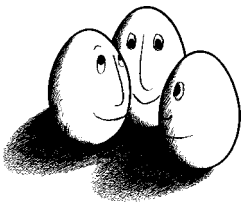
**PG-Seminar**

# Informationsextraktion

Christian Hüppe



PG-Seminar
am Fachbereich Informatik
der Universität Dortmund

14. November 2001

**Betreuer:**

Prof. Dr. Katharina Morik
Stefan Haustein

# Inhaltsverzeichnis

# 1 Einleitung

Heutzutage gibt es mehr elektronische Dokumente als je zuvor und die Anzahl erhöht sich täglich. Allein das Internet stellt uns unzählige Mengen an Texten zur Verfügung. Das Problem hierbei besteht darin, dass die Informationen die in diesen Texten enthalten sind in der Regel nur vom Menschen verstanden werden können. Ein Computer fast die Dokumente nur als eine Aneinanderreihung von Zeichen auf, die eigentliche Information ist für ihn nicht erkennbar.
In diesem Referat soll die Methode der Informationsextraktion vorgestellt werden durch die der oben genannte Mißstand zumindest zum Teil behoben werden könnte.

# 2 Dokumentensuche

Die zunehmende Verarbeitung von elektronischen Informationsverarbeitungs- und Speichermedien läßt die Menge der Daten und Dokumente, die digital zur Verfügung stehen, immer schneller anwachsen. Gleichzeitig werden die Daten durch die zunehmende Vernetzung für eine steigende Zahl von Personen zugänglich. Diese Daten können aber nur genutzt werden, wenn sie auch erschlossen sind, dh. wenn diejenigen, die sie nutzen wollen auch wissen, wo und wie sie die richtigen Informationen finden, was diese bedeuten und wie sie verwendet werden können. Dokumentensuchsysteme dienen nun dazu, aus einer Sammlung von Dokumenten diejenigen auszuwählen, die den Informationbedarf der Nutzer möglichst gut befriedigen [RP99].

## 2.1 Information Retrieval

Information Retrieval Systeme haben die Aufgabe, aus einer Menge von Dokumenten eine Teilmenge auszuwählen, die am besten zu einer Benutzeranfrage paßt. Die Benutzeranfrage geschieht mit Hilfe von Stichwörtern die durch den Benutzer eingegeben werden. Das System vergleicht die Anfrage mit den Dokumenten und wählt die Dokumente aus in denen diese Suchbegriffe vorkommen, also zum Thema passen könnten. Der Benutzer erhält als Ergebnis eine geordnete Menge relevanter Texten. So müssen nicht viele irrelevante Dokumente von Hand durchsucht werden.

# 3 Informationsextraktion

Bei der Informationsextraktion werden domänspezifische Informationen aus einen, dem System zuvor unbekannten, Text extrahiert (herausgezogen). Meistens handelt es sich bei diesen Dokumenten um natürlichsprachliche Texte. Hierbei wird nach bestimmten, vorher festgelegten Informationen gesucht und gleichzeitig irrelevanten Informationen überlesen. Was als relevant gilt, wird dabei durch vordefinierte domänspezifische Lexikoneinträge oder Regeln dem System fest vorgegeben [Neu01].
[CL96] formuliert das Ziel eines Informationsextraktionssystems folgendermaßen:
*The goal of IE research is to build systems that find and link relvant information while ignoring extraneous and irrelevant information.*

