Universität zu Köln, Institut für Linguistik, Sprachliche Informationsverarbeitung
Hauptseminar: “Intelligente Systeme” bei Prof. Dr. Jürgen Rolshoven, WS 2006/2007

Paradigmen als Merkmale zur Textklassifikation

Entwicklung eines korpusbasierten Lernverfahrens

Fabian Steeg

26. Februar 2007

steeg@netcologne.de
Matrikelnummer 3598900
Liebigstr. 43
50823 Köln

Inhaltsverzeichnis

1 Überblick **1**

2 Textklassifikation **1**

3 Korpuslinguistik und maschinelle Sprachverarbeitung **2**

3.1 Korpuslinguistik 2

3.2 Korpora 3

3.3 Korpusannotation 4

3.4 Lernen und Evaluieren 5

4 Das Web als Basis für Korpora **6**

5 Klassifikation mit Paradigmen **7**

5.1 Maschinelles Lernen 8

5.2 Paradigmen und Suffixbäume 8

5.3 Programmstruktur und Vorgehen 9

5.4 Evaluierung 12

6 Software Architecture for Language Engineering **14**

7 Fazit **15**

Abbildungsverzeichnis

1 Generische Architektur eines Systems zur Textklassifikation 2

2 Screenshot der in Delicious zusammengestellten Korpora 7

3 Wissenserwerb, Klassifikation und Evaluierung 8

4 Kybernetischer Regelkreis des Lernens 9

5 Suffixbaum 10

6 Präfixbaum 10

7 UML-Klassendiagramm der Implementierung 11

8 Statusanzeige beim Ausführen der Jar-Datei 12

9 Properties-Datei zur Konfigurierung 12

10 Ergebnisse der ersten Evaluierung 13

11 Komponenten beim Wissenserwerb 14

12 Komponenten bei der Klassifikation 14

1 Überblick

Gegenstand dieser Arbeit ist die Beschreibung von Entwurf und Implementierung eines Verfahrens zur Textklassifikation, basierend auf einem unüberwachten, korpuslinguistischen Lernverfahren. Als Merkmale zur Klassifikation dienen Paradigmen,[1] die mithilfe von Suffixbäumen effizient ermittelt werden. Die Paradigmen erlauben dabei eine Form von lexikalischer Disambiguierung durch Berücksichtigung von Kontexten. Erste Evaluierungen mit kleinen Korpora ergeben einen Recall von 80-90% und eine Precision von 40-50%.

Die Arbeit ist wie folgt gegliedert: Abschnitt 2 führt in die Texklassifikation ein; Abschnitt 3 beschreibt die enge Beziehung zwischen maschineller Sprachverarbeitung und Korpuslinguistik, insbesondere im maschinellen Lernen. In Abschnitt 4 wird das verwendete, Web-basierte Korpus beschrieben; in Abschnitt 5 Konzept, Implementierung und Evaluierung des Verfahrens. Abschnitt 6 schließlich argumentiert aufbauend auf eine Diskussion möglicher Verbesserungen für den Einsatz einer *Software Architecture for Language Engineering* in der maschinellen Sprachverarbeitung.

2 Textklassifikation

Textklassifikation ist eine Form maschineller Sprachverarbeitung, bei der Texte vorgegebenen Klassen zugeordnet werden. Sie unterscheidet sich von der Schlüsselwort-Extraktion (*keyword extraction*), bei der die für den Text relevantesten Wörter im Text ermittelt werden. Hier sind die resultierenden Kategorien selbst immer als Wort im Text enthalten, während bei der Textklassifikation die Kategorien unabhängig vom Text sein können – so muss etwa ein Text der Kategorie 'Politik' nicht das Wort *Politik* enthalten. Das Ergebnis beider Ansätze ist dabei gleich: sie klassifizieren automatisch Texte, einer Zielsetzung mit großem praktischen Potential, insbesondere in Anbetracht der heute verfügbaren Menge an maschinenlesbaren Dokumenten.

Textklassifikationssysteme bestehen konzeptuell aus zwei Hauptkomponenten: dem Wissenserwerb und der eigentlichen Klassifikation. Der Wissenserwerb wiederum gliedert sich in drei weitere Teile: erstens der Merkmalsberechnung, bei der die Merkmale, anhand derer klassifiziert werden soll, ermittelt werden; zweitens der Merkmalsauswahl, bei der die relevanten Merkmale ausgewählt werden und schließlich drittens der eigentlichen Modellbildung. Das so im Wissenserwerb gebildete Modell wird dann bei der eigentlichen Klassifikation verwendet (siehe Abb. 1). Weitere Informationenen zu verschiedenen Textklassifikationsverfahren finden sich etwa in Goller *et al.* (2000).

1 *Paradigma* meint im Kontext dieser Arbeit eine Menge von Wörtern, die zueinander in paradigmatischer Relation stehen, d.h. die in gemeinsamen Kontexten vorkommen und damit gegeneinander austauschbar sind (siehe etwa Lyons 1968 : 70)

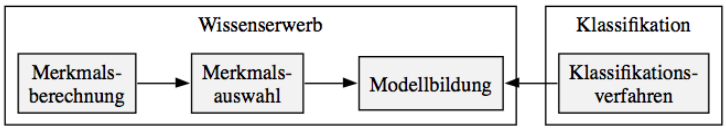

Abbildung 1: Generische Architektur eines Systems zur Textklassifikation (nach Brückner 2001).

3 Korpuslinguistik und maschinelle Sprachverarbeitung

Das nachfolgend beschriebene Verfahren beruht im Wissenserwerb auf einem korpuslinguistischen, examplarbasierten Lernverfahren. McEnery (2003) bezeichnet Textkorpora als "the raw fuel of NLP" und damit als Grundlage und Voraussetzung der maschinellen Sprachverarbeitung. Der Grund hierfür besteht darin, dass annotierte Korpora eine maschinelle Reproduktion menschlicher Intuition ermöglichen (siehe Abschnitt 3.4). So bilden annotierte Korpora die Grundlage für maschinelles Lernen in der Sprachverarbeitung (McEnery 2003 : 459).

