI-Wert**
mode	1455	1,56
zoom	1126	3,19

Tabelle 4: Vergleich von absoluter Häufigkeit und PMI-Wert für **mode** und **zoom** und die Produktklasse **Digital Cameras**

Obwohl **mode** sogar häufiger als **zoom** auftritt, erhält es einen sehr viel niedrigeren PMI-Wert, da keine hohe Kollokation zwischen **mode** und **Digital Cameras** besteht.

C-Value

Wie unter 2.4.2 gesehen, ist der C-Value für einen zusammengesetzten Begriff ein Maß dafür, ob dieser ein tatsächliches Kompositum ist oder nicht. Es erscheint interessant, dieses Maß auf Unigramm-Begriffe, also Einzelwörter, zu erweitern. Dies würde bspw. im folgenden Satz helfen, zu beurteilen, ob es sich bei der Nominalphrase **optional battery grip** um ein Feature handelt, oder ob eher ein Teil davon ein Feature ist (**optional battery**, **battery grip**, **battery** oder **grip**).

(9) It balances best with the **optional battery grip** [...] (B00008VE6L)

Bei der Berechnung für Unigramme ist allerdings eine Anpassung der Formel aus Kap. 2.4.2 erforderlich, um den Multiplikanden 0 als Resultat aus $log_2(1)$ zu vermeiden. Dazu erhöhen wir den Numerus des Logarithmus um Eins, sodass der Zweck der Logarithmierung, das Abschwächen des positiven Effekts eines Begriffs mit vielen Einzelwörtern, erhalten bleibt.

$$\text{Modified-C-value}(A) = log_2(|A| + 1) \cdot \left(f(A) - \frac{\sum_{i=1}^{n} B_i}{\text{Anzahl der } B_i} \right)$$

Anhand einer manuellen Kontrolle der C-Values für eine große Anzahl Featurekandidaten konnte die Anwendung auf Unigramme als sinnvoll bestätigt werden. Als Beispiel seien hier die zwei Featurekandidaten **shutter** und **shutter speed** genannt. Da **shutter** selbst kein Feature, aber Teilbegriff des Features **shutter speed** ist, ist der C-Value, relativ zur Gesamthäufigkeit, klein. Bei **shutter speed** ist er hingegen relativ groß, da dieser Begriff nicht Teil eines weiteren Features ist.

[55] „[...] our index now provides access to over 20 billion items“, `http://www.ysearchblog.com/archives/000172.html`, zuletzt geprüft: 2008-06-06

Featurekandidat	Häufigkeit	C-Value (gerundet)
shutter	450	386
shutter speed	94	149

Tabelle 5: Vergleich von absoluter Häufigkeit und C-Value für **shutter** und **shutter speed** und die Produktklasse **Digital Cameras**

Produktnamen

Sobald ein Featurekandidat einen Produktnamen enthält, ist dies ein Indikator dafür, dass es sich vermutlich nicht um ein Feature handelt. In der Komponente zum Aufbauen des Korpus (Kap. 3.3) werden ebenfalls alle abgefragten Produktnamen gespeichert. Ein Beispiel-Produktname ist:

(10) Sony Alpha DSLRA350K 14.2MP Digital SLR Camera with Super Steady-Shot Image Stabilization DT 18-70mm f/3.5-5.6 Zoom Lens

Ein Featurekandidat, der den Begriff **Sony** enthält, soll also durch eine Prüffunktion zum Produktnamen abgewertet werden. Dazu wird zu einer Produktklasse eine Liste aller Produktnamen erstellt. Um allerdings zu vermeiden, dass Begriffe wie **Zoom Lens** in diese Liste aufgenommen werden, annotieren wir zuvor die Produktnamen in einer weiteren UIMA-Pipeline.[56] Dabei betrachten wir ausschließlich Substantive, und von diesen nehmen wir nur solche in die Liste auf, die keinen Eintrag in WordNet besitzen, d. h. vermutlich Eigennamen sind. Beim o. a. Produktnamen führt dies zu:

(11) sony dslra350k steadyshot

In diesem Fall würden die drei Wörter **sony**, **dslra350k** und **steadyshot** einzeln in die Liste der Produktnamen aufgenommen.

Bewertungsfunktion

Die Bewertungsfunktion für Featurekandidaten kombiniert die drei Ranglistenpositionen eines bewerteten Featurekandidaten für die Maße *absolute Häufigkeit*, *PMI-Wert* und *C-Value*, jeweils auf eine Skala zwischen 0 und 1 normalisiert und zu gleichen Teilen gewichtet (jeweils 1/3). Dadurch werden die o. g. Indikatoren für Features allesamt berücksichtigt. Falls im Namen des Featurekandidaten ein Produktname auftritt, führt dies zu einer abschließenden Abwertung um 20 %.[57]

Resultat dieser Bewertungsfunktion ist, wie in Kap. 3.5 erläutert, ein Wert zwischen 0 und 1, der eine Klassifizierung in Features und Nicht-Features ermöglicht.

[56] Diese kleine, eigenständige Systemkomponente trägt den Namen *ProductNameAnnotation* und ist in Kap. 6.1 dokumentiert.

[57] Für den Fall, das die Feature-Extraktions-Ergebnisse manuell nachgeprüft werden sollen, wird ein Untersuchen der Begriffe mit Bewertungen kleiner als $0,8$ empfohlen (dies ist der maximale Wert für Begriffe, die Produktnamen enthalten), um eventuelle Fehler mit falsch erkannte Produktnamen zu erkennen.

4 Evaluation

4.1 Vorgehensweise

Wir wählen zur Evaluation drei verschiedene Produktklassen aus. Für jede Produktklasse wird eine große Anzahl von Review-Dokumenten von Amazon.com gespeichert. Über die folgenden beiden Komponenten extrahieren wir aus möglichst vielen dieser Dokumente alle Featurekandidaten und führen eine Evaluation durch, um eine Liste von möglichen Features zu erhalten.[58] Für jede Produktklasse setzen wir einen Schwellenwert fest, den wir zur Klassifizierung in Features und Nicht-Features nutzen (zu dieser Vorgehensweise siehe Kap. 3.5).[59]

Für die auf diese Art gewonnenen Features lassen sich die allgemeinen Gütemaße *Recall* und *Precision* bzw. *F-Maß* (siehe Kap. 2.5.1) berechnen. Dabei substrahieren wir für die Anzahl der gefundenen Features als Nenner der *Precision*-Berechnung die Anzahl nicht automatisch gefundener Synonymbezeichnungen von der Größe der Feature-Liste, um diese nicht mitzuzählen.

Die Ergebnisse der Extraktion, also die Listen der Features zu jeder Produktklasse, finden sich im Anhang (Kap. 6.3).

Zur Berechung der o. g. Gütemaße müssen allerdings die Anzahl und der Umfang der tatsächlichen Features bekannt sein. Dafür lassen wir durch drei Personen Listen erstellen, die aus deren Sicht (es sind keine Domänenexperten) die wesentlichen Features zu der jeweiligen Produktklasse umfassen. Dazu haben diese Personen teilweise externe Quellen verwendet, wie z. B. Webseiten von Herstellern von Produkten oder zu Produkt-Reviews.

