Website-Klassifikation und Informationsextraktion aus Informationsseiten einer Firmenwebsite

Yeong Su Lee

München 2008

Website-Klassifikation und Informationsextraktion aus Informationsseiten einer Firmenwebsite

Yeong Su Lee

Dissertation
am Centrum für Informations- und Sprachverarbeitung (CIS)
der Ludwig-Maximilians-Universität
München

Vorgelegt von

Yeong Su Lee
aus Puan, Korea

München, den 20. 12. 2007

Erstgutachter: Herr Prof. Dr. Franz Guenthner
Zweitgutachter: Herr Prof. Dr. Klaus Schulz
Tag der mündlichen Prüfung: 31. 01. 2008

Danksagung

Vor allem danke ich meinem Betreuer, Herrn Prof. Dr. Franz Guenthner, der mir die Gelegenheit gegeben hat, am CIS (Centrum für Informations- und Sprachverarbeitung) arbeiten zu dürfen. Seine motivierenden und wegweisenden Anregungen inspirierten meine Forschung. Des Weiteren hat mir meine Tätigkeit am CIS finanzielle Freiheit gegeben, so dass ich mich voll auf die Arbeit konzentrieren konnte. Ferner konnte ich das Thema mit den wunderbaren Kollegen konstruktiv diskutieren.
Außerdem bedanke ich mich bei Dr. Gerhard Rolletschek, mit dem ich zusammen arbeiten durfte. Sein Engagement hat mich dazu bewegt, dass sich meine Arbeit wesentlich verbesserte. Auch Dr. Sandra Bsiri danke ich sehr, da ich mit ihr besonders detailiert über das gesamte System diskutieren konnte. Ihr Wissen und ihre praktischen Vorschläge waren für mich sehr aufschlussreich, so dass ich die Arbeit auf den heutigen Stand bringen konnte. Mithilfe von Michaela Geierhos, Annette Gotscharek und Uli Reffle war es mir möglich, die Arbeit viel besser zu gestalten. Ihnen allen danke ich. Besonderer Dank geht an Michaela. Ihre Hilfsbereitschaft hat mich aus dem Not gerettet. Ebenfalls danke ich Sebastian Nagel, an den ich mich jederzeit wenden konnte, wenn Probleme bei meiner Arbeit auftraten.
Mein Dank gilt auch all denen, die konstruktiv zur Behandlung meines Themas beigetragen haben. Ohne sie wäre die Arbeit nicht in diesem Umfang zustande gekommen. Insbesondere bedanke ich mich bei Andreas Neumann, Magda Gerristen und Ulrich Real.
Nicht zuletzt bedanke ich mich bei meiner Familie, die mich mit viel Geduld unterstützt und stets ermutigt hat.
Vielen Dank!

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Abkürzungen und Akronyme

DFA	Deterministic Finite-State Automaton
DOM	Document Object Model
DNS	Domain Name System
IE	Information Extraction
ISO	International Organization for Standardization
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
NFA	Non Deterministic Finite-State Automaton
SGML	Standard Generalized Markup Language
SLD	Secondary Level Domain
TLD	Top Level Domain
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WWW	World Wide Web
XML	Extensible Markup Language
ccTLD	country code Top Level Domain
gTLD	generic Top Level Domain

Kapitel 1

Einleitung

So wie die Anzahl der Webseiten ständig wächst, wird auch Menge der Informationen auf den Webseiten immer größer und vielfältiger. Daraus entsteht ein enormes Interesse daran, gezielt nützliche Informationen aus Webseiten zu extrahieren.

Dabei ist vor allem problematisch, dass sich Domain-Inhaber sowie Seiteninhalte häufig ändern und täglich eine nicht unerhebliche Menge neuer Domains registriert wird[1]. Deshalb ist es generell schwierig, Informationen aktuell zu halten.

Gerade Webseiten unterscheiden sich von einfachem Text, der grammatisch ist und in den Aufgabenbereich der traditionellen natürlichen Sprachverarbeitung fällt. So wäre es zwar wünschenswert, wenn Webseiten auf XML (eXtensible Markup Language) -ähnlicher Struktur basieren würden, was aber nicht der Fall ist.

