Informationsextraktion

Dr. Günter Neumann
DFKI GmbH
neumann@dfki.de

1 Was ist Informationsextraktion?

Mit der rasanten Verbreitung des Internet tritt das Problem der Informationsüberflutung immer stärker in den Vordergrund: Je mehr Texte on-line zur Verfügung stehen, desto schwieriger wird es, das Informationspotential gezielt zu nutzen, d.h. relevante Informationen zu finden, zu extrahieren und in kompakter Form zu repräsentieren.

Um die Informationsüberflutung adäquat meistern zu können, wird bereits fieberhaft nach neuen Technologien für zukünftige intelligente Informationsmanagementsysteme geforscht. Eine sich neu etablierte Forschungsrichtung ist die Erforschung und Realisierung von Systemen zur *Informationsextraktion* (IE). Das Ziel der IE ist die Konstruktion von Systemen, die gezielt domänspezifische Informationen aus freien Texten aufspüren und strukturieren können, bei gleichzeitigem "Überlesen" irrelevanter Information. IE-Systeme versuchen keine umfassende Analyse des gesamten Inhaltes aller Textdokumente, sondern sollen nur die Textpassagen analysieren bzw. "verstehen", die relevante Information beinhalten. Was als relevant gilt, wird dabei durch vordefinierte domänenspezifische Lexikoneinträge oder Regeln dem System fest vorgegeben. Dieses Wissen muss dabei so detailliert und genau wie möglich festlegen, welche Typen von Information von einem IE-System extrahiert werden soll, damit eine umfangreiche und zugleich präzise Extraktion ermöglicht wird.

Typischerweise modelliert die vorgegebene Information komplexe, zusammenhängende Antwortmuster bezüglich *wer*, *was*, *wem*, *wann*, *wo* und eventuell *warum*. Sie werden in Form von Templates spezifiziert, also Bündeln von Attribut/Wert–Paaren, z.B. Firmen- und Produktinformationen, Umsatzmeldungen, Personalwechsel, Stellenausschreibungen. Die Kernfunktionalität eines IE-Systems läßt sich dann kurz wie folgt charakterisieren:

- Eingabe: Spezifikation des Typs der relevanten Information in Form von Templates (Menge von Attributen) und eine Menge von freien Textdokumenten (Pressemitteilungen, Internet-Dokumente etc.)

- Ausgabe: eine Menge von instanziierten Templates (Werte für Attribute), die mit den als relevant identifizierten und normalisierten Textfragmenten gefüllt sind.

Die so extrahierten Daten können vielseitig eingesetzt werden, z. B. zur feinkörnigen Textfilterung oder -klassifikation, als Einträge für Datenbanken, zur Unterstützung von Text Mining und Antwortextraktionssystemen, oder als Ausgangspunkt für eine Textzusammenfassung.

2 Ein Beispiel

Folgendes Beispiel soll das gerade Erläuterte illustrieren. Wir betrachten die Aufgabe, Informationen über Personalwechsel aus Online-Dokumenten zu extrahieren. Insbesondere wollen wir wissen, welche Person (*PersonOut*[1]) welche *Position* welcher *Organisation* wann (*TimeOut*) verlassen hat, und welche neue Person (*PersonIn*) wann (*TimeIn*) diese *Position* wieder aufgenommen hat. Das dazugehörige Template hat folgende Form:

[*PersonOut PersonIn Position Organisation TimeOut TimeIn*]

Für den folgenden Text:

> Dr. Hermann Wirth, bisheriger Leiter der Musikhochschule München, verabschiedete sich heute aus dem Amt. Der 65jährige tritt seinen wohlverdienten Ruhestand an. Als seine Nachfolgerin wurde Sabine Klinger benannt. Ebenfalls neu besetzt wurde die Stelle des Musikdirektors. Annelie Häfner folgt Christian Meindl nach.

ergibt sich dann beispielsweise das gefüllte (instanziierte) Template:

PersonOut	Dr. Hermann Wirth
PersonIn	Sabine Klinger
Position	Leiter
Organization	Musikhochschule München
TimeOut	heute
TimeIn	