Für jede Produktklasse erstellen wir aus diesen drei Listen eine „aggregierte Liste“ der Features mit hoher Übereinstimmung (mehr als zweimal genannt) und fügen in Einzelfällen auch offensichtliche Features hinzu, falls diese weniger als zweimal genannt wurden. Diese „aggregierten“ Listen mit „tatsächlichen Features“ für jede Produktklasse werden ebenfalls im Anhang (Kap. 6.4) zur Verfügung gestellt, zusammen mit Verweisen auf die Webseiten, die zum Finden der tatsächlichen Features verwendet wurden.

In einem zweiten Evaluationsansatz lassen wir stichprobenhaft für drei Review-Dokumente jeder Produktklasse die Features manuell durch eine Person annotieren und vergleichen diese Ergebnisse mit denen der automatischen Annotation, auch hier auf der Grundlage der Maße Recall/Precision bzw. F-Maß.

Anschließend wollen wir das Extraktionsverfahren auf die Plausibilität des Bewertungsparameters aus Kap. 3.5 hin prüfen. Dieser dient zur Beurteilung, ob Begriffe, die Teile längerer Komposita sind, als Features in Frage kommen. Dazu markieren wir aus der Liste der extrahierten Features (Kap. 6.3) solche, die fälschlicherweise als Feature identifiziert wurden.

[58] Genaue Zahlen zum Umfang des bewerteten Materials stehen in den Tabellen zur Geschwindigkeit ab Seite 29.

[59] Die Schwellenwerte entsprechen Positionen von Featurekandidaten in dieser Liste. Wir legen sie nach Augenmaß fest; alle Featurekandidaten, die in der Liste darüber stehen, werden als Features betrachtet.

Als letztes Evaluationskriterium messen wir die benötigten Laufzeiten der einzelnen Komponenten. Diese sind auch abhängig von den Hardware-Voraussetzungen, die mit angegeben werden.

4.2 Ergebnisse

4.2.1 Vergleich mit extern erstellten Listen für jede Produktklasse

Die Ergebnisse des Vergleichs zwischen den von drei Personen manuell erstellten Listen von „tatsächlichen Features“ und den automatisch extrahierten Features werden in Tabelle 6 angegeben.

Produktklasse	**Recall-Wert**	**Precision-Wert**	**F-Maß**
Digital Cameras	$31/50 = 0,62$	$31/56 = 0,55$	$0,58$
Cell Phones	$30/54 = 0,56$	$30/53 = 0,57$	$0,56$
Folding Knives	$16/32 = 0,5$	$16/24 = 0,67$	$0,57$

Tabelle 6: Ergebnisse für Recall, Precision und F-Maß beim Vergleich der automatischen Feature-Extraktion mit extern erstellten Listen für jede Produktklasse

4.2.2 Vergleich mit manueller Annotation von Review-Dokumenten

Die Ergebnisse für den Vergleich der manuellen und der automatischen Annotation von Features für drei Review-Dokumente pro Produktklasse werden in Tabelle 7 (absolute Werte) und Tabelle 8 (Recall, Precision und F-Maß) angegeben. Dabei wurde ein Schwellenwert von **0.8** für die Klassifikation von Features und Nicht-Features festgelegt (Erläuterungen zur Klassifikation und zum Schwellenwert in Kap. 3.5).

Legende zu Tabelle 7:
korr. = Anzahl korrekt identifizierte Features
falsch = Anzahl falsch identifizierte Features
tats. = Anzahl tatsächliche Features

Review-Dokument	**korr.**	**falsch**	**tats.**
Digital Cameras: erstes Review	12	6	18
Digital Cameras: zweites Review	5	1	10
Digital Cameras: drittes Review	7	1	9
$\sum$ **Digital Cameras**	24	8	37
Cell Phones: erstes Review	8	7	19
Cell Phones: zweites Review	7	5	22
Cell Phones: drittes Review	5	8	1
$\sum$ **Cell Phones**	20	20	42
Folding Knives: erstes Review	0	2	6
Folding Knives: zweites Review	2	1	5
Folding Knives: drittes Review	4	2	4
$\sum$ **Folding Knives**	6	7	15

Tabelle 7: Absolute Ergebnisse für den Vergleich der manuellen und der automatischen Annotation von Features für drei Review-Dokumente pro Produktklasse

Produktklasse	**Recall-Wert**	**Precision-Wert**	**F-Maß**
Digital Cameras	$24/37 = 0,65$	$24/(24+8) = 0,75$	$0,7$
Cell Phones	$20/42 = 0,48$	$20/(20+20) = 0,5$	$0,49$
Folding Knives	$6/15 = 0,4$	$6/(6+7) = 0,46$	$0,43$

Tabelle 8: Ergebnisse für Recall, Precision und F-Maß für den Vergleich der manuellen und der automatischen Annotation von Features für drei Review-Dokumente pro Produktklasse

4.2.3 Prüfung von Teilen von Komposita

In Tabelle 9 sind die Ergebnisse der oben erläuterten Plausibilitätsprüfung angegeben, ob fälschlicherweise Features identifiziert wurden, die Teile längerer Komposita sind. Zur Illustration: Bei der Produktklasse **Folding Knives** wurde **opener** identifziert, obwohl es nur Teil von **bottle opener** oder **can opener** ist, aber kein eigenständiges Feature.

Legende zu Tabelle 9:
extr. = Anzahl extrahierte Features
falsch = Anzahl fälschlicherweise extrahierte Features, die Teil längerer Komposita sind

Produktklasse	**extr.**	**falsch (Komposita)**
Digital Cameras	75	3
Cell Phones	65	1
Folding Knives	26	1

Tabelle 9: Ergebnisse für fälschlicherweise extrahierte Features, die Teil längerer Komposita sind

4.2.4 Geschwindigkeit

Die Laufzeit für jede Komponente wurde auf einem System mit 512 MB RAM und 1-GHz-CPU (AMD Sempron) unter dem Betriebssystem Kubuntu Linux 7.10 getestet. Dabei wurden die Komponenten jeweils mit der Option `java -Xms32m -Xmx512m` aufgerufen, die der Java Virtual Engine mehr Arbeitsspeicher zuweist, sodass teilweise die swap-Partition in Anspruch genommen wurde.

Produktklasse	**# Produkte**	**# Reviews**	**Zeit**
Digital Cameras	280	19992	66 min.
Cell Phones	470	6345	21 min.
Folding Knives	3991	974	32 min.

Tabelle 10: Geschwindigkeit `amazonToDatabase`

Produktklasse	**# Reviews**	**# Featurekandidaten**	**Zeit**
Digital Cameras	5000	22396	142 min.
Cell Phones	4022	32918	108 min.
Folding Knives	974	5860	14 min.

Tabelle 11: Geschwindigkeit `featureCandidateExtraction`

Produktklasse	**# evaluierte Featurekandidaten**	**Zeit**
Digital Cameras	1000	32 min.
Cell Phones	1000	24 min.
Folding Knives	1000	38 min.