## 3.1 Informationsextraktion vs. Information Retrieval

In diesem Abschnitt wird das Informationsextraktionsverfahren mit dem Information Retrieval-Verfahren verglichen. Zur besseren Erläuterung vergleiche ich die beiden Verfahren wie in [Cun99] mit Hilfe eines Beispielszenarios.
Ein Benutzter sucht Informationen über Aktienkurse von Firmen mit Besitz in Bolivien. Um die entsprechenden Daten auszuwerten werden sie anschließend in einer Tabellenkalkulation verarbeitet.
Der Anwender einer Information Retrieval Systems gibt eine Menge relevanter Suchbegriffe ins System ein und erhält als Ergebnis eine geordnete Menge relevanter Dokumente. Dies können z. B. Zeitungsartikel sein, welche die Wörter enthalten, die mit den Suchbegriffen übereinstimmen oder zumindest Ähnlichkeiten aufweisen. Anschließend müssen die gefundenen Texte von einem Menschen durchgesehen, die signifikanten Dokumente ausgewählt und die erforderlichen Informationen entnommen werden. In einem weiteren Schritt könnte man die so gefundenen Daten in einer Tabellenkalkulation auswerten und dadurch die bestmöglichste Aktie finden. Hierbei sollte aber erwähnt werden, dass durch die Weiterverarbeitung eine Bewertung der Daten vorgenommen wird. Dies hört allerdings nicht zu den Aufgaben eines Information Retrieval Systems.
Im Gegensatz dazu kann ein richtig konfiguriertes Informationsextraktionssystem automatisch die relevanten Texte finden und die entsprechenden Informationen aus diesen Dokumenten extrahieren, also nur die spezifischen Daten herausziehen, bei gleichzeitigem überlesen von relevanter Information. Dem Benutzer bleibt das mühsame Durchsehen der Dokumente, wie beim Information Retrieval System, erspart. [Cun99]

## 3.2 Vor- & Nachteile

Es gibt viele verschiedene Vor- und Nachteile die mit den, im Kapitel 3.1, vorgestellten Informationsextraktionssystem zusammenhängt. Zum einen ist zu erwähnen, dass Informationsextraktionssysteme aufwendig sind. Dies hängt mit der Tatsache zusammen, dass die Systeme sehr wissensinteniv sind, dh. sie können meist nur durch Experten bedient und gewartet werden.
Ein sehr schwerwiegender Nachteil bei der Informationsextraktion ist, dass das System in der Regel domänspezifisch ist, es muß also an verschiedene Bereiche und Szenarien angepaßt werden. Zum Beispiel kann ein System das Informationen aus Zeitungsartikeln aus dem Bereich Luft und Raumfahrt extrahieren soll nicht mit Artikel über Wertpapiere angewendet werden.
Das Defizit, dass Informationsextraktionssysteme rechenintensiver als Information Retrieval Systeme sind, kann mit der Tatsache, dass bei langen Texten erheblich Zeit eingespart wird, behoben werden. Wie schon erwähnt, entfällt hier das Durchsehen der Dokumente von Hand [Cun99].

# 4 Informationsextraktionsschritte

Die Informationsextraktion wird in verschiedenen Schritten durchgeführt. Dabei werden in jeder Etappe verschiedene Informationen und Erkenntnisse über den Text gewonnen. Insgesamt gibt es nach [Cun99] folgende fünf Schritte:

- Named Entity recognition (NE)

- Coreference Resolution (CO)
- Template Element construction (TE)
- Template Relation construction (TR)
- Senario Template production (ST)

## 4.1 Named Entity recognition (NE)

Beim NE-Schritt werden Namen und Orte erkannt und klassifiziert. Die Namen beziehen sich dabei auf etwas Bestimmtes, sind also Einzeldinge. Siehe Punkt 5.

## 4.2 Coreference Resolution (CO)

Im CO-Schritt werden Referenzen zwischen den NE-Objekten identifiziert. Hier wird also erkannt, welche Dinge sich aufeinander beziehen. Siehe Punkt 5.

## 4.3 Template Element construction (TE)

Während dieser Phase werden den, im ersten Schritt gefundenen, NE-Objekten beschreibende Informationen angehängt. Beispielsweise wird das NE-Objekt `Auto` durch die Information `rot` ergänzt. Siehe Punkt 5.

## 4.4 Template Relation construction (TR)

Dieser Arbeitsschritt identifiziert Beziehungen zwischen den NE-Objekten. Es könnte erkannt werden, dass beispielsweise `Herr Meier` der Besitzer des `Autos` ist. Siehe Punkt 5.

## 4.5 Senario Template production (ST)

Hier werden die mittlerweile schon komplexeren TE- und TR-Objekte mit einem speziellen Ereignis-Szenario in Beziehung gesetzt. Siehe Punkt 5.

# 5 Anwendungsbeispiel

In diesem Abschnitt wird ein Beispiel gegeben, um die in Punkt 4 auf der vorherigen Seite genannten Phasen zu veranschaulichen:

> Die glänzend rote Rakete wurde am Dienstag gestartet. Sie ist die Erfindung von Dr. Hans Müller. Dr. Müller ist ein Wissenschaftler bei Raketenwerke Inc.