Außerdem werden Webseiten durch Hyperlinks miteinander vernetzt und sind meistens in HTML (Hyper Text Markup Language) kodiert. HTML ist in erster Linie für die Präsentation der Inhalte gedacht und nicht für die Extraktion von Informationen. Diese Charakteristik der Webseiten macht eine automatische Extraktion von Informationen schwierig.

In der Vergangenheit wurden zur automatischen Extraktion von Informationen aus Webseiten viele Wrapper entwickelt und verwendet. Viele bisher

[1] siehe Domain-Statistik bei `http://icannwiki.org/Domain_Statistics` und `http://www.denic.de/de/domains/statistiken/domainvergleich_tlds/index.html`.

entwickelte Wrapper sind oft nur für Datensätze sinnvoll, die aus einem Template erzeugt worden sind, weil sie sich an wiederholenden Strukturen orientieren oder mit Tree-Edit-Messungen arbeiten. Sie eignen sich daher nicht zum Extrahieren von Informationen aus einer Menge verschieden konzipierter Webseiten.

Daraus ergibt sich die Problematik, Informationen zu bestimmten Zwecken aus einer Menge von Webseiten automatisch zu extrahieren und zu aktualisieren.

Es können die unterschiedlichsten Informationen aus sehr vielen Bereichen verfügbar sein, angefangen bei Firmen- und Produktinformationen bis zu Verkaufsinformationen und auch schwierigen Fachbegriffen. Es gibt jedoch keine Universallösung, all diese verschiedenen Informationen extrahieren zu können. Die vorliegende Arbeit beschränkt sich daher auf die Extraktion von Firmeninformationen.

In unserer Internet-Gesellschaft nehmen die Firmenwebseiten einen bedeutenden Stellenwert ein. Das Web (World Wide Web) bietet Firmen die Möglichkeit, sich potenziellen Kunden, Bewerbern und Geschäftspartnern zu präsentieren und umfassende, immer aktuelle Informationen über die Firma, ihre Produkte bzw. Dienstleistungen bereitzustellen.

Im Gegenzug dafür greifen Verbraucher und Bewerber immer häufiger auf die Online-Informationen der Firmen zu. So stellte sich bei Bsiri (2007) [14] heraus, dass 70% der Franzosen das Web als primäre Quelle für ihre Jobsuche nutzen. Immer öfter greifen nun Kunden auf das Angebot im Web zurück.

Als Reaktion auf dieses Defizit entstanden viele Firmenverzeichnisse (z.B. Gelbe Seiten). Diese werden jedoch in der Regel manuell erstellt und gewartet. Jedoch hat die manuelle Verwaltung solcher Listen ihre Grenzen, da leicht Fehler bezüglich Aktualität und Vollständigkeit entstehen können.

Um den manuellen Aufwand möglichst gering zu halten und den Anspruch auf Aktualität und Vollständigkeit erfüllen zu können, ist die Automatisierung dieses Verfahrens unerlässlich.

In der vorliegenden Arbeit wird nun versucht, die gesuchten Firmeninformationen aus Webseiten automatisch zu extrahieren, zu vervollständigen und zu aktualisieren. Dies soll anhand der strukturellen und kontextuellen Eigenschaften der HTML-Seiten erfolgen.

1.1 World Wide Web und HTML

Die Informationsquellen im Internet werden meistens über das Web (World Wide Web) angeboten und in HTML (Hyper Text Markup Language), welches auf SGML (Standard Generalized Markup Language, ISO 8879) basiert, kodiert.

Das Web ist ein internetbasiertes Computernetzwerk, das Benutzern die Informationen von anderen über ein weltweites Netzwerk (genannt Internet) zugänglich macht. Es basiert auf dem Konzept von Hypertext, das von Ted Nelson[2] im Jahr 1965 eingeführt und von Tim Berners-Lee vom CERN im Jahr 1992 auf das Web angewandt wurde. Im Hypertext kann eine Verknüpfung (Link) auf einen Text verweisen, der nicht linear aufgebaut ist.