Tabelle 12: Geschwindigkeit `featureCandidateEvaluation`

4.3 Auswertung und Vergleich zu existierenden Verfahren

Die Ergebnisse bei den Extraktionsgütemaßen *Recall* und *Precision* bzw. *F-Maß* ordnen sich im Vergleich zu anderen Feature-Extraktionsverfahren relativ niedrig ein. So messen [HL04] bspw. einen Recall von durchschnittlich $0,71$ und eine Precision von durchschnittlich $0,72$; [PE05] geben sogar eine Precision von $0,94$ und einen Recall von $0,77$ an.

Wir analysieren die Ergebnisse zu beiden Maßen für unser Feature-Extraktionssystem einzeln.

4.3.1 Precision

Bei der Beurteilung des Precision-Wertes in Tabelle 6 ist zu berücksichtigen, dass in den von Dritten erstellten Listen der sog. „tatsächlichen Features“ (Kap. 6.4) eine große Anzahl von Begriffen vorkommt, die in den Review-Dokumenten überhaupt nicht oder

nur selten auftreten. Solche Features werden bei unserem Verfahren schlechter bewertet bzw. überhaupt nicht gefunden.[60]

Außerdem stellt die Granularität von Features einen wesentlichen Einflussfaktor für die Precision dar. Beispielsweise ist unklar, ob die folgenden Begriffe als eigene tatsächliche Features betrachtet oder, als feinere Ausprägungen bestehender Features, unter diese subsummiert werden sollen:

size / height / width / dimensions / thickness

So haben wir beim Zählen für die o. a. Resultate genau auf diese Granularitäten geachtet. Beispielsweise wurde der Begriff *display* als falsch annotiertes Feature gezählt, obwohl *display size* an dieser Textstelle auftauchte; ebenso mit *battery* und *battery type*. Eine weniger strikte Zählweise würde hierbei den Precision-Wert erhöhen.

Bei vergleichbaren Ansätzen wird dieses Problem nicht erwähnt – nur [Liu06] meint zum Problem der Granularität, dieses könne starke Auswirkungen auf die Qualität des Feature-basierten Opinion Mining haben. Auch die Listen der Feature-Begriffe zur Berechnung von Recall und Precision werden bei vergleichbaren Verfahren selten zur Verfügung, sodass die Qualität der Feature-Extraktion dort nicht direkt nachvollzogen werden kann.

4.3.2 Recall

Der niedrige Recall-Wert resultiert, nach Beobachtungen, zu einem Teil aus nicht eliminierten Produktnamen. Dabei ist bei Tests auf einem größeren Korpus ein besserer Wert zu erwarten, da damit die Anzahl der Produktnamen steigt. Zur Einordnung: In diesem Kapitel lagen den Tests bei der Produktklasse **Digital Cameras** knapp 20.000 Reviews zu 280 Produkten zugrunde; für die Extraktion wurden 5.000 Reviews verwendet. Bei Amazon.com sind aber ca. 7.500 Produkte mit insgesamt mehr als 60.000 Reviews verfügbar.

Weitere Tests auf größeren Korpora und mit mehr Extraktionen könnten die Ergebnisse also noch verändern.

Ein weiterer Punkt, der zum hier gemessenen niedrigen Recall-Wert beiträgt, ist das Auftreten von Features in impliziter Form (vgl. Kap. 2.3.3), die in unserem Verfahren nicht erkannt werden. Im folgenden Beispiel aus der Produktklasse **Folding Knives** (Klappmesser) wurde keins der Features **red** (implizit **color**), **shiny** (implizit **surface**), **new** (implizit **age**) und **stocked** (implizit **versatility**) extrahiert:

(12) It's a very nice knife, all **red** and **shiny** and **new**. Also, very well **stocked**. (B0009KF4GG)

Dass auch beim Vergleich der automatischen mit manuellen Annotationen kein wesentlich höherer Recall erreicht werden konnte, liegt vor allem an der mangelhaften Zusammenführung von Variationen (siehe Kap. 2.3.2: alternative Schreibweisen). So wird im folgenden Beispiel aus der Produktklasse **Cell Phones** der Begriff **Wi-Fi** nicht als Feature erkannt, obwohl **wifi** mit einer Bewertung von $0,9$ in der Liste der Features (alle Begriffe oberhalb $0,8$) auftaucht.

[60] vgl. Kap. 2.4.2: *infrequent features* werden benachteiligt, sowie Kap. 3.5 zur Bewertung über absolute Häufigkeit

(13) In the end I sent the device back because it's **Wi-Fi** module failed. (B000QTWT7W)

Über eine Verbesserung des Normalisierungskomponente (Kap. 3.4.3) wäre vermutlich ein höherer Recall möglich.

4.3.3 Modifizierter C-Value

Die Prüfungen, wieviele Features fälschlicherweise als solche markiert wurden, obwohl es sich eindeutig um Teile längerer Komposita handelt und nicht um eigenständige Features, zeigt zufriedenstellende Ergebnisse: Bei den drei getesteten Produktklassen kam dies nur in geringem Maße vor (4 %, 1,5 % und 3,8 %).

Anhand dieser Stichproben können wir für das in Kap. 3.5 vorgestellte Maß, den modifizierten C-Value als Erweiterung auf Unigramm-Wörter, eine vorläufige Beurteilung vornehmen: Es ist offensichtlich vorerst kein gegenteiliger Effekt feststellbar, und die manuelle Kontrolle der C-Value-Werte für einzelne Begriffe bestätigt dabei den gewünschten Effekt der Berechnung, dass Begriffen mit geringer „termhood“[61] ein ebenfalls geringer Wert zugewiesen wird. Es sind allerdings noch weitere Tests und theoretische Begründungen für dieses Maß notwendig.

4.3.4 Geschwindigkeit

Eine Einordnung der gemessenen Werte für die Laufzeit jeder Komponente ist schwierig, da es an Vergleichswerten fehlt. Außerdem ist die Laufzeit stark abhängig vom System und von externen Werkzeugen. So ist für die UIMA-Komponente **featureCandidateExtraction** zu beachten, dass ein Großteil der Laufzeit auf die NLP-Annotatoren entfällt, die von Dritten entwickelt wurden (vgl. Kap. 3.4.1). Des Weiteren läuft die Komponente **featureCandidateEvaluation** läuft wesentlich schneller, sobald die Ergebnisse der Yahoo!-Suchanfragen in der Datenbank gepuffert sind (vgl. Kap. 3.5). Zudem wurde in den o. a. Tests häufig die swap-Partition des Testsystems in Anspruch genommen, wodurch Programmausführungen um ein Vielfaches verlangsamt wurden.

[61] Begriff von [FAM00], vgl. Kap. 2.4.2: Komposita (S. 11)

5 Zusammenfassung und Ausblick

Wir haben verschiedene Ansätze zur Feature-Extraktion im Rahmen von Product Review Mining vorgestellt.

Als wesentliche Probleme stellen sich dabei das Finden von möglichen Features („Featurekandidaten“), das Auflösen von Koreferenzen und Variationen sowie die Beurteilung, ob ein Featurekandidat ein Feature ist, dar.

