Zeitschrift: bulletin.ch / Electrosuisse
Herausgeber: Electrosuisse
Band: 94 (2003)
Heft: 1

Artikel: Informationsbeschaffung im Internet
Autor: Jantke, Klaus P.
DOI: https://doi.org/10.5169/seals-857508

Informationsbeschaffung im Internet

Lerntechnologien für die Extraktion von Information aus semistrukturierten Dokumenten

Wissen verbirgt sich in enormem Umfang in betrieblichen Dokumenten und im Internet. Aber wie soll man an das Wissen herankommen? Die effiziente Extraktion von Wissen kann nicht von Hand erfolgen. Computerprogramme müssen Information aus Dokumenten automatisch extrahieren und in der vom Benutzer gewünschten Form ablegen. Die induktive Inferenz formaler Sprachen erweist sich dabei als Kerntechnologie, wenn es darum geht, solche Extraktionsprogramme automatisch zu lernen und im Dialog mit dem Benutzer den Bedürfnissen anzupassen. Die Technologie ist nicht auf HTML beschränkt, sondern funktioniert auch für Formate wie LaTeX, XML und PDF.

Klaus P. Jantke

Unter induktiver Schlussfolgerung versteht man die Herleitung einer allgemeinen Regel aus einer Reihe von bekannten Tatsachen. Analog dazu behandelt die sogenannte *induktive Inferenz* (vgl. engl. inference, Schlussfolgerung) Probleme, bei denen die Hauptschwierigkeit darin besteht, aus unvollständiger Information zu lernen. Dabei geht es vor allem darum, Verfahren zu entwickeln und zu implementieren, die erlauben, Computern das induktive Lernen beizubringen. Schon vor 35 Jahren wurde nachgewiesen, dass bereits relativ einfache Aufgabenstellungen, die zum Lernen formaler Sprachen gehören, algorithmisch unlösbar sind [1]. So gibt es zum Beispiel kein Verfahren, das in der Lage wäre, ganz beliebige reguläre Sprachen (Kasten) zu lernen, wenn aus der jeweiligen Zielsprache nur positive Beispiele[1] vorliegen[2].

Die Aufgabe, reguläre Sprachen – und eventuell sogar Sprachen höherer Komplexität – zu lernen, ist eine der Teilaufgaben, die sich stellt, wenn man computerunterstützt Information aus Internetquellen extrahieren will. Zur Bewältigung dieses an sich unlösbaren Problems wurde das Forschungs- und Entwicklungsprojekt LExIKON[3] lanciert, wobei LExIKON für «Learning for Extraction of Information respectively Knowledge from open Networks» steht. Das Ziel war, Information auf neuartige Art und Weise aus dem Internet – mittlerweile werden auch Quellen in XML und PDF verwendet – zu extrahieren.

Informationsextraktion am Beispiel illustriert

Im Internet findet man zahlreiche Bibliografien. In vielen Fällen, wie etwa bei der COLT-Bibliografie[4], kann man Files ganz verschiedener Art herunterladen (linkes Fenster in Bild 1a).

Natürlich kann man die gefundenen Texte vom Bildschirm abschreiben, einzeln per Drag and Drop kopieren oder aus ihnen die gewünschte Information mittels eines kleinen, selbst programmierten Programms extrahieren und in ein File oder in eine relationale Datenbank schreiben. Eine solche Programmieraufgabe ist ein typischer Fall für die künstliche Intelligenz. Anhand des Quelldokuments und des vom Benutzer vorgelegten Beispiels synthetisiert das LExIKON-System ein Extraktionsprogramm und führt dieses umgehend aus. Bild 1b zeigt die Eingabe eines Beispiels durch den Benutzer, Bild 1c das Extraktionsergebnis. Derartige Extraktionsprogramme werden *Wrapper* genannt.

Sollte der Benutzer mit dem Ergebnis nicht zufrieden sein, teilt er dies dem LExIKON-System mit, indem er exemplarisch aufzeigt, was ihm noch nicht gefällt: Er kann beispielsweise auf ein vom System extrahiertes Ergebnis zeigen und dieses zurückweisen oder aus dem Dokument ein weiteres Beispiel heranziehen, das bei der Extraktion übersehen worden ist. Dem LExIKON-System wird dann die Aufgabe gestellt, anhand der neuen Information einen verbesserten Wrapper zu generieren. Dieser Prozess kann iteriert werden.

Reguläre Sprache

Unter einer Sprache versteht man eine Menge von Wörtern über einem endlichen Alphabet. Nach dieser Definition ist die Menge {apfel, pflaume, birne, ein, zwei, drei} eine Sprache über dem Alphabet {a,b,c,d,e,f,i,l,m,n,p,r,u,w,z}.
Was aber ist eine reguläre Sprache? Wer sich schon mit Suchalgorithmen auseinander gesetzt hat, kennt den Begriff des regulären Ausdrucks. Reguläre Ausdrücke sind beispielsweise:

R_1 = a | b | c | d | e | f | i | l | m | n | p | r | u | w | z
R_2 = a | b*

wobei die Operatoren | für ODER und * für «beliebig repetiert» stehen (etwas vereinfachte Darstellung!).
Der Ausdruck R1 bedeutet somit «beliebiger Buchstabe unseres (verkürzten) Alphabets», der Ausdruck R2 «a ODER null, ein oder mehrere b». Man könnte auch schreiben R2= {a, b, bb, bbb,...}. Da sich R2 aus unendlich vielen Wörtern zusammensetzt, beschreibt R2 eine «unendliche» Sprache.
Eine reguläre Sprache ist somit eine Sprache, die sich durch einen regulären Ausdruck beschreiben lässt.

