

Meike Klettke, Mathias Bietz, Ilvio Bruder, Andreas Heuer, Denny Priebe, Günter Neumann, Markus Becker, Jochen Bedersdorfer, Hans Uszkoreit, Alexander Maedche, Steffen Staab, Rudi Studer

GETESS - Ontologien, Objektrelationale Datenbanken und Textanalyse als Bausteine einer Semantischen Suchmaschine

In diesem Artikel wird dargestellt, wie Verfahren aus der Wissensrepräsentation, der Computerlinguistik, dem Information Retrieval und aus dem Bereich Datenbanken eingesetzt werden können, um für Suchmaschinen und in Dokumentenserver neue Funktionalitäten bereitzustellen.

1 Einführung

Die meisten Leser dieser Zeitschrift werden über praktische Erfahrungen mit Suchmaschinen verfügen. Dabei werden sie sowohl die Erfahrung kennen, dass man einige gesuchte Informationen sehr schnell finden kann. Bei anderen Anfragen bemerkt man aber auch die Grenzen der eingesetzten Methoden. Ursache dafür ist, dass die gegenwärtig verfügbaren Suchmaschinen meist Verfahren des Information Retrieval einsetzen. Hierbei handelt es sich um sehr ausgefeilte statistische und überwiegend wortbasierte arbeitende Standard-Verfahren, die - wie man im praktischen Einsatz sehen kann - sehr effizient arbeiten.

Für einige Arten von Anfragen reichen diese statistischen und auf Einzelworten basierenden Verfahren nicht aus, man wünscht sich oft, dass die Suchmaschinen sowohl die Anfragen, die man an sie stellt, als auch die Dokumente der Ergebnismenge besser »verstanden« hätten.

Im BMBF-Projekt GETESS (German Text Exploitation and Search System) wurde aufgrund dieser Erfahrungen der Prototyp einer Suchmaschine entwickelt, der Methoden aus der Computerlinguistik und der Wissensrepräsentation in eine Suchmaschine integriert. Das gleiche Szenario kann auch als Dokumentenserver eingesetzt werden. Hierbei werden die Dokumente nicht über Agenten im Internet gesucht und analysiert, sondern die jeweiligen Dokumente sind lokal verfügbar.

Suchmaschinen und lokale Dokumentenserver stellen komplexe Systeme dar. Zur intelligenten Suche in den Dokumenten werden Technologien aus verschiedenen Gebieten wie Wissensrepräsentation, Computerlinguistik, Information Retrieval und Datenbanken integriert. Gerade aus dem Zusammenwirken dieser verschiedenen Technologien traten dabei Synergieeffekte auf.

Einige der dabei entwickelten Verfahren werden wir in diesem Artikel vorstellen. Dabei soll ein besonderer Schwerpunkt darauf liegen, welche Technologien aus verschiedenen Forschungsgebieten der Informatik sich auf welche Weise ergänzen und gegenseitig bereichern.

Das BMBF-Projekt GETESS (German Text Exploitation and Search System, Projektlaufzeit: September 1998 - Juni 2001) wurde in Zusammenarbeit

zwischen dem DFKI Saarbrücken, der Universität Karlsruhe, der Universität Rostock und der GECKO mbH Rostock bearbeitet. Im Rahmen des Projektes entstand eine Konzeption für eine Suchmaschine oder einen Dokumentenserver. Dazu wurden die notwendigen Basistechnologien konzipiert, implementiert, integriert und an die konkreten Anforderungen des Gesamtprojektes angepaßt. Es wurden zwei Anwendungsgebiete (Tourismus, Finanzinformationen) betrachtet.

2 Überblick

Abbildung 1 zeigt die Gesamtkonzeption des Projektes GETESS in etwas vereinfachter Form. Dabei gibt es im System zwei Hauptprozesse, das Einsammeln, Aufbereiten und Speichern der Informationen in einer Art Index und die Entwicklung einer Schnittstelle für die Suche in den Indexinformationen.

Neben diesen Prozessen existieren weitere wie zum Beispiel das Erstellen oder Verändern der Ontologien, der Aufbau von Wörterbüchern, das Erstellen von Datenbank-Entwürfen, usw.

Im Folgenden wollen wir einen »Blick ins Innere« dieses Tools werfen. In Abbildung 1 ist zu sehen, dass Ontologien an zentraler Stelle eingesetzt werden

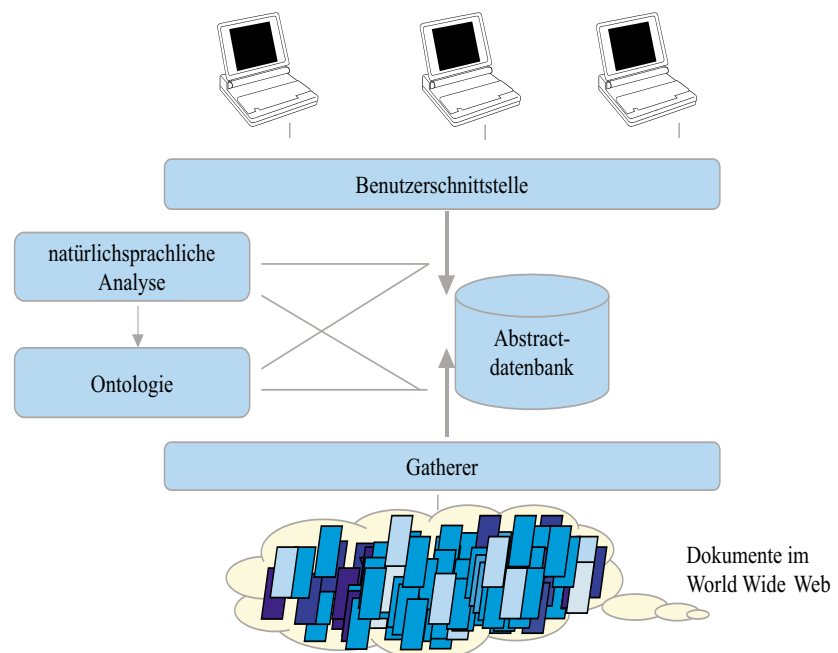


Abb. 1: Informationsfluss in der GETESS-Suchmaschine

und von vielen anderen Komponenten genutzt werden. Im folgenden Abschnitt werden Ontologien detaillierter vorgestellt und der konkrete Einsatz im Projekt GETESS beschrieben.

3 Ontologie

Ontologien sind formale Modelle einer Anwendungsdomäne, die dazu dienen, den Austausch und das Teilen von Wissen zu erleichtern [Guarino 1998]. Auf der methodischen Seite werden Techniken der objektorientierten Modellierung konsequent so weiterentwickelt, dass die Modelle nicht bloß zur Strukturierung von Software dienen, sondern auch ein explizites Element der Benutzerschnittstelle darstellen und zur Laufzeit verwendet werden. Auf der sozio-kulturellen Seite erfordern Ontologien daher die Eignung einer Gruppe von Anwendern auf die jeweiligen Begriffe und deren Zusammenhänge.

