

**Universität Ulm
Fakultät für Informatik**

Proseminar SS2004

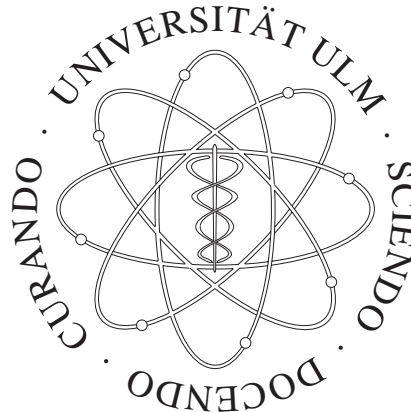
**ExtrAns: Verarbeitung natürlicher, schriftlicher
Sprache**

Christian Bohnacker

Betreuer

**Prof. F. v. Henke
Dr. Holger Pfeifer**

Abteilung Künstliche Intelligenz



ExtrAns ist ein experimentelles Computerprogramm, das den Umgang mit technischen Dokumenten erleichtern soll. Anwender haben die Möglichkeit, Fragen in natürlicher Sprache, anstatt in irgendeiner formalen Abfragesprache, zu technischen Dokumenten zu stellen. ExtrAns sucht nach Antworten in den technischen Dokumenten. Das Antwort-Extraktions-Verfahren von ExtrAns wurde an der Universität Zürich entwickelt.

1. Einleitung

Für die meisten Firmen und Organisationen sind technische Dokumente wichtige Wissensquellen, weil sie das Know-how und die Erfahrung der Fachleute in bestimmten Gebieten kombinieren. Um dem optimalen Gebrauch von diesen Dokumenten in den spezifischen Problemsituationen zu garantieren, müssen Leute in der Lage sein, exakte und zuverlässige Informationen schnell zu finden. Antwort-Extraktion ist eine neue Computertechnologie, die Benutzern hilft, exakte Antworten zu ihren Fragen in den technischen Dokumenten zu finden.

Diese Arbeit beschreibt das ExtrAns („EXTRActing ANSwers“) System.

ExtrAns arbeitet als Eingabe für die Wissensbasis mit einer Ansammlung von technischen Dokumenten. Jeder Satz der Eingabe wird einer vollen syntaktischen Analyse unterzogen und die resultierende Syntaxstruktur in eine logische Darstellung (eine so genannte logische Form) der Bedeutung des Satzes umgewandelt. Dann wird die logische Form, zusammen mit einem Zeiger zum ursprünglichen Satz, der Wissensbasis hinzugefügt.

Wenn ein Benutzer eine Frage stellt, leitet ExtrAns davon, genau in der gleichen Weise wie zur Erstellung der Wissensbasis, eine logische Form ab und versucht dann, alle logischen Formen in der Wissensbasis zu finden, die zu der logischen Form der Frage passen.

Die Ausgabe des Systems für den Benutzer ist eine geordnete Liste von Sätzen aus der Wissensbasis, die diese logische Form enthalten.

Auf diese Weise erhält also die Antwort-Extraktion (AE) Textteile auf der Grundlage ihrer Bedeutung anstatt auf der Grundlage ihrer oberflächlichen Form.

2. ExtrAns Komponenten

Die Erkennung von natürlicher Sprache ist eine komplexe Thematik. Eine der größten Schwierigkeiten der Erkennung von natürlicher Sprache ist es, dass Sätze verstanden werden müssen. Es sind nicht nur einzelne Wörter für sich wichtig, da z. B. ein Verneinungswort die Aussage eines Satzes umkehren kann. Die Analyse der Sätze wird von bestimmten Modulen durchgeführt. Für jede Aufgabe bei der Sprachverarbeitung gibt es ein so genanntes linguistisches Modul.