Das Web ist dokumentorientiert, deshalb kann jedes Dokument durch eine URI (Uniform Resource Identifier) identifiziert werden. Die Internet-Adresse (URL, Uniform Resource Locator) wird oft für sie eingesetzt, wobei ein Dokument verschiedene, beliebige Datentypen, wie Text, Hypertext, Grafik usw. enthalten kann.

Webimplementierungen erfolgen über ein Benutzer-Server-Modell. Der Benutzer benötigt ein spezielles Programm (z.B. einen Web-Browser wie Firefox, Internet Explorer usw.), um mittels HTTP (Hyper Text Transfer Protocol) durch die Daten auf den Servern navigieren zu können. Dafür senden Web-Browser die Anfragen an entfernt liegende Server, welche ihnen mit in HTML kodierten Dokumenten (Dateien) antworten. Der Web-Browser interpretiert diese nun und zeigt sie auf dem Bildschirm des Benutzers an.

HTML ist Text, der um HTML-Tags erweitert wurde. Die Tags definieren Dokument, Textformatierungen, Hyperlinks und vieles mehr. Sie sind einfach und flexibel, um das Dokument nach Belieben gestalten zu können. Jedoch ist die Verwendung nicht strikt genug und kann so leicht missbraucht werden und sehr komplizierte Implementierungen[3] erlauben.

Da HTML nicht rein textbasiert ist, kann die Extraktion von Informationen

[2] Sein Hypertext-System "Xanadu" kann über `http://xanadu.com/` heruntergeladen werden.

[3] Anders als XML (eXtensible Markup Language), welches strikt öffnende und schließende Tags und eine regelgerecht geschachtelte Struktur vorschreibt, ist HTML in dieser Hinsicht flexibel und Implementierungen können unübersichtlich werden. XHTML ist die XML-Variante von HTML.

nicht auf Grundlage herkömmlicher linguistischer Techniken erfolgen. Andererseits kann man sich nicht wie bei XML auf die Struktur verlassen.

1.2 Textart und Informationsextraktion

Informationsextraktionssysteme (IE-Systeme) werden nach Textart unterschieden, wobei die Texte nach ihrer Strukturiertheit[4] aufgeteilt werden.

Auf die natürlichen unstrukturierten Plain-Texte werden Systeme angewendet, die eine linguistische Analyse ermöglichen. Sie wenden morphologische und syntaktische Analysen auf die gegebenen Texte an. Danach werden die gesuchten Informationen extrahiert. Diese Vorgehensweise ist sehr aufwendig und manchmal überflüssig, weil die gesuchten Informationen oft anhand bestimmter einfacher Muster gefunden werden können.

Tabellen und relationale Datenbanken sind typische Beispiele für strukturierte Informationen. Für sie wird keine linguistische Analyse benötigt. Es muss lediglich die Struktur erkannt werden, um die gesuchten Informationen zu finden.

HTML-Texte können meist als semi-strukturiert bezeichnet werden: Teilweise sind sie durch HTML-Tags markiert, und teilweise sind sie natürliche Texte. Diesbezüglich stellen semi-strukturierte Texte eine große Herausforderung für IE-Systeme dar.

IE-Systeme für Webseiten beschäftigen sich mit semi-strukturierten Texten, denn sie müssen die HTML-Struktur und Textmuster erkennen. HTML-Tags sind ein wichtiger Hinweis auf die Struktur, da sehr viele Informationen im Web z.B. in Tabellen dargestellt werden. Somit sind Tags für Tabellen oft eine große Hilfe. Aber man kann sich nicht nur auf die Tags verlassen, da viele Informationen in Textform präsentiert werden, und auch wenn Daten in einer Tabelle dargestellt werden, sind sie oft nicht so strukturiert, wie man sich es wünschen würde.

[4] Strukturiertheit wird im Kapitel 5 definiert.

1.3 Vorgehensweise

Die Arbeit basiert auf folgenden Überlegungen.

