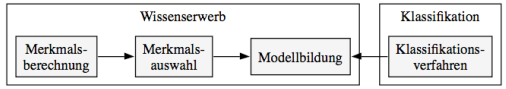
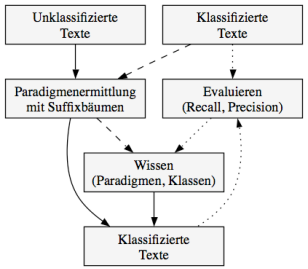
Exemplarbasierte Textklassifikation: Handout zum Referat im Hauptseminar "Intelligente Systeme", WS 2006-2007

Textklassifikation: Automatische Zuordnung von Texten zu vordefinierten Kategorien. Zwei Hauptkomponenten: Wissenserwerb (bestehend aus Merkmalsberechnung, Merkmalsauswahl und Modellbildung) und die eigentliche Klassifikation:



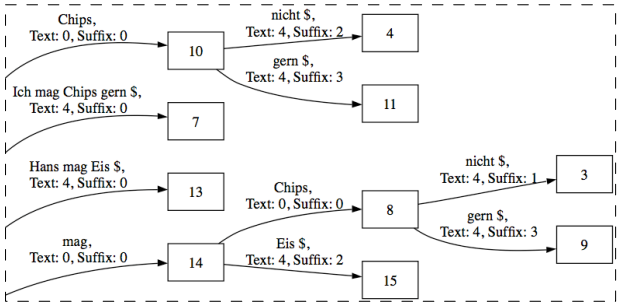
Unser Grundgedanke: Texte haben das gleiche Thema, wenn sie die gleichen Wörter in den gleichen Kontexten enthalten (d.h. wenn sie ähnliche Paradigmen enthalten).

z.B. wenn in einem Text *Chips* und *Eis* im gleichen Kontext auftauchen, geht es vermutlich um Essen. In einem Text über Mikrochips kommt das Wort *Chips* auch vor, allerdings nicht im gleichen Kontext wie *Eis*, daher sollte der Text nicht der gleichen Klasse zugeordnet werden.



Als Eingabe dienen schon klassifizierte Texte, daraus wird gelernt (gestrichelte Linie). Dann können unklassifizierte Texte klassifiziert werden (durchgezogene Linie). Zur Evaluation werden klassifizierte Texte neu klassifiziert und das Ergebnis mit der ursprünglichen Klassifikation verglichen (gepunktete Linie).

Suffixbäume enthalten in ihrer Struktur Paradigmen: Beschriftungen der Kanten von inneren Knoten zu ihren Kindern stehen zueinander in paradigmatischer Beziehung (siehe auch http://www.spinfo.uni-koeln.de/space/Forschung/Weitere/Suffixbäume).



Wissenserwerb und Klassifikation

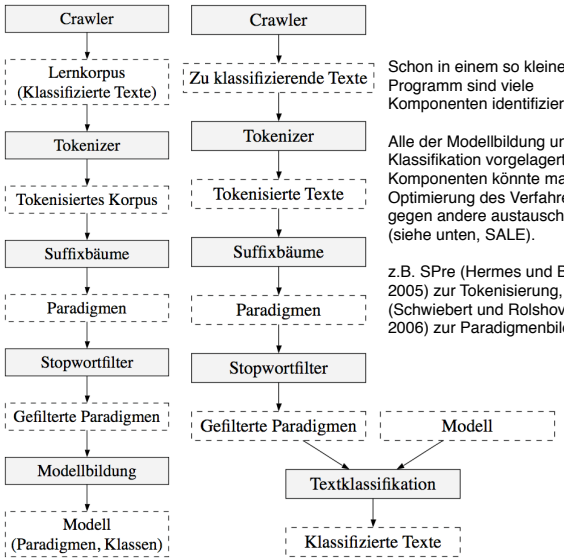
Merkmalsberechnung: Erstellung von Paradigmen durch Suffixbäume.

Merkmalsauswahl: Die besten Paradigmen (Filtern durch Stopwort-Listen).