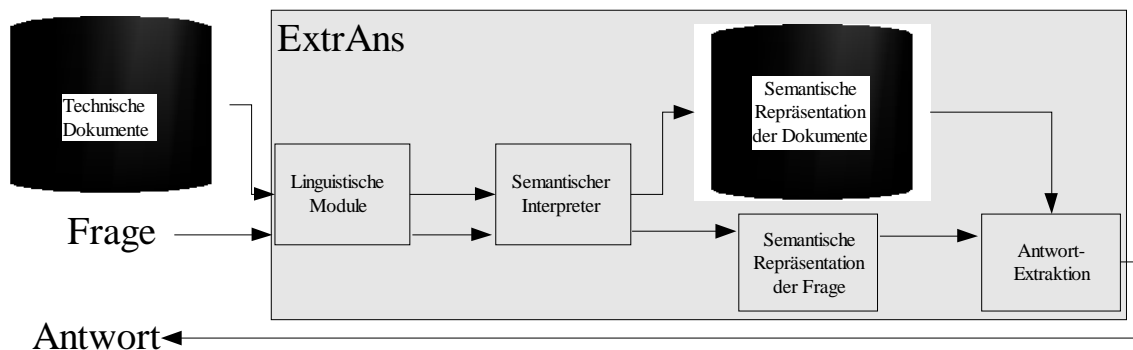


Abbildung 1: Schematische Darstellung des ExtrAns Systems.

Zu Abbildung 1: Als Eingabe für die zu erstellende Wissensbasis benötigt ExtrAns technische Dokumente. Diese werden von einer Vielzahl von linguistischen Modulen analysiert und von dem semantischen Interpreter Satz für Satz in logische Formen (siehe Kapitel 2) überführt. So entsteht die semantische Repräsentation der Dokumente, die der Bedeutung der einzelnen Sätze der Quelltexte entsprechen.

Mit den gleichen linguistischen Modulen werden auch die Fragen vom Benutzer an das System bearbeitet und schließlich in eine logische Form überführt.

Die Antwort-Extraktion besteht nun darin, zu der Frage passende logische Formen in der Sammlung der logischen Formen der Wissensbasis zu finden. Die Antworten sind die, zu den gefundenen logischen Formen, jeweils passenden Quellsätze. Diese werden nach Relevanz sortiert ausgegeben.

2.1. Linguistische Module

Aus verschiedenen Tests ergab sich, dass für eine erfolgreiche Antwort-Extraktion normalerweise nur Nomen, Verben, Adjektive, Adverbien, Präpositionen und die Beziehung dieser Wörter zueinander relevant sind. So ist es also für ExtrAns wichtig diese erkennen zu können und unwichtige Wörter wie zum Beispiel Artikel herausfiltern zu können.

Um grammatikalische Informationen über die Sätze zu bekommen, wurde in Versuchen mit Link Grammar (<http://www.link.cs.cmu.edu/link/>) gearbeitet. Link Grammar ist ein frei verfügbarer Parser mit breiter Abdeckung der englischen Grammatik. Durch dieses Programm war es ExtrAns möglich, unnötige Informationen in der Wissensbasis oder in den Fragen der Benutzer zu erkennen und vor der weiteren Verarbeitung herauszufiltern.

Einen Einblick und mehr Informationen zu diesem Programm bekommen man bei

<http://www.link.cs.cmu.edu/link/submit-sentence-4.html>

Nachdem nur noch die relevanten Wörter behandelt werden, bearbeitet ExtrAns noch folgende linguistische Eigenschaften der Eingaben:

- Synonymie (inhaltliche Übereinstimmung von verschiedenen Wörtern oder Konstruktionen), und Lemmatisation (mit Stichwörtern versehen und ordnen), siehe dazu WorldNet im Kapitel 2.2. Folgerungen.
- Disambiguierung (Auflösung von Mehrdeutigkeiten)
- Anapherauflösung (Wiederholung eines Wortes oder mehrerer Wörter zu Beginn aufeinander folgender Sätze oder Satzteile). Hierbei kann man sich zum Beispiel Folgendes vorstellen.

Quellsatz:

„A beep sounds if the static inverter is activated. It sounds again if the static inverter is deactivated.“

Für die Wissensbasis wird nun „It“ im zweiten Satz durch „A beep“ ersetzt.

Nach allen Modifizierungen der Quellsätze durch die linguistischen Module wird von dem semantischen Interpreter von ExtrAns ein sinngemäßes, logisches Abbild (logische Form) generiert und in der Wissensbasis gespeichert. Diese linguistischen Module tragen somit zu großen Teilen dazu bei, dass die Antwort-Extraktion exakt und präzise funktioniert. Ein weiterer wichtiger Aspekt sind Folgerungen.

