Extraktion und Anreicherung von Merkmalshierarchien durch Analyse unstrukturierter Produktrezensionen

Robin Küppers
Institut für Informatik
Heinrich-Heine-Universität
Universitätsstr. 1
40225 Düsseldorf, Deutschland
kueppers@cs.uni-duesseldorf.de

ABSTRACT

Wir präsentieren einen Algorithmus zur Extraktion bzw. Anreicherung von hierarchischen Produktmerkmalen mittels einer Analyse von unstrukturierten, kundengenerierten Produktrezensionen. Unser Algorithmus benötigt eine initiale Merkmalshierarchie, die in einem rekursiven Verfahren mit neuen Untermerkmalen angereichert wird, wobei die natürliche Ordnung der Merkmale beibehalten wird. Die Funktionsweise unseres Algorithmus basiert auf häufigen, grammatikalischen Strukturen, die in Produktrezensionen oft benutzt werden, um Eigenschaften eines Produkts zu beschreiben. Diese Strukturen beschreiben Obermerkmale im Kontext ihrer Untermerkmale und werden von unserem Algorithmus ausgenutzt, um Merkmale hierarchisch zu ordnen.

Kategorien

H.2.8 [**Database Management**]: Database Applications—*data mining*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*text analysis*

Schlüsselwörter

Text Mining, Review Analysis, Product Feature

1. EINLEITUNG

Der Einkauf von Waren (z. B. Kameras) und Dienstleistungen (z. B. Hotels) über Web-Shops wie Amazon unterliegt seit Jahren einem stetigen Wachstum. Web-Shops geben ihren Kunden (i. d. R.) die Möglichkeit die gekaufte Ware in Form einer Rezension zu kommentieren und zu bewerten. Diese kundengenerierten Rezensionen enthalten wertvolle Informationen über das Produkt, die von potentiellen Kunden für ihre Kaufentscheidung herangezogen werden. Je positiver ein Produkt bewertet wird, desto wahrscheinlicher wird es von anderen Kunden gekauft.
Der Kunde kann sich so ausführlicher über die Vor- und Nachteile eines Produkts informieren, als dies über redaktionelle Datenblätter oder Produktbeschreibungen möglich wäre, da diese dazu tendieren, die Vorteile eines Produkts zu beleuchten und die Nachteile zu verschweigen. Aus diesem Grund haben potentielle Kunden ein berechtigtes Interesse an der subjektiven Meinung anderer Käufer.
Zudem sind kundengenerierte Produktrezensionen auch für Produzenten interessant, da sie wertvolle Informationen über Qualität und Marktakzeptanz eines Produkts aus Kundensicht enthalten. Diese Informationen können Produzenten dabei helfen, die eigene Produktpalette zu optimieren und besser an Kundenbedürfnisse anzupassen.
Mit wachsendem Umsatz der Web-Shops nimmt auch die Anzahl der Produktrezensionen stetig zu, so dass es für Kunden (und Produzenten) immer schwieriger wird, einen umfassenden Überblick über ein Produkt / eine Produktgruppe zu behalten. Deshalb ist unser Ziel eine feingranulare Zusammenfassung von Produktrezensionen, die es erlaubt Produkte dynamisch anhand von Produktmerkmalen (*product features*) zu bewerten und mit ähnlichen Produkten zu vergleichen. Auf diese Weise wird ein Kunde in die Lage versetzt ein Produkt im Kontext seines eigenen Bedürfnisses zu betrachten und zu bewerten: beispielsweise spielt das Gewicht einer Kamera keine große Rolle für einen Kunden, aber es wird viel Wert auf eine hohe Bildqualität gelegt. Produzenten können ihre eigene Produktpalette im Kontext der Konkurrenz analysieren, um z. B. Mängel an den eigenen Produkten zu identifizieren.
Das Ziel unserer Forschung ist ein Gesamtsystem zur Analyse und Präsentation von Produktrezensionen in zusammengefasster Form (vgl. [3]). Dieses System besteht aus mehreren Komponenten, die verschiedene Aufgaben übernehmen, wie z.B. die Extraktion von Meinungen und die Bestimmung der Tonalität bezüglich eines Produktmerkmals (siehe dazu auch Abschnitt 2). Im Rahmen dieser Arbeit beschränken wir uns auf einen wichtigen Teilaspekt dieses Systems: die Extraktion und Anreicherung von hierarchisch organisierten Produktmerkmalen.
Der Rest dieser Arbeit ist wie folgt gegliedert: zunächst geben wir in Abschnitt 2 einen Überblick über verwandte Arbeiten, die auf unsere Forschung entscheidenen Einfluss hatten. Anschließend präsentieren wir in Abschnitt 3 einen Algorithmus zur Extraktion und zur Anreicherung von hierarchisch organisierten Produktmerkmalen. Eine Bewertung des Algorithmus wird in Abschnitt 4 vorgenommen, sowie einige Ergebnisse präsentiert, die die Effektivität unseres Algorithmus demonstrieren. Die gewonnenen Erkenntnisse werden in Abschnitt 5 diskutiert und zusammengefasst. Des