Hier ist zu beachten, dass die gesamte Information über zwei Sätze verteilt ist, wobei im zweiten relevanten Satz sogar ein anaphorischer Ausdruck aufgelöst werden muss ("seine Nachfolgerin"). Wenn wir eine satzorientierte Verarbeitung annehmen (vgl. Abschn. 4), in der in einem ersten Schritt die relevante Information pro Satz bestimmt wird, dann müssen in einem nachfolgenden Verarbeitungsschritt die verschiedenen Teile zusammengefügt werden (vgl. Abschn. 4). Darüberhinaus ist der konkrete Wert für das Attribut *TimeOut* im Text nur relativ genannt ("heute"). Der Wert für das Attribut *TimeIn* ist explizit nicht genannt, so dass er nur über zusätzliche, eventuell sehr spezifische Annahmen ableitbar wäre, was wir für das Beispiel nicht tun wollen. Da nicht alle Attribute mit Werten belegt werden können, spricht man auch von einer *partiellen* Instanz. Übrigens enthält der Beispieltext noch eine weitere Templateinstanz:

[1] Die Attribute sind kursiv hervorgehoben.

$$\begin{bmatrix} \texttt{PersonOut} & \text{Christian Meindl} \\ \texttt{PersonIn} & \text{Annelie Häfner} \\ \texttt{Position} & \text{Musikdirektors} \\ \texttt{Organization} & \text{Musikhochschule München} \\ \texttt{TimeOut} & \\ \texttt{TimeIn} & \end{bmatrix}$$

Bei diesem Beispiel ist zu beachten, dass wir für das Attribut *Organization* denselben Wert benutzen, wie im ersten Beispiel. Das ist nur möglich, wenn sehr spezifische Regeln definiert sind. Da solche Regeln Informationen aus der näheren Umgebung überprüfen (z.B. "übernehme den Wert des Attributes Organization aus dem zuletzt gefüllten Template"), sind sie im allgemeinen fehleranfällig. Dies gilt auch für die Belegung der beiden Zeitattribute, die in unserem Beispiel unspezifiziert bleiben. Übrigens wäre es auch möglich, dass einzelne Attribute selbst eine eigene Templatestruktur besitzen, die dann entsprechend gefüllt würde. Beispielsweise kann eine normalisierte Form für Personennamen wie folgt lauten: [*Nachname Vorname Titel*].

3 Evaluationskriterien für IE

Die Güte eines IE-Systeme wird mittels der Maße Präzision und Vollständigkeit beurteilt.[2] Im Rahmen der "Message Understanding Conference (MUC)"–Reihe werden IE-Systeme sogar wettbewerbsmäßig systematisch evaluiert, MUC98.

Die Präzision P (engl. Precision) bezeichnet den Anteil der korrekt gewonnenen Wissenseinheiten (WE) (Templates oder einzelne Attribut–/Wert–Paare) im Vergleich zu den insgesamt gefundenen WE. Eine hohe Präzision bedeutet daher, dass fast alle gefundenen WE relevant sind. Die Vollständigkeit V (engl. Recall) bezeichnet den Anteil der korrekt gewonnenen WE im Vergleich zu den insgesamt gewinnbaren WE. Eine hohe Vollständigkeit bedeutet daher, dass fast alle relevanten WE extrahiert wurden.

Es ist schwierig, beide Parameter gleichzeitig zu optimieren: Wird eine Suche auf eine hohe Präzision hin optimiert, so steigt die Wahrscheinlichkeit, dass möglicherweise relevante Wissenseinheiten nicht erkannt werden. Optimiert man andererseits die Vollständigkeit, so steigt die Gefahr, dass Wissenseinheiten mit in das Ergebnis aufgenommen werden, die irrelevant sind. Um ein zusammenfassendes Maß für die Güte des IE-Prozesses zu schaffen, wurde das F-Maß definiert (in der Regel wird in der genannten Gleichung β=1 gesetzt):[3]

$$F = \frac{(\beta^2 + 1) * P * V}{\beta^2 P + V}$$

[2] Die erwähnten Maße werden auch in anderen Anwendungen eingesetzt, wie z.B. Information Retrieval und Textklassifikation.

[3] Im wesentlichen definiert die Formel ein geometrisches Mittel, dass über den Parameter β gewichtet werden kann. Die Abweichung von 1 legt fest, ob dabei P oder V stärker gewichtet werdne soll.