2.2. Folgerungen

Das Extrahieren von Antworten erfordert auch manchmal, dass ein System Folgerungen oder Spekulationen macht. Im Moment macht ExtrAns nur wenige Spekulationen. Eine dieser Spekulationen ist über die Verteilung des Zusammenhangs (distributivity of conjunction) in Sätzen. Zum Beispiel wird zu dem Satz:

“The static inverter is activated and a beep sounds”

von ExtrAns zwei Informationen entsprechend dem Inhalt:

“The static inverter is activated”

“a beep sounds”

zu der logischen Form dieses Satzes in der Wissensbasis gespeichert. Es ist also möglich, auf Fragen zu den beiden Teilsätzen, diesen Satz als Antwort zu finden. Denn bei Fragen ist es oftmals der Fall, dass zum Beispiel von einem Techniker etwa wahrgenommen wird:

“a beep sounds” oder “The static inverter is activated” nicht aber unbedingt beide Informationen.

Der Zusammenhang dieser beiden Fakten ist dem System aber bekannt und es gibt dem Techniker den Satz aus. Der Techniker kann dann überprüfen, ob die Information relevant ist, indem er kontrolliert, ob die beiden Informationen zutreffen.

Eine andere besonders nützliche Art der Folgerungen ist Synonymie. Zum Beispiel kann auf der Basis von WorldNet (<http://www.cogsci.princeton.edu/~wn/>) eine Auflistung sinnverwandter Wörter (Thesaurus) erstellt und gespeichert werden. Jede Gruppe von sinnverwandten Wörtern (bei

ExtrAns als „synsets“ bezeichnet) wird dann mit einem eindeutigen String („synset identifier“) identifiziert. Dieses linguistische Modul ersetzt schließlich alle im Thesaurus definierten Wörter in der abschließenden, logischen Form des Fragesatzes mit ihren synset Bezeichnungen.

Probleme können mit mehrdeutigen Wörtern auftreten, die nicht eindeutig einem synset zugeordnet werden können. Um das System zu vereinfachen, kann man bei Mehrdeutigkeit in folgendermaßen vorgehen: Wenn ein Wort zu zwei oder mehr synsets gehört, wählt ein Algorithmus nach dem Zufallsprinzip ein synset für dieses Wort aus. Da Wörter innerhalb spezieller technischer Dokumente nur begrenzt mehrdeutig sind, sind deswegen kaum Probleme mit der Wortbedeutung zu erwarten.

2.3. Anwendungen

Die erste ExtrAns Anwendung, um willkürliche Benutzerfragen zu den “Unix documentation files” (man pages) zu beantworten ist online verfügbar (<http://www.ifi.unizh.ch/CL/extrans/>). Das dortige Testsystem umfasst mehr als 500 unbearbeitete man pages. Hier können Fragen wie zum Beispiel:

“Which command copies files?”

gestellt werden. Bei dieser relativ kleinen Wissensbasis arbeitet ExtrAns gut.

Momentan wird ExtrAns unter anderem verwendet, um Fragen zu dem Flugzeugwartungs-Handbuch des Airbus A320 zu beantworten (<http://www.ifi.unizh.ch/cl/webextrans/>). Hierbei handelt es sich um etwa 120Mbytes Dokumente einer sehr technischen Domäne. Im Vergleich mit den insgesamt etwa 270Kbytes großen Manpages bietet das Flugzeugwartungs-Handbuch eine wichtige Testumgebung für die Skalierbarkeit von ExtrAns.

Anhand dieser Tests konnte man unter anderem sehen, dass ExtrAns auf zu viele Fragen keine Antwort findet, falls logische Einschränkungen bei der Suche nach Antworten nicht gelockert werden.

2.4. Antwortfindung

Im striktesten Fall muss die logische Form einer Frage genau zu einer logischen Form eines Satzes in der Wissensbasis passen, damit dieser Satz als Antwort ausgegeben wird. Falls ExtrAns keine direkte Antwort in der Wissensbasis auf eine Frage findet, kann es logische Einschränkungen bei der Suche nach Antworten Schritt für Schritt lockern, bis eine Antwort möglich ist.