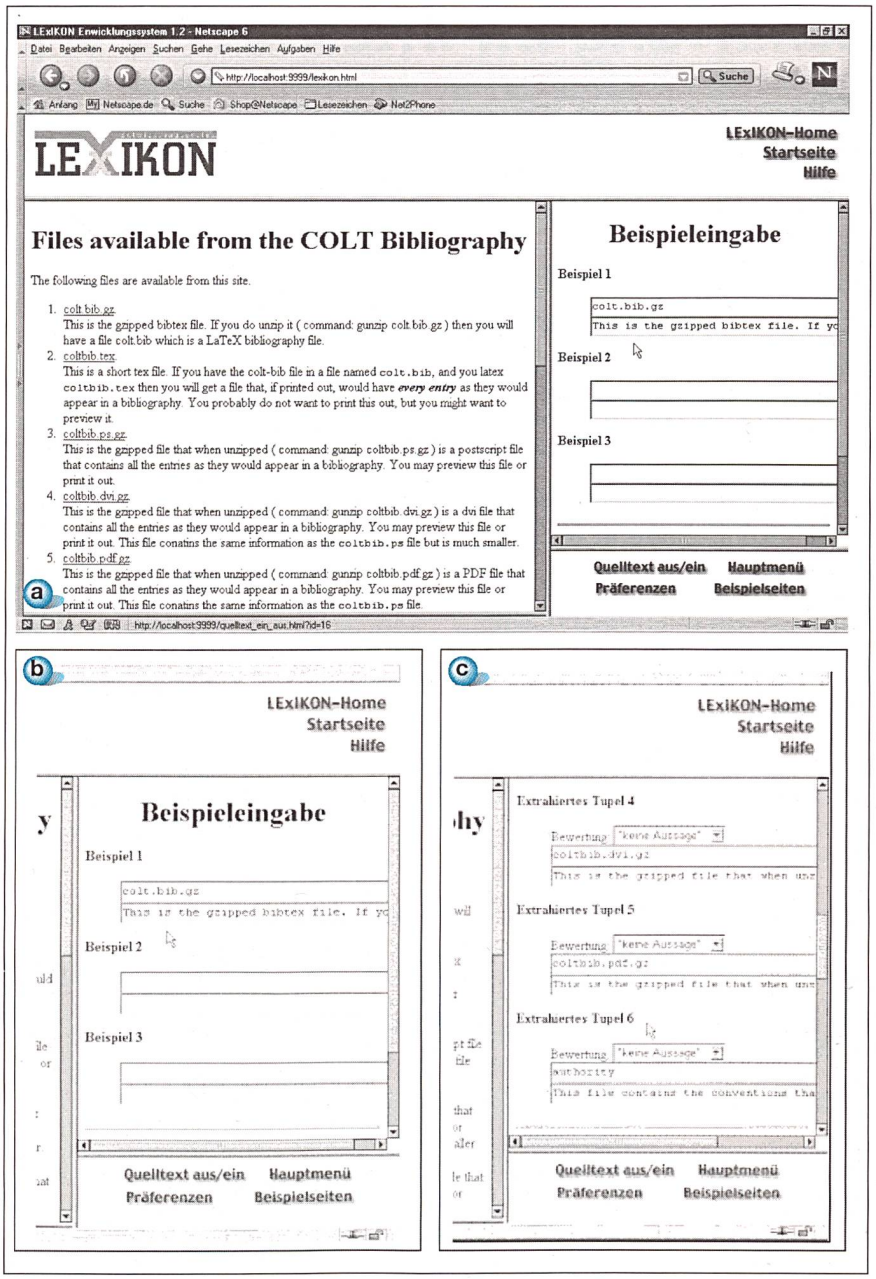

Bild 1 Eingabemaske des LExIKON-Systems

a: Gesamtansicht der Eingabemaske; b: rechte Spalte von Bild 1a mit Eingabe eines Suchtextes durch den Benutzer; c: von LExIKON gelieferte Information (rechte Spalte von Bild 1a nach Suchprozess)

Informationsextraktion und induktive Programmierung

Ein Anwender kann in einem Dokument beispielsweise an Daten wie Produktbezeichnungen, Händlernamen und Preisen interessiert sein. Jeder einzelne Datensatz ist dann ein Tripel mit drei solchen Angaben. Zeigt der Benutzer nun auf einige wenige solcher Beispiele, bekommt er vom System eine ganze Liste von Tripel zurückgeliefert – eine Relation.

Wenn der Eindruck entsteht, das LExIKON-System würde in einem Dialog mit dem Quelldokument eine Relation «lernen» (Bild 2), dann ist das nicht richtig. Das System lernt bzw. generiert keine Relationen, sondern Programme (Wrappers).

Wenn man die Folge der Interaktionen von Mensch und Maschine etwas genauer betrachtet, ergibt sich die Darstellung von Bild 3, wobei jeweils D_i ein Dokument und T_i eine Menge von bewerteten Beispielen bezeichnet. Intern wird durch das LExIKON-System in jedem Schritt ein Wrapper w_i gelernt, der auf das jeweils aktuelle Dokument D_i angewendet wird. Das Ergebnis dieser Extraktion wird ausgegeben. Man beachte, dass T_i nicht für ein Beispiel, sondern für eine Menge von Beispielen steht, die auf ein bestimmtes Dokument angewendet werden, bis der Wrapper in Bezug auf dieses Dokument eine befriedigende Arbeit leistet.

Der algorithmische Kern der Arbeit des LExIKON-Systems besteht demnach darin, dass Wrappers aus Informationsfolgen der Form (D_1,T_1), (D_2,T_2), (D_3,T_3), ... das gewünschte Extraktionsverhalten lernen. Bild 4 zeigt, wie Lernen und Extraktion zusammenspielen.

Dokumente und zugehörige Beispiele werden dem Lernmodul übergeben, welches auf der Grundlage des gewünschten Input-Output-Verhaltens einen neuen Wrapper konstruiert. Der neue Wrapper und das Dokument werden danach dem Extraktionsmodul übergeben, welcher dem Benutzer ein (mehr oder weniger gutes) Extraktionsergebnis liefert.