25^{th} GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), 28.05.2013 - 31.05.2013, Ilmenau, Germany.

Weiteren geben wir einen Ausblick auf unsere zukünftige Forschung.

2. VERWANDTE ARBEITEN

Dieser Abschnitt gibt einen kurzen Überblick über verwandte Arbeiten, die einen Einfluss auf unsere Forschung hatten. Die Analyse von Produktrezensionen basiert auf Algorithmen und Methoden aus verschiedensten Disziplinen. Zu den Wichtigsten zählen: Feature Extraction, Opining Mining und Sentiment Analysis.
Ein typischer Algorithmus zur merkmalsbasierten Tonalitätsanalyse von Produktrezensionen ist in 3 unterschiedliche Phasen unterteilt (vgl. [3]):

1. Extraktion von Produktmerkmalen.
2. Extraktion von Meinungen über Produktmerkmale.
3. Tonalitätsanalyse der Meinungen.

Man unterscheidet zwischen impliziten und expliziten Merkmalen[3]: explizite Merkmale werden direkt im Text genannt, implizite Merkmale müssen aus dem Kontext erschlossen werden. Wir beschränken uns im Rahmen dieser Arbeit auf die Extraktion expliziter Merkmale.
Die Autoren von [3] extrahieren häufig auftretende, explizite Merkmale mit dem a-priori Algorithmus. Mit Hilfe dieser Produktmerkmale werden Meinungen aus dem Text extrahiert, die sich auf ein Produktmerkmal beziehen. Die Tonalität einer Meinung wird auf die Tonalität der enthaltenen Adjektive zurückgeführt. Die extrahierten Merkmale werden - im Gegensatz zu unserer Arbeit - nicht hierarchisch modelliert.
Es gibt auch Ansätze, die versuchen die natürliche Hierarchie von Produktmerkmalen abzubilden. Die Autoren von [1] nutzen die tabellarische Struktur von Produktbeschreibungen aus, um explizite Produktmerkmale zu extrahieren, wobei die hierarchische Struktur aus der Tabellenstruktur abgeleitet wird. Einen ähnlichen Ansatz verfolgen [5] et. al.: die Autoren nutzen ebenfalls die oftmals hochgradige Strukturierung von Produktbeschreibungen aus. Die Produktmerkmale werden mit Clusteringtechniken aus einem Korpus extrahiert, wobei die Hierarchie der Merkmale durch das Clustering vorgegeben wird. Die Extraktion von expliziten Merkmalen aus strukturierten Texten ist (i. d. R.) einfacher, als durch Analyse unstrukturierter Daten.
Die Methode von [2] et. al. benutzt eine Taxonomie zur Abbildung der Merkmalshierarchie, wobei diese von einem Experten erstellt wird. Diese Hierarchie bildet die Grundlage für die Meinungsextraktion. Die Tonalität der Meinungen wird über ein Tonalitätswörterbuch gelöst. Für diesen Ansatz wird - im Gegensatz zu unserer Methode - umfangreiches Expertenwissen benötigt.
Die Arbeit von [8] et. al. konzentriert sich auf die Extraktion von Meinungen und die anschließende Tonalitätsanalyse. Die Autoren unterscheiden zwischen subjektiven und komparativen Sätze. Sowohl subjektive, als auch komparative Sätze enthalten Meinungen, wobei im komparativen Fall eine Meinung nicht direkt gegeben wird, sondern über einen Vergleich mit einem anderen Produkt erfolgt. Die Autoren nutzen komparative Sätze, um Produktgraphen zu erzeugen mit deren Hilfe verschiedene Produkte hinsichtlich eines Merkmals geordnet werden können. Die notwendigen Tonalitätswerte werden einem Wörterbuch entnommen.