Es wurden linguistische und statistische Methoden untersucht, die diese speziellen Herausforderungen lösen können. Von diesen wurden mehrere Vorgehensweisen als Basis für ein eigenes Verfahren ausgewählt. Dazu zählt eine linguistische Analyse der Review-Dokumente auf Grundlage bestimmter Muster. In diversen Normalisierungsschritte werden einheitliche Formen gefunden und als irrelevant erachtete Featurekandidaten ausgeschlossen. In einem Bewertungsschritt werden mehrere Parameter, die Indikatoren für Features repräsentieren, zu einer Bewertungsfunktion kombiniert. Zu diesen zählen der PMI-Wert zum Bestimmen der Kollokation zwischen einem Featurekandidaten und der Produktklasse sowie eine modifizierte Version des C-Value, über den festgestellt werden soll, ob ein Featurekandidat ein eigenständiges Feature oder nur Teil eines ihn umfassenden Kompositums ist, welches ein Feature darstellt.

Teil des ursprünglichen Ansatzes war, ähnlich wie beim Verfahren von [CNZ05], externe Listen von Produktfeatures heranzuziehen und in die Extraktion mit einfließen zu lassen. Dazu wurden bei der Komponente zum Aufbauen des Korpus (Kap. 3.3) von Amazon.com mitgelieferte Produktattribute gespeichert und verarbeitet. Diese Listen (jeweils pro Produktklasse) stellten sich jedoch als unzuverlässig für die Indikation von Features heraus, da sie zuwenige tatsächliche Attribute zur Produktklasse enthielten. Deshalb wurde dieser Ansatz wieder verworfen.

Viele neue Ideen, wie das Verwenden von Produktnamen zum Ausschluss von Featurekandidaten oder das Prüfen auf wertende Adjektive, entstanden während der Entwicklung des Systems.

In Form einer prototypischen Umsetzung leistet das vorgestellte System eine akzeptable Feature-Extraktion. Die Funktionsweise wurde detailliert vorgestellt und begründet. Die Programmkomponenten des Systems werden zusammen mit dieser Arbeit zur Verfügung gestellt. Sie sind wie im Anhang (Kap. 6.1) dokumentiert einfach verwendbar und der Quellcode kann weiter verwendet werden. Ich hoffe, dass das System im weiteren Verlauf des im Aufbau befindlichen Projekts zum Product Review Mining von Nutzen ist.

5.1 Offene Punkte

Die unter 2.3 beschriebenen Herausforderungen werden nicht alle berücksichtigt. So ist eine Erweiterung der Extraktion auf implizite Features wünschenswert, wobei unklar ist, über welche Mittel diese eindeutig auf explizite Features abgebildet werden können. Auch das Auflösen von Koreferenzen kann noch verbessert werden, z. B. über erweiterte Synonymerkennung ähnlich wie bei [CNZ05] oder Erkennen von Ähnlichkeiten bei alternativen Schreibweisen.

Momentan erfolgt die Ausgabe der Features in Listenform. Die Anordnung der Features in einer hierarchischen Struktur würde einen Mehrwert bei der Ausgabe darstellen – dies

ist allerdings nicht ohne Weiteres zu erreichen, da unklar ist, wie erkannt werden soll, in welche Kategorie der Beziehung[62] zum Produkt ein Feature fällt.

Um die Ergebnisse der Evaluation zu festigen, wäre ein weiterer Evaluationslauf mit einer größeren Datenbasis und mit weiteren Personen, die Listen von „tatsächlichen Features“ erstellen sowie Review-Dokumente manuell annotieren, sehr hilfreich.

Ein kritischer offener Punkt ist die Tatsache, dass in der Komponente *Bewertung und Klassifizierung* (Kap. 3.5) für die Zahl der Gesamtsuchergebnisse über den verwendeten Webservice von Yahoo! der Datentyp `Integer` verwendet wird.[63] Für Begriffe, bei denen die Anzahl der Suchergebnisse größer als der größtmögliche Integer-Wert $(2^{31} - 1)$ ist, kann die PMI-Berechnung aus Kap. 3.5 nicht korrekt durchgeführt werden. Daher wird momentan in diesen (selten auftretenden Fällen) der PMI-Wert -1 gesetzt. Generell ist jedoch eine andere Vorgehensweise wünschenswert, das korrekte Suchergebniszahlen liefert.

Zuletzt müssen die Bewertungsfunktion und deren Parameter aus Kap. 3.5 kritisch hinterfragt werden. Die Gewichtung der drei einfließenden Maße zu jeweils 1/3 und der Abzug von 20 % des Wertes bei Auftreten eines Produktnamens basiert allein auf heuristischen Annahmen über mögliche Indikatoren, ob ein Begriff ein Feature ist. Beispielsweise muss überlegt werden, ob absolute Häufigkeit und C-Value geringer gewichtet werden, da erstere als Faktor in den C-Value eingeht, oder die absolute Häufigkeit sogar komplett vernachlässigt werden sollte.

[62] Teil, Eigenschaft, Eigenschaft eines Teils etc.

[63] Quelle: Schemadatei für die Ergebnisse des Webservice, abgerufen von: `http://search.yahooapis.com/WebSearchService/V1/WebSearchResponse.xsd`, 2008-06-09

6 Anhang

6.1 Benutzung des Feature-Extraktionssystems

Voraussetzungen

Folgende Systemvariable muss gesetzt werden: `OM_HOME=/my/path/`

Dabei muss `/my/path/` auf das Verzeichnis verweisen, in dem die folgenden Unterverzeichnisse vorhanden sind:

- `/my/path/desc/` beinhaltet die UIMA-Descriptor-Dateien für die Komponenten `FeatureCandidateExtraction` und `ProductNameAnnotation`.
- `/my/path/dist/` beinhaltet die zum Starten jeder Komponente erforderlichen Dateien (jar-Datei und Bibliotheken).
- `/my/path/resources/` enthält benötigte externe Ressourcen, wie WordNet- oder NLP-Parser-Modelldateien.
- `/my/path/conf/` enthält Konfigurationsdateien wie SQL-Skripte und ini-Dateien für die Komponenten `AmazonToDatabase` und `FeatureCandidateEvaluation`.

Vor dem Aufruf der ersten Komponente muss eine MySQL-Datenbank auf Grundlage des SQL-Skripts `sqlCreate.sql` angelegt werden. Dazu muss ein Datenbankbenutzer mit Lese- und Schreibrechten angelegt werden.

Logging

Die Ausgabe des Systems zu Informationen, Fehlern und Warnungen wird über das Loggingsystem *log4j* gesteuert.[64] Dazu muss die Datei `log4j.xml` im Verzeichnis `conf` bearbeitet werden. Dort lassen sich u. a. die Ausgabeart einstellen (z. B. Konsole oder Datei) sowie das Logging-Level (z. B. `FATAL`, `ERROR`, `WARN`, `INFO`, `DEBUG` oder `TRACE`).

Starten der einzelnen Komponenten

Das System besteht aus vier einzelnen Komponenten, die separat aufrufbar sind, aber deren Aufruf teilweise von vorhergehenden Komponenten abhängt. Das Ablaufdiagramm in Abbildung 6 zeigt die möglichen Reihenfolgen für den Aufruf aller Komponenten auf; die Pfeile stellen dabei die Voraussetzungen dar.

Jede einzelne Komponente wird als jar-Archiv bereitgestellt und kann über die Java Virtual Machine mit folgendem Aufruf gestartet werden:

```
java -Xms32m -Xmx512m -jar $OM_HOME/dist/<Komponentenname>.jar
```

Dabei wird der Java Virtual Engine mehr Arbeitsspeicher zugewiesen, um ressourcenintensive Vorgänge ohne Fehler ausführen zu können.