Neue Konzepte zur Fokussierung von Lernaufgaben

Es sei daran erinnert, dass die oben beschriebene Lernaufgabe im Allgemeinen algorithmisch unlösbar ist [2]. Um dennoch zu einer Lösung zu gelangen, wurde

Bild 2 Extraktion von relationalem Wissen vermittels LExIKON

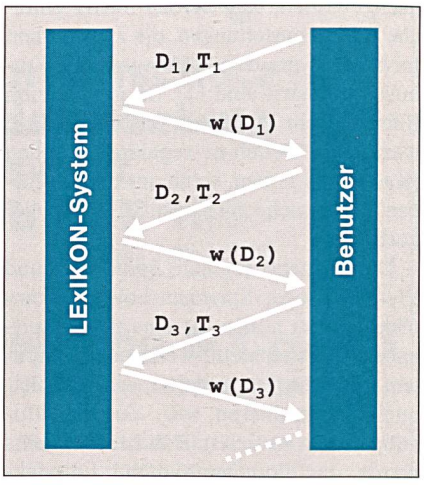

Bild 3 LExIKON-Lern-Dialog

TSM®

Informationstagung für Betriebselektriker

neu auch in Bern

Zürich Kongresshaus
Dienstag 11., Dienstag 25. oder Mittwoch 26. März 2003
Bern Kursaal
Donnerstag 20. März 2003

Tagungsort
Kongresshaus Zürich, Gotthardstrasse 5, 8002 Zürich
Kursaal Bern, Kornhausstrasse 3, 3000 Bern 25

Zielgruppen
Betriebselektriker mit Bewilligung für sachlich begrenzte Installationsarbeiten und deren Vorgesetzte, Kontrollorgane und weitere Elektrofachleute

Tagungsziel
Weiterbildung von Betriebselektrikern für ihre beruflichen Aufgaben, Pflichten und Verantwortung sowie Information über den neusten Stand der Technik (Vorschriften).

Tagungsleiter
Jost Keller
Weiterbildung,
Electrosuisse, Fehraltorf

Unterlagen
Tagungsband mit allen Referaten

Kosten
Teilnahmekarte (inbegriffen sind Tagungsband, Pausenkaffee, Mittagessen mit einem Getränk und Kaffee)

Nichtmitglieder	Fr. 400.--
Einzelmitglieder	Fr. 300.--
Mitarbeiter von Kollektivmitgliedern	Fr. 360.--
Mitarbeiter von Vertragskunden	Fr. 300.--

Anmeldung
Senden Sie das beigelegte Anmeldeblatt an Electrosuisse, Anlassorganisation, Luppmenstrasse 1, 8320 Fehraltorf, oder per Fax auf die Nr. 01 956 16 75.
Anmeldung über Internet:
www.sev-weiterbildung.ch
Anschliessend erhalten Sie eine Rechnung und die Teilnahmeunterlagen.

Für weitere Informationen wenden Sie sich bitte an die Electrosuisse,
Telefon direkt 01 956 11 75.

Electrosuisse
Luppmenstrasse 1
8320 Fehraltorf

Tel. +41 (0) 1 956 11 11
Fax +41 (0) 1 956 11 22
info@electrosuisse.ch
www.electrosuisse.ch

electrosuisse

Programm

09.00 Erfrischungen

09.30 Begrüssung
Serge Michaud
Electrosuisse, Fehraltorf

Einführung in die Themen
Jost Keller
Electrosuisse, Fehraltorf

Frequenzumrichter und Sanftanlasser in der Praxis
Yvan Bürgisser
Danfoss AG, Frenkendorf

Technik und Funktionsweise, typische Anwendungsgebiete, Installationshinweise sowie Hinweise auf EMV-Probleme mit Lösungsansätzen

Uebertragungskapazität in LAN Netzen
Roland Chervet
Nexans Schweiz AG, Cortaillod

Die technologischen Grenzen für Kupfer und Glasfaser. Technik allgemein, Interface-Geräte, typische Einsatzgebiete, Praxisbeispiele

10.45 Pause mit Erfrischungen

NIV aus der Sicht des Betriebselektrikers
Willi Berger
Electrosuisse, Fehraltorf

Erste Erfahrungen, Aufgaben und Pflichten, Schlussprotokoll, Sicherheitsnachweis

NIN - Umsetzung
André Moser
Electrosuisse, Fehraltorf

Umsetzung der Teile 3, 4 und 5: Wahl und Anordnung der Betriebsmittel aufgrund der Schutzmassnahmen und der charakteristischen Merkmale einer Anlage.
Teil 6: Messungen Erstprüfung

12.20 Mittagessen

14.00 Kalibrieren von Mess- und Prüfmitteln
Beat Schär
Electrosuisse, Bern

Sinn und Zweck, Sicherheitsaspekte, Festlegen der Kalibrierwürdigkeit und der notwendigen Intervalle

Arbeiten unter Spannung
Herbert Keller
Electrosuisse, Fehraltorf
Fridolin Kuhn
Glomar AG, Goldach

Wichtige Punkte aus Vorschriften und Richtlinien wie Starkstromverordnung, EN 50 110 sowie STI-Mitteilung 407. Praxisbeispiele, Demonstration mit isoliertem Werkzeug und geeigneten Körperschutzmitteln

Aus Unfällen lernen
Werner Berchtold
Electrosuisse, Fehraltorf

Aktuelle Unfallereignisse und wichtige Schlussfolgerungen für den sicheren Umgang mit Elektrizität

Organisation bei Nothilfeleistungen
Ruedi Lang
Electrosuisse, Fehraltorf

Erste Hilfe heute, die technischen Hilfsmittel (wie z.B. Defibrillator), Demonstration einer einfachen Erste Hilfe – Massnahme

16.15 Schlusswort

Luppmenstrasse 1
CH-8320 Fehraltorf

Tel. +41 (0) 1 956 11 11
Fax +41 (0)1 956 11 22
info@electrosuisse.ch
www.electrosuisse.ch

SEV Verband für Elektro-, Energie- und Informationstechnik

electrosuisse

Informationstagung:

für Betriebselektriker

Dienstag 11. Dienstag 25. oder 26. März 2003, Kongresshaus Zürich
Donnerstag 20. März 2003, Kursaal Bern

Von:	An: Electrosuisse Luppmenstrasse 1 8320 Fehraltorf
Fax Nr.	Fax Nr. 01 956 16 75
Tel. Nr.	Tel.Nr. 01 956 11 75
E-Mail:	E-Mail: daniela.kneubuehler@electrosuisse.ch

Anmeldung

Bitte mit Maschine oder in Blockschrift ausfüllen

Teilnehmer

Name	Vorname	Preiskategorien				Teilnahme-Datum			
		V	**E**	**K**	**N**	11.03.	20.03.	25.03.	26.03.