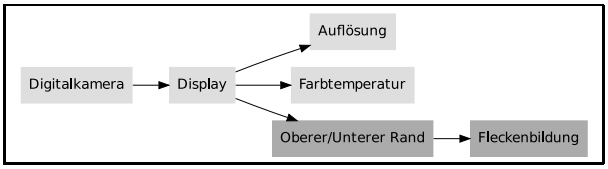

Abbildung 1: Beispielhafte Merkmalshierarchie einer Digitalkamera.

Wir haben hauptsächlich Arbeiten vorgestellt, die Merkmale und Meinungen aus Produktrezensionen extrahieren, aber Meinungsanalysen sind auch für andere Domänen interessant: z. B. verwenden die Autoren von [7] einen von Experten annotierten Korpus mit Nachrichten, um mit Techniken des maschinellen Lernens einen Klassifikator zu trainieren, der zwischen Aussagen (Meinungen) und Nicht-Aussagen unterscheidet. Solche Ansätze sind nicht auf die Extraktion von Produktmerkmalen angewiesen.

3. ANREICHERUNG VON MERKMALSHIERARCHIEN

Dieser Abschnitt dient der Beschreibung eines neuen Algorithmus zur Anreicherung einer gegebenen, unvollständigen Merkmalshierarchie mit zusätzlichen Merkmalen. Diese Merkmale werden aus unstrukturierten kundengenerierten Produktrezensionen gewonnen, wobei versucht wird die natürliche Ordnung der Merkmale (Unter- bzw. Obermerkmalsbeziehung) zu beachten.
Die Merkmalshierarchie bildet die Basis für weitergehende Analysen, wie z.B. die gezielte Extraktion von Meinungen und Tonalitäten, die sich auf Produktmerkmale beziehen. Diese nachfolgenden Analyseschritte sind nicht mehr Gegenstand dieser Arbeit. Produkte (aber auch Dienstleistungen) können durch eine Menge von Merkmalen (*product features*) beschrieben werden. Produktmerkmale folgen dabei einer natürlichen, domänenabhängigen Ordnung. Eine derartige natürliche Hierarchie ist exemplarisch in Abbildung 1 für das Produkt **Digitalkamera** dargestellt. Offensichtlich ist **Display** ein Untermerkmal von **Digitalkamera** und besitzt eigene Untermerkmale **Auflösung** und **Farbtemperatur**.
Hierarchien von Produktmerkmalen können auf Basis von strukturierten Texten erzeugt werden, wie z. B. technische Datenblättern und Produktbeschreibungen (vgl. [5]). Diese Datenquellen enthalten i. d. R. die wichtigsten Produktmerkmale. Der hohe Strukturierungsgrad dieser Datenquellen erlaubt eine Extraktion der Merkmale mit hoher Genauigkeit ($\approx$ 71% [5]). Allerdings tendieren Datenblätter und Produktbeschreibungen dazu, ein Produkt relativ oberflächlich darzustellen oder zu Gunsten des Produkts zu verzerren. Zum Beispiel enthält die Hierarchie in Abbildung 1 eine Reihe von Merkmalen, wie sie häufig in strukturierten Datenquellen zu finden sind (helle Knoten). Allerdings sind weitere, detailliertere Merkmale denkbar, die für eine Kaufentscheidung von Interesse sein könnten. Beispielsweise könnte das **Display** einer Digitalkamera zur Fleckenbildung am unteren/oberen Rand neigen. **Unterer/Oberer Rand** wird in diesem Fall zu einem Untermerkmal von **Display** und Obermerkmal von **Fleckenbildung** (dunkle Knoten).
Eine derartige Anreicherung einer gegebenen, unvollständigen Merkmalshierarchie kann durch die Verarbeitung von

kundengenerierten, unstrukturierten Rezensionen erfolgen. Wir halten einen hybriden Ansatz für durchaus sinnvoll: zunächst wird eine initiale Merkmalshierarchie mit hoher Genauigkeit aus strukturierten Daten gewonnen. Anschließend wird diese Hierarchie in einer zweiten Verarbeitungshase mit zusätzlichen Produktmerkmalen angereichert.
Für den weiteren Verlauf dieses Abschnitts beschränken wir uns auf die zweite Analysephase, d.h. wir nehmen eine initiale Merkmalshierarchie als gegeben an. Für die Evaluation unseres Algorithmus (siehe Abschnitt 4) wurden die initialen Merkmalshierarchien manuell erzeugt.
Unser Algorithmus wurde auf der Basis einer Reihe von einfachen Beobachtungen entworfen, die wir bei der Analyse von unserem Rezensionskorpus gemacht haben.