- **NE:** Rakete, Dienstag, Erfindung, Dr. Müller, Erfindung, Wissenschaftler, Raketenwerke Inc.
- **CO:** z. B. "Sie" bezieht sich auf die Rakete

- **TE:** Rakete = glänzend rote Erfindung von Dr. Müller
- **TR:** Müller arbeitet für die Raketenwerke Inc.
- **ST:** Raketenstart ist ein Ereignis, in welches die verschiedenen Entities verwickelt waren

# 6 Qualitätskriterien: Präzision und Vollständigkeit

Die Qualität eines Informationsextraktions Systems wird mit Hilfe der beiden Maße *Präzision* oder *Relevanz* und *Vollständigkeit(eng. Recall)* bestimmt.

## 6.1 Präzision

Präzision ist der relevante Teil der Ergebnismenge einer Abfrage. Wenn beispielsweise bei einer Informationsabfrage 17 von 20 Ergebnissen relevant sind, ist das eine gute Präzision, werden allerdings nur drei von 20 relevant sind, handelt es sich um eine schlechte Präzision.
Wie in Abbildung 1 zusehen ist, ergibt sich die Formel 1 für die Präzision P, wenn man die Schnittmenge der ausgegebenen und der relevantentexte Texte durch die Menge alle ausgegeben Texte teilt.

| | relevante Texte | irrelevante Texte |
|---|---|---|
| ausgegebene Texte | A | B |
| nicht ausgegebene Texte | C | D |

Abbildung 1: Präzision / Vollständigkeit

$$P = \frac{A}{A \cup B} \tag{1}$$

## 6.2 Vollständigkeit

Die Vollständigkeit gibt den Anteil der relevanten Dokumente, die gefunden wurden, gegenüber allen Dokumenten an. An der Abbildung 1 kann erneut erkannt werden, dass sich die Formel 2 aus der Schnittmenge der ausgegebenen und der relevantentexte Texte dividiert durch die Menge aller relevanten Dokumente. Wenn in einer Datenbank insgesamt 100 relevante Datensätze enthalten sind, es werden aber nur 15 gefunden, so ist die Vollständigkeit gering.

$$V = \frac{A}{A \cup C} \tag{2}$$

Die Präzision und die Vollständigkeit hängen unmittelbar zusammen. Nehmen wir an man will die Vollständigkeit erhöhen und gibt einfach alle Dokumente aus. Dadurch erhält man dann eine Vollständigkeit von 100%. Das hätte allerdings zur Folge, dass die Präzision sehr gering ausfallen würde.
Abbildung 2 veranschaulicht, dass die beiden Maße gegenläufig sind, dh. bei guter Vollständigkeit ergibt sich eine geringe Präzision und umgekehrt.

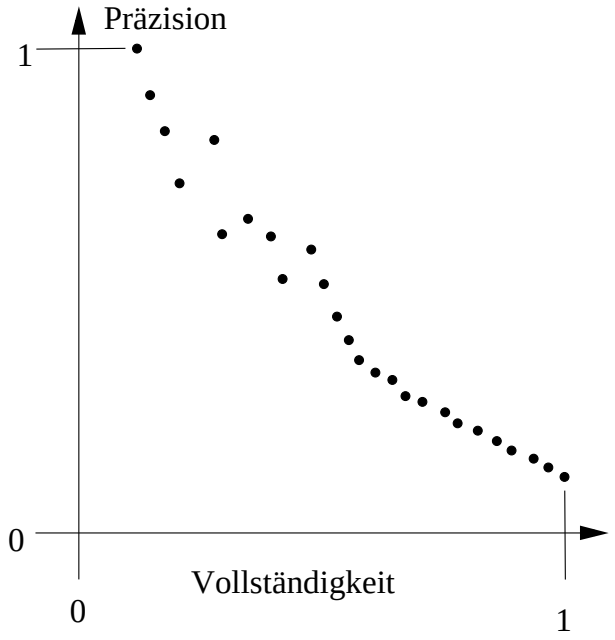



Abbildung 2: Präzision-Vollständigkeits-Diagramm [RP99]

## 6.3 F-Maß

Wie im Abschnitt 6.2 auf der vorherigen Seite schon angedeutet, ist es schwierig die Parameter Präzision und Vollständigkeit 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 ß=1 gesetzt) [Neu01]:

$$F = \frac{(\beta^2 + 1.0) \cdot P \cdot V}{(\beta^2 \cdot P) + V} \tag{3}$$

Im wesentlichen definiert die Formel ein geometrisches Mittel, dass über den Parameter ß gewichtet werden kann. Die Abweichung von 1 legt fest, ob dabei P oder V stärker gewichtet werden sollte.