3.1 Korpuslinguistik

Korpuslinguistik ist eine sprachwissenschaftliche Arbeitsweise und umfasst verschiedene Tätigkeiten, die mit der Erstellung und der Auswertung von Textkorpora (Köhler 2005) sowie mit der Erstellung von Werkzeugen für die beiden ersten Tätigkeiten zu tun haben. McEnery (2003) definiert ein Textkorpus als "a large body of linguistic evidence". Damit sind Korpora eine sehr wertvolle Quelle für die linguistische Arbeit. Korpuslinguistische Methoden werden nicht nur in der maschinellen Sprachverarbeitung und der Computerlinguistik[2] angewendet, sondern auch in anderen Bereichen der Sprachwissenschaft, etwa in der allgemeinen Sprachwissenschaft oder den Philologien. Somit ist Korpuslinguistik kein Teilbereich der Linguistik, sondern eine Arbeitsweise, die innerhalb der genannten Teilbereiche verwendet wird und auf verschiedene Ebenen der Sprache (wie Morphologie, Syntax, Semantik oder Pragmatik) angewendet werden kann (McEnery & Wilson 1996 : 2). Da ein großer Teil der Auswertung von Korpora mithilfe von statistischen Verfahren erfolgt, wird diese Art der Arbeit mit Korpora auch als *Quantitative Linguistik* bezeichnet (siehe Köhler 2005, Manning & Schütze 1999). Köhler (2005) betont dabei die Notwendigkeit, linguistische Fragestellungen in die Sprache der Statistik zu überführen und Ergebnisse in die Sprache der Linguistik zurückzuübersetzen und beklagt ein bislang zu wenig ausgeprägtes Methodenbewusstsein im Bereich statistischer, korpuslinguistischer Arbeit.

2 Eine scharfe Trennung der Bereiche *Computerlinguistik* und *maschinelle Sprachverarbeitung* (Natural Language Processing, NLP) ist schwierig; mitunter wird maschinelle Sprachverarbeitung als angewandte Computerlinguistik charakterisiert, mitunter als eine von der Computerlinguistik zu unterscheidende Ingenieursdisziplin.

In der heutigen, computerisierten Form gibt es Korpuslinguistik seit den späten 1940er Jahren (McEnery 2003). Ein Beispiel für frühe, nicht computerisierte Korpuslinguistik stellen Bibelkonkordanzen dar, die alle Wörter der Bibel alphabetisch sortiert in ihrem Kontext auflisten und ab dem 13. Jahrhundert entstanden; diese zeigen den radikalen Wandel durch die Entwicklung des Computers: Während das Erstellen der ersten Bibelkonkordanz 14 Jahre dauerte,[3] ist es durch die Entwicklung des Computers möglich geworden, innerhalb von Sekundenbruchteilen Konkordanzen beliebiger Texte automatisch zu erstellen; das Erstellen eines solchen Programms selbst ist innerhalb von Tagen oder Stunden möglich. So hat die Erfindung des Computers die Korpuslinguistik im heutigen Sinn erst ermöglicht. Schon durch ihre Enstehung ist die Korpuslinguistik also auf das Engste mit der maschinellen Sprachverarbeitung verbunden.

3.2 Korpora

Korpora sind durch verschiedene Eigenschaften gekennzeichnet. Zunächst sind Korpora wie oben beschrieben heute typischerweise maschinenlesbar. Ein Korpus ist darüber hinaus im Hinblick auf eine bestimmte Fragestellung organisiert. Dies bezeichnet McEnery (2003) als *sampling frame* des Korpus, innerhalb dessen es ausgewogen und repräsentativ sein soll. Als Beispiel nennt McEnery (2003) die Entwicklung eines Dialogsystems zum Verkauf von Eintritts- und Fahrkarten. Wenn hierfür ein Korpus zusammengestellt werden soll, sollten zum einen nur relevante Texte verwendet werden, d.h. Dialoge von Kartenverkäufen (dies entspricht dem gewählten *sampling frame*). Es sollten dabei jedoch verschiedene Arten von Verkaufsgesprächen verwendet werden, etwa von Bus- und Flugzeugtickets sowie von Telefon- und Schalterverkäufen. Außerdem sollten in jedem Bereich Gespräche verschiedener Personen verwendet werden, um Idiosynkrasien einzelner Sprecher zu vermeiden. So erhält man ein ausgewogenes und repräsentatives Korpus. Diesen Grundgedanken folgend wurden die Korpora für das implementierte Verfahren zusammengestellt (siehe Abschnitt 4).

McEnery (2003) unterscheidet verschiedene Arten von Korpora: *Monolinguale Korpora* sind Korpora mit Texten in einer einzigen Sprache. *Vergleichbare Korpora* sind Korpora aus Texten in verschiedenen Sprachen, die in Bezug auf *sampling frame*, Ausgewogenheit und Repräsentativität vergleichbar sind. Diese Korpora eignen sich etwa für den kontrastiven Sprachvergleich. *Parallele Korpora* sind Korpora, die zunächst aus Texten einer einzigen Sprache zusammengestellt und dann in andere Sprachen übersetzt werden. Solche Korpora eignen sich etwa für Systeme zur maschinellen Übersetzung, die aus Beispielübersetzungen lernen. *Monitorkorpora* werden laufend aktualisiert und dienen etwa zur Beobachtung von Sprachwandel. Neben diesen Unterscheidungen können gesprochene von reinen Schriftkorpora unterschieden werden. Gesprochene Korpora (die monolingual, vergleichbar oder parallel sowie Monitorkorpora sein können)

3 `http://de.wikipedia.org/wiki/Bibelkonkordanz`

enthalten Informationen über die gesprochene Form des Textes. Dies kann bedeuten, dass es sich um akustisches Material handelt, oder auch um transkribierte Sprachdaten. Da diese beiden Formen Vor- und Nachteile haben,[4] gibt es zunehmend Korpora, die aus akustischem und aus transkribiertem Material bestehen und deren Inhalte aufeinander abgestimmt sind (*time alignment*), wodurch es möglich ist, auf die jeweils andere Form zuzugreifen (McEnery 2003 : 451). Solche multimodale Korpora (Evert & Fitschen 2001) können darüber hinaus auch Informationen über Mimik oder Gestik enthalten, etwa in Form von Videomaterial. Die Multimodalität der Korpora kann auch als eine Form von Korpusannotation gesehen werden (siehe Abschnitt 3.3).

Das implementierte Verfahren verwendet monolinguale Schriftkorpora. Eine Zusammenstellung eines *vergleichbaren* (s.o.) Korpus wäre im beschriebenen Vorgehen durch die Nutzung von Texten entsprechender Ressorts der englischsprachigen Ausgabe von Spiegel-Online relativ einfach realisierbar (siehe Abschnitt 4).