Hyponyme (Wort, Lexem, das in einer untergeordneten Beziehung zu einem anderen Wort, Lexem steht, aber inhaltlich differenzierter, merkmalthaltiger ist, z. B. *essen* zu *zu sich nehmen*, *Tablette* zu *Medikament* [DU82]) werden zuerst betrachtet. Diese müssen natürlich (zum Beispiel von Hand) definiert sein.

Falls Hyponymien in der Frage gefunden werden, werden zu der logischen Form der Frage die logischen Formen der synsets der Hyponyme als Disjunktionen hinzugefügt. Technisch gesehen ist diese resultierende logische Form mit der ursprünglichen logischen Form gleichwertig, denn der Sinn bleibt im Wesentlichen gleich. So ist es aber möglich, Antwortsätze zu finden, die ohne Ausnutzen von Hyponymieeigenschaften nicht hätten finden lassen.

Falls dann noch keine Antworten gefunden werden können, sucht ExtrAns in der Wissensbasis nach Sätzen, die am wahrscheinlichsten zur Frage passen. Es wird also nach Sätzen gesucht, die möglichst gut zu der logischen Form der Frage passen. Das bedeutet, dass in einem Satz der Wissensbasis wenige Wörter, die nicht zu der logischen Form der Frage passen würden, nicht beachtet werden. Bei der Ausgabe werden dann die besten Treffer zuerst angezeigt.

Schließlich, falls auch diese Methode fehlschlägt, versucht das System “Keyword matching”. Es wird nicht mehr auf syntaktische Kriterien geachtet, sondern nur noch Informationen über Wortklassen verwendet um gesuchte Stichwörter in der Wissensbasis zu finden. Dies ähnelt einer klassischen Stichwort-Suche von Suchmaschinen, bei der mit unter noch zusätzlich nach synonymen der Stichwörter gesucht wird.

3. Logische Formen

Um den Inhalt von natürlichen Sätzen logisch verarbeiten zu können, verwendet ExtrAns logische Formen. Es ist einerseits ein Formalismus nötig, der mit problematischen Sätzen fertig wird (z. B. lange Sätze, Sätze, die von Link Grammar nicht korrekt erkannt wurden oder Sätze mit Rechtschreibfehlern), und andererseits einfach zu erstellen und zu verwenden sein sollte. Um diese Ziele zu erreichen, wurden die logischen Formen in einer flachen Notation erstellt.

3.1. Flache Notation

Für gewöhnlich enthalten logische Formen eingebettete Ausdrücke. Zum Beispiel

```
a=sqrt(exp(2));
```

Hier ist der Ausdruck `exp(2)` in dem Ausdruck `sqrt(...)` eingebettet.

Das ExtrAns Team entschied sich, eine flache Darstellung zu benutzen, um die Ableitung der logischen Formen von komplizierten Sätzen zu erleichtern und ihre schnelle Verarbeitung zu ermöglichen. Um eingebettete Ausdrücke zu vermeiden, kann man zusätzliche Argumente einführen. Es würde eine flache, logische Form von der 1. Instruktion eine Sequenz wie zum Beispiel 2. haben:

1. `A:=(factorial(25)-exp(12)*2);`
2. `factorial(X,25);`
`exp(Y,12);`
`A:=(X-Y)*2;`

Die zusätzlichen Argumente X und Y speichern die Ergebnisse von `factorial` und `exp`. Auf ähnliche Weise kann man auch den logischen Ausdruck "John ate an apple quickly" (1) flach darstellen (2):

1. $\exists a(\text{quick}(\text{eat}(j, a)) \wedge \text{apple}(a))$
2. $\exists a, e(\text{eat}(e, j, a) \wedge \text{quick}(e) \wedge \text{apple}(a))$

Es wird die neue logische Variable `e` verwendet, um auszudrücken, dass das Ereignis essen schnell ist (`quick(e)`). Dieser Vorgang, das Verdinglichung abstrakter Konzepte, wird mit Reification bezeichnet.

3.2. Reification

Reification ist eine technische Bezeichnung aus dem Gebiet der künstlichen Intelligenz für das Einführen neuer Entitäten, die sich auf abstrakte Konzepte beziehen. Der semantische Interpreter von ExtrAns verwendet Reification um flache, logische Formen zu erstellen. Es werden für die Antwort-Extraktion Objekte, Ereignisse und Eigenschaften reifiziert.