1. Eine Webpräsenz besteht aus vielen Webseiten, die miteinander verlinkt sind. Einfache Crawler versuchen alle diese verlinkten Seiten zu holen und zu indizieren. So kann bei umfangreichen Webpräsenzen leicht die verfügbare Speicherkapazität und die Bandbreite des Netzes überfordert werden, was sehr zeitaufwendig ist. Da sich Webseiten ständig ändern können, muss dieser Vorgang regelmäßig wiederholt werden.

   Oft befinden sich alle gewünschten Informationen auf nur einigen Webseiten einer Webpräsenz. Wenn bekannt ist, welche Webseiten die gesuchten Informationen enthalten könnten, müssen nur noch diese Seiten geholt und analysiert werden. Deshalb wurden Methoden zum gezielten Crawlen entwickelt, sodass sich Speicherplatzbedarf, benötigte Bandbreite und Zeitbedarf enorm dadurch reduzieren.

   Diesbezüglich wird in dieser Arbeit versucht, einen fokussierten Crawler in Bezug auf Firmeninformationen zu erstellen. Aus einer Menge von Trainingsdaten wird gelernt, welche Links und Anchor-Texte zu einer Informationsseite führen, wodurch sich eine statistische Bewertung von Links und Anchor-Texten ergibt. Danach wird die Statistik auf die Testdaten angewandt.

2. Eine Einstiegsseite (Homepage) kann ihre Webpräsenz charakterisieren, weil viele wichtige Merkmale auf der Homepage zu finden sind. Um eine Webpräsenz zu kategorisieren, müssen nicht alle Webseiten analysiert werden. Die wichtigen Merkmale einer Homepage können Aufschluss über ihre Webpräsenz besser geben. Es muss lediglich überlegt werden, welche Merkmale auf einer Firmen-Homepage entscheidend sind.

   Auf diese Weise wird versucht, alle aussagekräftigen strukturellen und textuellen Merkmale aus der Homepage zu extrahieren.

3. HTML-Text ist semi-strukturiert, da stets eine gewisse Struktur vorgegeben sit, die durch HTML-Tags repräsentiert wird, welche wiederum auf eine Baumstruktur abgebildet werden können. Die Repräsentation der HTML-Seite durch eine Baumstruktur dient der Erkennung von

textuellen Einheiten. Dadurch wird die Erkennung der Informationseinheiten erleichtert.

4. Die gesuchten Informationen kommen in einem bestimmen Bereich der Seite vor. Nachdem die HTML-Seite auf ihre Baumstruktur abgebildet wurde, wird versucht, den Datenbereich mit der Depth-First-Traverse (Tiefensuche) zu bestimmen. Dabei werden die irrelevanten Bereiche automatisch vom Baum abgeschnitten.

5. Die Informationensdichte im Datenbereich ist sehr hoch, wobei jede Klasse oft durch einen Delimiter abgegrenzt wird. Wird ein Attribut als solches erkannt, dann tritt der Wert entweder im benachbarten Text oder zwischen den Delimitern auf. Somit kann auf ein großes Lexikon verzichtet werden. Stattdessen werden für jede Klasse interne und externe Indikatoren (Attribute) zusammengestellt, welche aus den Trainingsdaten abgeleitet werden.

   Der Datenbereich wird beim Attribut-Wert-Verfahren traversiert, wofür die vorhandenen Klassenattribute verwendet werden können. Für den jeweiligen Klassenwert kann ein regulärer Ausdruck gebildet werden.

   Auf diese Weise ist es möglich, die Klassenattribute leicht zu erweitern, und die Daten werden separat über eine Datei verwaltet.

6. Die extrahierten Daten sollen erweitert und aktuell gehalten werden.

7. Das entwickelte System soll auf andere Sprachen übertragbar sein.

Die vorliegende Arbeit ist wie folgt strukturiert: In Kapitel 2 wird die Website-Klassifikation behandelt. Dabei werden entsprechende Kategorien und Merkmale für die Entscheidungsfindung diskutiert und festgelegt. Hierfür werden die Merkmale nur fokussiert auf die gerade untersuchte Homepage analysiert, wobei zwischen strukturellen und textuellen Merkmalen unterschieden wird. Für die Klassifikation werden zunächst die strukturellen Merkmale verwendet, dagegen werden die textuellen Merkmale beim Naive-Bayes-Klassifikator benutzt.