3.3 Korpusannotation

Annotierte Korpora sind Korpora, die mit verschiedenen Arten linguistischer Information angereichert sind (McEnery & Wilson 1996 : 24). Ein Beispiel für Korpusannotationen ist etwa die Kennzeichnung von Wortarten durch Part-of-Speech-Tags (POS-Tags). Andere Annotationen[5] enthalten etwa Informationen über Stammformen (als Ergebnis einer Stammformenreduktion, auch *Stemming* oder *Lemmatisierung* genannt), über die syntaktische Struktur (solche Korpora werden auch *Baumbanken* genannt) sowie semantische, stilistische oder Informationen zur Diskursstruktur.[6] Diese Anreicherung basiert immer auf einer bestimmten Interpretation der Daten (McEnery 2003). Beim Annotieren ist daher zu beachten, dass der ursprüngliche Text auch nach einer Annotation noch in seiner Rohform verfügbar sein sollte, etwa durch *dynamische Annotation* (siehe Benden & Hermes 2004, Hermes & Benden 2005). McEnery (2003) nennt vier Vorteile von Korpusannotationen: Zunächst eine verbesserte Nutzbarkeit der Korpora für eine größere Anzahl von Benutzern, seien dies Menschen, die einer bestimmten Fremdsprache unkundig sind oder Computerprogramme, die Sprache verarbeiten sollen. In beiden Fällen werden die Annotationen benötigt, können jedoch von dem, der sie benötigt, nicht selbst erstellt werden. Darüber hinaus ist ein annotiertes Korpus aber auch für Benutzer, die die Analysen selbst vornehmen könnten viel schneller zu verwenden als ein nicht-annotiertes Korpus. Ein zweiter Vorteil ist die Wiederverwertbarkeit der bei der Analyse gewonnen Daten, etwa der ermittelten POS-Tags,

4 Korpora mit akustischem Material sind schwerer zu erschließen, etwa zur Suche nach bestimmten Wörtern, während es bei transkribierten Daten etwa zum Verlust prosodischer Feinheiten kommt.

5 Beispiele verschiedener Korpusannotationen und weiteres Material findet sich unter: `http://bowland-files.lancs.ac.uk/monkey/ihe/linguistics/corpus2/2fra1.htm`

6 In abnehmender Häufigkeit und Verbreitung, d.h. morphologische sind verbreiteter als syntaktische Annotationen, diese wiederum verbreiteter als semantische oder pragmatische Annotationen.

der Stammformen, syntaktischen Strukturen etc. Als weitere Vorteile nennt McEnery (2003) die Multifunktionalität der Daten sowie die explizite Formulierung und objektive Erfassung von Ergebnissen einer Analyse.

Korpusannotationen können automatisch, von Hand oder in einer kombinierten Form erstellt werden. Für Aufgaben wie das POS-Tagging oder Lemmatisieren für große Sprachen wie Englisch, Französisch, Deutsch oder Spanisch können Annotationen automatisch erstellt werden. Häufig aber wird das Ergebnis der maschinellen Annotation von Hand kontrolliert und erfolgt damit halbautomatisch, was immer noch ein schnelleres Annotieren als komplett von Hand ermöglicht. Einige Bereiche erfordern aber auch rein manuelle Erstellung der Annotationen, etwa zur Darstellung anaphorischer und kataphorischer Relationen (McEnery 2003). Als Argument gegen manuelle und halbautomatische Annotation ist vorgebracht worden, dass diese nur eine geringe Konsistenz aufweisen könne, da menschliche Annotatoren vergleichbare Stellen im Korpus nicht immer übereinstimmend annotieren. Es wurden jedoch Untersuchungen unternommen, die gezeigt haben, dass bei geschulten Annotatoren der leichte Rückgang der Konsistenz mehr als kompensiert wurde durch eine gesteigerte Genauigkeit der Annotationen (McEnery 2003 : 457).

3.4 Lernen und Evaluieren

Annotierte Korpora ermöglichen es Computerprogrammen, die Intuitionen von Experten, die die Annotationen erstellt oder verbessert haben, in Bezug zu den Texten zu setzen und so menschliche Intuitionen zu reproduzieren. Ein Beispiel hierfür wäre etwa ein auf Hidden-Markov-Modellen basierender POS-Tagger, der aus annotierten Korpora lernt und dadurch in die Lage versetzt wird, neue Texte zu annotieren (siehe etwa Manning & Schütze 1999). Analog ist das nachfolgend beschriebene Verfahren: aus klassifizierten Texten wird induktiv gelernt und anhand des erworbenen Wissens werden neue Texte klassifiziert. In diesem Sinn ist das beschriebene Verfahren ein korpusbasiertes, induktives Lernverfahren (siehe Abschnitt 5.1). Ein solches maschinelles Lernen von Strukturen aus Korpora wird auch als *Grammatik-Bootstrapping* bezeichnet. Die obigen Beispiele (POS-Tagging und Textklassifikation) deuten an, dass es in der maschinellen Sprachverarbeitung und der Computerlinguistik zahlreiche Bereiche gibt, die als korpuslinguistische Verfahren charakterisiert werden können. Dies umfasst neben den oben genannten Beispielen etwa die Volltextsuche (*Information Retrieval*), die Mustersuche in Texten (*Text Data Mining* oder *Text Mining*) sowie verwandte Gebiete wie die Informationsextraktion.

Ein weiterer, wichtiger Einsatzbereich für Korpora bildet die gemeinsame Evaluierung von Systemen zur maschinellen Sprachverarbeitung durch die Verwendung eines gemeinsamen Korpus, wie etwa bei den *Message Understandig Conferences* (siehe etwa Grishman & Sundheim 1996). Im nachfolgend beschriebenen Verfahren wird ein dem Lernkorpus vergleichbares, kleineres Kor-

pus zur Evaluierung eingesetzt (siehe Abschnitt 5.4). Software, die die Arbeit mit annotierten Korpora zur Entwicklung und Evaluierung von sprachverarbeitenden Komponenten unterstützt wird als *Software Architecture for Language Engineering* bezeichnet (siehe Abschnitt 6).