[64] Details zum Loggingsystem *log4j* unter `http://logging.apache.org/log4j/` (zuletzt geprüft: 2008-06-05)

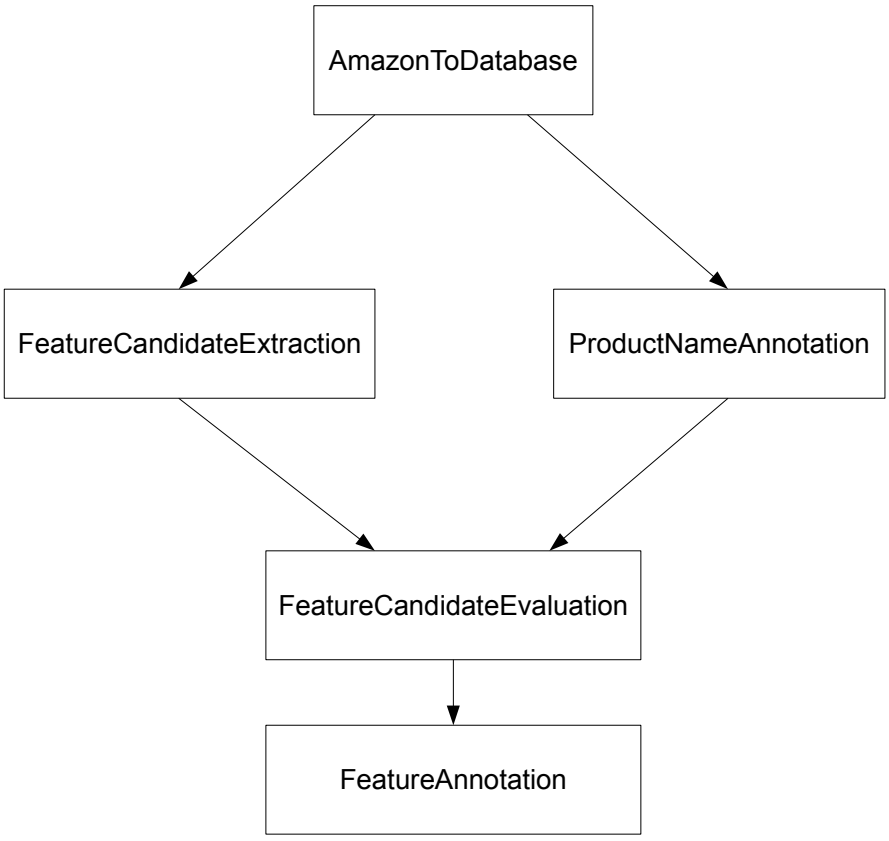

Abbildung 6: mögliche Abfolgen für den Aufruf der Komponenten

AmazonToDatabase

In der Datei `amazonToDatabase.ini` sind folgende Parameter anzugeben:

- **JDBC_URL** – URL für die Datenbank-Verbindung. Beispiel: `jdbc:mysql://localhost/my_db?user=me&password=pw`
- **ACCESS_KEY_ID** – Amazon-Benutzer-Lizenzschlüssel, der bei jedem Aufruf des Webservice angegeben werden muss.
- **BROWSE_NODE_ID** – Amazon-ID einer Produktklasse. Beispiel: `281052` für Digitalkameras
- **PRODUCT_CLASS_NAME** – Bezeichnung der Produktklasse für die Speicherung in der Datenbank. Beispiel: `Digital Cameras`
- **SEARCH_TITLE** – Suchbegriff. Beispiel: `""` (leerer String) bewirkt, dass sämtliche Produkte einer Produktklasse beim Abrufen der Review-Texte berücksichtigt werden.

FeatureCandidateExtraction

In `FeatureExtractionCollectionProcessingEngine.xml` (UIMA-Descriptor-Datei) sind folgende Parameter anzugeben:

- **JDBCUrl** – Datenbank-URL, muss für den MySQLCollectionReader und den FeatureCandidateConsumer angegeben werden. Beispiel: `jdbc:mysql://localhost/my_db?user=me&password=pw`
- **ProductClass** – Bezeichnung der Produktklasse. Dies ist für den MySQLCollectionReader, den FeatureCandidateAnnotator und den FeatureCandidateConsumer einzeln anzugeben. Beispiel: `Digital Cameras`
- **LowerLimit** und **UpperLimit** – Untere und obere Grenze für zu verarbeitende Review-Dokumente in der Datenbank. Beispiele: `100` und `102`
- **NonAnnotated** – Einstellung, ob bereits annotierte Dokumente ignoriert werden sollen. Beispiel: `true` bewirkt, dass nur Dokumente verarbeitet werden, die noch nicht annotiert sind.
- **...ModelFile** – mehrere Parameter für die Verzeichnispfade zu den Modelldateien der NLP-Werkzeuge. Beispiel: `resources/opennlp_models/sentdetect/EnglishSD.bin.gz` (für die Modelldateien des Satzgrenzen-Annotators)
- **stopwordsFile** – Pfad zur Datei mit Stoppwörtern. Dies muss für den FeatureCandidateAnnotator sowie den FeatureCandidateNormalizer angegeben werden (vgl. Kap. 3.4.2 und 3.4.3). Beispiel: `resources/stopwords.txt`

ProductNameAnnotation

In `ProductNameAnnotationCollectionProcessingEngine.xml` (UIMA-Descriptor-Datei) sind folgende Parameter anzugeben:

- **JDBCUrl** – Datenbank-URL, muss für den MySQLCollectionReader und den ProductNameAnnotationConsumer angegeben werden. Beispiel: `jdbc:mysql://localhost/my_db?user=me&password=pw`
- **ProductClass** – Bezeichnung der Produktklasse. Dies ist ebenfalls für den MySQLCollectionReader und den ProductNameAnnotationConsumer einzeln anzugeben. Beispiel: `Digital Cameras`
- **...ModelFile** – mehrere Parameter für die Verzeichnispfade zu den Modelldateien der NLP-Werkzeuge. Beispiel: `resources/opennlp_models/sentdetect/EnglishSD.bin.gz` (für die Modelldateien des Satzgrenzen-Annotators)

FeatureCandidateEvaluation

In der Datei `featureCandidateEvaluation.ini` sind folgende Parameter anzugeben:

- **JDBC_URL** – URL für die Datenbank-Verbindung. Beispiel: `jdbc:mysql://localhost/my_db?user=me&password=pw`
- **PRODUCT_CLASS_NAME** – Bezeichnung der Produktklasse. Beispiel: `Digital Cameras`
- **LIMIT** – Anzahl der zu evaluierenden Featurekandidaten. Beispiel: `1000`

FeatureAnnotation

In `FeatureAnnotationCollectionProcessingEngine.xml` (UIMA-Descriptor-Datei) sind folgende Parameter anzugeben:

- **JDBCUrl** – Datenbank-URL, muss für den MySQLCollectionReader und den FeatureAnnotationConsumer angegeben werden. Beispiel: `jdbc:mysql://localhost/my_db?user=me&password=pw`
- **ProductClass** – Bezeichnung der Produktklasse. Dies ist ebenfalls für den MySQLCollectionReader und den FeatureConsumer einzeln anzugeben. Beispiel: `Digital Cameras`
- **Limit** – Grenze für die Anzahl der zu annotierenden Dokumente. Beispiel: `500`
- **Threshold** – Grenzwert der Bewertung eines Featurekandidaten. Nur Featurekandidaten oberhalb dieses Grenzwertes werden als Features annotiert. Beispiel: `8` (Anmerkung: Vorerst müssen Integer-Werte angegeben werden, die als Double-Werte interpretiert werden. Ein Wert `8` bedeutet einen Grenzwert von `0,8`.)