Die Güte des Ergebnisses hängt stark von der Schwierigkeit der Aufgabe ab. So berichten GS96 im Rahmen der MUC-7, dass bei einer einfachen Aufgabe, wie der Erkennung von Namen, die meisten Systeme sowohl bei der Vollständigkeit als auch bei der Präzision Werte von über 90% erreichten (das beste System erreichte V=96%, P=97%). Die schwierigste Aufgabe war das Auffüllen von Szenario-Templates, d.h. von Templates, die ein bestimmtes Szenario beschreiben, wie etwa eines terroristischen Anschlags oder das Beispiel in Abschnitt 2. Die Ergebnisse lagen bei V=40 bis 50% und P=60 bis 70% (das beste System erreichte V=47%, P=70%).

AI97 zeigen, dass bei komplexen Aufgaben die Güte des Ergebnisses in der Regel nicht besser als 60% bezüglich des F-Maßes ist. Dies ist jedoch nicht so schlecht, denn auch Menschen sind nicht in der Lage, bei der Analyse komplexer Texte 100% Vollständigkeit und Präzision zu erreichen. Ein Mensch erreicht bei der Erkennung von Eigennamen ein F-Maß von etwa 97% (vgl. MUC98).

4 Ein generisches IE-System

Das Beispiel in Abschnitt 2 zeigt, dass eine Reihe von linguistischen Teilaufgaben zu lösen sind, wie z.B. lexikalische Analyse, Namenserkennung, partielle Syntaxanalyse, Referenzauflösung. Es wäre prinzipiell möglich, hierzu ein generisches Textanalysesystem zu verwenden, das eine vollständige, tiefe Analyse der Bedeutung eines gesamten Textes durchführt. Aber selbst wenn es möglich wäre, die komplette Grammatik (inklusive Lexikon) einer Sprache zu formalisieren und in solch einem System zu repräsentieren, so würde das System immer noch ein Höchstmaß an Robustheit und Effizienz benötigen, um große Mengen von freien Texten verarbeiten zu können. Um aber mit dem durch die Internet–Revolution ausgelösten enorm steigenden Bedarf an Sprachtechnologie schritthalten zu können, arbeiten Forscher gerade im Bereich der IE an Methoden, die einen Kompromiß zwischen theoretischen Ansprüchen und pragmatischen Anforderungen darstellen.

Dies hat zur Entwicklung von *flachen* Textverarbeitungmethoden geführt. Hier werden bestimmte generische Sprachregularitäten, von denen bekannt ist, dass sie Komplexitätsprobleme verursachen, entweder nicht oder ganz pragmatisch behandelt, z.B. durch Beschränkung der Rekursionstiefe auf Basis einer Korpusanalyse oder durch Verwendung von Heuristiken ("präferiere längstmögliche Teilketten"). Diese ingenieursmäßige Sichtweise auf Sprachverarbeitung hat zu einer Renaissance und Weiterentwicklung von bekannten Technologien geführt, insbesondere von endlichen Zustandsautomaten oder Transduktoren in der Syntaxanalyse, RS96. Im Unterschied zu endlichen Automaten (die ein Erkennen von regulären Mustern in einer Eingabesequenz erlauben) definieren Transduktoren zusätzlich eine Relation zwischen möglichen Sequenzen von Eingabesymbolen (z. B. Wörtern und ihren zugeordneten lexikalischen Merkmalen) und zugeordneten Ausgabestrukturen (z.B. Phrasen in Form einer Dependenzstruktur). Obwohl die Syntax einer natürlichen Sprache mindestens kontextfrei ist, zeigen aktuelle Arbeiten im Bereich der IE, dass bereits mit den einfacheren Transduk-

toren sehr praktikable Systeme realisierbar sind.

Transduktoren können kaskadiert werden, indem ein Transduktor auf dem Ausgabeband eines anderen operiert. Dies erlaubt eine feine Modularität eines Systems in verschiedene Komponenten:

Tokenscanner: Auf der Basis von regulären Ausdrücken identifiziert diese Komponente die Textstruktur (z.B. Paragraphen, Einrückungen, Titelzeile) und speziellen Zeichenketten (Tokens), wie z.B. Datums- und Zeitangaben, Abkürzungen, Wortgrenzen und Interpunktionszeichen. Hier können auch HTML– oder XML–Parser zur Analyse entsprechend markierter Texte eingesetzt werden.