V	Mitarbeiter von Kunden mit Beratungs- und Kontrollvertrag Fr. 300.-	**E**	Einzelmitglied Fr. 300.-	**K**	Kollektivmitglied Fr. 360.-	**N**	Nichtmitglied Fr. 400.-
		Nr.		Nr.			

Ab 5 Teilnehmern der selben Firma bei gleichzeitiger Buchung wird ein Rabatt von 5% gewährt.

Liefer- und / oder Rechnungsadresse

Firma	
Abteilung	
Strasse / Nr.	
PLZ / Ort	

Rechnungsadresse (falls nicht identisch mit obiger Adresse):

Firma	
Abteilung	
Strasse / Nr.	
PLZ / Ort	

Datum: _____ Unterschrift: _____

Bei Abmeldungen zwischen 1 und 20 Tagen vor Tagungsbeginn werden Fr. 50.- in Rechnung gestellt.
Bei Nichterscheinen ist der volle Teilnehmerbetrag zu entrichten

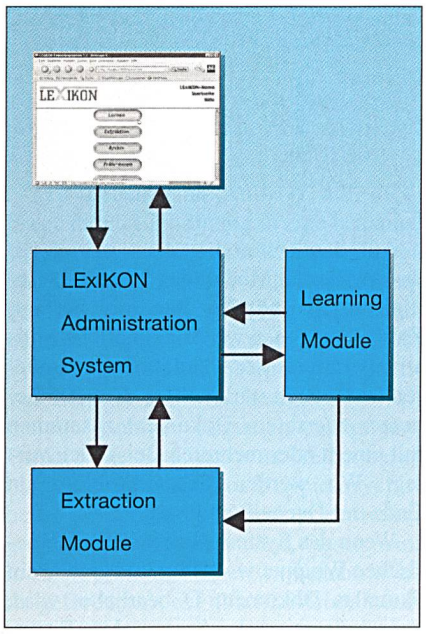

Bild 4 LExIKON-Architektur

im Projekt LExIKON folgendermassen vorgegangen:

- In einem ersten Schritt werden diejenigen Stellen lokalisiert, an denen man vor algorithmisch nicht beherrschbaren Problemen steht.
- Anschliessend werden verwandte Aufgaben gesucht, die eventuell zwar schwächer, das heisst eingeschränkter oder etwas anders formuliert sind, deren algorithmische Lösung man aber kennt.
- In einer Synthese wird schliesslich versucht, eine Version der ursprünglichen Aufgabenstellung zu formulieren, bei der man an den kritischen Stellen Modifikationen im Sinne der zuerst gefundenen lösbaren Teilprobleme vornimmt.

Diese Ausführungen sind sehr allgemein. Sie sollen aber am Beispiel des Projekts LExIKON detaillierter beschrieben werden. Die drei oben genannten Schritte werden nachfolgend als Analyse von Engpässen, Analyse verfügbarer Verfahren und Synthese des (neuen) Lösungsansatzes bezeichnet.

Analyse von Engpässen

Eine konzeptionelle Schwachstelle der Extraktion von Information aus vorgelegten Dokumenten liegt darin, dass man im Allgemeinen nicht erwarten kann, dass negative Information (Gegenbeispiele) vorgelegt werden[1]. Die Grenzen des Lernens aus nur positiven Beispielen sind hinreichend gut bekannt [1, 2, 3], was zu Forschungen über die Möglichkeiten des Lernens unter restriktiven Annahmen motiviert [4, 5, 6] hat. Im LExIKON-Ansatz wird daher versucht, entweder auf natürliche Art und Weise negative Beispiele zu erlangen oder sich auf Strukturen zu fokussieren, die nur aus positiven Beispielen lernbar sind.

Verfügbare Verfahren

Text-Pattern (Textmuster) im Sinne von [4] sind auf vielfältige Art und Weise lernbar, auch wenn nur positive Beispiele vorgelegt werden, und es gibt Lernverfahren mit erstaunlichen Eigenschaften [7]. Darüber hinaus sind spezielle theoretische Konzepte wie «Bounded Finite Thickness» entwickelt worden, die als Grundlage für das Lernen aus nur positiven Beispielen dienen können [5]. Sogenannte «Elementary Formal Systems» (EFS) verwenden Pattern, um formale Sprachen darzustellen [8]. Mit Hilfe von EFS hat man eine Reihe von Lernbarkeitsergebnissen für Fälle erzielt, bei denen nur positive Beispiele verfügbar sind [5, 6, 9].

Synthese des Lösungsansatzes

Leider lassen sich in den Lernaufgaben zur Wissensextraktion aus HTML-Seiten und ähnlichen Dokumenten keine Pattern finden. Wrappers kann man als Vorschriften dafür auffassen, wie Information in Dokumenten bestimmten Typs zu «verpacken» ist (daher der Name). Dual dazu erklären sie, wie man «verpackte» Information wieder «auspacken» kann [10, 11, 12, 13, 14].

Bild 5 illustriert das Problem, zu Texten, die in einem Dokument markiert werden, die syntaktischen Begrenzer (manchmal entsprechen diese den Tags, oft sind es nur Teile von Tags oder ganz andere Zeichenketten.) zu finden. Die Ausdehnung der Texte ist nur vage bestimmt. Das LExIKON-System bildet dynamisch hypothetische Beschreibungen der zulässigen Begrenzer-Sprachen für alle im Dokument relevanten Stellen. Es handelt sich tatsächlich um formale Sprachen, das heisst um Mengen von Zeichenketten. Diese können unendlich sein und werden dann z.B. durch reguläre Terme beschrieben.