# 7 Dokumentenanalyse

Dokumente enthalten verschiedene Informationen. Diese sind allerdings nicht nur rein durch die Bedeutung des Textes, d.h. rein inhaltlich, gegeben. Es gibt weiterhin eine ganze Reihe anderer Informationen die ein Dokument beinhaltet. Zum Beispiel können Informationen durch Anordnung von Textteilen im Gesamtdokument Auskunft über die Relevanz dieser Textfrakmente geben. [Fre98] betrachtet die folgenden fünf Sichtweisen um Informationen aus Dokumenten zu extrahieren.

- Terms View
- Mark-Up View
- Layout View
- Typographic View
- Linguistic View

Die Abbildung 5 auf der nächsten Seite zeigt eine Email, die ein Seminar in einer Universität ankündigt. Diese soll nun im Folgenden durch verschiedene Sichtweisen untersucht werden.

## 7.1 Terms View

Beim Terms View werden die im Text vorkommenden Wörter (terms) alphabetisch als Vektor angelegt (siehe Abbildung 3). Hierbei wird nun gezählt, wie oft die einzelnen Wörter im Gesamtdokument vorkommen (term frequency).

| 1 | 1 | 3 | 2 | 2 | 4 | 1 | 1 | 1 | ... | ... | ... | ... | term frequency |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| abstract | and | andrew | at | auditorium | cmu | computerized | concrete | construction | ... | ... | ... | ... | |

Abbildung 3: Die term frequency

Während des nächsten Schritts wird analysiert, wie häufig die einzelnen Wörter in allen Dokumenten auftreten (siehe Abbildung 4 auf der nächsten Seite).
Dieses Vorgehen gibt uns die Möglichkeit festzustellen, welche Wörter wichtig bzw. ausschlaggebend für die jeweiligen Dokumente sind. Kommt beispielsweise ein Wort sehr häufig in einem Dokument vor aber nur selten in der Gesamtheit aller Dokumente, dann kann davon ausgegangen werden, dass dieses Wort kennzeichnend für den entsprechenden Text ist.

```
<0.21.3.95.14.12.11.ed47+@andrew.cmu.edu.0>
Type: cmu.andrew.official.cmu-news
Topic: ECE Seminar
Dates: 30-Mar-95
Time: 4:00 - 5:00 PM
Place: Scaife Hall Auditorium
PostedBy: Edmund J. Delaney on 21-Mar-95 at 14:12 from andrew.cmu.edu
Abstract:

   COMPUTERIZED TESTING AND SIMULATION OF CONCRETE CONSTRUCTION

                        FARRO F. RADJY, PH.D.


                          President and Founder
                        Digital Site Systems, Inc.
                            Pittsburgh, PA


 DATE: Thursday, March 30, 1995
 TIME: 4:00 - 5:00 P.M.
 PLACE: Scaife Hall Auditorium
REFRESHMENTS at 3:45 P.M.

-----------------------
```

Abbildung 5: Email zur Seminaranmeldung

| 4 | n | 3 | n | 2 | 1 | 2 | 1 | 1 | ... | ... | ... | ... | n/doc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| abstract | and | andrew | at | auditorium | cmu | computerized | concrete | construction | ... | ... | ... | ... | |