1. Ein Produktmerkmal wird häufig durch ein Hauptwort repräsentiert.
2. Viele Hauptwörter können dasselbe Produktmerkmal beschreiben. (Synonyme)
3. Untermerkmale werden häufig im Kontext ihrer Obermerkmale genannt, wie z. B. "das Ladegerät der Kamera".
4. Textfragmente, die von Produktmerkmalen handeln, besitzen häufig eine sehr ähnliche grammatikalische Struktur, wie z.B. "die Auflösung der Anzeige" oder "die Laufzeit des Akkus", wobei Unter- und Obermerkmale gemeinsam genannt werden. Die Struktur der Fragmente lautet [DET, NOUN, DET, NOUN], wobei DET einen Artikel und NOUN ein Hauptwort beschreibt.

Der Rest dieses Abschnitts gliedert sich wie folgt: zunächst werden Definitionen in Unterabschnitt 3.1 eingeführt, die für das weitere Verständnis notwendig sind. Anschließend beschreiben wir unsere Analysepipeline, die für die Vorverarbeitung der Produktrezensionen verwendet wurde, in Unterabschnitt 3.2. Darauf aufbauend wird in Unterabschnitt 3.3 unser Algorithmus im Detail besprochen.

3.1 Definitionen

Für das Verständnis der nächsten Abschnitte werden einige Begriffe benötigt, die in diesem Unterabschnitt definiert werden sollen:

Token. Ein Token t ist ein Paar $t = (v_{word}, v_{POS})$, wobei v_{word} das Wort und v_{pos} die Wortart angibt. Im Rahmen dieser Arbeit wurde das Universal Tagset [6] benutzt.

Merkmal. Wir definieren ein Produktmerkmal f als ein Tripel $f = (S, C, p)$, wobei S eine Menge von Synonymen beschreibt, die als textuelle Realisierung eines Merkmals Verwendung finden können. Die Elemente von S können Worte, Produktbezeichnungen und auch Abkürzungen enthalten. Die Hierarchie wird über C und p kontrolliert, wobei C eine Menge von Untermerkmalen und p das Obermerkmal von f angibt. Das Wurzelelement einer Hierarchie beschreibt das Produkt/die Produktgruppe selbst und besitzt kein Obermerkmal.

POS-Muster. Ein POS-Muster q ist eine geordnete Sequenz von POS-Tags $p = [tag_1, tag_2, \ldots, tag_n]$, wobei n die Musterlänge beschreibt. Ein POS-Tag beschreibt eine Wortart, z.B. steht DET für einen Artikel, NOUN für ein Hauptwort und ADJ für ein Adjektiv. Weitere Informationen über das Universal Tagset finden sich in [6].

3.2 Analysepipeline

Für die Verarbeitung und Untersuchung der Produktrezensionen haben wir eine für den NLP-Bereich (Natural Language Processing) typische Standardpipeline benutzt: die Volltexte der Rezensionen sind für unsere Zwecke zu grobgranular, so dass in einer ersten Phase der Volltext in Sätze zerteilt wird. Anschließend werden die Sätze tokenisiert und die Wortarten der einzelnen Worte bestimmt. Des Weiteren werden Stoppworte markiert - dafür werden Standard-Stoppwortlisten benutzt. Wir beenden die Analysepipeline mit einer Stammformreduktion für jedes Wort, um die verschiedenen Flexionsformen eines Wortes auf eine kanonische Basis zu bringen.
Für die Bestimmung zusätzlicher Produktmerkmale aus Produktrezensionen, sind vor allem Hauptworte interessant, die i. d. R. keine Stoppworte sind. Allerdings ist uns aufgefallen, dass überdurchschnittlich viele Worte fälschlicherweise als ein Hauptwort erkannt werden - viele dieser Worte sind Stoppworte. Wir nehmen an, dass die variierende, grammatikalische Qualität der Produktrezensionen für die hohe Anzahl falsch bestimmer Worte verantwortlich ist. Die Stoppwortmarkierung hilft dabei, diesen Fehler etwas auszugleichen.