Für die Beschreibung des «Einpackens» an einer bestimmten Stelle in einem Dokument sind Pattern durchaus adäquat; sie können den Zusammenhang

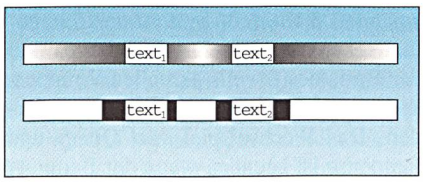

Bild 5 Lokalisieren und Hypothetisieren von Begrenzern

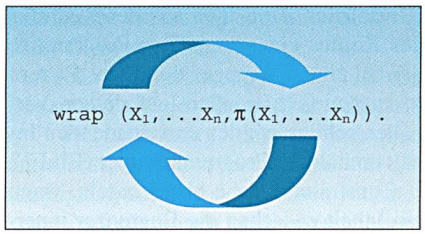

Bild 6 Das Wrapper-Prinzip

```
wrap (X1,...Xn,π(X1,...Xn)):-
    „grammars",
    „constaints".
```

Bild 7 Das Prinzip der (A)EFS

zwischen dem eigentlichen Inhalt und der umgebenden Syntax gut formalisieren. EFS (siehe oben) sind geeignet, diese Form der Wissensrepräsentation mit zusätzlichen Einschränkungen (Constraints) zu versehen. Wenn allerdings an mehreren Stellen in einem Dokument Informationen gleicher Art verpackt sein können, wie zum Beispiel mehrere Literaturangaben auf einer Seite, dann muss der Algorithmus in der Lage sein, unzulässige Vermischungen – etwa die Zuordnung falscher Seitenangaben zu Publikationen – auszuschliessen. Technisch ausgedrückt benötigt man die logische Negation.

Als Fundierung der LExIKON-Technologie wurde eine Erweiterung des Konzepts der EFS, die so genannten «Advanced Elementary Formal Systems» (AEFS) entwickelt [12]. Wrappers sind spezielle Programme, die sich als AEFS darstellen lassen. Zur Einführung der wichtigsten Begriffe wird nachfolgend eine logische Schreibweise verwendet; die Implementation der Wrappers erfolgt allerdings in Java oder C++.

Aus logischer Sicht sind EFS und AEFS logische Programme, in denen statt Termen Text-Pattern (im Sinne von [4]) vorkommen.

Ein Wrapper ist ein Programm, das spezifiziert, wie bestimmte Inhalte (die Belegung der Variablen X1,...,Xn im Bild 6) in ein Pattern eingepackt werden sollen; das Entpacken ist dual dazu. Solch ein Pattern π kann für zwei Variablen X1 und X2 zum Beispiel die Form π = Z1L1X1R1Z2L2X2R2Z3 haben. Mit geeigneten Constraints kann ausgedrückt werden, dass der Text beliebig beginnt (Z1), dann folgt ein linker Begrenzer (L1) für die erste Komponenten des Inhalts (X1), dann ein rechter Begrenzer (R1), dann ein beliebiger Zwischentext (Z2) usw.

Natürlich kann man aus der Kenntnis des Kopfes eines logischen Programms, der bei einem Wrapper durch das Pattern π determiniert ist, nicht auf dessen Verhalten schliessen; den entscheidenden Inhalt enthält der Programmkörper (Bild 7).

Constraints (siehe oben) regeln, wann ein Inhalt zwischen die Begrenzer innerhalb eines Pattern eingesetzt werden darf. Welche Begrenzer vorkommen können, wird durch Grammatiken für formale Sprachen angegeben. Solange der Körper des Programms keine Negation enthält, handelt es sich hierbei um EFS. Weil die logische Negation ein kritischer Punkt beim Abarbeiten logischer Programme (oder ihrer Pendants in irgendeiner anderen Programmiersprache) darstellt, ist die Art und Weise der Negation, die man verwendet und zulässt, der Dreh- und Angelpunkt. Im LExIKON-Ansatz muss ausgedrückt werden, dass ganz bestimmte Begrenzer nicht in anderen Zeichenketten vorkommen. Das führt zu einer Erweiterung der EFS, die AEFS heisst und im Bild 8 illustriert wird.

Wenn ein Benutzer mit dem LExIKON-System in Interaktion tritt, dann stellt sich dem Computersystem die Aufgabe, AEFS zu lernen (Bild 7). Gegenstand des Lernens kann dabei sein:
- das Pattern im Kopf des AEFS;
- die Stelligkeit des Patterns (Anzahl und ggf. Typ der Variablen) bei gegebener Grundstruktur;
- die Grammatik-Regeln;
  die Constraint-Regeln.

Selbst wenn alle diese Aufgaben zu lösen sind, steht man vor nachweislich unlösbaren Problemen. Erst durch geeignete Einschränkung erhält man lösbare Lernaufgaben. Das LExIKON-System löst vornehmlich die Aufgabe, formale Sprachen für die in Frage kommenden Begrenzer zu lernen.

Theoretische Fundierung der Reichweite der Technologie

Die LExIKON-Technologie beruht also auf folgenden Kernkonzepten und algorithmischen Ideen:
- Wrapper-Konzepte;
- Repräsentation von Wrappers durch AEFS;
- induktive Lernbarkeit von AEFS.

Die Frage nach der Reichweite der Technologie ist einerseits die Frage nach der Adäquatheit ihrer Konzepte und nach der Ausdrucksfähigkeit ihrer Beschreibungsmittel sowie andererseits die Frage nach der generellen Lösbarkeit der algorithmischen Aufgaben und – im positiven Fall der Lösbarkeit – nach ihrer Komplexität, beziehungsweise nach dem Leistungsumfang der zur Verfügung stehenden Verfahren. Die Frage nach Begriffen und Ausdrucksfähigkeit ist in [12] hinreichend umfassend beantwortet worden. Daher soll nachfolgend die Frage nach der Reichweite der Verfahren diskutiert werden [13].

```
wrap (X1,...Xn,A1L1X1R1A2L2X2R2A3...nXnRnAn+1) :-
   l1(L1) :- ...
   Ri not in Xi ...
   Li not in Ai ...
```

Bild 8 AEFS für Island-Wrappers

Streng formal gesehen, erreicht man mit LExIKON, dass Programme aus Input-Output-Beispielen ein intendiertes Verhalten lernen, welches im vorliegenden Fall immer die Extraktion von Tupeln von Zeichenketten aus Dokumenten bedeutet. Aus theoretischen Untersuchungen sind komplexe Hierarchien von lösbaren Lernaufgaben dieser Art bekannt [2, 15]. Wenn es gelingt, das was LExIKON tut, in diese schon gut bekannten und gründlich studierten Hierarchien einzuordnen, dann kann man die Möglichkeiten und Grenzen der LExIKON-Technologie besser verstehen und kommunizieren. Im günstigsten Fall lassen sich aus solchen Charakterisierungen sogar Richtungen der weiteren Arbeit ableiten.