Lexikalische Analyse: Bei der lexikalischen Verarbeitung erfolgt eine morphologische Analyse der potentiellen Wortformen, d.h. die Bestimmung der Wortart (Part-of-Speech, POS) und der Flexionsform (z.B. Plural oder Singular). Speziell zur Verarbeitung der deutschen Sprache muss auf dieser Ebene auch eine Analyse von Komposita und Hyphenkoordination (*An- und Verkauf*) durchgeführt werden, da gerade die Kompositabildung im Deutschen sehr produktiv ist, NBB$^+$97. Anschließend werden morphosyntaktisch mehrdeutige Wörter (*Ich meine meine Tasche*) mittels POS-Taggern desambiguiert.

Eigennamenerkennung findet und normalisiert spezielle Ausdrücke wie Personen-, Firmen-, Produktnamen, komplexe Datums-, Zeit-, und Maßausdrücke. Da auch Eigennamen sehr produktiv sind, werden sowohl spezielle Eigennamenlisten als auch Automatengrammatiken (zur Behandlung nicht-lexikalisierter Ausdrücke) verwendet. Wichtig auf dieser Ebene ist die Behandlung von Referenzen zwischen Eigennamen (*EN-Koreferenz*), um festzustellen, dass z.B. “Bundeskanzler Schröder”, “G. Schröder” oder “Schröder” in einem Text dieselbe Person bezeichnet (vgl. PN00).

Parsing: Wie in der Einleitung bereits erläutert, wird in den meisten IE-Systemen keine vollständige syntaktische Analyse durchgeführt, sondern eine flache, fragmentarische Analyse, die einfach mit endlichen Automaten und nichtambigen Grammatiken erreicht werden kann. Die Parsingaufgabe wird stark modularisiert durch explizite Trennung in Phrasen- (NP, PP, VG) und Satzstruktur. Dadurch kann eine strikt bottom-up gesteuerte, kaskadierte Parsingstrategie realisiert werden (auch als *Chunk-Parsing* bekannt), wobei zuerst einfache, nichtrekursive Phrasen erkannt werden und dann in einer nächsten Phase zu komplexeren Einheiten kombiniert werden (z.B. NP-Koordination). Diese sehr modulare Vorgehensweise erlaubt es, eine einfache, aber domänenunabhängige Phrasenanalyse mit sehr domänenspezifischen Regeln zur Erkennung von komplexen (Satz-)Einheiten zu kombinieren.

Koreferenzauflösung: Die zentrale Aufgabe ist es festzustellen, ob unterschiedliche linguistische Objekte auf dieselbe Templateinstanz Bezug nehmen. Im Rahmen der IE sind folgende Koreferenzprobleme zu lösen: 1) EN-Koreferenz (siehe Paragraph zur Eigennamenerkennung), 2) Pronominale Referenz, d.h. Referenzen zwischen Pronomonen (“er”, “sie”, etc), ENs und NPs, 3) Referenzen zwischen Designatoren (“die Firma”, “der Detroiter Autohersteller”) und anderen Instanzen (“Ford”). Ähnlich wie bei den flachen Parsingstrategien hat sich gerade im Umfeld der IE-Forschung gezeigt, dass eine flache Koreferenzauflösung (die also keine vollständige Syntaxverarbeitung voraussetzt) bereits

sehr gute Ergebnisse liefert.

Erkennung domänenrelevanter Muster: Dies ist der kritische Teil eines IE-Systems, da hier die Regeln definiert werden, die die Struktur von Templateinstanzen bestimmen. Sie bauen unmittelbar auf den vorher genannten Komponenten auf. Im Prinzip führen sie ein domänenspezifisches Sammeln gemäß der einem IE-System bekannten Templatestruktur durch. Die Regeln sind so definiert, dass sie Merkmale der Köpfe der extrahierten Phrasen überprüfen (z.B. syntaktische Eigenschaften, Eintrag im Domänenlexikon). Ein Beispiel solch einer Regel findet sich im nächsten Abschnitt. Meist sind diese Regeln satzbasiert, d.h. sie liefern (partielle) Templateinstanzen, die sich nur auf Informationen eines (möglicherweise sehr langen) Satzes beziehen.