- **Objekte (objects)**
 Ein Nomen, wie zum Beispiel "contactor" (Kontaktgeber, aus dem Flugzeugwartungs-Handbuch des Airbus A320) wird logisch durch das Prädikat `object(contactor, o1, [x1])` dargestellt. `o1` ist nun das reifizierte Konzept, dass das Objekt `x1` ein `contactor` ist. Die neue Entität `o1` kann nun zum Beispiel in Konstrukten mit Adjektiven verwendet werden.
- **Ereignisse (events)**
 Ein Verb wie "installs" führt `evt(install, e1,[x1,x2])` ein. `e1` ist das Konzept, dass `x2` von `x1` installiert wird. Ereignis-Reification wird verwendet, um Ereignisse durch Adverbien und Präpositionen modifizieren zu können.

- **Eigenschaften (properties)**
Hierzu zählen Adjektive und Adverbien. Ein Adjektiv wie etwa blue führt `prop(blue,p1,[x1])` ein. Das Konzept p1 bedeutet, dass x1 blau ist.

3.3. Flache, logische Ausdrücke

Anhand einiger Beispiele kann man sehr gut sehen, wie flache, logische Ausdrücke von natürlichen Sätzen aussehen. Der Satz aus dem Flugzeugwartungs-Handbuch des Airbus A320

„The ECAM contactor is located in the left frame.“

wird folgendermaßen logisch dargestellt:

```
holds(e4),
object('ecam_contactor',o2,[x2]),
evt(locate,e4,[x4,x2]),
object(anonymous_object,o4,[x4]),
object(frame,o7,[x7]),
prop(left,p2,[x7]),
prop(in,p5,[e4,x7]).
```

Diese logische Form beschreibt folgenden Sachverhalt:

Es gibt drei Objekte: Einen `ecam_contactor` (x2), einen Rahmen (`frame`, x7) und ein anonymes Objekt (x4).

Das anonyme Objekt (o4) wurde erstellt, da aus dem Satz des Handbuches nicht hervorging, mit was der `ecam_contactor` an dem linken Rahmen angebracht ist.

Die Eigenschaft p2 beschreibt, dass der Rahmen (x7) rechts ist.

Dass der `ecam_contactor` (x2) sich in dem Rahmen (x7) befindet, wird durch das Ereignis `locate` (e4) zum Ausdruck gebracht.

Dass `holds` ein Synonym für `locate` ist, gibt `holds` (e4) an.

Ein weiterer Beispielsatz:

The static inverter is activated if the CSM/G is unavailable.

```
if(e5,p10),
prop(static,p2,[x3]),
object(inverter,o3,[x3]),
evt(activate,e5,[x5,x3]),
object(anonymous_object,o5,[x5]),
object('csm/g',o8,[x8]),
prop(unavailable,p10,[x8]).
```

Hier ist interessant, dass das Objekt `inverter` (x3) durch ein anonymes Objekt (x5) aktiviert wird (`activate` (e5)), falls die Eigenschaft `unavailable` (p10) auf das Objekt `csm/g` (x8) zutrifft.

Diese Beispiele sind absichtlich einfach gehalten, um die Notation zu veranschaulichen. Bei komplexeren Sätzen gibt es möglicherweise auch verschiedene Interpretationsmöglichkeiten und es können auch nicht alle Wörter klassifiziert werden. Bei der folgenden Interpretation sind Wörter, die von dem ExtrAns Parser nicht als Nomen, Verben, Adjektive oder Adverbien klassifiziert werden konnten, als *keyw*(..) gekennzeichnet. Unbekannte Wörter werden hier *kursiv* dargestellt:

In normal flight configuration, each IDG supplies its own distribution network via its Line Contactor (GLC).

```
holds(a1), prop(in, p1, [a1, x4]),
object(configuration, a2, [x4, x3, a5]), evt(configure, x4, [x3, a5]),
object(flight, a3, [x3]), explication(x4, x6), object(each, a4, [x6]),
prop(normal, p2, [x4]), evt(supply, e8, _), object(idg, a6, [x7]),
object(network, a7, [x12]), compound_noun(x11, x12),
object(distribution, a8, [x11, a9, a10]),
evt(distribute, x11, [a9, a10]), prop(via, p12, [x12, x16]),
object('Contactor', a11, [x6]), explication(x16, x18),
object(glc, a12, [x18]), prop(own, p10, _), keyw('Line').
```