Formalisierung als Basis der Charakterisierung

Für Untersuchungen wie in [2, 15] wurden die Lernaufgaben in mathematisch präzise Termini gefasst, um für die Ergebnisse klare Beweise in den Händen zu haben. Grundlagen dafür findet man in [5] oder in anderen Standard-Referenzen. Aber wie formalisiert man den im LExIKON-System ablaufenden Lernprozess angemessen, so dass diese Formalisierung die Realität der Wissensverarbeitung mit LExIKON hinreichend genau widerspiegelt und darüber hinaus eine mathematisch fundierte Einordnung erlaubt? In [13] ist, in Anlehnung an [16, 17], ein Vorschlag ausgearbeitet worden, der zum Erfolg führt: Das Lernen von Wrappers durch das LExIKON-System wird als eine Form von «Lernen durch Befragen» aufgefasst.

Das LExIKON-System befragt den Benutzer. Der Benutzer beantwortet sogenannte Queries des Systems. Die Antwort des Benutzers, als Response bezeichnet, dient dem System dazu, hypothetische Wrappers zu generieren. Auf der Basis des jeweils aktuellen Wrappers kann wieder eine Query formuliert werden. Das Wechselspiel von Query und Response ist beendet, wenn der Benutzer zufrieden ist. Dann verfügt das System über einen induktiv gelernten Wrapper.

Auf den ersten Blick ist irritierend, dass in diesem Modell das System die Initiative hat und den Benutzer befragt. Zum einen geben wir prinzipiell die Initiative nicht aus der Hand, und zum anderen ist es doch wirklich so, dass der Benutzer zuerst ein Dokument zusammen mit einem oder mehreren Beispielen vorlegt. Wir werden dieses Problem am Ende der Darstellung lösen.

Wenn das System über einen hypothetischen Wrapper w verfügt und gerade ein aktuelles Dokument D bearbeitet wird, dann präsentiert das System dem Benutzer, wie schon erörtert, nicht w selbst, sondern das Extraktionsergebnis w(D). Betrachtet man das Paar (D,w(D)) als Query, so ist der Benutzer zufrieden, wenn das Extraktionsergebnis mit seiner Sicht R auf das Dokument übereinstimmt, also $w(D) = R$ gilt. Andernfalls muss der Benutzer die Query beantworten. Eine sinnvolle Antwort ist entweder das Zurückweisen eines aus Sicht des Benutzers zuviel extrahierten Beispiels (d.h. Eingabe eines $t \in w(D) \backslash R$) oder der Nachtrag eines noch fehlenden Beispiels (d.h. $t \in R \backslash w(D)$). Im Allgemeinen ist im Fall $w(D) \neq R$ ein zulässiges Response die Eingabe eines Beispiels aus der symmetrischen Differenz: $t \in R \Delta w(D)$. Ist der Benutzer dagegen mit dem Extraktionsergebnis auf dem vorliegenden Dokument zufrieden, kann er entweder den Prozess insgesamt beenden oder zu einem neuen Dokument übergehen.

Nun wird auch deutlich, dass die Eingabe des ersten Dokuments mit einem oder mit mehreren Beispielen in diesen formalen Rahmen passt. Wir unterstellen, dass das System zu Beginn des Lernvorgangs keinen Wrapper hat bzw. nur einen solchen Wrapper w_0, der nichts extrahiert. Es ist also stets $w_0(D) = \varnothing$. Tatsächlich ist dann ein erstes Beispiel, welches ein Benutzer vorlegt, aus der genannten symmetrischen Differenz: $t \in R \Delta w_0(D) = R$.

In diesem Szenario verläuft die Interaktion von Mensch und Maschine also als eine Folge von Query-Response-Paaren der Form $((q_0, r_0),(q_1, r_1), \ldots)$, wobei jede Query die Form $q_i=(D_i, w_i(D))$ hat und jedes Response r_l die akzeptierende Beendigung der Interaktion bedeutet oder im Vorlegen eines Dokuments mit einem

Beispiel gemäss der obigen Constraints besteht, d.h. $r_i=(D_i, t_i)$.

Charakterisierung der Reichweite der Technologie

Wir geben eine Antwort auf diese Frage durch Einordnung in eine bekannte Hierarchie, welche wir [15] entlehnen und hier nur skizzieren. Alle Bezeichnungen beziehen sich auf [15] und werden nicht weiter motiviert, wohl aber inhaltlich erläutert. Alle Probleme des induktiven Lernens von berechenbaren (in formalen Termini: allgemein-rekursiven) Funktionen, die lösbar sind, indem beliebig viele Input-Output-Beispiele verarbeitet werden und schliesslich ein richtiges Programm gelernt wird, fassen wird in *Lim* zusammen. Hier wird nichts weiter gefordert als der abschliessende Lernerfolg. Wenn man ausserdem verlangt, dass jedes zwischenzeitlich als Hypothese generierte Programm die Information widerspiegelt, aus der es generiert worden ist (was man als *Konsistens* bezeichnet), dann wird die Klasse aller so lösbaren Lernprobleme mit *Cons* bezeichnet.

Lernen, das nicht immerzu auf die gesamte Historie des Lernprozesses zurückblickt, sondern seine nächste Hypothese immer nur aus der vorangehenden Hypothese und dem gerade aktuellen Beispiel konstruiert, heisst iteratives Lernen und führt zur Problemklasse *It*. Verlangt man von jeder Zwischenhypothese, dass sie selbst ein total definiertes Programm ist, was aus Gründen der Rekursionstheorie (vgl. [18]) eine recht einschneidende Forderung darstellt, dann bezeichnen wir alle derart lösbaren Lernprobleme mit *Total*. Wenn man sogar noch entscheiden kann, ob und wann ein Lernprozess beendet ist, spricht man von finitem Lernen. Die Klasse aller derart lösbaren Probleme heisst *Fin*.

Diese klassischen Lernbegriffe gehen von einer irgendwie fixierten Reihenfolge der zum Lernen vorgelegten Information aus. Hebt man diese Einschränkung auf, so notiert man das durch einen oberen Index *arb* für arbiträre Informationsangebote. Es entstehen die Problemklassen Lim^{arb}, $Cons^{arb}$, It^{arb}, $Total^{arb}$ und Fin^{arb}.