4 Das Web als Basis für Korpora

Neben linguistisch aufbereiteten Korpora, auf die der Zugriff häufig beschränkt ist (etwa rechtlich oder technisch), steht mit dem Web eine große, öffentliche, maschinenlesbare Textsammlung zur Verfügung, die allerdings weder linguistisch organisiert (siehe Abschnitt 3.2) noch aufbereitet ist (siehe Abschnitt 3.3). Das Web ist dennoch eine attraktive Quelle für die Zusammenstellung von Korpora, dessen Erschließung aber abhängig von Werkzeugen ist, etwa zur Suche (z.B. Google) oder zur Kategorisierung (z.B. Delicious, s.u.); insbesondere Ressourcen mit Formen von sozialer Intelligenz – wie etwa die Kategorien in Delicious oder die kollaborative Arbeit an der Wikipedia – haben großes Potential als Quelle für das maschinelle Lernen in der Sprachverarbeitung. Auf dieser Grundlage wird im vorgestellten System Delicious zur Zusammenstellung von Korpora verwendet. Mit Delicious lassen sich Web-Bookmarks verwalten; diese können klassifiziert werden, wobei mehrere Klassen frei vergeben werden können (*tagging*); die Klassen lassen sich wiederum zu *bundles* zusammenfassen, d.h. ein *bundle* umfasst dann mehrere Klassen, denen je mehrere Webseiten zugeordnet sind (siehe auch Abb. 2). Konzeptuell entsprechen die *bundles* einem Korpus mit einem gewünschten *sampling frame*; die Klassen können als Korpusannotation betrachtet werden und die Grundlage des zu lernenden Wissens darstellen, und sind damit die menschliche Intuition, die von der Maschine reproduziert werden soll (vgl. Abschnitt 3.3).

Delicious stellt eine öffentliche, Web-basierte API für den Dienst bereit, für die es eine Java-API gibt, welche im vorgestellten System verwendet wird.[7] Daneben besteht die Möglichkeit des Exports in das Netscape-Bookmark-HTML-Format, das vom implementierten System als alternative Quelle der Links eingelesen werden kann. Zum Parsen der verlinkten Seiten und damit zum Einlesen der eigentlichen Texte wird mit NekoHTML[8] ein korrigierender HTML-Parser verwendet. Das Korpus wird also aus den Inhalten der in Delicious angegebenen Webseiten zusammengestellt. Ein solches Programm, das mehrere Dateien abruft und verarbeitet wird allgemein als *Crawler* bezeichnet. Ein Crawler, der aus dem Internet Informationen abruft und zusammenführt, wird auch als *Bot* oder spezieller als *Aggregator* bezeichnet (vgl. Heaton 2002). Im implementierten Verfahren wird zur Evaluierung ein Testkorpus aus Online-Nachrichten von Spiegel-Online verwendet. Zum Training des Systems wird dementsprechend ein Korpus verwendet, das aus ebensolchen Artikeln besteht (vgl. Abschnitt 3.2). Die Ausgewogenheit wird in diesem Fall durch die Tatsache sichergestellt, dass Artikel verschiedenener Ressorts im Lern-

7 `http://sourceforge.net/projects/delicious-java/`
8 `http://people.apache.org/~andyc/neko/doc/`

Abbildung 2: Screennshot der in Delicious zusammengestellten Korpora für Training und Evaluierung des Systems.

korpus vorhanden sind (siehe auch Abb. 2). Es handelt sich hierbei um eine experimentelle Umsetzung der geschilderten Gedanken in kleinem Maßstab; bei einer praxisnahen Umsetzung würden etwa Artikel verschiedener Zeitungen verwendet um Ausgewogenheit und Repräsentativität des Korpus zu gewährleisten.

5 Klassifikation mit Paradigmen

Grundgedanke des implementierten Verfahrens ist es, Paradigmen als Merkmale zur Klassifikation zu verwenden. So könnte etwa das Paradigma [Eis, Chips] als Merkmal für die Klasse 'Essen' verwendet werden. Würde nun etwa Text (1) klassifiziert, würde das Paradigma [Eis, Chips] gefunden und der Text entsprechend klassifiziert.

(1) *Hans mag Eis. Anna mag Chips nicht. Ich mag Chips gern.*

Text (2) dagegen würde nicht der Klasse 'Essen' zugeordnet, da hier zwar die Wörter *Chips* und *Eis* vorkommen, jedoch nicht im gleichen Kontext. Dies ermöglicht eine Form von lexikalischer Disambiguierung, hier etwa für *Chips* als 'Kartoffelchips' im Gegensatz zu 'Mikrochips'. Das Verfahren kann aufgrund der symbolischen Natur des verwendeten Wissens – nämlich der

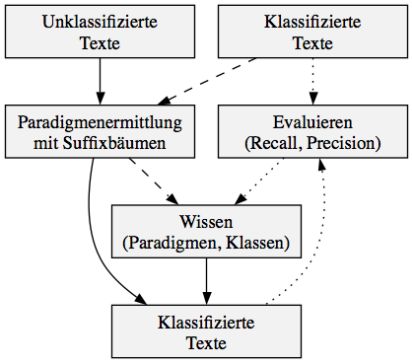

Abbildung 3: Wissenserwerb, Klassifikation und Evaluierung im implementierten Verfahren; die gestrichelte Linie beschreibt den Wissenserwerb, die durchgezogene Linie die Klassifikation und die gepunktete Line die Evaluierung.

Paradigmen – als symbolisches Verfahren bezeichnet werden. Eine Übersicht des Systems und der einzelnen Schritte findet sich in Abb. 3.

(2) *In Dresden produziert Infineon Chips. Die Produktion liegt zur Zeit auf Eis.*

5.1 Maschinelles Lernen

Der Wissenserwerb des Systems stellt damit eine Form von maschinellem Lernen dar. Genaugenommen handelt es sich um ein unüberwachtes, induktives, exemplarbasiertes Lernverfahren mit einer symbolischen Wissensrepräsentation. Abb. 4 stellt das Verfahren in den Kontexts des Lernbegriffs in der Kybernetik. Verschiedene maschinelle Lernverfahren zur Textklassifikation (siehe Brückner 2001) verwenden numerische Wissenrepräsentationen. Für die Verwendung des beschriebenen Verfahrens in einem solchen Kontext wären dies Werte, die ausdrücken, ob und wie sehr ein Paradigma für eine Klasse relevant ist, repräsentiert in einen Merkmalsvektor pro Klasse, mit Werten, die die Relevanz der Klasse für jedes Paradigma kennzeichnen.