Hier erkennt man, dass bei ExtrAns nicht versucht wird, die vollständige semantische Information zum Ausdruck zu bringen. Informationen wie modale Hilfsverben, die grammatikalische Zeit, Plural und Quantifizierer werden in der logischen Form ausgeschlossen. Die Entwickler von ExtrAns sprechen bei diesem Ansatz deswegen auch von der minimalen logischen Form.

3.4. Minimale, logische Form

Minimale logische Formen erleichtern das Finden von Antworten, da sie keine überflüssigen Informationen enthalten. Die minimale, logische Form des Frage-Satzes

“Where is the ECAM contactor located?”

würde folgendermaßen aussehen:

```
object('ecam_contactor',01,Y), evt(locate,E2,[X,Y])
```

Die Antwort:

„The ECAM contactor is located in the left frame.“

wird folgendermaßen logisch dargestellt:

```
holds(e4),
object('ecam_contactor',o2,[x2]),
evt(locate,e4,[x4,x2]),
object(anonymous_object,o4,[x4]),
object(frame,o7,[x7]),
prop(left,p2,[x7]),
prop(in,p5,[e4,x7]).
```

Die logische Form der Antwort auf diese Frage hat aber eigentlich nicht das holds Prädikat und drückt auch nicht das anonyme Objekt aus. So verwendet ExtrAns Unifikation, um die Deckung zwischen zwei logischen Formen festzustellen. Sätze, die überlappende Terme enthalten, sind gute Kandidaten für die Antwort. Außerdem müssen die Variablen kompatibel sein. Der Satz “The ECAM contactor is located in the left frame” ist eine mögliche Antwort, falls die Variable 01 mit o2 der Antwort übereinstimmen. Die Variable X stimmt mit der Konstanten x4 der Antwort, und so weiter überein.

Auf die Frage “What activates the static inverter?” wäre eine Antwort “The static inverter is activated if the CSM/G is unavailable.”. Jedoch ist die Aktivierung des statischen Inverters konditionell an eine Bedingung gebunden. Ob der CSM/G verfügbar ist oder nicht, kann ExtrAns nicht wissen. In solch einem Fall gibt ExtrAns diese Antwort an den Benutzer, so dass der Benutzer entscheiden kann, ob die Antwort passt.

3.5. Terminologie

Für die syntaktische Darstellung von Sätzen muss ExtrAns Ausdrücke wie “electrical centralized aircraft monitor” oder “electrical connector” als zusammengehörige Einheiten erkennen. In Ausdrücken wie “avionics compartment” ist beispielsweise nur das Wort “compartment” wichtig. Hier sollte dann das Wort “avionics” ignoriert werden. Auch bei Fragen an das System wie “Where is the electronic centralized aircraft monitor?” sollten Antworten gefunden werden, die Ausdrücke wie “centralized aircraft monitor” oder “ECAM” enthalten.

Um solche Fragen auch effizient beantworten zu können muss das System bestimmen können, welche Ausdrücke in Beziehung zu einander stehen. In diesem Gebiet gibt es einige Ansätze, doch kann dieser Schritt momentan nicht voll automatisch durchgeführt werden. Es muss eine World-Net ähnliche Hierarchie von Subtypen erstellt werden, dass ExtrAns bestimmen kann, welche Ausdrücke spezifische Typen von anderen Ausdrücken sind. Anhand von Statistiken können zusätzlich einige unbedeutende Wörter herausgefiltert werden. Einige Subtyp-Relationen kann ExtrAns aber auch mit einem recht einfachen Algorithmus selbstständig erkennen. Es ist einfach zu bestimmen, dass “electronic centralized aircraft monitor” ein Subtyp von “aircraft monitor” oder “First Officer seat” ein Subtyp von “seat” ist, da nur noch beschreibende Wörter an das Objekt angefügt werden.