Zwischen diesen Klassen von Lernproblemen herrscht der folgende bekannte Zusammenhang (Bild 9), wobei die weiter unten stehenden Klassen in den oberen echt enthalten sind.

In den Zusammenhang von Bild 9 soll eingeordnet werden, was man mit der LExIKON-Technologie leisten kann. Klassen von Wrappers, die man so wie in diesem Beitrag dargestellt interaktiv lernen kann, werden in der Problemklasse *LEx* zusammengefasst. Wenn wir ausserdem vom LExIKONSystem eine inkrementelle Arbeitsweise verlangen wollen, bei der es keine Möglichkeit gibt, auf frühere Responses zurückzugreifen, notieren wir das als LExInc.

In Anlehnung an [13] ergibt sich die in Bild 10 dargestellte Einordnung.

Die Einordnung ist relativ klar, so dass wir uns hier auf die Erörterung von zwei Phänomenen beschränken können. Erstens wird deutlich, dass der Ansatz von LExIKON offenbar restriktiv ist. Man muss bedenken, dass Bild 9 nur einen kleinen Ausschnitt der in der Theorie bekannten Typen von Lernproblemen zeigt. Wenn man in diesem recht begrenzten Ausschnitt der Welt die Klassen von Programmen, also von Wrappers im Sinne der Wissensextraktion, einordnet, wie das in Bild 10 geschehen ist, liegen diese relativ weit unten. Das heisst, mit anderen Worten, dass es eine Reihe von durchaus lösbaren Lernproblemen gibt, die umfassender sind als die Aufgaben, die im Moment mit der LExIKON-Technologie gelöst werden können. Das ist nicht nur als Hinweis auf die Grenzen der Technologie zu verstehen, zeigt es doch auch, dass man versuchen kann, die Reichweite von LExIKON auszuweiten, und dass es dafür Chancen gibt.

Auch die Abgrenzung von It^{arb} ist von Bedeutung. Viele Theoretiker sehen in It^{arb} die Inkarnation dessen, was man unter realitätsnahem Lernen aus unvollständiger Information verstehen möchte: keine Anforderungen an die Speicherung der Historie des induktiven Lernprozesses und keine Anforderungen irgendwelcher Art, wie die Information vorzulegen sei. Wie auch immer, *LEx* und It^{arb} sind miteinander unvergleichbar. Dies heisst auch, dass es Lernprobleme gibt, die LExIKON löst, die aber über die Grenzen von It^{arb} hinausgehen.

Zweitens ist mit dem Beweis von Total = $Total^{arb}$ = *LEx* in [13], wo es noch einige detailliertere Ergebnisse dieser Art gibt, eine aussagekräftige Charakterisierung gelungen. Diese Äquivalenz sagt unter anderem, dass man bei der Generierung von Wrappern zur Extraktion von Information aus semistrukturierten Dokumenten stets erwarten kann, dass der hypothetisch generierte Wrapper auf beliebigen Dokumenten arbeitet (auch wenn er ggf. nichts extrahiert). Jedenfalls braucht es keinen Fall zu geben, in dem ein Wrapper auf einem Dokument «abstürzt».

So lassen sich aus der theoretischen Charakterisierung Anforderungen an das Systemverhalten ableiten.

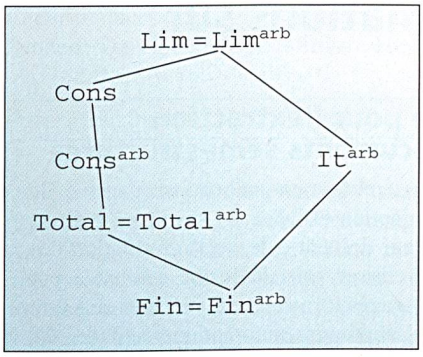

Bild 9 Hierarchie von klassischen Lernproblemen

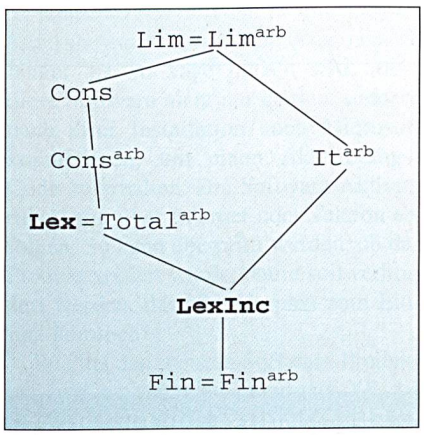

Bild 10 Einordnung der LExIKON-Technologie in bekannte Probleme

Die Technologie im Einsatz: Möglichkeiten, Grenzen und künftige Entwicklungen

Das LExIKON-Entwicklungssystem beinhaltet gegenwärtig drei unterschiedliche Lernverfahren und ein Verfahren zur Extraktion, das heisst zur Anwendung von Wrappern auf Dokumenten.

Entwicklung der Kerntechnologie

Es wird daran gearbeitet, weitere Verfahren in das System zu integrieren. Damit wird die Kerntechnologie weiter ausgearbeitet. Die LExIKON-Technologie ist ja in keiner Weise auf HTML-Dokumente beschränkt. Sie funktioniert immer dann, wenn weitest gehend freie Inhalte in syntaktische Regelmässigkeiten eingebettet sind. Mittelfristig ist daher vorgesehen, neben HTML-Quellen gleichermassen andere Dokumentformate wie zum Beispiel LaTeX, XML und PDF zuzulassen. Das liegt alles noch im Rahmen der bisherigen Technologieentwicklung.

Erweiterung der Kerntechnologie

Der Ansatz von LExIKON ist dadurch gekennzeichnet, dass im Zuge der ma-

schinellen Informationsverarbeitung gerade der Inhalt, der einen Benutzer interessiert, vom System nicht bearbeitet wird. Das System befasst sich nur mit der einbettenden Syntax.

Daher ist die Funktionsweise dieser Technologie vollkommen unabhängig
- von der zugrunde liegenden natürlichen Sprache und
- vom zugrunde liegenden Anwendungsbereich.