3.3 Der Algorithmus

In diesem Abschnitt beschreiben wir einen neuen Algorithmus, um eine initiale Hierarchie von Produktmerkmalen mit zusätzlichen Merkmalen anzureichern, wobei die natürliche Ordnung der Merkmale erhalten bleibt (siehe Algorithmus 1). Der Algorithmus erwartet 3 Parameter: eine 2-dimensionale Liste von Token T, die sämtliche Token für jeden Satz enthält (dabei beschreibt die erste Dimension die Sätze, die zweite Dimensionen die einzelnen Wörter), eine initiale Hierarchie von Merkmalen f und eine Menge von POS-Mustern P. Da der Algorithmus rekursiv arbeitet, wird zusätzlich ein Parameter d übergeben, der die maximale Rekursionstiefe angibt. Der Algorithmus bricht ab, sobald die vorgegebene Tiefe erreicht wird (Zeile 1-3).

Kandidatensuche (Zeile 4-11). Um geeignete Kandidaten für neue Produktmerkmale zu finden, werden alle Sätze betrachtet und jeweils entschieden, ob der Satz eine Realisierung des aktuell betrachteten Merkmals enthält oder nicht. Wenn ein Satz eine Realisierung hat, dann wird die Funktion $applyPatterns$ aufgerufen. Diese Funktion sucht im übergebenen Satz nach gegebenen POS-Mustern und gibt – sofern mindestens ein Muster anwendbar ist – die entsprechenden Token als Kandidat zurück, wobei die Mustersuche auf das unmittelbare Umfeld der gefundenen Realisierung eingeschränkt wird, damit das korrekte POS-Muster zurückgeliefert wird, da POS-Muster mehrfach innerhalb eines Satzes vorkommen können.
Im Rahmen dieser Arbeit haben wie die folgenden POS-Muster verwendet:

- [DET, NOUN, DET, NOUN]
- [DET, NOUN, VERB, DET, ADJ, NOUN]

Algorithm 1: refineHierarchy

Eingabe : T : Eine 2-dimensionale Liste von Token.
Eingabe : P : Ein Array von POS-Mustern.
Eingabe : f : Eine initiale Merkmalshierarchie.
Eingabe : d : Die maximale Rekursionstiefe.
Ausgabe: Das Wurzelmerkmal der angereicherten Hierarchie.

```
1  if d = 0 then
2  |  return f
3  end
4  C ← {} ;
5  for Token[] T' ∈ T do
6  |  for Token t ∈ T' do
7  |  |  if t.word ∈ f.S then
8  |  |  |  C ← C ⋃ applyPattern(T', P) ;
9  |  |  end
10 |  end
11 end
12 for Token[] C' ∈ C do
13 |  for Token t ∈ C' do
14 |  |  if t.pos ≠ NOUN then
15 |  |  |  next ;
16 |  |  end
17 |  |  if t.length ≤ 3 then
18 |  |  |  next ;
19 |  |  end
20 |  |  if hasParent(t.word, f) then
21 |  |  |  next ;
22 |  |  end
23 |  |  if isSynonym(t.word, f.S) then
24 |  |  |  f.S ← t.word ;
25 |  |  |  next ;
26 |  |  end
27 |  |  f.C ← f.C ⋃ ({t.word}, {}, f) ;
28 |  end
29 end
30 for Feature[] f' ∈ f.C do
31 |  refineHierarchy(T, f', P, d − 1);
32 end
```

Validierungsphase (Zeile 12-29). Die Validierungsphase dient dazu die gefundenen Kandidaten zu validieren, also zu entscheiden, ob ein Kandidat ein neues Merkmal enthält. Man beachte, dass es sich bei diesem neuen Merkmal um ein Untermerkmal des aktuellen Produktmerkmals handelt, sofern es existiert. Für die Entscheidungsfindung nutzen wir eine Reihe von einfachen Heuristiken. Ein Token t ist **kein** Produktmerkmal und wird übergangen, falls $t.v_{word}$:

1. kein Hauptwort ist (Zeile 14-16).
2. keine ausreichende Länge besitzt (Zeile 17-19).
3. ein Synonym von f (oder eines Obermerkmals von f) ist (Zeile 20-22).
4. ein neues Synonym von f darstellt (Zeile 23-26).