6.2 Datenbankschema für das Feature-Extraktionssystem

Zu Informationszwecken wird hier eine visuelle Darstellung des verwendeten Datenbankschemas bereitgestellt. Hinweis: Die Abbildung ist ein Screenshot aus dem Programm *MySQL Workbench*.

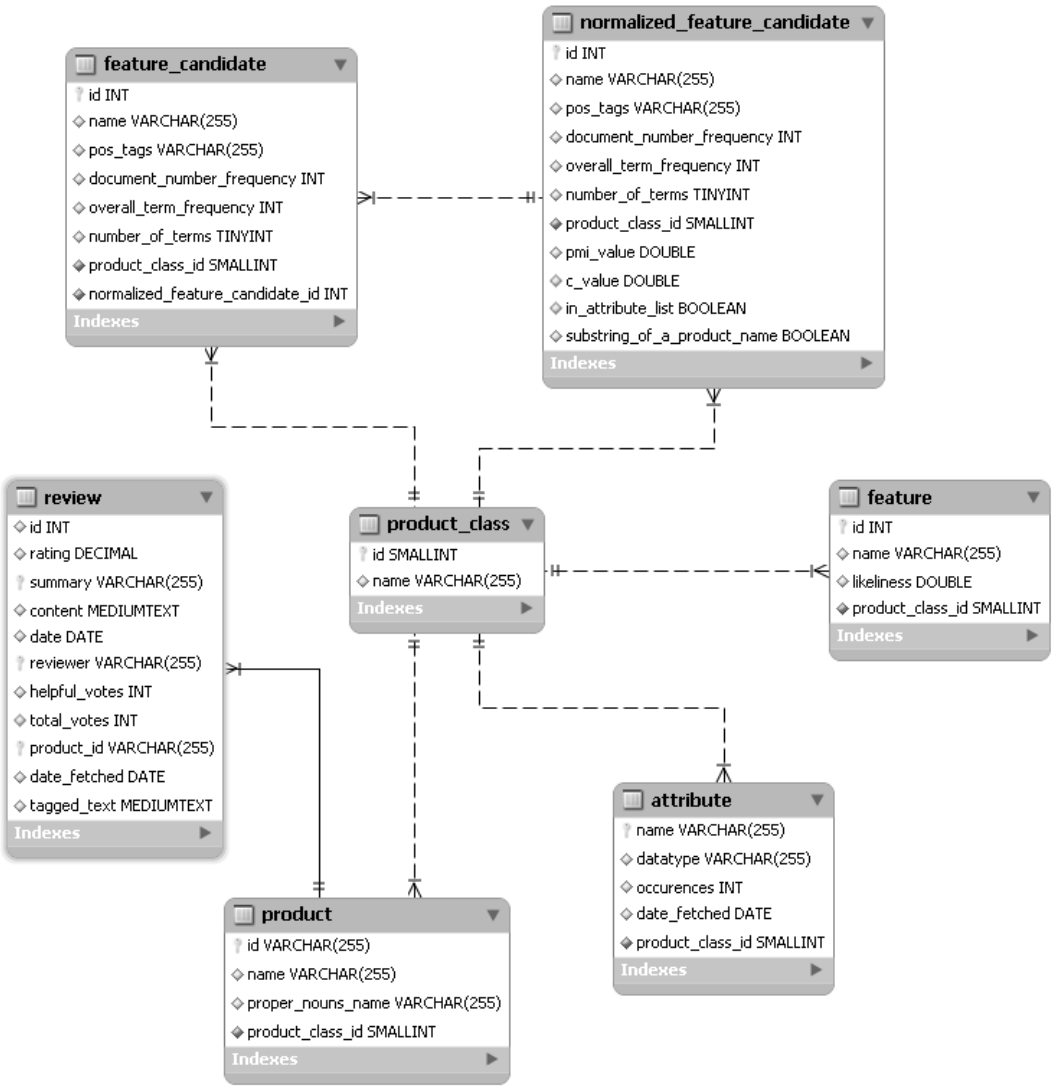

Abbildung 7: Datenbankschema

6.3 Liste der extrahierten Features

Abbildung 8 zeigt die aus den Review-Dokumenten extrahierten Features für die Produktklassen **Digital Cameras**, **Cell Phones** und **Folding Knives**. Auf Grundlage dieser Extraktionen wurde das Feature-Extraktion-System in Kap. 4 bewertet.

Digital Cameras			**Cell Phones**		**Folding Knives**
image stabilization	zoom	inch	Mp3 player	screen	knife
SD1000	pocket	format	battery life	touch screen	blade
optical zoom	rebel xt	feature	amazon	speaker	swiss army knife
slr	auto mode	zoom lens	camera	player	scissors
dslr	shutter speed	price range	mp3	pc	pocket
viewfinder	xt	lens cap	blackberry	laptop	toothpick
camcorder	sensor	video quality	headset	digital camera	screwdriver
point and shoot	G9	iso	battery	color	sheath
optical viewfinder	autofocus	shooting mode	call quality	pda	bottle opener
lcd	white balance		sim card	texting	pocket knife
megapixels	face detection		keypad	ring tone	pliers
canon	stabilization		song	sprint	nail file
image quality	screen		Samsung	data plan	key chain
picture quality	flash		ringtones	blackberry pearl	handle
battery life	compact		card	cable	army
lens	SD870		memory card	text messaging	opener
SD750	extra battery		charger	earpiece	lock
amazon	image stabilizer		device	text message	clip
lcd screen	iso setting		sd card	sound	large blade
kit lens	manual control		memory	feature	steel
shutter	manual mode		palm	devices	pen
memory card	card		pocket	extended battery	pocket clip
movie mode	aperture		music player	fm radio	pouch
point-and-shoot	shoot		ipod	pro	grip
battery	aa battery		gps	talk time	sharp edge
wide angle lens	SLRs		windows	warranty	corkscrew
memory	pixel		keyboard	trackball	
canon SD1000	speed		AT&T	iTunes	
sd card	point shoot		dropped call	replacement	
color	photography		reviewer	antenna	
shutter lag	shutter button		wifi	interface	
rebel	angle		adapter	alarm	
Elph	live view		ringtone		

Abbildung 8: Liste der extrahierten Features

6.4 Liste der tatsächlichen Features

Abbildung 9 zeigt die tatsächlichen Features für die Produktklassen **Digital Cameras**, **Cell Phones** und **Folding Knives**. Diese wurden unter Mithilfe von weiteren Personen und unter Verwendung externer Quellen [Dig08, CNE08, Wik08a, Pho08, GSM08, Wik08c, Wik08b, Vic08] erstellt und finden Verwendung bei der Evaluation (Kap. 4).