Template–Unifikation: Es ist klar, dass ein einzelner Satz nicht alle notwendigen Informationen zur Instanziierung eines Templates enthalten muss, sondern dass die Information über mehrere Sätze verteilt sein kann (vgl. Abschn. 2). Daher ist es nötig, Informationen aus unterschiedlichen Templateinstanzen zu vereinigen. Im Allgemeinen ist dies eine sehr schwierige Aufgabe, die in IE-Systemen meist mittels einer einfachen Template–Unifikationsstrategie behandelt wird: Wenn zwei Templates identische Information in mindestens einem Attribut haben, dann wird die gesamte Information beider Templates mittels Unifikation vereinigt. Dabei wird für je zwei typkompatible Attribute überprüft, ob eine Koreferenzbeziehung besteht, ob sie semantisch kompatibel sind (via Domänenlexikon) oder ob sie in einem Subsumptionsverhältnis stehen. Daneben werden auch sehr anwendungsspezifische Heuristiken eingesetzt.

Noch dominieren IE-Systeme zur Verarbeitung von englischen Texten. In den letzten Jahren wurden aber auch IE-Systeme zur Verarbeitung anderer Sprachen realisiert, z.B. Chinesisch, Japanisch, Italienisch und Deutsch. Dabei sind aufgrund sprachspezifischer Aspekte Anpassungen bzw. Erweiterungen an der generischen Architektur nötig. So zeigt sich bei der freien Verarbeitung von deutschen Texten, neben der komplexen Kompositaanalyse, dass ein rein bottom-up orientiertes Parsing wegen der freien Wortstellung ungeeignet ist, eine gemischte top-down/bottom-up Strategie dagegen vielversprechender, NBP00.

5 Maschinelle Lernverfahren für IE

Da die zwei letztgenannten Komponenten sehr domänenspezifisch sind, müssen sie für ihren Einsatz an eine neue Aufgabe und Domäne adaptiert werden. Dies ist in der Regel sehr aufwändig und muss in der Regel von Experten durchgeführt werden. Daher steht die Entwicklung von maschinellen Lernverfahren im Fokus der aktuellen IE-Forschung, um gerade den Prozess der Domänenmodellierung möglichst zu automatisieren. Wir wollen daher zum Schluß des Artikels kurz auf den aktuellen Stand in der Forschung eingehen.

Die Mehrzahl der aktuellen Ansätze sind Varianten von überwachten induktiven Lernverfahren. Ausgehend von einer Trainingsmenge von bereits mit den Ergebnissen annotierten Textdokumenten ist es das Ziel, automatisch Regeln zum Füllen von Templates zu induzieren. Dazu werden schrittweise die durch

die Trainingsmenge vorgegebenen Templateinstanzen verallgemeinert, so dass sie auch auf nichtannotierte, neue Dokumente anwendbar sind. Um zu diesen domänenspezifischen Regeln zu gelangen, werden die Dokumente mit den flachen Verarbeitungskomponenten vorverarbeitet.[4] Die meisten aktuellen Verfahren lernen Regeln zum Füllen einzelner Attribute, neuere Ansätze erlernen sogar Regeln zum Füllen ganzer Templates, CM98.

Das folgende Beispiel soll einen kleinen Einblick in die Wirkungsweise eines induktiven IE-Lernverfahren geben (entnommen aus Huf96). Ausgangspunkt sei folgendes annotiertes Trainingsbeispiel:

⟨PNG⟩ Sue Smith ⟨/PNG⟩, 39, of Menlo Park, was appointed

⟨TNG⟩ president ⟨/TNG⟩ of ⟨CNG⟩ Foo Inc. ⟨/CNG⟩

wobei PNG für "person name group", TNG für "title name group" und CNG für "company name group" steht. Zusammen mit einer linguistischen Vorverarbeitung kann folgende Prolog-ähnliche Template-Regel abgeleitet werden:

noun-group(PNG, head(isa(person))), noun-group(TNG, head(isa(title))),
noun-group(CNG,head(isa(company))), prep(PREP, head(of OR at OR by)),
verb-group(VG, type(passive), head(named OR elected OR appointed)),
subject(PNG,VG), object(VG,TNG),
post-nominal-prep(TNG,PREP), prep-obj(PREP,CNG)
$\Longrightarrow$ management-appointment(person(PNG), title(TNG), company(CNG)).