In Kapitel 3 werden die Charakteristiken von Firmen-Homepages und Informationsseiten beschrieben. Anschließend wird in Kapitel 4 eine Übersicht zum entwickelten System gezeigt.

In Kapitel 5 wird die hier vorgestellte Methode zur Extraktion von Informationen ausführlich beschrieben. Dafür wird das Dokument auf eine Baumstruktur abgebildet und darauf das Attribut-Wert-Verfahren angewandt. Das Verfahren erlangt besonderen Stellenwert beim Einsatz mit verschiedenen Tabellentypen.

Nachdem die Extraktionsmethode beschrieben wurde, wird in Kapitel 6 näher auf die Extraktion der einzelnen Klasseninformation eingegangen. Im Anschluß an die Klassenaufteilung wird über die verschiedenen IE-Methoden und über ihre Anwendbarkeit auf die Informationsseiten diskutiert. Für die Extraktion einzelner Klasseninformationen werden die internen und externen Indikatoren intensiv genutzt.

Das entwickelte System wird in Kapitel 7 evaluiert, wofür sich der Präzision, dem Recall und dem F1-Maß bedient wird.

In Kapitel 8 wird die Struktur der Datenbanken veranschaulicht. Ferner wird gezeigt, wie ein Alias-Domain-Name erkannt werden kann.

Zum Schluss wird in Kapitel 9 die Arbeit zusammengefasst. Außerdem werden die zukünftigen Erweiterungsmöglichkeiten des Systems diskutiert.

Kapitel 2

Klassifikation von Websites

Da Firmendaten für uns besonders von Interesse sind, ist es wichtig, dass eine gegebene Website von Anfang an als solche erkannt wird. Als Website bezeichnet man einen Webauftritt eines Domainnamens. Diese beinhaltet im Normalfall eine Homepage bzw. zumindest eine Einstiegsseite. Im Internet-Lexikon „itwissen“ wird „Website“ wie folgt definiert[1]:

> „Die Website ist die Standort-Präsenz innerhalb des Web. Es ist das komplette Angebot eines Unternehmens, einer Organisation oder Verwaltung, einer Universität oder Forschungseinrichtung, eines Vereins oder einer Privatperson, die sich hinter dem Domain-Namen verbirgt, wobei eine Website in aller Regel aus vielen, in aller Regel hierarchisch angeordneten Web-Seiten besteht. Eine davon ist die Homepage, von der aus sich die Web-Seiten-Hierarchie eröffnet[2].“

Da eine Website aus vielen Webseiten besteht, ist es wahrscheinlicher, dass eine Website charakteristisch verschiedene Webseiten beinhalten kann. Eine Webseite kann eine ausführliche textuelle Anleitung eines Produktes beschreiben, während eine andere Webseite derselben Website nur Video- und Audiodaten zu Präsentationszwecke vorhält. Trotz dieser unterschiedlichen Webseiten kann sie dennoch der Webauftritt einer Firma sein.

[1] Eine genauere Definition folgt in Abschnitt 2.2.

[2] `http://www.itwissen.info/definition/lexikon/__website_website.html`.

Aufgrund der extrem schnell wachsenden Menge an Webdokumenten ist die Klassifikation von Websites und Webdokumenten in den Fokus des Information-Retrieval und der Informationsextraktion gerückt. Durch vorheriges Kategorisieren von Websites kann eine Suchmaschine die benötigte Bandbreite reduzieren und für die Benutzer einen besser geeigneten Service bieten. Zudem wird die Extraktion von Informationen gezielter und präziser.

Jedoch ist die Klassifikation von Websites von ihrem Zweck abhängig. Für Webverzeichnisse sind z.B. thematisch klassifizierte Dokumente, für Gelbe Seiten nach Branchen klassifizierte Websites besser geeignet.

In Pierre (2001) [108] erfolgt die Klassifikation von Websites nach 1997 NAICS (North American Industrial Classification System). Seine Klassifikation entspricht den 21 NAICS-Kategorien, wofür er Inhaltsmerkmale aus Meta-Informationen und Body-Text verwendet. Er suchte dabei durch gezieltes Crawlen nach der Inhaltsseite einer Website.