3.1 Referenz und Bedeutung

Ontologien dienen der Verbesserung der Kommunikation zwischen menschlichen und maschinellen Akteuren. Hierbei befinden sich die Akteure (ob mit oder ohne Ontologie) in einer Kommunikationssituation, deren herausragende Eigenschaften durch das semiotische Dreieck [Ogden, Richards 1923] aufgezeigt werden. Das semiotische Dreieck illustriert die Interaktion zwischen Worten (oder allgemeiner: Symbolen), Begriffen und realen Dingen in der Welt (vgl. Abbildung 2). Worte, die benutzt werden, um Informationen zu übertragen, können die Essenz einer Referenz, das ist der Begriff oder das referenzierte Ding in der Welt, nicht vollständig erfassen. Dennoch gibt es eine Korrespondenz zwischen Wort, Begriff und Ding.

Die Auswahl einer bestimmten Korrespondenz aus der Vielzahl a priori möglicher Korrespondenzen geschieht durch den Empfänger einer Nachricht. Hierbei benutzen verschiedene Empfänger unter Umständen verschiedene Begriffsbildungen und haben einen variierenden Erfahrungshintergrund, was wiederum zu verschiedenen Resultaten bezüglich der Korrespondenz zwischen einem Wort

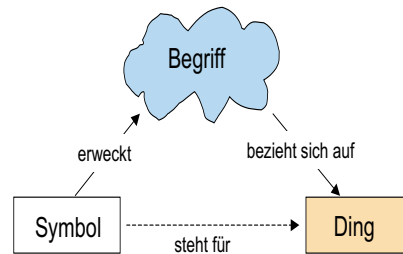


Abb. 2: Semiotisches Dreieck (in der Tradition von Peirce, Saussure und Frege)

und den möglichen Begriffen und Dingen in der Welt führen kann.

Eine Ontologie wird durch eine logische Theorie ausgedrückt, die sich aus einem Vokabular und einer Menge von logischen Aussagen zu der jeweils interessierenden Anwendungsdomäne zusammensetzt. Die logische Theorie spezifiziert Beziehungen zwischen Worten (allgemeiner: Symbolen) und schränkt dabei die Menge der möglichen Interpretationen für Worte und ihren zugehörigen Beziehungen ein. Auf diese Weise reduziert eine Ontologie die Anzahl möglicher Korrespondenzen zwischen Worten und Dingen, die der Empfänger einer Nachricht, der sich auf eine Ontologie festgelegt hat, als gültig interpretieren kann. Idealerweise bleibt im Kontext von Kommunikationssituation und Ontologie für jedes Wort aus dem Vokabular genau eine Korrespondenz mit Begriffen und

Dingen in der Welt übrig.

3.2 Die Ontologie in GETESS

Wie oben beschrieben, stellen Ontologien semantische Vokabulare dar, welche den Inhalt einer bestimmten Domäne (wie zum Beispiel Tourismus) strukturieren. In diesem Sinne verbindet die Ontologie in GETESS die anderen Module des Systems (die Datenbank, das Sprachverarbeitungssystem und die Benutzerschnittstelle) und fungiert als eine Art Mediator zwischen den Modulen.

Im folgenden Beispiel soll die Mediatorfunktion der Ontologie exemplarisch dargestellt werden. Abbildung 3 stellt die Interaktion zwischen der Ontologie und dem Sprachverarbeitungssystem dar. Die Sprachverarbeitung erkennt syntaktische Relationen zwischen Wörtern und Phrasen. Ob die syntaktische Relation in eine semantische Relation überführt werden kann, wird in der Ontologie entschieden, da geprüft wird, ob eine adäquate konzeptuelle Relation in der Ontologie existiert.

Im Beispiel findet die Sprachverarbeitungskomponente eine syntaktische Relation zwischen Musikfestival und Usedom in der Phrase Usedoms Musikfestival. Über die Referenz von Lexikon auf Ontologie wird das AbstractWort Musikfestival auf die In-

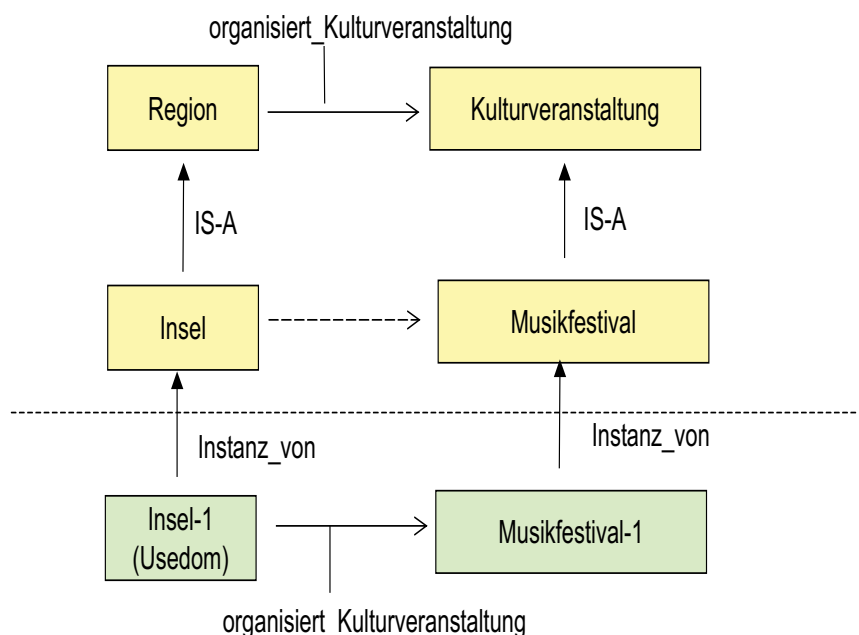


Abb. 3: Interaktion zwischen Ontologie und Sprachverarbeitung

stanz Musikfestival-1 des Begriffs MUSIKFESTIVAL und der Eigenname Usedom auf die Instanz Insel-1 auf den Begriff INSEL abgebildet. In der Ontologie existiert eine konzeptuelle Relation

organisiert_Kulturveranstaltungen zwischen Regionen und Kulturveranstaltungen, welche aufgrund der Vererbungssemantik auch für die beiden Begriffe INSEL und MUSIKFESTIVAL gültig ist. Als Ergebnis der Verarbeitung wird schließlich der relationale Fakt `organisiert_Kulturveranstaltungen(Insel-1, Musikfestival-1)` in der Datenbank gespeichert. Dieses Beispiel zeigt, wie die Ontologie den Verarbeitungsprozess zwischen Sprachverarbeitung und Speicherung in der Datenbank bestimmt.

Wie oben gesehen kann die Ontologie durch ihre zentrale Rolle auch zu einem »Engineering-Flaschenhals« werden. Im Rahmen des Projektes GETESS wurden deshalb auch Methoden und Tools zum semi-automatischen Engineering entwickelt, welche den Prozess der Ontologieentwicklung und -instandhaltung auf Basis existierender Daten unterstützen [Macedche, Staab 2001].