Diese Regel besteht aus zwei Teilen: dem linken Bedingungsteil *Bed* und dem rechten Templateinstanziierungsteil $TInst$. Ausgangspunkt für die Bestimmung von *Bed* ist das Ergebnis der linguistischen Vorverarbeitung mit dem Trainingssatz, für den eine Phrasenanalyse durchgeführt wird. In dem vorliegenden Fall wird die erste NP, wenn ihr Kopfelement zur semantischen Klasse Person gehört, dem Attribut PNG zugewiesen. Die Eigenschaften der Verbgruppe werden unmittelbar in die Regel übernommen. Wir haben in diesem Beispiel bereits angenommen, dass die gleiche Regel bereits für Trainingsbeispiele mit den Verben "elected" und "named" abgeleitet wurde, so dass die drei Regeln in eine überführt wurden. Analoges gilt für die aufgeführte Präposition.

Die aktuellen Verfahren zeigen bereits erstaunlich gute Ergebnisse. So berichtet Huf96 für sein System ein F-Maß von 85.2%. Für eine IE-Anwendung im Bereich der Online-Stellenangebote berichten CM98 von 87.1% P und 58.8% V. Sehr gute Ergebnisse werden auch in dem Bereich der multilingualen Eigennamenerkennung berichtet, Gal96; BMSW97. Die meisten aktuellen Verfahren verlangen noch eine recht umfangreiche Menge an bereits annotiertem Trainingsmaterial, deren Erstellung sehr zeitaufwendig ist. Daher versuchen neuartige Methoden, auch ohne annotiertes Trainingsmaterial Templateregeln abzuleiten, Ril96; YGTH00.

[4] Hier zeigt sich ganz deutlich der Vorteil von verfügbaren domänenunabhängigen linguistischen Modulen. In aktuelle Verfahren kommen dabei unterschiedliche Module zum Einsatz. So verwendet Fre98 nur Tokenization, CM98 zusätzlich POS-Tagging, Huf96 Phrasenerkennung und Ril96 flache Satzanalyse.

Literatur

Appelt, D. und D. Israel: *Building information extraction systems.* Tutorial during the 5th ANLP, Washington, 1997. http://www.ai.sri.com/~appelt/ie-tutorial/.

Bikel, D. M., S. Miller, R. Schwartz und R. Weischedel: *Nymble: a High-Performance Learning Name-finder.* In: *Proceedings of 5th ANLP*, Washington, USA, March 1997.

Califf, M. und R. Mooney: *Relational Learning of Pattern-Match Rules for Information Extraction.* In: *Proceedings of the AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, 1998.

Freitag, D.: *Information Extraction From HTML: Application of a General Learning Approach.* In: *Proceedings of the 15th AAAI*, 1998.

Gallippi, A.: *Learning to Recognize Names Across Languages.* In: *34th ACL*, Santa Cruz, California, USA, 1996.

Grishman, R. und B. Sundheim: *Message Understanding Conference – 6: A Brief History.* In: *Proceedings of the 16th COLING*, Kopenhagen, Denmark, Europe, 1996.

Huffman, S.: *Learning information extraction patterns from examples.* In: Wermter, Riloff und Scheler (Herausgeber): *Connectionist, Statistical, and Symbol Approaches to Learning for Natural Language Processing*, Band 1040 der Reihe *LNAI*, Berlin, Springer, 1996.

Proceedings of the Seventh Message Understanding Conference (MUC-7), (Fairfax, VA, April 1998), 1998. Morgan Kaufmann Publishers.

Neumann, G., R. Backofen, J. Baur, M. Becker und C. Braun: *An Information Extraction Core System for Real World German Text Processing.* In: *Proceedings of 5th ANLP*, Washington, USA, March 1997.

Neumann, G., C. Braun und J. Piskorski: *A Divide-and-Conquer Strategy for Shallow Parsing of German Free Texts.* In: *Proceedings of the 6th ANLP*, Seattle, USA, April 2000.

Piskorski, J. und G. Neumann: *An Intelligent Text Extraction and Navigation System.* In: *Proceedings of the 6th RIAO.* Paris, April 2000.

Riloff, E.: *Automatically Generating Extraction Patterns from Untagged Text.* In: *13th AAAI*, 1996.

Roche, R. und Y. Schabes: *Finite State Devices for Natural Language Processing.* MIT Press, Cambridge MA, 1996.

Yangarber, R., R. Grishman, P. Tapanainen und S. Huttunen: *Unsupervised Discovery of Scenario-Level Patterns for Information Extraction.* In: *Proceedings of the 6th ANLP*, Seattle, USA, April 2000.