Abbildung 4: Die inverse document frequency

## 7.2 Mark-Up View

Beim Mark-Up View werden Informationen aus dem Dokument gewonnen, indem nicht nur die einzelnen Wörter (terms), sondern zusätzlich auch noch sogenannte Meta-terms betrachtet werden. Diese Meta-terms geben Informationen über die Rolle, die ein Wort oder Textteil in einem Gesamtdokument spielt. Beispielweise enthält XML oder HTML spezielle Meta-terms (tags) [Fre98]. In HTML kann mit Hilfe dieser Meta-terms z. B. angegeben werden, wie der Titel einer Homepage lautet. Die Meta-terms können also spezielle Textteile kennzeichnen. Dadurch kann die Bedeutung von Textteilen, die mit

solchen Metazeichen gekennzeichnet sind, leicht zu interpretieren werden.
In Abbildung 6 erkennt man durch die tags <title> und </title> den Namen der Homepage. Da in der ersten großen Überschrift (durch <h2> und </h2> markiert) erneut die Wörter "Dayne Freitag" auftauchen, kann man davon ausgehen, dass es sich hierbei um den Besitzer der Homepage oder um ein Hauptthema auf der dieser Seite handelt.

```
<html>
<head>
<title>Dayne Freitag s Home Page</title>
</head>

<body bgcolor="#FFFFFF">

<center><h2>Dayne Freitag</h2>
<hr>
<h3><font face="Helvetica">Contents</font></h3></center>

<center>

<table>

<tr><td>
<font face="Courier">
Introduction.....................................
<a href="intro.html"><i>intro.html</i></a>
</font>
```

Abbildung 6: Der Mark-Up View

## 7.3 Layout View

Der Layout View betrachtet die 2-dimensionale Anordnung und Größe von Wörtern. Dadurch können wichtige Textobjekte bzw. Abschnitte erkannt werden. Z. B. Paragraphen, Überschriften oder auch Emailköpfe [Fre98].
Bei dieser Betrachtung wird so vorgegangen, dass einfach alle Zeichen, mit Ausnahme der Leerstellen, z. B. durch Sterne ersetzt werden. Mit geübtem Auge kann man in Abbildung 7 auf der nächsten Seite beispielsweise den Mailkopf erkennen, auch wenn man gar nicht weiß um was für einen Text es sich hierbei handelt.