4 Natürlichsprachige Analyse

Die zentrale Aufgabe des DFKI LT-Labs im Rahmen des GETESS-Projektes ist die Erforschung und Entwicklung von natürlichsprachlichen Methoden zur:

- Extraktion bzw. Generierung von Abstracts aus Texten (ein Abstract ist eine Menge von instanziierten Templates bzw. domainspezifischen Konzeptbeschreibungen),
- linguistischen Analyse von Benutzeranfragen im Rahmen des GETESS-Dialogsystems,
- natürlichsprachlichen Generierung von Zusammenfassungen zur benutzerfreundlichen Präsentation von Suchergebnissen (i.a. eine Menge von Abstracts),
- Multilingualität (Deutsch, Englisch) in allen Bereichen mit dem Schwerpunkt auf der linguistischen Analyse und der natürlichsprachigen Generierung von Zusammenfassungen.

Um den notwendigen Grad an Robustheit und Effizienz zu erreichen, werden in allen Punkten Methoden der flachen Sprachverarbeitung und der Informationsextraktion eingesetzt. Ein weiterer Schwerpunkt ist die Realisierung einer systematischen Verbindung des linguistischen Wissens mit Domänenwissen.

Ausgangspunkt der flachen Textverarbeitung in GETESS ist das am LT-Lab entwickelte System SMES, ein Kernsystem zur Informationsextraktion, vgl. [Neumann et al. 1997], [Neumann et al. 2000]. Die relevanten Eigenschaften von SMES sind:

- robuste und effiziente Verarbeitung von großen Mengen freier deutscher Texte,
- umfangreiche linguistische Wissensquellen (u.a. sehr große Stammlexika, Eigennamengrammatiken, Phrasen- und Satzgrammatiken),
- Erweiterbarkeit der linguistischen Wissensquellen durch deklarative Formalismen,
- hoher Grad an Modularität,
- Adaptierbarkeit der Kernfunktionalität auf andere Sprachen.

Im Wesentlichen basiert die Verarbeitung in SMES auf dem Einsatz von gewichteten endlichen Automaten zur lexikalischen und syntaktischen Analyse. SMES verfügt über innovative top-down/bottom-up gemischte Chunk-Parsingstrategien, die gerade bei Anwendungen, wie die GETESS-Suchmaschine, rein bottom-up orientierten Chunk-Parsern oder tiefen Parsingstrategien (noch) überlegen sind.

4.1 Bilinguale NLP-Kernmaschine

SMES wurde im Rahmen des GETESS-Projektes in wesentlichen Bereichen verbessert und erweitert. So konnte eine volle bilinguale NL-Kernmaschine entwickelt werden, die die Verarbeitung von Deutschen und Englischen, sogar gemischtsprachlichen Texten ermöglicht. Um den hohen Anforderungen des multilingualen Grammar-Engineerings gerecht zu werden, verfügt die neue SMES Version über einen sehr hohen Grad an Modularität und Deklarativität. Dies wird im Wesentlichen über eine systematische

Trennung in syntaktischen Erkennungsteil (Eingabe) und domänenspezifische Normalisierung/Strukturierung (Ausgabe) erreicht.

Eine zentrale Aufgabe bei der domänenspezifischen Extraktion ist der Einsatz von Verfahren zur Auflösung von Referenzen zwischen extrahierten Termen (vgl. »Microsoft steigerte seinen Umsatz. Der Softwarehersteller ...«, wo Microsoft und Softwarehersteller auf dieselbe Instanz verweisen). Im Rahmen des GETESS-Projektes wurden hierbei sehr robuste auf flachen Methoden basierende Strategien entwickelt. Bemerkenswert ist hier, dass diese Verfahren kein volles Parsing voraussetzen, sondern bereits auf der Phrasenanalyse aufsetzen können, wobei lokale Präferenzmechanismen ein Ranking der möglichen Kandidatenmenge berechnen.

Auf allen Stufen können die Ergebnisse in Form von XML-Strukturen zwischen-gespeichert werden, die auch als externe Schnittstellen zu anderen Komponenten des GETESS-Systems dienen (zum Beispiel Dialogsystem und DBMS). Hierbei wird ein uniformes Annotationsschema verwendet, das eine volle Verknüpfung der Ergebnisse auf allen Ebenen ermöglicht.

4.2 Abstractgenerierung

Ausgehend von einer domänenspezifischen Ontologie (Tourismus, Finanzen) ist das zentrale Ziel die Identifikation und Extraktion von relevanten Textstellen und die Bestimmung ihrer domänenspezifischen Beziehungen. Somit stellt ein Abstract eine kondensierte, semantische Repräsentation der relevanten Teile eines Textes dar. Bei der Berechnung von Abstracts verfolgen wir im GETESS-Projekt eine optimistische bottom-up Strategie. Zuerst werden einfache Beziehung zwischen relevanten Wörtern und Konzepten mittels eines Domänenlexikons, das eine unmittelbare Beziehung zu Wortstämmen von SMES und Konzepten der Ontologie definiert, hergestellt. In der aktuellen Version von GETESS werden hierbei im Wesentlichen Nomen betrachtet. Nach der syntaktischen Analyse des Textes können nun alle nominalen Terme, die einen Bezug zum Domänenlexikon aufwei-

sen, extrahiert werden (Konzeptspotting). Während der Termextraktion werden Instanznamen vergeben, mit Hilfe derer textlinguistische Bezüge modelliert werden. Zentraler Aspekt bei der Abstractgenerierung ist die Erstellung relationaler Beziehungen zwischen nominalen Termen, die prinzipiell durch Inferenz über der Ontologie möglich ist. Daher werden in einem nächsten Schritt alle möglichen Paare über der Menge der nominalen Terme gebildet (Kandidatenmenge) und der Inferenzmaschine übergeben. Sie überprüft, für welche Paare welche Relationen definiert sind und führt damit eine Filterung durch. Alle dadurch relationierten Termpaare (relationale Tupel) definieren das Abstract für ein Textdokument. Um die Größe der Kandidatenmenge sinnvoll zu beschränken, werden aus den Ergebnissen der domänenspezifischen Termextraktion Heuristiken eingesetzt, u.a. linguistische Heuristiken (z.B. minimal Attachment) und Heuristiken, die die HTML Struktur berücksichtigen. Hier ein Beispiel für ein Abstract für den Satz: Wir bieten ein Hotel mit ruhigen Zimmern.

```
<list>
  <tuple name='GETESS-output'>
    <fragment>
      <type>Zimmer</type>
      <inst>Zimmer_193</inst>
      <name>zimmer</name>
    </fragment>
    <fragment>
      <type>Zimmer</type>
      <inst>Zimmer_193</inst>
      <name>ruhig</name>
    </fragment>
    <constraint
      type='HEURISTIC'>
      MODIFIER
    </constraint>
  </tuple>
  <tuple name='GETESS-output'
    rel='hat_Zimmer'>
    <fragment>
      <type>Hotel</type>
      <inst>Hotel_192</inst>
      <name>hotel</name>
    </fragment>
    <fragment>
      <type>Zimmer</type>
      <inst>Zimmer_193</inst>
      <name>zimmer</name>
    </fragment>
    <constraint
      type='HEURISTIC'>
      SENTENCE-HEURISTIC
      NP-PP-HEURISTIC
    </constraint>
```

```
</tuple>
</list>
```