Digital Cameras	Cell Phones	Folding Knives
accessory	adapter	belt
aperture	alarm	blade
battery / aa battery / extra battery	antenna	blade arresting
battery capacity / battery life	availability	blade material
battery type	battery	blade size
button / shutter button	battery life	bottle opener
case / robustness	bluetooth	can opener
color / color support / color depth	cable	clip / pocket clip
connections / usb / pc	call quality / voice quality	corkscrew
digital zoom	camera / digital camera	design / decorations / format / model
display / lcd / lcd monitor	camera resolution	file / nailfile
display resolution / lcd resolution / lcd monitor resolution	charger	grip / handle
display size / lcd size / lcd monitor size	color	hook
expansion slot / memory slot	design	key chain
exposure / exposure time	device type / palm / blackberry / pda / personal digital assistant	lock
face detection	display / screen / lcd	magnifying glass
flash / external flash	display colors / screen solors / lcd colors	material / steel
flash memory / flash memory card	display resolution / screen resolution / lcd resolution	pen / ballpoint pen
focus / manual focus / auto focus	ergonomics	pliers
format / file format	form factor / slider / bar / flip	pouch
image quality / picture quality / photo quality	games	price
lens / wide angle lens / macro lens / telephoto lens / tele lens / long-focus lens / zoom lens	gps	saw
lens cap	handsfree	scissors
light sensitivity / iso / iso settings	headset / headphones / earpiece	screwdriver / phillips-head screwdriver
manufacturer	infrared port	sheath
material	internet / wap / gprs	size / height / width / dimensions / thickness
memory / storage / internal memory	java	stud / thumb stud / nail-nick / slot
memory card / card / sd card	keys / keyboard / keypad	toothpick
modes / shooting modes / auto mode / manual mode	memory / memory types / memory card slot	tweezers
optical zoom	memory card / card / sd card	versatility / functions / tools
point and shoot / point and shoot camera / point-and-shoot / point-and-shoot camera	mms	warranty
price / price range	mp3 player / jukebox / media player / music player	weight
red eye reduction	network bands / multimode	
resolution / pixels / megapixels	network providers / data plan	
self timer / delayed shuttering	operating system / windows / palm os	
sensor	price	
shutter	radio / fm radio	
shutter speed / shutter time / shutter lag	ringtones	
size / height / width / dimensions / thickness	sim card	
slr camera / single lens reflex camera dslr / digital single lens reflex camera	size / height / width / dimensions / thickness	
software	sms / texting / text messaging / text message	
stabilization / image stabilization / image stabilizer	software	
usability / ergonomics	sound	
video / movie / video quality / movie quality	speaker	
video mode / movie mode	standby time	
viewfinder / optical viewfinder / electronic viewfinder	talk time	
warranty	touchscreen	
weight	trackball	
white balance	usability	
zoom	usb	
	video / video recorder	
	wallpapers	
	warranty	
	weight	
	wlan / wi-fi	

Abbildung 9: Liste der tatsächlichen Features

6.5 Liste der Penn-Treebank-POS-Tags

Zu Informationszwecken folgt hier eine Liste der Part-of-speech-Tags (Markierungen für Wortarten) des Penn-Treebank-Projekts des *Computer and Information Science Department* der *University of Pennsylvania*.[65] Diese Syntax findet Verwendung im vorgestellten Feature-Extraktionssystem dieser Arbeit.

Tag	**Description**	**Tag**	**Description**
CC	Coordinating conjunction	PRP$	Possessive pronoun
CD	Cardinal number	RB	Adverb
DT	Determiner	RBR	Adverb, comparative
EX	Existential there	RBS	Adverb, superlative
FW	Foreign word	RP	Particle
IN	Preposition or subordinating conjunction	SYM	Symbol
JJ	Adjective	TO	to
JJR	Adjective, comparative	UH	Interjection
JJS	Adjective, superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBG	Verb, gerund or present participle
NN	Noun, singular or mass	VBN	Verb, past participle
NNS	Noun, plural	VBP	Verb, non-3rd person singular present
NNP	Proper noun, singular	VBZ	Verb, 3rd person singular present
NNPS	Proper noun, plural	WDT	Wh-determiner
PDT	Predeterminer	WP	Wh-pronoun
POS	Possessive ending	WP$	Possessive wh-pronoun
PRP	Personal pronoun	WRB	Wh-adverb

Tabelle 13: Penn-Treebank-POS-Tags

[65] abgerufen am 02.06.2008 von `http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`

Literatur

[Apa08a] Apache UIMA: *UIMA Overview & SDK Setup*. Version: 2008. http://incubator.apache.org/uima/downloads/releaseDocs/2.2.2-incubating/docs/html/overview_and_setup/overview_and_setup.html. – Elektronische Ressource, zuletzt geprüft: 2008-06-05

[Apa08b] Apache UIMA: *What is UIMA?* Version: 2008. http://incubator.apache.org/uima/. – Elektronische Ressource, zuletzt geprüft: 2008-05-08

[Baz07] Bazaarvoice: *Bazaarvoice Press Release: Bazaarvoice Achieves Profitability with Unprecedented Client Growth*. Version: Mai 2007. http://www.bazaarvoice.com/press050207.html. – Elektronische Ressource, zuletzt geprüft: 2008-05-06

[Baz08] Bazaarvoice: *Customer generated content for your website*. Version: 2008. http://www.bazaarvoice.com/productOverview.html. – Elektronische Ressource, zuletzt geprüft: 2008-05-06

[Bes07a] Best, David: *Information Extraction: KnowItAll*. Version: Januar 2007. http://www.david-best.de/uni/UnscharfeDaten/presentation.pdf. – Elektronische Ressource, zuletzt geprüft: 2008-05-05. – Folien zum Vortrag im Seminar Verwaltung Unscharfer Daten

[Bes07b] Best, David: *Informationsextraktion auf dem Korpus World Wide Web anhand des Systems KnowItAll*. Version: März 2007. http://www.david-best.de/uni/UnscharfeDaten/IE_Systems.pdf. – Elektronische Ressource, zuletzt geprüft: 2008-05-05. – Seminararbeit im Seminar Verwaltung Unscharfer Daten

[Bm02] Bussmann, Hadumod: *Lexikon der Sprachwissenschaft (Kröners Taschenausgaben)*. 3., aktual. u. erw. A. Kröner, 2002. – ISBN 3520452030

[CNE08] CNET: *CNET Reviews: Pentax K20D (body only) Specs. Digital cameras Specifications*. Version: 2008. http://reviews.cnet.com/digital-cameras/pentax-k20d-body-only/4507-6501_7-32825568.html. – Elektronische Ressource, zuletzt geprüft: 2008-06-04

[CNZ05] Carenini, Giuseppe ; Ng, Raymond T. ; Zwart, Ed: Extracting knowledge from evaluative text. In: *K-CAP '05: Proceedings of the 3rd international conference on Knowledge capture*. ACM. – ISBN 1595931635, 11-18