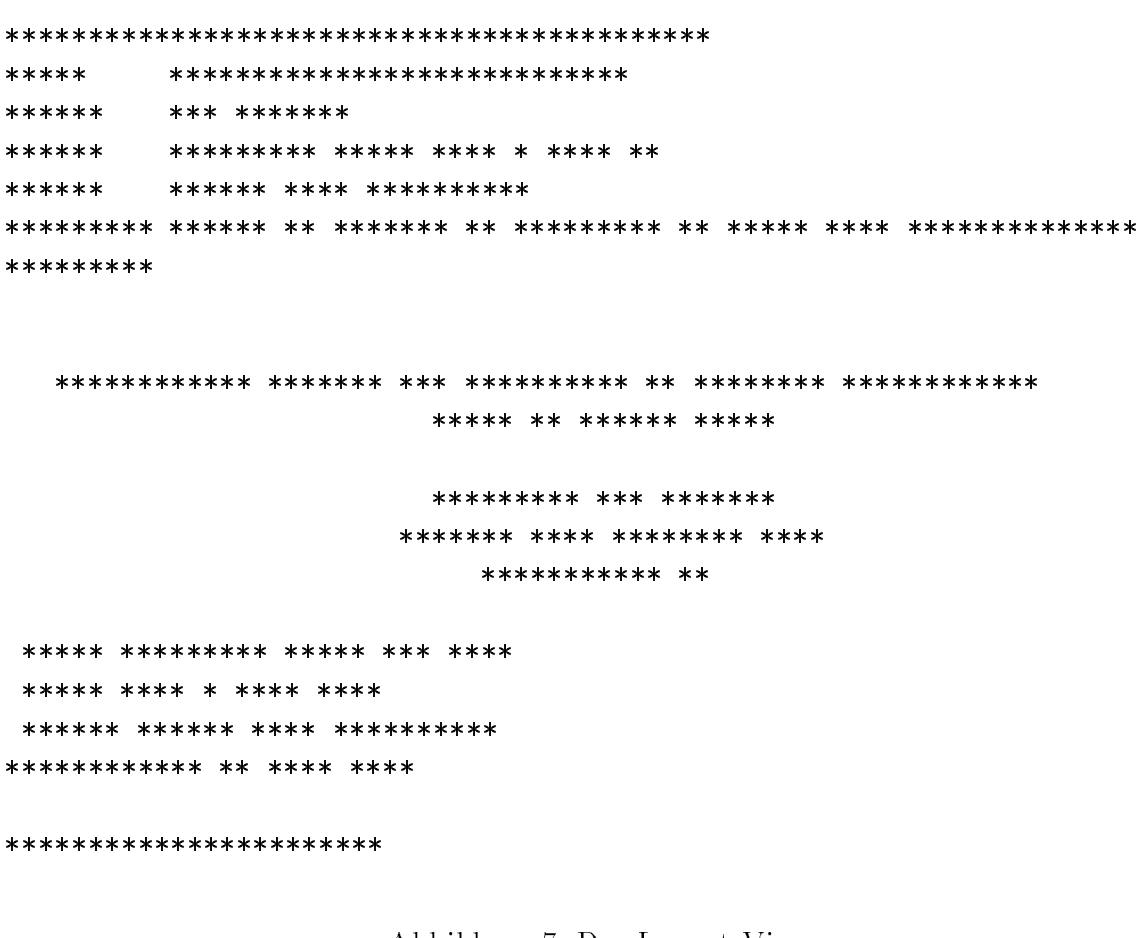


Abbildung 7: Der Layout View

## 7.4 Typographic View

Betrachtet man die Typographie eines Textes, unterteilt man die Zeichen in unterschiedliche Klassen [Fre98]:

- **Nummerisch:**

  Zahlen werden durch ‹9› ersetzt

- **Zeichensetzung:**

  Bei der Zeichensetzung erfolgt keine Änderung

- **Großbuchstaben:**

  Großbuchstaben werden durch ein ‹A› ersetzt. Die sonstigen Buchstaben ersetzt man durch ‹a›.

Halt man diese Umwandelungen durchgeführt, dann können, wie in Abbildung 8 auf der nächsten Seite erkennbar, z. B. Uhrzeiten, Datumsangaben und wichtige Textstellen (hier nun in Großbuchstaben) erkannt werden.

```
<9.99.9.99.99.99.99.aa99+@aaaaaa.aaa.aaa.9>
Aaaa:     aaa.aaaaaa.aaaaaaaa.aaa-aaaa
Aaaaa:    AAA Aaaaaaa
Aaaaa:    99-Aaa-99
Aaaa:     9:99 - 9:99 AA
Aaaaa:    Aaaaaa Aaaa Aaaaaaaaaa
AaaaaaAa: Aaaaaa A. Aaaaaaa aa 99-Aaa-99 aa 99:99 aaaa aaaaaa.aaa.aaa
Aaaaaaaa:


   AAAAAAAAAAAA AAAAAAA AAA AAAAAAAAAA AA AAAAAAAA AAAAAAAAAAAA

                          AAAAA A. AAAAA, AA.A.
                          Aaaaaaaaa aaa Aaaaaaa
                        Aaaaaaa Aaaa Aaaaaaa, Aaa.
                            Aaaaaaaaaa, AA


 AAAA: Aaaaaaaa, Aaaaa 99, 9999
 AAAA: 9:99 - 9:99 A.A.
 AAAAA: Aaaaaa Aaaa Aaaaaaaaaa
AAAAAAAAAAAA aa 9:99 A.A.
```

Abbildung 8: Der Typographic View

## 7.5 Linguistic View

Beim linguistische Betrachtung (Linguistic View) wird ein Satz in seine Bestandteile zerlegt. Dazu wird als erstes einmal ein Lexikon benötigt. Ein Lexikon ist eine Liste von Wörtern, die jedem Wort bestimmte Eigenschaften zuordnet. Siehe hierzu Abbildung 9 auf der nächsten Seite. Hieraus kann nun erkannt werden, um was für Wortarten es sich bei den einzelnen Wörtern handelt. Sind die einzelnen Wortarten identifiziert können die Wörter zu Gruppen zusammengefaßt werden. Die Abbildung 10 auf der nächsten Seite zeigt wie beispielsweise ein abstraktes Nomen (AN), eine Konjunktion und ein weiteres abstraktes Nomen zu einer Konjunktionsgruppe zusammengefaßt werden. Diese bildet mit einem Adjektiv zusammen eine abstrakte Nomengruppe (ANG).