5.2 Paradigmen und Suffixbäume

Zur Ermittlung der Paradigmen im Lernkorpus und in den zu klassifizierenden Texten kommen Suffixbäume zum Einsatz,[9] die eine effiziente Ermittlung der Paradigmen ermöglichen. Ein Suffixbaum ist mit linearer Laufzeit- und Speicherplatzkomplexität konstruier- und nutzbar,[10]

9 `http://stnl.sourceforge.net/`

10 Weitere Informationenen zu Konstruktion und Nutzung von Suffixbäumen findet sich etwa in Gusfield (1997).

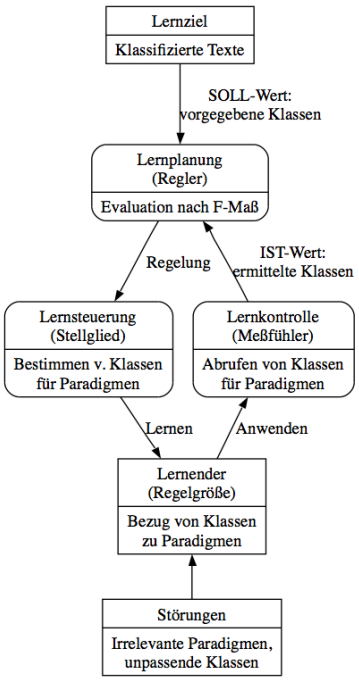

Abbildung 4: Kybernetischer Regelkreis des Lernens (von Cube 1965), im unteren Teil der Kästen ergänzt für das beschriebene Verfahren.

während etwa eine Ermittlung von Paradigmen über einen paarweisen Vergleich aller Sätze über die Levenshtein-Distanz eine quadratische Laufzeitkomplexität aufweist. Ein Suffixbaum mit Wörtern als Symbole enthält Informationen über Paradigmen im repräsentierten Text in seiner Struktur: Da mehrfach auftretende Teilstrings im Baum nur einmal vorkommen und er an den Stellen verzweigt, an denen sich die Sätze unterscheiden, stehen alle Beschriftungen von Kanten, die von inneren Knoten ausgehen, zueinander in paradigmatischer Relation. In einem Suffixbaum können Paradigmen mit gemeinsamen Kontexten *vor* den Paradigmen (siehe Abb. 5), in einem Suffixbaum für den umgekehrten Text – einem Präfixbaum – können Paradigmen mit gemeinsamem Kontext *nach* den Paradigmen ermittelt werden (siehe Abb. 6).

5.3 Programmstruktur und Vorgehen

Die Struktur der Implementierung basiert auf dem grundlegenden Aufbau aus Wissenserwerb und Klassifikation; mit letzterer ist die Evaluierung assoziiert. Hinzu kommen Klassen für das eigentliche Programm, das Crawling, die Vorverarbeitung in Form von HTML-Parsing und Stopwortlisten-Filtern sowie eine Klasse zur Repräsentation von kategorisierten Texten. Ein

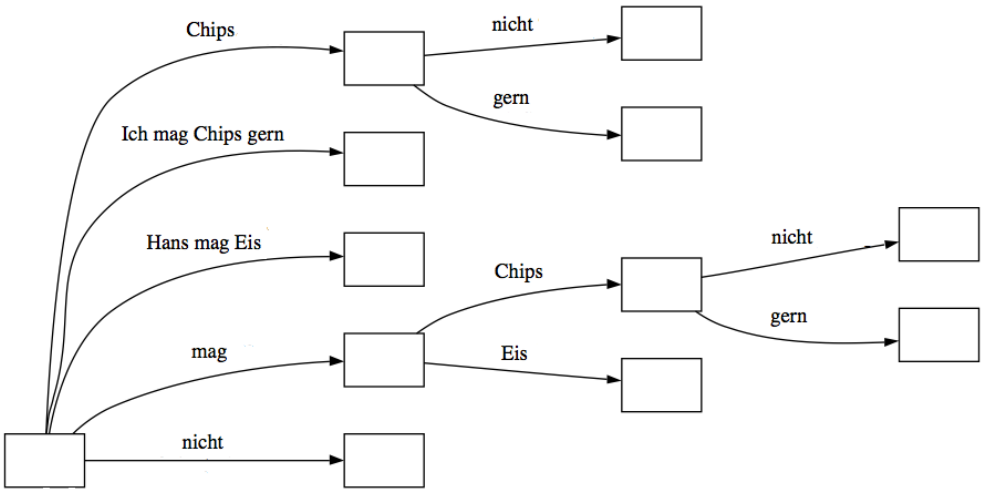

Abbildung 5: Ausschnitt des Suffixbaums für Text (1)

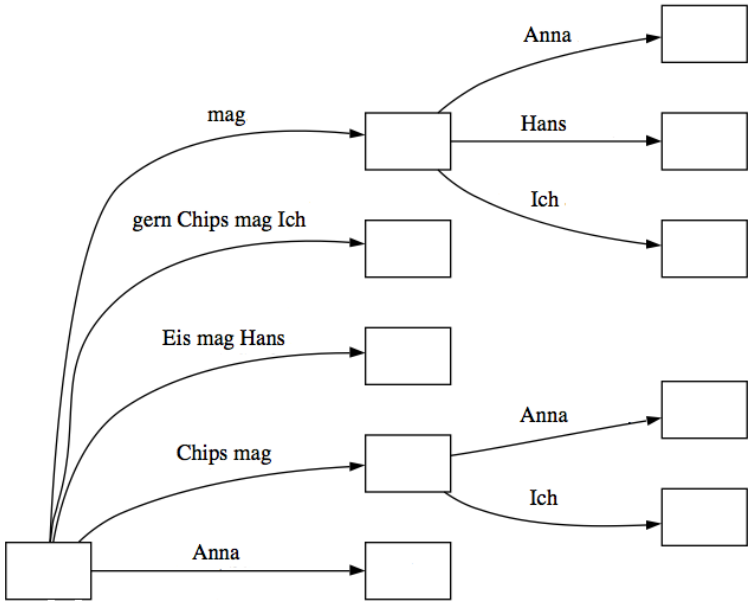

Abbildung 6: Ausschnitt des Präfixbaums für Text (1)