[Dai96] Daille, Béatrice: Study and Implementation of Combined Techniques for Automatic Extraction of Terminology. Version: 1996. http://citeseer.ist.psu.edu/587322.html. In: Klavans, Judith (Hrsg.) ; Resnik, Philip (Hrsg.): *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*. The MIT Press, 49–66

[Dig08] Digital Photography Review: *Buying Guide: Digital Cameras Features Search (Digital Photography Review)*. Version: 2008. http://www.

dpreview.com/reviews/compare.asp. – Elektronische Ressource, zuletzt geprüft: 2008-06-04

[DLP03] Dave, Kushal ; Lawrence, Steve ; Pennock, David M.: Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: *WWW '03: Proceedings of the 12th international conference on World Wide Web*. New York, NY, USA : ACM, 2003. – ISBN 1–58113–680–3, S. 519–528

[ES06] Esuli, Andrea ; Sebastiani, Fabrizio: SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In: *Proceedings of LREC-06, the 5th Conference on Language Resources*

[Etz04] Etzioni, Oren u. a.: *WebScale Information Extraction in KnowItAll (Preliminary Results)*. Version: May 2004. http://www.cs.washington.edu/research/knowitall/papers/www-paper.pdf. – Elektronische Ressource, zuletzt geprüft: 2007-03-08

[FAM00] Frantzi, Katerina ; Ananiadou, Sophia ; Mima, Hideki: Automatic recognition of multi-word terms:. the C-value/NC-value method. In: *International Journal on Digital Libraries* V3 (2000), Nr. 2, 115–130. http://dx.doi.org/10.1007/s007999900023

[Fel98] Fellbaum, Christiane (Hrsg.): *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press http://amazon.com/o/ASIN/026206197X/. – ISBN 026206197X

[GCHH91] Gale, W.A. ; Church, K.W. ; Hanks, P. ; Hindle, D.: Using Statistics in Lexical Analysis. In: Zernik, U. (Hrsg.): *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*. Hillsdale, N.J. : Lawrence Erlbaum Associates, 1991, S. 115–164

[GSM08] GSMArena: *Phone Finder - search for a phone by feature - GSMArena.com*. Version: 2008. http://www.gsmarena.com/search.php3. – Elektronische Ressource, zuletzt geprüft: 2008-06-04

[Hip08] Hipp, Laura: *Austin Business Journal: Bazaarvoice growing rapidly*. Version: Februar 2008. http://austin.bizjournals.com/austin/stories/2008/02/18/story8.html. – Elektronische Ressource, zuletzt geprüft: 2008-05-06

[HL04] Hu, Minqing ; Liu, Bing: Mining and summarizing customer reviews. In: *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA : ACM, 2004. – ISBN 1–58113–888–1, S. 168–177

[Jod08] Jodange: *Jodange Products: Top of Mind*. Version: 2008. http://www.jodange.com/products.html. – Elektronische Ressource, zuletzt geprüft: 2008-05-06

[KU96] Kageura, Kyo ; Umino, Bin: Methods of automatic term recognition: a review. In: *Terminology* 3 (1996), Nr. 2, 259–289. http://citeseer.ist.psu.edu/kageura96methods.html

[Liu06] Liu, Bing: *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. 1. Springer, 2006. – ISBN 3540378812

[MKSW99] Makhoul, J. ; Kubala, F. ; Schwartz, R. ; Weischedel, R.: *Performance measures for information extraction*. 1999

[MS99] Manning, Christopher D. ; Schuetze, Hinrich: *Foundations of Statistical Natural Language Processing*. 1. The MIT Press, 1999. – ISBN 0262133601

[NAM04] Nenadié, Goran ; Ananiadou, Sophia ; McNaught, John: Enhancing automatic term recognition through recognition of variation. In: *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*. Morristown, NJ, USA : Association for Computational Linguistics, 2004, S. 604

[Opi08] Opinmind: *Advertisers*. Version: 2008. `http://opinmind.com/advertisers.html`. – Elektronische Ressource, zuletzt geprüft: 2008-05-06

[PE05] Popescu, Ana M. ; Etzioni, Oren: Extracting product features and opinions from reviews. (2005), 339-346. `http://www.cs.washington.edu/homes/etzioni/papers/emnlp05_opine.pdf`

[Pho08] Phone Scoop: *Phone Finder - search database of cell phone specs & features (Phone Scoop)*. Version: 2008. `http://www.phonescoop.com/phones/finder.php`. – Elektronische Ressource, zuletzt geprüft: 2008-06-04

[PLV02] Pang, Bo ; Lee, Lillian ; Vaithyanathan, Shivakumar: Thumbs up?: sentiment classification using machine learning techniques. In: *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*. Morristown, NJ, USA : Association for Computational Linguistics, 2002, S. 79–86

[Pow07] PowerReviews: *Solutions Overview*. Version: 2007. `http://www.powerreviews.com/social-shopping/solutions/`. – Elektronische Ressource, zuletzt geprüft: 2008-05-06

[SB87] Salton, Gerard ; Buckley, Chris: Term Weighting Approaches in Automatic Text Retrieval. (1987)

[SM08] SentiMetrix: *The company*. Version: 2008. `http://sentimetrix.com/`. – Elektronische Ressource, zuletzt geprüft: 2008-05-06

[UMS94] Ungerer, Friedrich ; Meier, Gerhard E. ; Schäfer, Klaus.: *A Grammar of Present- Day English. (Lernmaterialien)*. Klett `http://amazon.com/o/ASIN/3125058007/`. – ISBN 3125058007

[Vic08] Victorinox: *Victorinox Swiss Army - MultiTools - SwissChamp*. Version: 2008. `http://www.swissarmy.com/MultiTools/Pages/Product.aspx?category=everyday&product=53501&`. – Elektronische Ressource, zuletzt geprüft: 2008-06-04

[Wik08a] Wikipedia: *Mobile phone features — Wikipedia, The Free Encyclopedia*. Version: 2008. http://en.wikipedia.org/w/index.php?title=Mobile_phone_features&oldid=210977484. – Elektronische Ressource, zuletzt geprüft: 2008-06-04

[Wik08b] Wikipedia: *Pocket knife — Wikipedia, The Free Encyclopedia*. Version: 2008. http://en.wikipedia.org/w/index.php?title=Pocket_knife&oldid=213348094. – Elektronische Ressource, zuletzt geprüft: 2008-06-04

[Wik08c] Wikipedia: *Swiss Army knife — Wikipedia, The Free Encyclopedia*. Version: 2008. http://en.wikipedia.org/w/index.php?title=Swiss_Army_knife&oldid=216614016. – Elektronische Ressource, zuletzt geprüft: 2008-06-04

[WWH04] Wilson, Theresa ; Wiebe, Janyce ; Hwa, Rebecca: *Just how mad are you? Finding strong and weak opinion clauses*. http://citeseer.ist.psu.edu/677957.html. Version: 2004

[YNBN03] Yi, Jeonghee ; Nasukawa, Tetsuya ; Bunescu, Razvan ; Niblack, Wayne: Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques. (2003). http://oucsace.cs.ohiou.edu/~razvan/papers/icdm2003.pdf

[Zie06] Ziegler, Cai: Die Vermessung der Meinung. (2006), Oktober, S. 106–109. – Artikel in iX 10/2006, Heise Zeitschriften Verlag Gmbh & Co. KG