4.3 Parsing von Benutzeranfragen

Die einheitliche Architektur für die Verarbeitung von domänenrelevanten Dokumenten und Anfragen sieht die Verwendung derselben Kern-Module von SMES bis hin zum Chunk Parser vor. Der Dialogkomponente werden allerdings nicht wie in der Abstractgenerierung relationierte Tupel angeboten, sondern alles analysierte Material, da bei einfachen Anfragen möglicherweise gar keine Relationen gefunden werden. Die XML-Schnittstelle muss dabei nicht nur gefundene Chunks repräsentieren, sondern auch Wörter, die nicht zu Phrasen zusammengefasst wurden, um eine robuste Anfrageanalyse zu gewährleisten. Folgendes Beispiel zeigt das Parsingergebnis für die Anfrage »Is there a hotel nearby?«:

```
<QUERY>
  <TOKEN string="Is">
    <READING stem="be"
      pos="V">
      <INFLECTION tense="PRES"
        form="FIN" person="3"
        number="S"/>
    </READING>
  </TOKEN>
  <TOKEN string="there">
    <READING stem="there"
      pos="ÄDV">
    </READING>
  </TOKEN>
  <NP i="2" j="4">
    <NPSEM>
      <DET>a</DET>
      <HEAD>hotel</HEAD>
      <DOMAIN-TYPE>Hotel
    </DOMAIN-TYPE>
    </NPSEM>
  </NP>
  <TOKEN string="nearby">
    <READING stem="nearby"
      pos="ÄDV">
    </READING>
  </TOKEN>
  <SENTENCE-END i="5" j="6">
    <MARKER>?</MARKER>
    <QU-ELE>T</QU-ELE>
  </SENTENCE-END>
</QUERY>
```

Die Token »Is« und »nearby« wurden keiner Phrase zugeordnet, deshalb wird das Ergebnis der morphologischen Anfrageanalyse geliefert. »a hotel« wurde

als NP mit dem Domäentypen HOTEL erkannt und das Fragezeichen als Satzendermarker mit Fragesemantik.

4.4 Generierung von Suchergebnissen

Wesentliche Aufgabe der Textgenerierung ist die Erstellung von natürlich-sprachlichen Texten auf Basis des Ergebnisses der Auswertung einer Datenbank-anfrage. In der Regel handelt es sich hierbei um relevante Stellen aus den aktivierten Dokumenten und ihre entsprechenden Abstracts, die im Rahmen der Textkondensation bestimmt wurden. In GETESS werden flache Generierungsverfahren auf der Basis von parametrisierbaren Textschablonen eingesetzt. Diese bestehen aus linguistischen Mustern (konkrete Wörter, Stämme, Phrasen) und Typen aus der Ontologie. Die Typen stellen Platzhalter für die durch den Datenbankzugriff bestimmten Instanzen der Abstracts dar. Beispiel: »[Stadt] hat viele [Hotel]«, »Das Hotel [Hotel] befindet sich in [Stadt].« oder »[Hotel] hat von [Start-Time] bis [End-Time] geöffnet.«

Anstatt die Textschablonen durch aufwendige Inspektion der Textmenge manuell zu erstellen, folgen wir einem reversiblen und lernbasierten Ansatz. Im aktuellen Kontext bedeutet Reversibilität die Verwendung von durch die Textkondensation berechneten Abstracts und ihrer linguistischen Zwischenergebnisse (auf Basis von XML). Zur Bestimmung von Textschablonen haben wir eine automatische Methode auf Basis des Explanation-based Learnings (EBL) entwickelt (basierend auf [Neumann 1997]) und in Java implementiert. Das Verfahren besteht aus 2 Schritten:

- Erstellung der Textschablonen,
- Erstellung eines semantischen Index auf die Patterns.

Der erste basiert auf einer Bestimmung der relevanten analysierten Textpassagen und ihrer approximativen Generalisierung. Im zweiten Schritt wird ein Entscheidungsbaum mit den domänen-spezifischen Typen aus den korrespondierenden Abstracts erstellt.

Diese dienen als Traversierungshilfen zu entsprechenden Generierungspatterns für entsprechende semantische Eingaben,

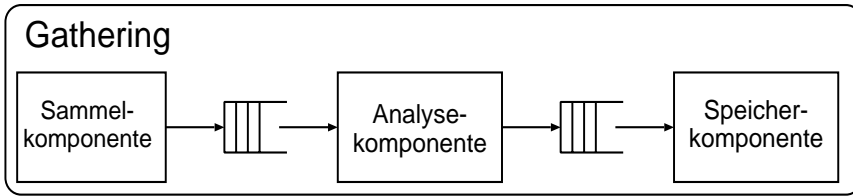


Abb. 4: Gatherer-Architektur

wie zum Beispiel:

```
<abstr>
  <concept
    name="Unterkunft">
    <name>Hotel Schoenborn
    </name>
    <slot name="in_Gebiet">
      <concept name="Stadt">
        <name>Rostock</name>
      </concept>
    </slot>
  </concept>
</abstr>
```

Dies führt zur Auswahl folgender Patterns (sortiert nach Frequenz):

```
[NP HeadUnterkunft]
[V liegt]
[PP in [NP HeadStadt]]
[NP Head$C1] [V liegt]
[PP in [NP Head$C2]]
[NP HeadHotel]
[V befindet sich in]
[NP HeadHafenstadt]]
```

Was letztlich zu folgenden Äußerungen führt:

```
[NP Hotel Schoenborn]
[V liegt] [PP in
[NP Rostock]]
[NP Hotel Schoenborn]
[V befindet sich in]
[NP Rostock]
```

Als Fazit kann gesagt werden, dass mit der Realisierung des Systems SMES bewiesen ist, dass eine robuste und effiziente NL Verarbeitung auf freien Deutschen Texten unter anwendungsnahen Bedingungen machbar ist. Am LT-Lab des DFKI sind wesentliche Komponenten für ein volles bilinguales IE-Kernsystem für freie Texte entstanden, das nicht nur eine Grundlage für robuste und effiziente Frage/Antwortsysteme darstellt, sondern durch seine systematische Integration

von flacher NLP und domänenspezifischen Ontologien auch große Bedeutung in Gebieten hat, wie dem Text Mining oder der Extraktion von Ontologien aus Texten.

5 Der Suchkern

Die Ontologie-Wissensbasis und eine linguistische Analyse sind zwei mächtige Werkzeuge für den Aufbau und die Konzeption der hier beschriebenen Internet-Suchmaschine. Der nächste Schritt beinhaltet nun die Integration dieser Werkzeuge in die Kernkomponente einer Suchmaschine.

Der Suchkern ist die Kernkomponente des Gathering-Prozesses. Der Suchkern ist mit den Gatherern herkömmlicher Internet-Suchmaschinen vergleichbar. Die Aufgaben eines Gatherers umfassen im Wesentlichen:

- das Einsammeln der Internet-Dokumente,
- das Vorverarbeiten der Dokumente, um beispielsweise Referenzen auf weitere Dokumente aufzulösen oder Metadaten zu sammeln,
- das Indexieren der Dokumentinhalte sowie
- das geeignete Ablegen bzw. Speichern der Daten.