Synonyme zu finden, ist eine etwas schwierigere Aufgabe. Man kann dafür FASTR verwenden. Informationen und der Quelltext dieses Programms können bei <http://www.limsi.fr/Individu/jacquemi/FASTR/index.html> eingesehen werden.

Mit diesen Werkzeugen, Umschreibungsregeln, den Synsets und einer morphologischen Datenbank (eine Datensammlung über die äußere Gestalt von Ausdrücken) kann ExtrAns folgende linguistische Variationen zwischen zwei Termen erkennen:

- Einfache Inversion (“generator control unit” ist eine Variante von “control unit for the generator”).
- Morphologische Variationen (“electric contactor” ist eine Variante von “electrical contactor”).
- Komplexe morphosyntaktische Variationen (“electrical generation equipment” ist eine Variante von “equipment for generating electricity”).
- Synonyme (“electrical fault” und “electrical defekt” oder “upright position” und “vertical position”).
- Kombinationen der oben aufgeführten Variationen (“functional test” und “operational check”).

Ein Fachmann in diesem technischen Gebiet muss noch unrelevante und unpassende Ausdrücke aus der Sammlung streichen.

4. Merkmale von ExtrAns

Bei der Antwort-Extraktion von ExtrAns spielen folgende Merkmale eine große Rolle:

- Die Terminologie bestimmter Fachgebiete wird berücksichtigt.
- Die Integration eines vollwertigen Parsers und eines robusten semantischen Interpreters, der auch komplexe und fehlerhafte Sätze verarbeiten kann.
- In der flachen, logischen Notation werden nur relevante Informationen gespeichert.
- Die Antworten werden übersichtlich und mit Verweisen auf den Quelltext angezeigt.

4.1. Beispiele

ExtrAns kann im Internet auf den Seiten der Universität Zürich (<http://www.ifi.unizh.ch/CL/extrans/>) an den Unix „man pages“ ausprobiert werden. Abbildung 2 zeigt eine Beispielausgabe des ExtrAns Systems in Anwendung auf das Flugzeug Wartungshandbuch des Airbus A320.

Man sieht die Ausgabe ((a) in Abbildung 2) auf die Frage “How is the distribution network supplied?”. Anhand der logischen Übereinstimmung werden die Antworten eingefärbt. Durch die Verknüpfung zum Originaltext im Wartungs-Handbuch kann direkt auf das Wartungs-Handbuch ((b) in Abbildung 2) zugegriffen werden (hier pgbk.24.22.00.00, grau unterlegt).