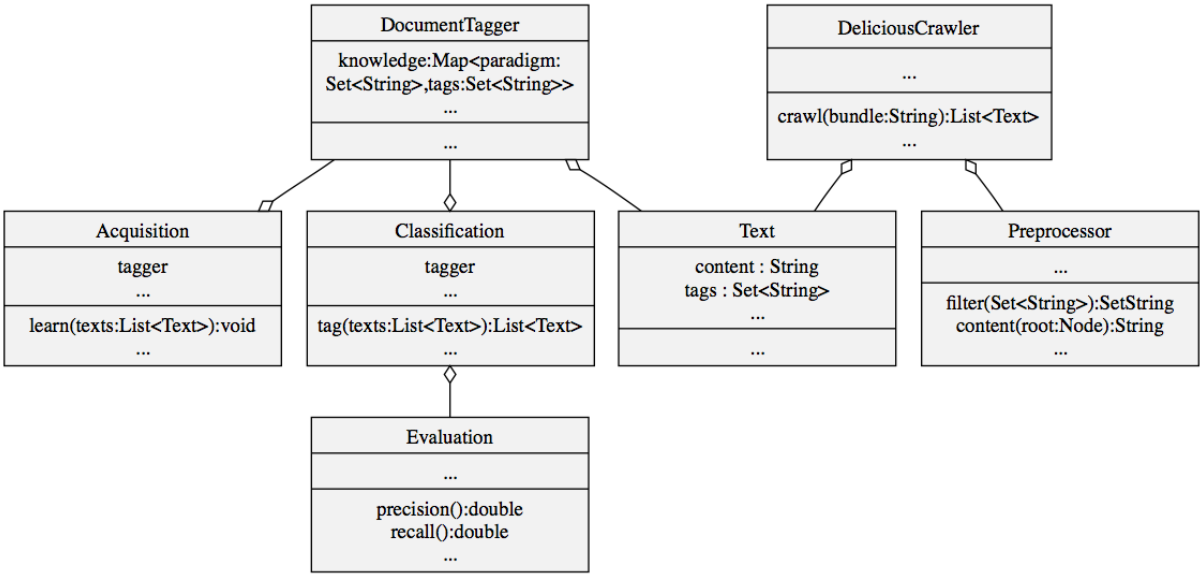

Abbildung 7: UML-Klassendiagramm der Implementierung

UML-Klassendiagramm der beschriebenen Implementierung findet sich in Abb. 7.

Das Vorgehen besteht im Wissenserwerb aus den drei in Abschnitt 2 beschriebenen Schritten: dies sind erstens die Merkmalsberechnung, hier in Form der Ermittlung von Paradigmen mit einem Suffixbaum, zweitens die Merkmalsauswahl, hier durch Filtern der Paradigmen mithilfe von Stopwortlisten und schließlich drittens die Modellbildung, hier das Ablegen der Paradigmen und der für diese relevanten Klassen in einer *Map*; hierbei bilden die Paradigmen die Schlüssel und die Klassen die Werte, etwa $[Chips, Eis] \rightarrow [Essen, Kino]$ oder $[Chips, Monitore] \rightarrow [Computer]$. Die Klassifikation besteht ebenfalls aus drei Schritten: erstens der Ermittlung der Paradigmen im zu klassifizierenden Text, zweitens wird für jede mögliche Klasse die beste Übereinstimmung eines der für diese Klasse relevanten Paradigmen mit einem Paradigma im zu klassifizierenden Text ermittelt. Bei einem Vergleich etwa von $[Chips, Eis]$ mit $[Chips, Cola]$ wäre die Übereinstimmung 50%. Drittens schließlich werden alle Klassen, deren beste Übereinstimmung einen bestimmten Schwellenwert überschreitet, als Ergebnis der Klassifikation ausgewählt; im Versuch wurde als Schwellenwert 50% verwendet.

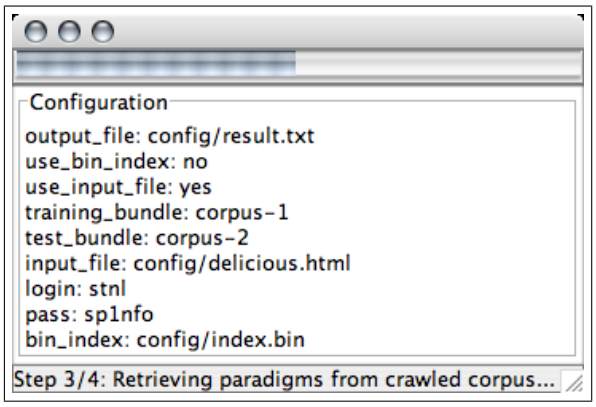

Abbildung 8: Statusanzeige

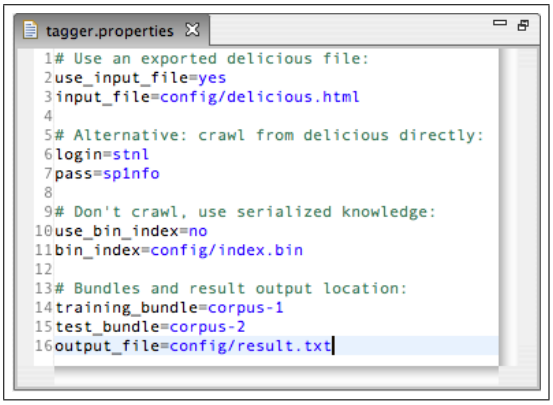

Abbildung 9: Properties-Datei

5.4 Evaluierung

Zur Evaluierung wurden zwei Korpora in einem eigenen Delicious-Account eingerichtet.[11] Die Software ist online verfügbar,[12] in Form einer ausführbaren Jar-Datei und als Quelltext. Nach Doppelklick der Jar-Datei öffnet sich ein Fenster, das die verwendeten Konfigurationswerte und den Fortgang der Analyse anzeigt (siehe Abb. 8). Dabei werden die angegebenen Texte zum Lernen und zum Klassifizieren verwendet, die Ergebnisse der Klassifikation werden in eine Textdatei geschrieben. Die Konfigurierung erfolgt über eine Java Properties-Datei (siehe Abb. 9). Eine solche *dynamische Konfigurierung* hat softwaretechnische Vorteile gegenüber einem Einbau der Details im Programmcode und kann als eine Form von Metaprogrammierung gesehen werden (siehe auch Hunt & Thomas 2003 : 135ff.). Dies erleichtert etwa eine Verwendung in anderen Zusammenhängen, etwa in einer SALE (siehe Abschnitt 6).