Die 3. Heuristik stellt sicher, dass sich keine Kreise in der Hierarchie bilden können. Man beachte, dass Obermerkmale, die nicht direkt voneinander abhängen, gleiche Untermerkmale tragen können.
Die 4. Heuristik dient zum Lernen von vorher unbekannten Synonymen. Dazu wird das Wort mit den Synonymen von f verglichen (z.B. mit der Levenshtein-Distanz) und als Synonym aufgenommen, falls eine ausreichende Ähnlichkeit besteht. Damit soll verhindert werden, dass die falsche Schreibweise eines eigentlich bekannten Merkmals dazu führt, dass ein neuer Knoten in die Hierarchie eingefügt wird.
Wenn der Token t die Heuristiken erfolgreich passiert hat, dann wird t zu einem neuen Untermerkmal von f (Zeile 27).

Rekursiver Aufruf (Zeile 30-32). Nachdem das Merkmal f nun mit zusätzlichen Merkmalen angereichert wurde, wird der Algorithmus rekursiv für alle Untermerkmale von f aufgerufen, um diese mit weiteren Merkmalen zu versehen. Dieser Vorgang wiederholt sich solange, bis die maximale Rekursionstiefe erreicht wird.

Nachbearbeitungsphase. Die Hierarchie, die von Algorithmus 1 erweitert wurde, muss in einer Nachbearbeitungsphase bereinigt werden, da viele Merkmale enthalten sind, die keine realen Produktmerkmale beschreiben (Rauschen). Für diese Arbeit verwenden wir die relative Häufigkeit eines Untermerkmals im Kontext seines Obermerkmals, um niederfrequente Merkmale (samt Untermerkmalen) aus der Hierarchie zu entfernen. Es sind aber auch andere Methoden denkbar, wie z.B. eine Gewichtung nach *tf-idf* [4]. Dabei wird nicht nur die Termhäufigkeit (*tf*) betrachtet, sondern auch die inverse Dokumenthäufigkeit (*idf*) mit einbezogen. Der *idf* eines Terms beschreibt die Bedeutsamkeit des Terms im Bezug auf die gesamte Dokumentenmenge.

4. EVALUATION

In diesem Abschnitt diskutieren wir die Vor- und Nachteile unseres Algorithmus. Um unseren Algorithmus evaluieren zu können, haben wir einen geeigneten Korpus aus Kundenrezensionen zusammengestellt. Unser Korpus besteht aus 4000 Kundenrezensionen von *amazon.de* aus der Produktgruppe **Digitalkamera**.
Wir haben unseren Algorithmus für die genannte Produktgruppe eine Hierarchie anreichern lassen. Die initiale Produkthierarchie enthält ein Obermerkmal, welches die Produktgruppe beschreibt. Zudem wurden häufig gebrauchte Synonyme hinzugefügt, wie z.B. **Gerät**. Im Weiteren präsentieren wir exemplarisch die angereicherte Hierarchie. Für dieses Experiment wurde die Rekursionstiefe auf 3 gesetzt, niederfrequente Merkmale (relative Häufigkeit $< 0,002$) wurden eliminiert. Wir haben für diese Arbeit Rezensionen in Deutscher Sprache verwendet, aber der Algorithmus kann leicht auf andere Sprachen angepasst werden. Die erzeugte Hierarchie ist in Abbildung 2 dargestellt. Es zeigt sich, dass unser Algorithmus – unter Beachtung der hierarchischen Struktur – eine Reihe wertvoller Merkmale extrahieren konnte: z. B. **Batterie** mit seinen Untermerkmalen **Haltezeit** und **Verbrauch** oder **Akkus** mit den Untermerkmalen **Auflad** und **Zukauf**. Es wurden aber auch viele Merkmale aus den Rezensionen extrahiert, die entweder keine echten Produktmerkmale sind (z.B. **Kompakt** oder eine falsche Ober-Untermerkmalsbeziehung abbilden (z. B. **Haptik** und **Kamera**). Des Weiteren sind einige Merkmale, wie z. B. **Qualität** zu generisch und sollten nicht als Produktmerkmal benutzt werden.