Modellbildung: Bezug von Klassen zu Paradigmen.

Der Wissenserwerb ist eine Form von masch. Lernen, hier: induktives, exemplarbasiertes Lernen (vgl. WdKW).

Verbreitete Verfahren zum maschinellen Lernen verwenden numerische Repräsentationen der Merkmale, hier etwa: ob und wie sehr ein Paradigma für eine Klasse relevant ist (ein Merkmalsvektor pro Klasse).

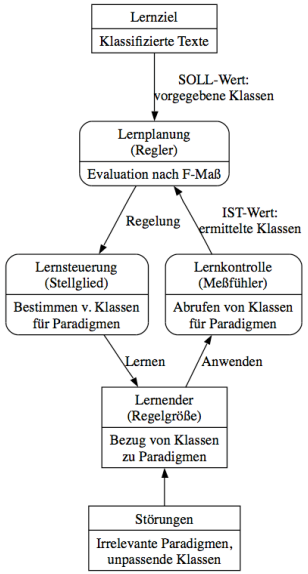


Schon in einem so kleinen Programm sind viele Komponenten identifizierbar.

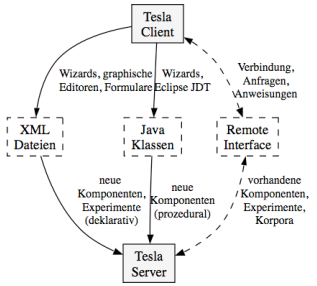
Alle der Modellbildung und der Klassifikation vorgelagerte Komponenten könnte man zur Optimierung des Verfahrens gegen andere austauschen (siehe unten, SALE).

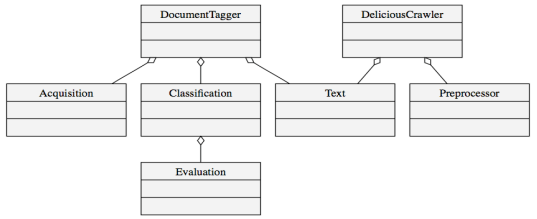
z.B. SPre (Hermes und Benden 2005) zur Tokenisierung, SOG (Schwiebert und Rolshoven 2006) zur Paradigmenbildung.

Unser Verfahren in der Darstellung von Felix v. Cube, Kybernetische Grundlagen des **Lernens**:



SALE, z.B. Tesla: Standardisierte Mechanismen für die Kommunikation zwischen Komponenten, Verwendung offener Standards (Java, XML), Reduzierung von Integrationskosten, Bereitstellung eines erweiterbaren Basis-Sets von Komponenten, die vom Benutzer (z.B. zu Vergleichszwecken) ausgetauscht werden können, Bereitstellung einer Entwicklungsumgebung zur Erleichterung der Implementierung von sprachverarbeitenden Komponenten und Anwendungen (siehe auch http://www.spinfo.uni-koeln.de/space/Forschung/Tesla).





Die **Programmstruktur** spiegelt grundsätzlich den Aufbau von Textklassifikationssystemen wie oben beschrieben dar, erweitert um Klassen zum Crawling und zur Vorverarbeitung (HTML, Filter).

Evaluation

Test mit kleinem Spiegel-Online-Korpus aus 120 Artikeln, Test mit 16 Artikeln, 10.000 Merkmalen (Paradigmen), zusammengestellt in Delicious (http://del.icio.us):

	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
Summe		14	6,85	8,78
Mittel		0,88	0,43	0,55

Literatur

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.

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.

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

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

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.

Schwiebert, Stephan und Jürgen Rolshoven (2006): 'SOG: Ein selbstorganisierender Graph zur Bildung von Paradigmen'. In: Rapp, Reinhard, Sedlmeier & Zunker-Rapp: *Perspectives on Cognition*. A Festschrift for Manfred Wettler. Lengerich: Pabst Science Publishers.

Dozent: Prof. Dr. Jürgen Rolshoven, Referenten: Valentina Rahmanian, Fabian Steeg, Sonja Subicin, Sprachliche Informationsverarbeitung, Universität zu Köln