Die Evaluierung erfolgt nach den Standardmaßen Precision, Recall und F-Maß. Die Precision drückt dabei hier aus, wie viele der ermittelten Klassen korrekt sind; der Recall drückt aus, wie viele der zu ermittelnden Klassen auch ermittelt wurden; das F-Maß erlaubt eine einheitliche Betrachtung von Recall und Precision. Über die Bildung des Mittels der Ergebnisse lässt sich das Verfahren selbst evaluieren. Versuche mit kleinen Korpora ergaben einen Recall von 80-90% und eine Precision von 40-50%. Ergebnisse der ersten Evaluierung mit 16 Spiegel-Online-Artikeln, ca. 10.000 Paradigmen als Merkmale und einer Klasse pro Text, ermittelt in einem Lernkorpus aus 120 Spiegel-Online-Artikeln finden sich in Abb. 10; weitere Experimente mit mehreren Klassen pro Dokument und einem anderen Lernkorpus von etwa doppelter Größe (211 Artikel), resultierend in etwa der doppelten Anzahl von Merkmalen (18.452) sowie einem Testkorpus mit

11 `http://del.icio.us/stnl`
12 `http://stnl.sourceforge.net/applications/`

Text	Recall	Precision	F-Maß
1	0	0	0
2	1	0,25	0,4
3	0	0	0
4	1	0,25	0,4
5	1	0,5	0,67
6	1	0,5	0,67
7	1	1	1
8	1	0,5	0,67
9	1	0,5	0,67
10	1	0,5	0,67
11	1	0,35	0,5
12	1	0,5	0,67
13	1	0,5	0,67
14	1	0,25	0,4
15	1	0,25	0,4
16	1	1	1
ϕ	0,88	0,43	0,55

Abbildung 10: Ergebnisse der ersten Evaluierung mit einer Klasse pro Dokument

ebenfalls verdoppelter Größe (31 Artikel) lieferten vergleichbare, leicht verbesserte Ergebnisse mit einem Recall von 91,7% und einer Precision von 52,1%. Bei heterogenen Korpora ist die Qualität geringer, was aber angesichts der grundsätzlichen Forderungen an Korpora (siehe Abschnitt 3.2) zu erwarten ist.

Es sind verschiedenen Ansätze für eine Verbesserung des Verfahrens denkbar: Als Verbesserungen, die die Sprachunanhängigkeit des Verfahrens erhalten sind eine Verbesserung der Qualität der Paradigmen, eine Anpassung von Größe und Beschaffenheit der Korpora sowie eine Anpassung des Algorithmus denkbar. Die Qualität der Paradigmen könnte etwa durch verbessertes Filtern oder eine Berücksichtigung von Mehrwort-Paradigmen geschehen. Die Korpora könnten aus anderen Quellen stammen, größer sein und anders vorverarbeitet[13] werden. Der Algorithmus könnte etwa in der Form angepasst werden, dass nicht ein fester Schwellenwert verwendet wird, sondern z.B. die besten 5% ausgewählt werden. Der Wert könnte darüber hinaus abhängig von Ergebnissen der Evaluierung angepasst werden (vgl. *Regelung* in Abb. 4). Als weitere, meist sprachspezifisch implementierte Verbesserungen wären etwa eine Stammformenreduktion und POS-Tagging denkbar. Eine Stammformenreduktion würde eine Berücksichtigung aller Wortformen für die Paradigmenermittlung ermöglichen und sollte damit die Qualität der Paradigmen verbessern, während POS-Tagging eine Unterscheidung verschiedener Wortarten mit derselben Wortform ermöglicht, was zu einer Verbesserung der Precision beitragen könnte. Für

13 Im implementierten Verfahren werden etwa lediglich die Inhalte aller Paragraph-Elemente der HTML-Dateien ausgelesen.

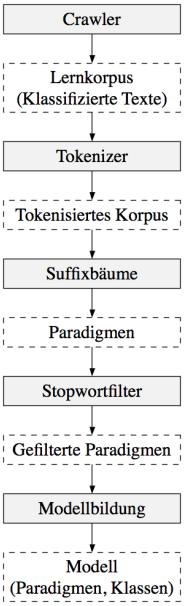

Abbildung 11: Wissenserwerb

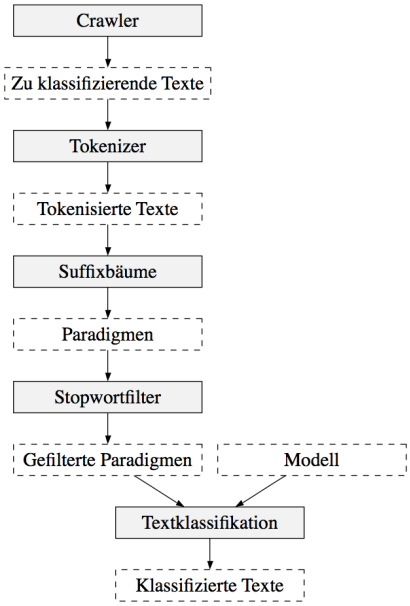

Abbildung 12: Klassifikation

eine Stammformenreduktion wäre ein korpusbasiertes Verfahren mit Suffixbäumen denkbar, das ähnlich wie das hier beschriebene Verfahren prinzipiell sprachunabhängig wäre.[14] Ein weiterer möglicher Ansatz bestünde in der Nutzung semantischer Relationen, etwa zur Berücksichtigung von Hyperonymen beim Vergleich von Paradigmen.

6 Software Architecture for Language Engineering

Der größte Teil der genannten Verbesserungen erfordert eine Integration mit anderen sprachverarbeitenden Komponenten. Zu diesem Zweck bietet sich der Einsatz einer *Software Architecture for Language Engineering* (SALE) an, einer Infrastruktur für die maschinelle Sprachverarbeitung (siehe Cunningham & Bontcheva 2006 sowie Köhler 2005). Eine solche Infrastruktur besteht aus Frameworks, Referenzarchitekturen und einer Entwicklungsumgebung und bildet damit eine Art Werkzeugkasten für die computerlinguistische Arbeit. Beispiele für SALEs sind etwa UIMA oder GATE. An der Abteilung für Sprachliche Informationsverarbeitung[15] am Institut für Linguistik der Universität zu Köln wird unter dem Namen *Tesla* (Text Engineering Software Laboratory) ein vergleichbares System entwickelt. Schwerpunkt der Arbeit bilden hier Wiederverwertbarkeit

14 Unter `http://stnl.sourceforge.net/applications/` findet sich eine Umsetzung eines solchen Verfahrens für das Spanische.