In der GETESS-Umgebung kommen nun weitere, teils konkretere Aufgaben dazu. Dazu gehören:

- das Integrieren spezieller Analysetechniken, hier linguistische Analyse in die Gathering-Architektur,
- das Vorbereiten der Dokumente auf die speziellen Analysetechniken, sowie
- das Sammeln und Ausnutzen von Strukturdaten und Metadaten für die speziellen Analysetechniken.

5.1 Komponenten des Suchkerns

Zu der für das Projekt GETESS konzipierten Gatherer-Variante ([Bruder et al. 2000]) gehören im Wesentlichen die Sammelkomponente, die Analysekomponente und die Speicherkomponente. In Abbildung 4 sind diese Komponenten in einer Architekturskizze dargestellt. Zwischen den Komponenten, die aufeinander folgende Aufgaben erledigen, sind Warteschlangen (sogenannte Queues) eingebaut. Diese ermöglichen eine halb-parallele Ausführung der Komponenten. Halbparallel deshalb, da für ein einzelnes Dokument die Komponenten ausschließlich sequentiell arbeiten können. Erst bei größeren Mengen mit unterschiedlich komplexen Dokumenten ergeben sich Zeiteinsparungen durch Nutzung der Queues. Die *Sammelkomponente* beinhaltet den Start eines sogenannten »robot«. Der »robot« bekommt eine oder mehrere URL-Adressen (Links) und sammelt die dahinter stehenden Dokumente ein. Die anschließende Dokumentvorverarbeitung extrahiert neue URL-Adressen und gibt diese dem »robot«. Die Links werden als Strukturdaten neben Metadaten wie Dokumenttyp und Einsammelzeitpunkt für die weitere Verarbeitung gespeichert.

Die *Analysekomponente* hat nun die Aufgabe, durch spezielle, wissensbasier-

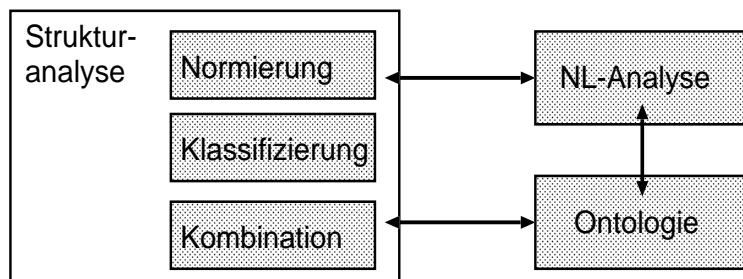


Abb. 5: Strukturanalyse-Architektur

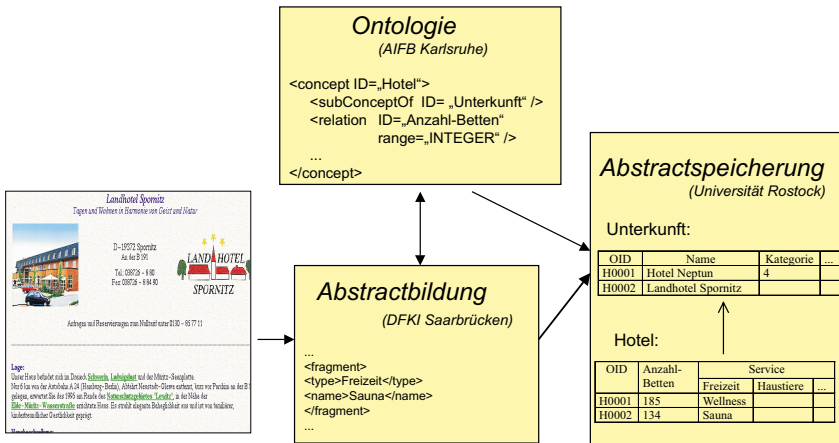


Abb. 6: Interaktion von Ontologie mit Datenbank und Sprachverarbeitung

te Analyse-Tools die Dokumente zu verarbeiten. Als Eingabe werden hierbei neben den Dokumenten selbst die gesammelten Strukturdaten und Metadaten übergeben. Das Ergebnis dieses Arbeitsschrittes sind *Abstracts*, die pro Dokument oder Dokumentengruppe erstellt werden können. Die Analysekomponente wird später noch genauer vorgestellt.

Der letzte Schritt im Gathering-Prozess ist das Speichern der Ergebnisse in einer objektrelationalen Datenbank. Die Eingabe besteht aus den Ergebnis-*Abstracts* und Metadaten zum ursprünglichen Dokument. Die Datenbankkomponente speichert dann indizierte, strukturierte Daten, die für komplexe Suchmaschinen-Anfragen sehr gut geeignet sind. Die Architektur der Datenbank und die Funktionsweise des Speicherns und Anfragens der Daten ist in Abschnitt 6 zu finden.

5.2 Analysekomponente

Die Analysekomponente besteht aus einer Strukturanalyse, einer linguistischen Analyse und einer semantischen Analyse. Obwohl sich eine Verzahnung der einzelnen Analyseschritte in der Praxis als sinnvoll herausgestellt hat, werden wir in dem Artikel diese drei Schritte einzeln betrachten. Die Strukturanalyse stellt dabei den ersten Schritt in der Analysekette dar und beinhaltet eine Normalisierung, Klassifikation und Kombination der Dokumente.

Abbildung 5 zeigt die Strukturanalyse mit den drei Komponenten und zeigt

die wichtigen Querverbindungen innerhalb des Analysevorgangs.

Bei der Normierung wird das Dokument Dokumenttyp-abhängig normalisiert. Dabei werden strukturell unwesentliche Elemente entfernt (Beispiel reine Layoutelemente) und semantisch ähnliche Elemente zusammengefaßt und vereinheitlicht.

Elemente, die zu viel Inhalt repräsentieren, wie beispielsweise das HTML-Tag `<META content='keyword'>`, die oftmals allgemeine, spezielle und für die Analyse verfälschende Daten (Häufig wird im Internet versucht, durch falsche Daten das Ranking in Suchmaschinen zu beeinflussen) gemischt enthalten, werden ebenfalls entfernt.

Bei der Klassifizierung wird zuerst mit Hilfe eines in GETESS entwickelten Sprachanalyseverfahrens ([Düsterhöft, Gröticke 2000]) die Sprache Deutsch oder Englisch erkannt. Die zweite Klassifizierung beruht auf charakteristischen Dokumentteilen (Bsp. Titel), die Auskunft über eine erste Einordnung gibt und damit zu speziellen Methoden in der weiteren Analyse führt. Der Schritt der Klassifizierung ist nur in Kombination mit der linguistischen Analyse sinnvoll.

Bei der Kombinationskomponente geht es darum, strukturelle Aspekte auszunutzen, um bestimmte inhaltliche Zusammenhänge im Dokument zu erkennen. Die Regeln dafür werden durch Strukturheuristiken definiert. Diese Heuristiken bilden zusammen mit den linguistischen Heuristiken die Entscheidungsgrundlage für die semantische Analyse.