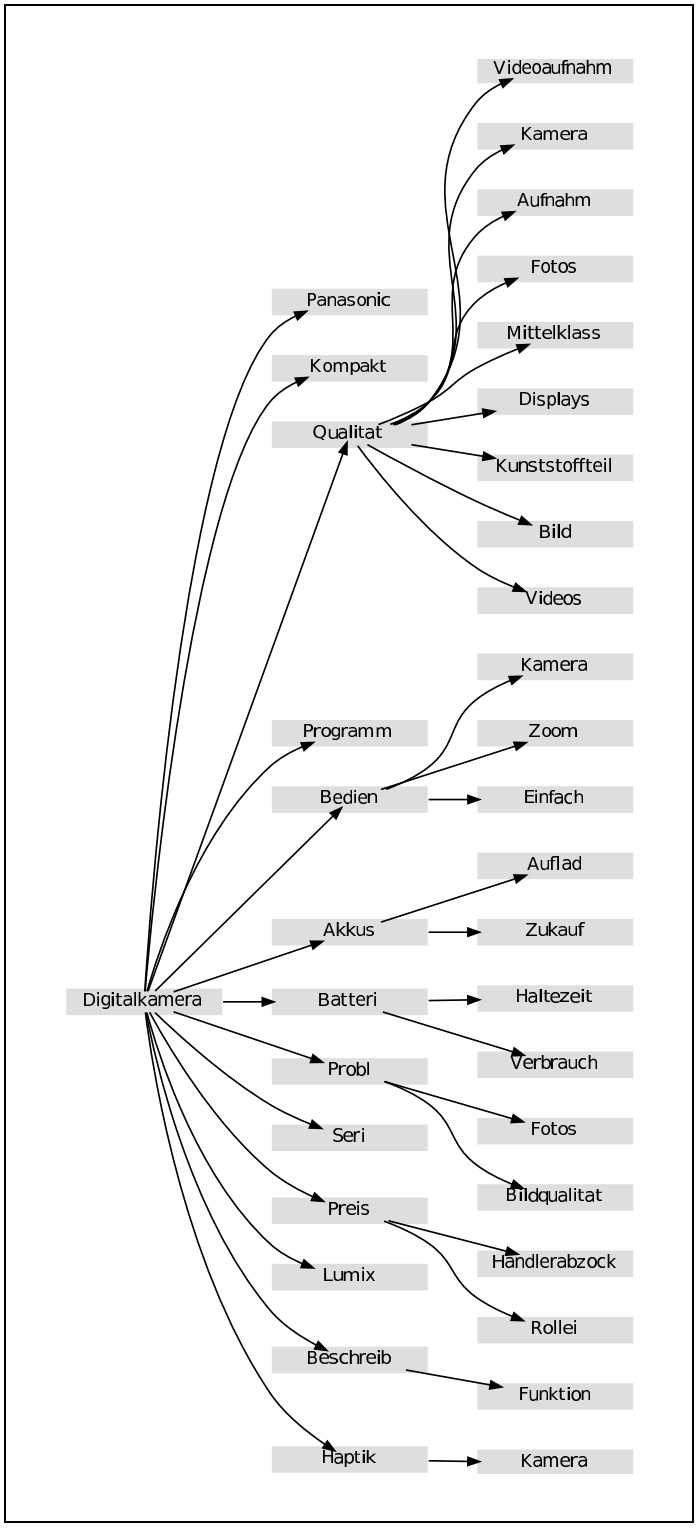

Abbildung 2: Angereicherte Hierarchie für die Produktgruppe Digitalkamera.