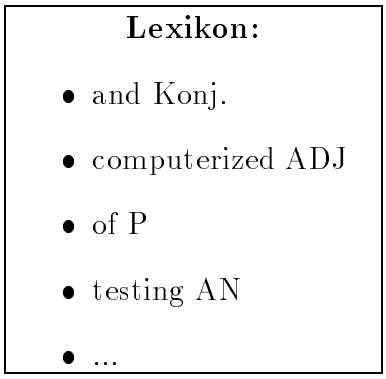



Abbildung 9: Lexikon

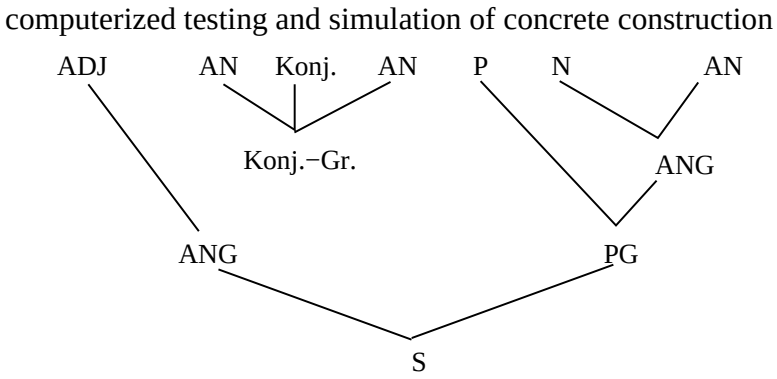



Abbildung 10: Linguistic View

# 8 Weitere Bereiche des Wissensmanagements

Es gibt mehrere Gründe Wissensmanagement zu betreiben. Zum einen erhofft man sich nach [Ort00] eine bessere Nutzung des in betrieblichen Datensammlungen vorhandenen Wissens, weiterhin soll beispielsweise Wissen, dass Mitarbeiter besitzen für die Firma erhalten bleiben, auch wenn Mitarbeiter den Betrieb verlassen. Außerdem sollen aus Daten weiterreichende Schlüsse gezogen werden können bzw. zukünftige Daten vorhersagen werden [Bel97].

Das Semantic Web realisiert die Version, dass man ein Netzwerk von Internetseiten entwickelt, die zusätzlich zu den für den Menschen verständlichen Informationen auch Informationen enthält, die vom Rechner interpretiert werden können. Diese zusätzlichen Informationen können mit Hilfe von speziellen Metadaten gegeben werden. Da allerdings ein Agent, der diese Informationen liest, keine Informationen darüber besitzt, was diese Zusatzangaben genau bedeuten und wie er sie interpretieren muß, werden Ontologien eingesetzt. Unter einer Ontologie versteht [Gru93] eine Spezifikation einer Konzeption. Ein Konzeption ist wiederum eine abstrahierte Sicht auf den Teil der Welt, der von Interesse ist. Mit Hilfe dieser Ontologie können nun Beziehungen und Zusammenhänge erkannt werden und somit z. B. Agenten verfügbar gemacht werden. Da bei der Informationsextraktion nur der Text eines Dokumentes analysiert wird, also auf Wörter, Satzbau und Wortzusammenhänge, kann hier von der Nutzung einer Ontologie abgesehen werden.

Sollen Zeitreihen analysiert werden, d.h. man will Daten analysieren die in einer zeitlichen Folge auftreten, werden die Verfahren nach [Höp01] oder [DLM+98] verwendet. Dieser Verfahren benötigen Daten, die natürlich auch in Dokumenten enthalten sein können. Mit Hilfe der Textextraktion können dann Daten aus diesen Dokumenten extrahiert und weiterverarbeitet werden.
Bei den Verfahren der Zeitreihenanalyse wird unteranderem der Apriori-Algorithmus nach [AMS+96] verwendet. Mit diesem Algorithmus werden Assoziationsregeln aufgestellt. Die Beziehungen sollen meist aus den Daten einer großen Datenbank erkannt werden. Ein Bezug zum Thema der Informationsextraktion besteht allerdings nicht.
RDT/DB nach [BM96] arbeitet direkt mit einer Datenbank zusammen. Mit diesem ILP-Wissensentdeckungswerkzeug ist es möglich bestimmte Regeln aus der Datenbank abzuleiten.
Eine weitere Möglichkeit große Datenmengen aus Datenbanken zu analysieren, ist die Verwendung von Data Cubes. Dies ist meist nötig wenn die Daten auf unterschiedlichen Detaillierungsstufen und über verschiedene Kombinationen von Attributen zusammengefaßt werden müssen. Eine Einführung in das Thema der Data Cubes wird in [GCB+97] und in [HRU96] geben. Eine Beziehung zum Thema dieses Referats ist nicht gegeben.