Die linguistische Analyse liefert linguistische Daten zu einzelnen Wörtern oder Wortgruppen. Die zugrundeliegenden Tools wurden in Abschnitt 4 dargestellt. Die semantische Analyse beruht auf dem vordefinierten Wissen und linguistisch oder strukturell getriebenen Heuristiken. Das benötigte Wissen ist zum einen die Ontologie mit ihren Konzepten und Relationen und zum anderen ein Domänenlexikon als Abbildung zwischen linguistischen Termen und den Konzepten. Bei der semantischen Analyse werden nun mittels Domänenlexikon die Terme eines Dokuments, die zu einem Konzept zuordenbar sind, herausgefiltert und aufgrund der Heuristiken für eine Relationierung vorgeschlagen. Die Relationierung ist dann der Vorgang, bei dem in der Ontologie nach einer semantischen, möglichen Verbindung zwischen zwei Termen gesucht wird. Das Ziel dieses Vorgangs ist eine Menge von solch relationierten Konzept-Wert-Paaren, aus denen letztendlich die *Abstracts* aufgebaut sind.

Probleme bestehender Suchmaschinenansätze, wie Skalierbarkeit und Performance, konnten auch in GETESS nicht prinzipiell gelöst werden. Aufgrund des Performance-Problems wurde eine halb-parallele Verarbeitung entwickelt, diese ist besonders notwendig, da die eingesetzten komplexen Analyseverfahren sehr zeitaufwendig sind.

6 Datenbanklösung

Nachdem jetzt der Einsatz von Ontologien im Projekt GETESS und die natürlichsprachige Analyse von Dokumenten und Benutzeranfragen, sowie das Zusammenwirken zwischen den computerlinguistischen Komponenten und der Wissensrepräsentation dargestellt wurde, folgt in diesem Abschnitt die Speicherung der analysierten Informationen.

6.1 Motivation des Datenbank-Einsatzes

GETESS ist zur Verarbeitung und Suche auf großen Dokumentmengen konzipiert worden. Das System kann als Suchmaschine oder Dokumentenserver eingesetzt werden. In beiden Anwendungsfäl-

len fallen große Mengen von Daten an, die eine indexähnliche Funktion haben.

6.2 Zusammenwirken der Datenbanken mit anderen Projektanteilen

Die Indexdaten werden durch die natürlichsprachige Analyse, die in Abschnitt 4 ausführlich dargestellt wurde, generiert. Dabei werden die im Original-Dokument vorkommenden Begriffe und die zugehörigen Ontologie-Einträge in Zusammenhang gesetzt. Die auf diese Weise ermittelten Inhalte eines HTML-Dokumentes werden als *Abstracts* bezeichnet.

Durch die zwei Ebenen von Informationen in den Abstracts (Begriffe des Dokumentes und Ontologie-Konzepte) lassen sich keine invertierten Listen als Indexstruktur einsetzen, es muß eine andere Form der Speicherung gefunden werden. Es soll für die Anwendung sichergestellt werden, dass komfortable Anfragen auf den Abstracts durchgeführt werden können, die über normale Suchfunktionen hinausgehen (zum Beispiel typabhängige Vergleiche, Aggregatfunktionen, Sortierung, u.v.m.). Weiterhin soll eine sichere und effiziente Speicherung erfolgen. Aufgrund dieser Anforderungen werden in GETESS Datenbanken zur Speicherung der Informationen eingesetzt.

Dabei werden die Datenbankentwürfe in Anlehnung an die Ontologie erstellt. Die Abstracts selbst werden in den auf diese Weise erstellten Datenbanken gespeichert.

Abbildung 6 zeigt das Zusammenwirken der Komponenten Ontologie, Gathering, Abstract-Generierung und Abstract-Speicherung bei der Indexbildung in der Suchmaschine.

6.3 Möglichkeiten zur Speicherung der Abstract-Informationen

Bei den auf diese Weise entstehenden Datenbanken treten typische Merkmale semistrukturierter Daten (nach [Abiteboul 1997], [Bunemann 1997]) auf - es entstehen Datenbanken mit einem sehr großen Datenbankschema, die Datenbanken sind jedoch oft nur schwach gefüllt, enthalten also viele Nullwerte.

Diese Eigenschaft ist durch die Anwendung begründet. Die Inhalte der Abstracts werden aus Webdokumenten extrahiert. Die Webdokumente sehen sehr unterschiedlich aus, das widerspiegeln auch die Abstracts, die ebenfalls semistrukturierte Eigenschaften aufweisen. Speichert man die Abstracts in Datenbanken, so weisen die so resultierenden Datenbanken sehr viele Nullwerte auf.

Abbildung 7 veranschaulicht die Verteilung von Tupeln in Abstract-Datenbanken für eine Testanwendung im Finanzbereich.

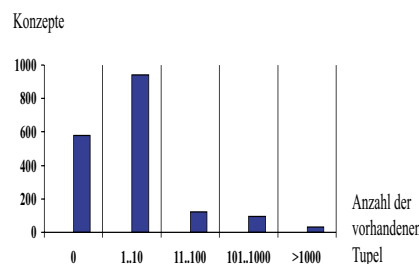


Abb. 7: Verteilung von auftretenden Konzepten für eine Finanzanwendung mit 1257 Konzepten

Aus diesem Aspekt resultierend wurden im Projekt GETESS zwei Möglichkeiten entwickelt, die Abstract-Daten in einer Datenbank zu speichern. Zum einen können objektrelationale Datenbanken eingesetzt werden, die Struktur dieser Datenbanken richtet sich dabei nach der in der Ontologie abgebildeten Struktur. Hierarchien der Ontologie werden in adäquater Weise auf die Datenbank abgebildet. Wie das aussieht, zeigt das folgende Beispiel für ein Konzept *Adresse*.

ID	Ort	PLZ	Straße
0001	Warnemünde	18119	Seestraße
0002	Warnemünde	18119	Alexandri- nenstraße

Nummer	Telefonvorwahl	Telefonnummer
12	0381	54340
115	0381	548210

Zum anderen können Konzepte der Ontologie zusammengefaßt werden und in einem XML-Attribut der Datenbank darge-

stellt werden. Dieser Fall wird angewendet, wenn Konzepte in den Abstracts nur sehr selten auftreten, also wenn diese Konzepte sehr speziell sind. Ein Beispiel dafür ist das folgende, das spezielle Merkmale eines Hotels zusammenfasst.

```
<DETAILS>
  <Ausstattung>
    <Schwimmbad>
      Meerwasserschwimmbad
    </Schwimmbad>
  </Ausstattung>
  <Ausstattung>
    <Gastronomie>
      Restaurant „Aquamarin“
    </Gastronomie>
  </Ausstattung>
  <Service>
    Original-Thalassozentrum
  </Service>
</DETAILS>
```

6.4 Optimierung der Datenbank-Entwürfe

Es ist nun möglich, beim Entwurf zu entscheiden, welches Ontologiekonzept auf welche Weise abgebildet werden kann. Wenn ein Datenbank-Entwerfer diese Entscheidung trifft, dann ist sie natürlich subjektiv gefärbt, deshalb wurde in [Klette, Meyer 2000] eine Methode entwickelt, mit der solche Entwurfsentscheidungen durch Auswertung von Statistiken über Daten und Anfragen getroffen werden können. Diese Methode wurde in GETESS implementiert und angewendet.

Damit ergibt sich das Szenario aus Abbildung 8, in deren Ergebnis optimierte Datenbank-Entwürfe aus den Ontologien erzeugt werden.