Natürlich liegen zusätzliche Reserven in einer Ausnutzung von linguistischem Wissen und von Besonderheiten der Domäne. In Abhängigkeit von den Interessen und Möglichkeiten verschiedener Kooperationspartner soll die LExIKON-Technologie langfristig mit anderen Technologien verzahnt werden. Das wird die Entwicklung von adäquaten Anwendungsszenarien erfordern und neue theoretische Fragen aufwerfen. Die Erprobung solcher bisher noch nicht unternommener Erweiterungen wird der gesamten Entwicklung wichtige Impulse geben.

Referenzen

[1] *E. Mark Gold:* Language Identification in the limit. Information and Control, (1967)14, pp. 447–474.

[2] *Dana Angluin and Carl H. Smith:* A survey of inductive inference: Theory and methods. Computing Surveys, (1983)15, pp. 237–269.

[3] *Thomas Zeugmann and Steffen Lange:* A guided tour across the boundaries of learning recursive languages. Algorithmic Learning for Knowledge-Based Systems, K.P. Jantke and S. Lange (eds.), Springer-Verlag, LNAI 961, 1995, pp. 190–258.

[4] *Dana Angluin:* Finding patterns common to a set of strings. J. Computer and Systems Science, 21(1980), pp. 2–32.

[5] *Takeshi Shinohara:* Inductive inference of monotonic formal systems from positive data. New Generation Computing, (1991)8, pp. 371–384.

[6] *Takeshi Shinohara:* Rich classes inferable from positive data. Information and Computation, 108, 1994, pp. 175–186, .

[7] *Steffen Lange and Rolf Wiehagen:* Polynomial-time inference of arbitrary pattern languages. New Generation Computing, (1991)8, pp. 361–370.

[8] *Setsuo Arikawa:* Elementary formal systems and formal languages – Simple formal systems. Memoirs of Faculty of Science, Kyushu University, Series A, Mathematics (1970)24, pp. 47–75.

[9] *Setsuo Arikawa, Takeshi Shinohara and Akihiro Yamamoto:* Learning elementary formal systems. Theoretical Computer Science 95, 1992, 97–113.

[10] *Boris Chidlovskii:* Wrapping Web information providers by transducer induction. 12th European Conference on Machine Learning, Freiburg, Germany, Sept. 2001, L. De Raedt and P.A. Flach (eds.), Springer-Verlag, LNAI 2167, 2001, pp. 61–72.

[11] *Boris Chidlovskii, Jon Ragetli and Maarten de Rijke:* Wrapper generation via grammar induction. 11th European Conference on Machine Learning, Barcelona, Catalonia, Spain, May/June 2000, Springer-Verlag, LNAI 1810, 2000, pp. 96–108.

[12] *Gunter Grieser, Klaus P. Jantke, Steffen Lange and Bernd Thomas:* A unifying approach to HTML wrapper representation and learning. Algorithmic Learning Theory, 12th Int. Conference, Washington, DC, USA, Nov. 2001, Springer-Verlag, LNAI 2225, 2001, pp. 332347.

[13] *Gunter Grieser, Klaus P. Jantke and Steffen Lange:* Consistency queries in information extraction. Algorithmic Learning Theory, 13th Int. Conference, Lübeck, Germany, Nov. 2002, Springer-Verlag, LNAI 2533, 2002, pp. 173-187.

[14] *Nicholas Kushmerick: Wrapper induction:* Efficiency and expressiveness. Artificial Intelligence, 118, 2000, pp. 15–68.

[15] *Klaus P. Jantke and Hans-Rainer Beick:* Combining Postulates of Naturalness in Inductive Inference. Elektronische Informationsverarbeitung und Kybernetik, (1981)17, pp. 465–484.

[16] *Dana Angluin:* Queries and Concept Learning. Machine Learning, (1988)2, pp. 319-342.

[17] *Dana Angluin:* Queries revisited. Algorithmic Learning Theory, 12th Int. Conference, Washington, DC, USA, Nov. 2001, Springer-Verlag, LNAI 2225, 2001, pp. 12–31.

[18] *Hartley Rogers jr.:* Theory of Recursive Functions and Effective Computability. McGraw-Hill. 1967.

Angaben zum Autor

Prof. Dr. Klaus P. Jantke ist seit Ende 1998 Principal Researcher und Projektleiter am Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI) in Saarbrücken. Er hat an verschiedenen deutschen Universitäten unterrichtet und war ordentlicher Professor an der Kuwait University, Kuwait City, sowie an der Hokkaido University, Sapporo, Japan. Gegenwärtig lehrt er an den Universitäten in Darmstadt und Saarbrücken. – *Kontakt: Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, 66123 Saarbrücken, jantke@dfki.de*

[1] Positives Beispiel: «Im Dokument sollten die Wörter Anfang und Ende enthalten sein.» Negative Beispiele kann man in einem Dokument nicht zeigen, aber man kann formulieren, was nicht vorkommen soll oder nicht vorkommen darf: «Das Dokument enthaelt keine Umlaute.»; «Bei diesem Menü kommen niemals zwei Gänge mit Fisch direkt nacheinander.»

[2] Im Gegensatz dazu hat D. Angluin in [2] (Konferenzbeitrag schon 1979) bewiesen, dass es umfangreiche Familien formaler Sprachen wie zum Beispiel Pattern-Sprachen gibt, für die man universelle Lernverfahren angeben kann, die allein aus positiven Beispielen in der Lage sind, jede mögliche Zielsprache zu erlernen.

[3] Das Projekt LExIKON ist in den Jahren 2000/2001 durch das Deutsche Bundesministerium für Wirtschaft und Technologie für 12 Monate unter dem Kennzeichen 01 MD 949 gefördert worden.

[4] COLT: Computational Learning Theory, www.learningtheory.org

L'obtention d'informations sur Internet

Technologies d'apprentissage pour l'extraction d'informations à partir de documents semi-structurés

Les documents d'exploitation et Internet renferment une masse énorme de savoir. Mais comment y accéder? Une extraction efficace du savoir ne peut se faire à la main. Les programmes d'ordinateur doivent extraire l'information des documents de manière automatique et la classer sous la forme souhaitée par l'utilisateur. L'interférence inductive des langages formels se révèle pour cela être une technologie centrale lorsqu'il s'agit d'apprendre automatiquement de tels programmes d'extraction et de les adapter aux besoins en dialogue avec l'utilisateur. Cette technologie n'est pas limitée à HTML mais fonctionne également avec des formats tels que LaTeX, XML et PDF.