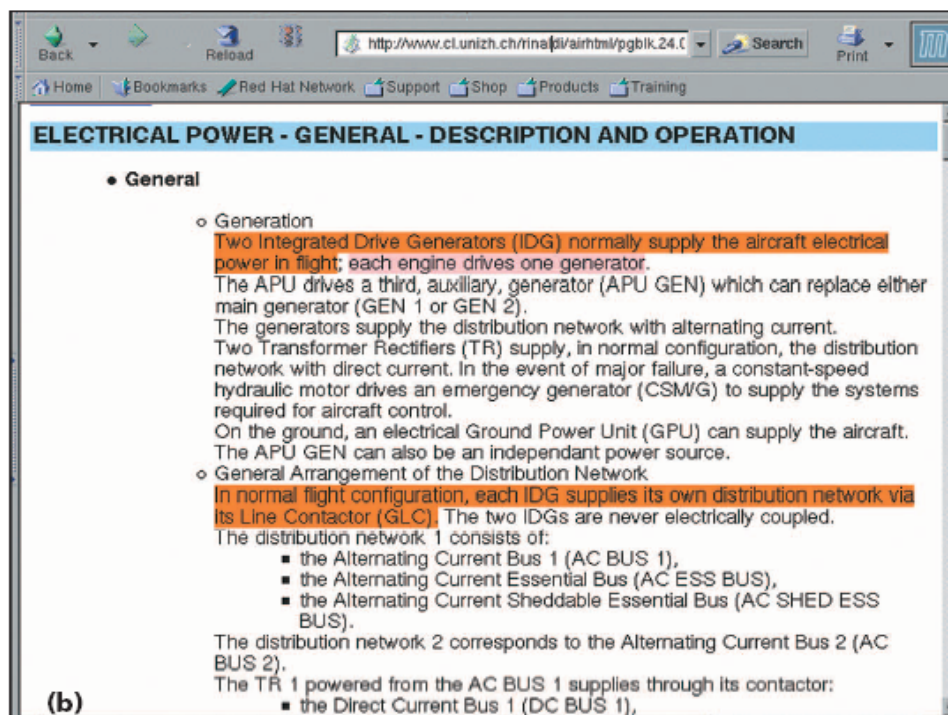
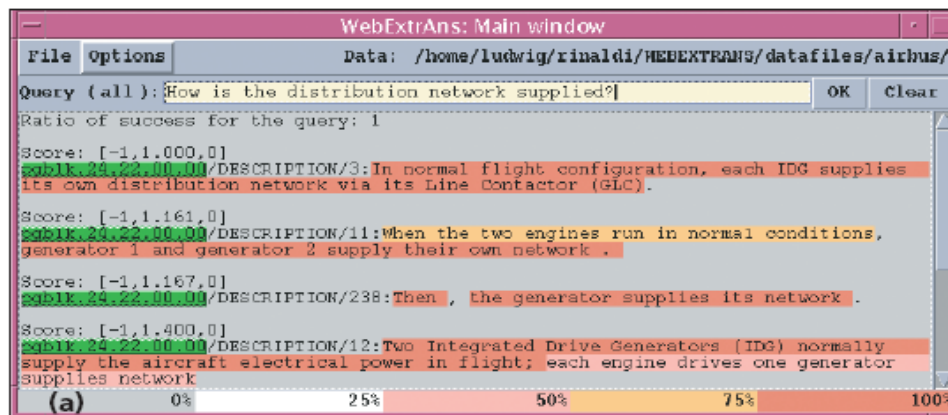


Abbildung 2: Beispielausgabe von ExtrAns.

4.2. Schlussbemerkung

Die natürliche Sprache mit logischen Formeln und intelligenten Algorithmen in den Griff zu bekommen, ist momentan noch eine Zukunftsvision. Doch ExtrAns und ähnliche Systeme sind ein großer Schritt dorthin. Das ExtrAns System verwendet für die Spracherkennung zum großen Teil externe Programme. Diese wurden hauptsächlich als Module eingebunden. Module tragen zu großen Teilen zu der Intelligenz von ExtrAns bei.

Eines der Module, Link Grammar, ist zuständig für das Erkennen von zusammengehörigen Einheiten in einem Satz. Bei einfach gebauten Sätzen funktioniert Link Grammar sehr zuverlässig. Sind die Sätze komplizierter Bauart, treten aber durchaus Fehler auf, die sich auf das System negativ auswirken.

Auch wird momentan bei ExtrAns bei Homonymen (Wort, das ebenso wie ein anderes geschrieben und gesprochen wird, aber verschiedene Bedeutung hat ... z. B. Schloss (Türschloss und Gebäude) [DU82]) nach dem Zufallsprinzip verfahren. Hier besteht noch die Aufgabe, intelligente Algorithmen zu finden. Probleme gibt es auch bei der Erkennung von Hyponymen (Wort, Lexem, das in einer untergeordneten Beziehung zu einem anderen Wort, Lexem steht, aber inhaltlich differenzierter, merkmalthaltiger ist, z. B. „essen“ zu „zu sich nehmen“, Tablette zu Medikament [DU82]) und Synonymen (vgl. 3.5. Terminologie).

Die Software ExtrAns wurde entwickelt, um in technischen Dokumenten Antworten zu finden. In technischen Gebieten, die nicht so sehr von alltäglichen, laxen Ausdrücken geprägt sind, funktioniert ExtrAns relativ gut.

Quellenangaben

- [IE03] Diego Mollá, Rolf Schwitter, Fabio Rinaldi, James Dowdall, Michael Hess; IEEE INTELLIGENT SYSTEMS, Extrans: Extracting Answers from Technical Texts, Seiten 12 ff., July/August 2003**
- [DU82] Dudenverlag Mannheim/Wien/Zürich, DUDEN Band 5 Fremdwörterbuch, 4., neu bearbeitete und erweiterte Auflage, 1982**