In der ersten Phase ist zu sehen, dass aus einer Ontologie ein Datenbank-Entwurf generiert wird. Diese Datenbank wird mit Abstract-Daten gefüllt. Die Aktion wird vom Gatherer initiiert, durch die natürlichsprachige Analyse werden dabei die Abstracts gebildet. Dieser Schritt kann in Anlehnung an Methoden der Künstlichen Intelligenz als *Trainingsphase* bezeichnet werden.

Im zweiten Schritt erfolgt eine *Optimierung des Datenbank-Entwurfes* aufgrund von Statistiken über den Daten. Aus diesen Statistiken wird die Entschei-

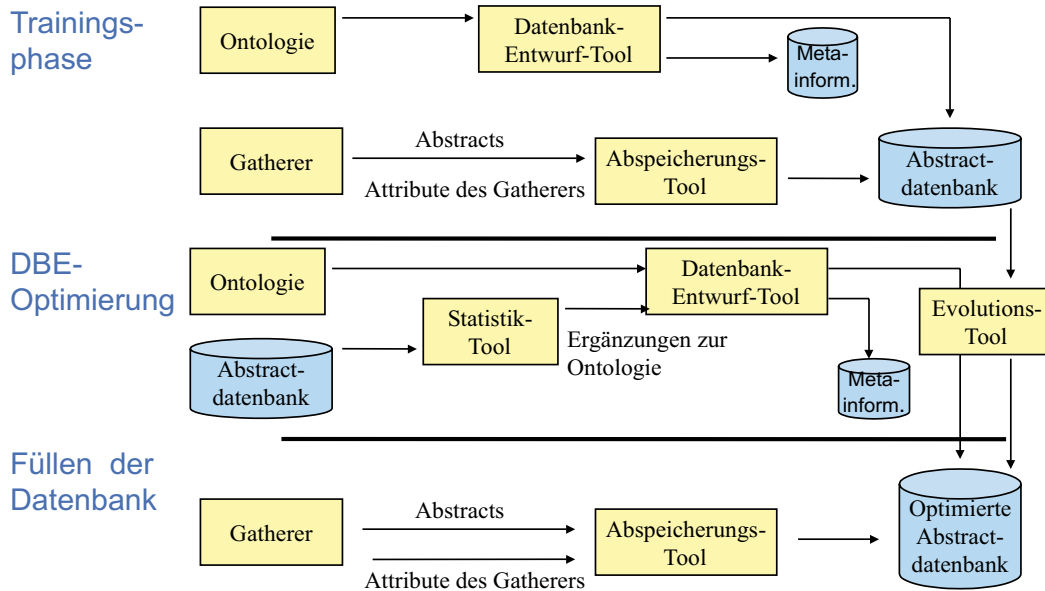


Abb. 8: Datenbankoptimierung

derung abgeleitet, welche Konzepte der Ontologie strukturiert in der Datenbank abgebildet werden und welche Konzepte zu XML-Attributen zusammengefaßt werden. Dazu wird ein Schwellwert festgelegt, der den Grad der Kompression des Datenbank-Schemas bestimmt. Alle Relationen der Datenbank, die weniger Tupel enthalten als der Schwellwert angibt, werden anschließend zu XML-Attributen zusammengefasst.

In der Beispielanwendung aus Abbildung 7 entstehen dabei bei Annahme eines Schwellwertes von 20 Tupeln 234 direkt abgebildete Relationen, die anderen 1023 Relationen werden in der optimierten Datenbank in XML-Attributen zusammengefaßt. Wählt man einen anderen Schwellwert, so läßt sich der Grad der Kompression verändern: (Schwellwert 10 Tupel: 317 Konzepte als Relationen, 940 als XML-Attribute; Schwellwert 100 Tupel: 125 Konzepte als Relationen, 1132 als XML-Attribute). In den auf diese Weise optimierten Datenbank-Entwurf werden die bereits eingesammelten und gespeicherten Daten aus der Trainingsphase übernommen.

Auf diese Weise können aus den Informationen, die in der Ontologie spezifiziert werden, die Datenbanken zur Abstract-Speicherung erstellt werden. Die Abstract-Datenbanken speichern die

durch die natürlichsprachige Analyse bereitgestellten Informationen.

Der Datenbank-Entwurf wird aufgrund einer Trainingsmenge von Abstracts optimiert und kann damit auf die konkreten Gegebenheiten angepaßt werden.

6.5 Anfragen an die Abstract-Datenbanken

Die Abstract-Datenbanken stellen die Verbindung zwischen den beiden Hauptprozessen der Suchmaschine oder des Dokumentenservers dar. Die während der Phase des Einsammelns und Aufbereitens der Informationen erzeugten Abstract-Informationen müssen für den Dialog mit dem Anwender angefragt werden.

Zur Beantwortung von Nutzeranfragen aufgrund der in Datenbanken gespeicherten Abstracts wurde im Projekt mit der Entwicklung einer Anfragesprache (IRQL) begonnen, die die Eigenschaften von Anfragesprachen für strukturierte Daten wie Attributierung, Ausnutzung von Typinformationen (Verwendung typspezifischer Operatoren, Verknüpfung sowie Restrukturierung) und Eigenschaften von Anfragesprachen für semistrukturierte Daten (vage Anfragen, inhaltsbasierte Anfragen, Ranking der Ergebnisse sowie Relevance Feedback) integriert.

Unser Ansatz sieht dabei vor, die Datenbankansprache SQL99 als Ausgangspunkt zu benutzen, zusätzliche Klauseln zur Unterstützung von Information-Retrieval-Suchaufträgen zu integrieren und die strenge Typisierung der Sprache aufzuheben, um auch Anfragen an semistrukturierte Daten realisieren zu können.

Zur Ausführung von IRQL-Anfragen werden die Möglichkeiten der Anfragesprachen existierender Datenbankmanagementsysteme ausgenutzt. Außerdem soll eine Unabhängigkeit vom im konkreten Fall verwendeten System zur Speicherung der Abstracts erreicht werden.

Ein weiterer Aspekt der Anfrageverarbeitung in GETESS ist die spezifische Ranking-Funktion, die in die IRQL integriert wurde. Die Ontologie leistet auch an dieser Stelle einen Beitrag. Sie kann dazu benutzt werden, aufgrund ihrer hierarchischen Struktur (Konzept-Baum) die Ähnlichkeit zweier Konzepte zu bestimmen. Zusammen mit klassischen Information-Retrieval-Techniken führt dieser Ansatz zu einer vielversprechenden Kombination.

7 Prototyp

Nach Vorstellung der Interna des entwi-

ckelten Prototypen ergibt sich die Frage, wie das Ganze für den Benutzer aussieht.