5. RESÜMEE UND AUSBLICK

In dieser Arbeit wurde ein neuer Algorithmus vorgestellt, der auf Basis einer gegebenen – möglicherweise flachen – Merkmalshierarchie diese Hierarchie mit zusätzlichen Merkmalen anreichert. Die neuen Merkmale werden automatisch aus unstrukturierten Produktrezensionen gewonnen, wobei der Algorithmus versucht die natürliche Ordnung der Produktmerkmale zu beachten.
Wir konnten zeigen, dass unser Algorithmus eine initiale Merkmalshierarchie mit sinnvollen Untermerkmalen anreichern kann, allerdings werden auch viele falsche Merkmale extrahiert und in fehlerhafte Merkmalsbeziehungen gebracht. Wir halten unseren Algorithmus dennoch für vielversprechend. Unsere weitere Forschung wird sich auf Teilaspekte dieser Arbeit konzentrieren:

- Die Merkmalsextraktion muss verbessert werden: wir haben beobachtet, dass eine Reihe extrahierter Merkmale keine echten Produktmerkmale beschreiben. Dabei handelt es sich häufig um sehr allgemeine Wörter wie z.B. **Möglichkeiten**. Wir bereiten deshalb den Aufbau einer Stoppwortliste für Produktrezensionen vor. Auf diese Weise könnte diese Problematik abgeschwächt werden.

- Des Weiteren enthalten die angereicherten Hierarchien teilweise Merkmale, die in einer falschen Beziehung zueinander stehen, z.B. induzieren die Merkmale **Akku** und **Akku-Ladegerät** eine Ober-Untermerkmalsbeziehung: **Akku** kann als Obermerkmal von **Ladegerät** betrachtet werden. Außerdem konnte beobachtet werden, dass einige Merkmalsbeziehungen alternieren: z.B. existieren 2 Merkmale **Taste** und **Druckpunkt** in wechselnder Ober-Untermerkmalbeziehung.

- Der Algorithmus benötigt POS-Muster, um Untermerkmale in Sätzen zu finden. Für diese Arbeit wurden die verwendeten POS-Muster manuell konstruiert, aber wir planen die Konstruktion der POS-Muster weitestgehend zu automatisieren. Dazu ist eine umfangreiche Analyse eines großen Korpus notwendig.

- Die Bereinigung der erzeugten Hierarchien ist unzureichend - die relative Häufigkeit eines Merkmals reicht als Gewichtung für unsere Zwecke nicht aus. Aus diesem Grund möchten wir mit anderen Gewichtungsmaßen experimentieren.

- Die Experimente in dieser Arbeit sind sehr einfach gestaltet. Eine sinnvolle Evaluation ist (z. Zt.) nicht möglich, da (unseres Wissens nach) kein geeigneter Testkorpus mit annotierten Merkmalshierarchien existiert. Die Konstruktion eines derartigen Korpus ist geplant.

- Des Weiteren sind weitere Experimente geplant, um den Effekt der initialen Merkmalshierarchie auf den Algorithmus zu evaluieren. Diese Versuchsreihe umfasst Experimente mit mehrstufigen, initialen Merkmalshierarchien, die sowohl manuell, als auch automatisch erzeugt wurden.

- Abschließend planen wir die Verwendung unseres Algorithmus zur Extraktion von Produktmerkmalen in einem Gesamtsystem zur automatischen Zusammenfassung und Analyse von Produktrezensionen einzusetzen.

6. REFERENZEN

[1] M. Acher, A. Cleve, G. Perrouin, P. Heymans, C. Vanbeneden, P. Collet, and P. Lahire. On extracting feature models from product descriptions. In *Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems*, VaMoS '12, pages 45–54, New York, NY, USA, 2012. ACM.

[2] F. L. Cruz, J. A. Troyano, F. Enríquez, F. J. Ortega, and C. G. Vallejo. A knowledge-rich approach to feature-based opinion extraction from product reviews. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, SMUC '10, pages 13–20, New York, NY, USA, 2010. ACM.

[3] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM.

[4] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[5] X. Meng and H. Wang. Mining user reviews: From specification to summarization. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 177–180, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[6] S. Petrov, D. Das, and R. McDonald. A universal part-of-speech tagset. In N. C. C. Chair), K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA).

[7] T. Scholz and S. Conrad. Extraction of statements in news for a media response analysis. In *Proc. of the 18th Intl. conf. on Applications of Natural Language Processing to Information Systems 2013 (NLDB 2013)*, 2013. (to appear).

[8] K. Zhang, R. Narayanan, and A. Choudhary. Voice of the customers: Mining online customer reviews for product feature-based ranking. In *Proceedings of the 3rd conference on Online social networks*, WOSN'10, pages 11–11, Berkeley, CA, USA, 2010. USENIX Association.