15 `http://www.spinfo.uni-koeln.de`

von Analysekomponenten und Ergebnissen durch dynamische Annotation (siehe Benden & Hermes 2004), die IDE-Integration in Eclipse sowie die Verteilbarkeit der Analysearbeit in einer Java-EE-Architektur zur Durchführung rechenaufwändiger Verfahren. Es fällt auf, dass bereits in einer solch kleinen, experimentellen Implementierung viele individuelle Komponenten identifizierbar sind, die verbessert und zu Evaluierungszwecken gegen andere ausgetauscht werden könnten. Im vorgestellten Verfahren etwa sind beim Wissenserwerb (siehe Abb. 11) alle der Modellbildung und bei der Klassifikation (siehe Abb. 12) alle der eigentlichen Textklassifikation vorgelagerte Schritte prinzipiell austauschbar und nicht spezifisch für das Verfahren. Denkbar wären hier etwa im Rahmen von Tesla der Einsatz von SPre zur Vorverarbeitung (Benden & Hermes 2004, Hermes & Benden 2005) und SOG zur Paradigmenbildung (Schwiebert 2005, Schwiebert & Rolshoven 2006).

7 Fazit

In der vorliegenden Arbeit wurde beschrieben, wie sich das Web als Grundlage für den Aufbau spezifischer Korpora für maschinelle Lernverfahren in der Sprachverarbeitung einsetzen lässt, implementiert in einem System zur Textklassifikation. Paradigmen als Klassifikationsmerkmale erlauben dabei eine Form von lexikalischer Disambiguierung durch Berücksichtigung von Kontexten. Zur effizienten Ermittlung der Paradigmen wird mit Suffixbäumen eine bisher in der Computerlinguistik wenig verwendete, vielseitige Datanstruktur eingesetzt. Eine experimentelle Evaluierung ergibt einen Recall von 80-90% und eine Precision von 40-50%. Verbesserte Vorverarbeitung, etwa in Form einer Stammformenreduktion oder POS-Tagging, könnte ohne großen Aufwand zu Verbesserungen führen; im Kontext möglicher Verbesserungen wurden Motivation und Mehrwert einer Integration verschiedener sprachverarbeitender Komponenten in einer *Software Architecture for Language Engineering* (SALE) beschrieben.

Literatur

Benden, C. & J. Hermes: 2004, 'Präprozessierung mit Nebenwirkungen: Dynamische Annotation', in E. Buchberger (ed.), *Beiträge zur 7. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS)*, Riegelnik, Wien, pp. 25–28.

Brückner, T.: 2001, 'Textklassifikation', in K. U. Carstensen, C. Ebert, E. Endriss, S. Jekat, R. Klabunde & H. Langer (eds.), *Computerlinguistik und Sprachtechnologie*, Spektrum, Heidelberg, Berlin, pp. 442–447.

von Cube, F.: 1965, *Kybernetische Grundlagen des Lernens und Lehrens*, Klett.

Cunningham, H. & K. Bontcheva: 2006, 'Computational Language Systems, Architectures', in K. Brown, A. H. Anderson, L. Bauer, M. Berns, G. Hirst & J. Miller (eds.), *The Encyclopedia of Language and Linguistics*, second edn., Elsevier, München.

Evert, S. & A. Fitschen: 2001, 'Textkorpora', in K. U. Carstensen, C. Ebert, E. Endriss, S. Jekat, R. Klabunde & H. Langer (eds.), *Computerlinguistik und Sprachtechnologie*, Spektrum, Heidelberg, Berlin, pp. 369–376.

Goller, C., J. Löning, T. Will & W. Wolff: 2000, 'Automatic document classification: A thorough evaluation of various methods', *7. Internationales Symposium für Informationswissenschaft* .

Grishman, R. & B. Sundheim: 1996, 'Message Understanding Conference - 6: A Brief History', in *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, Kopenhagen, pp. 466–471.

Gusfield, D.: 1997, *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*, Cambridge University Press.

Heaton, J.: 2002, *Programming Spiders, Bots and Aggregators in Java*, Sybex.

Hermes, J. & C. Benden: 2005, 'Fusion von Annotation und Präprozessierung als Vorschlag zur Behandlung des Rohtextproblems', in B. Fisseni, H.-C. Schmitz, B. Schröder & P. Wagner (eds.), *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen. Beiträge zur GLDV-Tagung 2005 in Bonn* (*Sprache, Sprechen und Computer* 8), Lang, Frankfurt a.M., pp. 78–90.

Hunt, A. & D. Thomas: 2003, *Der Pragmatische Programmierer*, Hanser, München, Wien.

Köhler, R.: 2005, 'Korpuslinguistik - zu wissenschaftstheoretischen Grundlagen und methodologischen Perspektiven', *GLDV-Journal for Computational Linguistics and Language Technology* **20**(2), 1–16.

Lyons, J.: 1968, *Introduction to Theoretical Linguistics*, University Press, Cambridge.

Manning, C. D. & H. Schütze: 1999, *Foundations of statistical natural language processing*, MIT Press, Cambridge, MA, USA.

McEnery, T.: 2003, 'Corpus Linguistics', in R. Mitkov (ed.), *The Oxford Handbook of Computational Linguistics*, Oxford Handbooks in Linguistics, Oxford University Press, Oxford, pp. 448–463.

McEnery, T. & A. Wilson: 1996, *Corpus Linguistics*, Edinburgh University Press, Edinburgh.

Schwiebert, S.: 2005, 'Entwicklung eines agentengestützten Systems zur Paradigmenbildung', in B. Fisseni, C, B. Schroder & P. Wagner (eds.), *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen. Beiträge zur GLDV-Tagung 2005 in Bonn* (*Sprache, Sprechen und Computer* 8), Lang, Frankfurt a.M., pp. 633–646.

Schwiebert, S. & J. Rolshoven: 2006, 'SOG: Ein selbstorganisierender Graph zur Bildung von Paradigmen', in Rapp, Reinhard, Sedlmeier & Zunker-Rapp (eds.), *Perspectives on Cognition. A Festschrift for*, Pabst Science Publishers, Lengerich.