Der Benutzer stellt seine Anfragen in natürlicher Sprache an eine Dialogkomponente, die diese Anfrage mit der in Abschnitt 4 vorgestellten natürlichsprachigen Analyse verarbeitet, Verknüpfungen zu Konzepten der Ontologie herstellt und daraus ein Frame berechnet, der den vom GETESS-System erkannten Teil der Benutzeranfrage repräsentiert. Aus diesem Frame werden dann die Anfragen an die Abstract-Datenbank generiert. Die Idee der Dialogkomponente ist es, dem Benutzer ein Werkzeug in die Hand zu geben, seine Suchanfrage zu präzisieren und so den Suchraum möglichst einzuschränken. Die dabei verwendete Dialogstrategie kennt dabei unterschiedliche Szenarien. Einmal wird die Anzahl der ermittelten Ergebnisse der Anfrage betrachtet, andererseits auch die Vollständigkeit der

durch den Benutzer gestellten natürlichsprachigen Anfrage. In einem Dialog mit dem Benutzer werden die durch die natürlichsprachige Analyse nicht erkannte Teile der Anfrage nachgefragt beziehungsweise detailliertere Informationen erfragt. So kann zum Beispiel bei der Suche nach »Hotels an der Ostseeküste«, die sehr viele Ergebnisse liefert, der Benutzer nach bestimmten gewünschten Ausstattungen, Zimmerarten, genaueren Adressinformationen, usw. gefragt werden. Dieses Wissen kommt aus der Ontologie, die Reihenfolge in der zusätzliche Informationen erfragt werden, wird aus den Datenbankstatistiken abgeleitet.

Ein Ausschnitt aus einem solchen Beispieldialog ist in Abbildung 9 dargestellt.

8 Zusammenfassung

Mit dem BMBF-Projekt GETESS wurde gezeigt, dass Techniken aus der Computerlinguistik, der Wissensrepräsentation und dem Datenbankbereich erfolgreich in Suchmaschinen oder Dokumentenserver integriert werden können. Die verschiedenen Technologien spielen dabei eng zusammen und ergänzen sich.

Im Projekt wurde ein vollständiger Prototyp entwickelt, der die in diesem Artikel beschriebenen Funktionalitäten enthält. Die Ontologien, Wörterbücher und linguistischen Analyseverfahren wurden dabei für zwei Anwendungsgebiete (Tourismus, Finanzinformationen) erstellt.

Literatur

- [Abiteboul 1997] Serge Abiteboul, Querying Semi-Structured Data. In 6th International Conference on Database Theory - ICDT 1997, Lecture Notes in Computer Science, 1186, Springer Verlag.
- [Böhm 1997] Klemens Böhm: Verwendung objektorientierter Datenbanktechnologie zur Verwaltung strukturierter Dokumente. Dissertation, Technische Hochschule Darmstadt, 1997.
- [Bruder et.a. 2000] Ilvio Bruder, Antje Düsterhöft, Markus Becker, Jochen Bedersdorfer, Günter Neumann: GETESS: Constructing a Linguistic Search Index for an Internet Search Engine. In LNCS: Natural Language Processing and Information Systems, NLDB 2000, Springer Verlag 2000.
- [Buneman, 1997] Peter Buneman: Semistructured Data. In Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, ACM Press, 1997.
- [Düsterhöft, Grötcke 2000] Antje Düsterhöft, Sherry Grötcke: A Heuristic Approach for Recognizing a Document's Language Used for the Internet Search Engine GETESS. In Proceedings of the 2nd International Workshop on Natural Language and Information Systems, 2000.
- [Heuer, Priebe 2000] Andreas Heuer, Denny Priebe: Integrating a Query Language for Structured and Semi-Structured Data and IR Techniques. In Proceedings of the 11th International Workshop on Database and Expert Systems Applications DEXA 2000, IEEE Computer Society Press.
- [Guarino 1998] N. Guarino: Formal Ontology and Information Systems. In: N. Guarino (ed.), Formal Ontology in Information Systems, Proc. of the 1st International Conference of Formal Ontology in Information Systems, Trento, Italy, 6-8 June 1998. IOS Press.
- [Klettke, Meyer 2000] Meike Klettke, Holger

The screenshot shows the GETESS user interface. At the top, there is a 'Dialog-History' section with a table:

No.	Ihre Anfrage	was wir daraus erkannt haben	Anz. Ergebnisse
1	Ich suche eine Unterkunft in Rostock	→ Unterkunft in Gebiet Hansestadt Name (rostock)	18

Below the history is the 'Dialog-Strategy' section, which lists various types of accommodations like 'Bauernhof', 'Campingplatz', etc., with an 'ignorieren' button.

The main part of the screenshot is a search result page for 'Tagungshotel' in Rostock. It features a 'Start' button and links for 'Interaktive Karte', 'Kühlungsborn', 'Suche', and 'English'. The results are organized into three columns:

- Hotel Neptun** (D-18119 Rostock-Warnemünde, Seestraße 19): Im Tagungszentrum stehen Ihnen variable Salons und Tagungsräume für bis zu 600 Personen zur Verfügung.
- Landhotel Wittenbeck** (D-18209 Wittenbeck, Strasse zur Kühlung 21a): Zwischen Ostseestrand und dem hügeligen und waldreichen Naturschutzgebiet Kühlung liegt das Landhotel in ruhiger, naturnaher und doch zentraler Lage an der Bäderstraße zum Ostseebad Kühlungsborn.

At the bottom, there is a 'MANET' logo and copyright information for September 2000.

Abb. 9: Dialogführung in GETESS

- Meyer: XML and Object-Relational Databases - Enhancing - Structural Mappings Based on Statistics. In WebDB 2000.
- [Maedche, Staab 2001] Alexander Maedche, Steffen Staab: Ontology Learning for the Semantic Web. In IEEE Intelligent Systems, 16(2), March/April, 2001.
- [Neumann et al. 1997] G. Neumann, R. Backofen, J. Baur, M. Becker und C. Braun: An Information Extraction Core System for Real World German Text Processing. In Proceedings der 5th International Conference of Applied Natural Language (ANLP), Washington, USA, 1997.
- [Neumann, 1997] G. Neumann: Applying Explanation-based Learning to Control and Speeding-up Natural Language Generation. In Proceedings des 34th Annual Meeting of the Association for Computational Linguistics (ACL), Madrid, Spain, 1997.
- [Neumann et al. 2000] G. Neumann, C. Braun und J. Piskorski: A Divide-and-Conquer Strategy for Shallow Parsing of German Free Texts. In Proceedings der 6th International Conference of Applied Natural Language (ANLP), Seattle, USA, 2000.
- [Ogden, Richards 1923] C.K. Ogden, I.A. Richards: The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism. Routledge & Kegan Paul Ltd., London, 10 edition, 1923.

Meike Klettke, Mathias Bietz, Ilvio Bruder,
Andreas Heuer, Denny Priebe
Universität Rostock
Fachbereich Informatik,
Lehrstuhl Datenbanken und
Informationssysteme
Albert-Einstein-Straße 21
1057 Rostock
meike@informatik.uni-rostock.de

Günter Neumann, Markus Becker,
Jochen Bedersdorfer, Hans Uszkoreit
DFKI Saarbrücken
Forschungsbereich Sprachtechnologie
Stuhlsatzenhausweg 3
66123 Saarbrücken
neumann@dfki.de

Alexander Maedche, Steffen Staab, Rudi Studer
Universität Karlsruhe
Institut für Angewandte Informatik und Formale
Beschreibungsverfahren (AIFB)
Universität Karlsruhe
76128 Karlsruhe
sst@aifb.uni-karlsruhe.de