Sollen Subgruppen in Daten erkannt werden, dann kann das MIDOS Verfahren nach [Wro97] angewendet werden. Hierbei handelt es sich um ein deskriptives Lernverfahren bei dem die Subgruppenerkennung über mehrere Relationen hinweg durchgeführt. Auf eine weiter Einführung wird hier auf Grund fehlender Bezüge zum Hauptthema verzichtet.

# 9 Aussichten

Das Problem das sich bei Standardinformationsextraktionssystemen ergibt, besteht darin, dass die Systeme sehr domänspezifisch sind. Eine Anpassung an andere Bereiche ist sehr schwierig und kann in der Regel nur durch Experten durchgeführt werden. Es ist also sinnvoll Systeme zu entwickeln, die auf verschiedenen Domänen angewendet werden können.

# Literatur


[AMS+96] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, chapter 12, pages 307–328. AAAI Press/The MIT Press, Cambridge Massachusetts, London England, 1996.

[Bel97] Gene Bellinger. *Knowledge Management: Emerging Thoughts by Gene Bellinger*. 1997.

[BM96] Peter Brockhausen and Katharina Morik. Direct access of an ILP algorithm to a database management system. In Bernhard Pfaringer and Johannes Fürnkranz, editors, *Data Mining with Inductive Logic Programming (ILP for KDD)*, MLnet Sponsored Familiarization Workshop, pages 95–110, Bari, Italy, jul 1996.

[CL96] Jim Cowie and Wendy Lehnert. Informations Extraktion. *Communications of the ACM*, 39(1):80–91, 1996.

[Cun99] Hamish Cunningham. *Informations Extraction a User Guide*. Institute for Language, Speech and Hearing (ILASH) and Department of Computer Sciene Universtity of Sheffield, UK, apr 1999. `http://www.dcs.shef.ac.uk/~hamish`.

[DLM+98] Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. Rule Discovery from Time Series. In Rakesh Agrawal, Paul E. Stolorz, and Gregory Piatetsky-Shapiro, editors, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pages 16 – 22, Ney York City, 1998. AAAI Press.

[Fre98] Dayne Freitag. *Maschine Learning for Information Extraction in Informal Domains*. PhD thesis, Computer Science Department Carnegie Mellon University Pittsburgh, PA, November 1998.

[GCB+97] Jim Gray, Surajit Chaudhuri, Adam Bosworth, Andrew Layman, Don Reichart, and Murali Venkatrao. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Mining and Knowledge Discovery*, 1(1):29 – 54, 1997.

[Gru93] T. R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*, pages 199–220, June 1993.

[Höp01] Frank Höppner. Learning temporal rules from state sequences. In Miroslav Kubat and Katharina Morik, editors, *Workshop notes of the IJCAI-01 Workshop on* Learning from Temporal and Spatial Data, pages 25–31, Menlo Park, CA, USA, 2001. IJCAI, AAAI Press. Held in conjunction with the International Joint Conference on Artificial Intelligence (IJCAI).

[HRU96] Venky Harinarayan, Anand Rajaraman, and Jeffrey D. Ullman. Implementing data cubes efficiently. pages 205–216, 1996.

[Neu01] Dr. Günter Neumann. *Informationsextraktion.* DFKI GmbH, 2001. `http://www.dfki.de/~neumann/publications/new-ps/ie.pdf`.

[Ort00] Erich Ortner. Wissensmanagement, teil 2: Systeme und werkzeuge. *Informatik Spektrum*, Juni 2000.

[RP99] Peter Rechenberg and Gustav Pomberger. *Informatik Handbuch*, pages 909–923. Carl Hanser Verlag München Wien, Zweite edition, 1999.

[Wro97] Stefan Wrobel. An algorithm for multi–relational discovery of subgroups. In J. Komorowski and J. Zytkow, editors, *Principles of Data Minig and Knowledge Discovery: First European Symposium (PKDD 97)*, pages 78–87, Berlin, New York, 1997. Springer.