In Amitay et al. (2003) [4] und Lindemann & Littig (2006, 2007) [85, 86] wurden Websites nach ihrer Funktion klassifiziert. Dabei wurden keine inhaltlichen, sondern nur strukturelle Merkmale verwendet. Sie teilten Websites nach ihrer Funktion in fünf bis acht Kategorien. Jedoch liegt der Hauptnachteil ihrer Methode darin, dass die Kategorie erst dadurch bestimmt wird, dass zuerst viele Webseiten einer Website gecrawlt und dann die Relationen zwischen ihnen berechnet werden.

Wie genau eine Website als solche kategorisiert, und welche Merkmale für die Klassifikation ausgewählt werden sollen, hängt vom Zweck ab. So hat in Bsiri (2007) [14] ein binäres Klassifikationssystem ausgereicht. Dabei wurde eine Website in Firmen- und Nicht-Firmen-Website klassifiziert, nachdem hauptsächlich die Anchor-Texte auf der Homepage eines Domain-Namens ausgewertet wurden.

Besonders zwei Faktoren sind für die Klassifikation von Websites wichtig: Es müssen Kategorien festgelegt und die zu bewertenden Merkmale ausgewählt werden. In diesem Kapitel werden diese beiden Faktoren untersucht und festgelegt.

Was die Merkmale betrifft, gehen wir davon aus, dass die Homepage weitgehend ihre Website charakterisieren kann. Das bedeutet, dass wir alle möglichen Merkmale aus der Homepage einer Website extrahieren können. Ist ein Domain-Name gegeben, dann wird wie bei Bsiri (2007) [14] zuerst nach der Home-

page gesucht. Aus der Homepage werden die strukturellen und verschiedenen domainnamenrelevanten Merkmale, die für die Klassifikation einer Website wichtig sein können, extrahiert. Danach wird versucht, die Website mithilfe dieser Faktoren zu klassifizieren.

Da eine Website über Domain-Namen zugeordnet werden kann, wird zuerst auf das Domain-Namen-System eingegangen. Danach werden natürliche Kategorisierungen der Domain-Namen anhand von Beispielen der ccTLD „uk“ gezeigt und die möglichen Kategorien der ccTLD „de“ diskutiert. Nachdem in die potenziellen Kategorien unterteilt wurde, wird die Klassifikation der Websites durchgeführt.

2.1 Website und Domain-Namen-System

Der Begriff „Website“ ist in erster Linie mit dem Begriff „Domain-Name“ verbunden. Alle Webseiten, die sich hinter einem „Domain-Namen“ verstecken, bilden zusammen diese eine Website.

Domain-Namen sind hierarchisch aufgebaut. Von rechts her wird zuerst die Wurzel („.“), dann die allgemeine und Länder-Domain vergeben. Jede Domain ist dabei durch einen Punkt („.“) getrennt, und die Wurzel wird in der Regel weggelassen. Die Hierarchie des Domain-Namen-Systems wird in Abbildung 2.1 veranschaulicht.

Die internationale Domain-Verwaltung erfolgt durch die IANA[3] und ICANN[4]. Sie verwalten generische Top-Level-Domains (gTLD) und Länder-Top-Level-Domains (ccTLD). Unter ccTLD können zweite Domain-Namen (SLD) registriert werden und diese werden im Normalfall als Website angesehen.

In Deutschland sind rund 10 Millionen SLDs registriert und werden durch den DENIC[5] verwaltet. Auf mehr als die ccTLD „de“-Domain verweist nur die gTLD „com“-Domain, darunter rund 60 Millionen.

Die bislang vergebenen gTLDs sind in Tabelle 2.1 angegeben.

Von den gTLDs interessieren wir uns für „com, info, net, org, biz, edu“, die auch in Deutschland ansässig sein können. Z.B. haben viele deutsche Firmen

[3] `http://www.iana.org`.
[4] `http://www.icann.org`.
[5] `http://www.denic.de`.

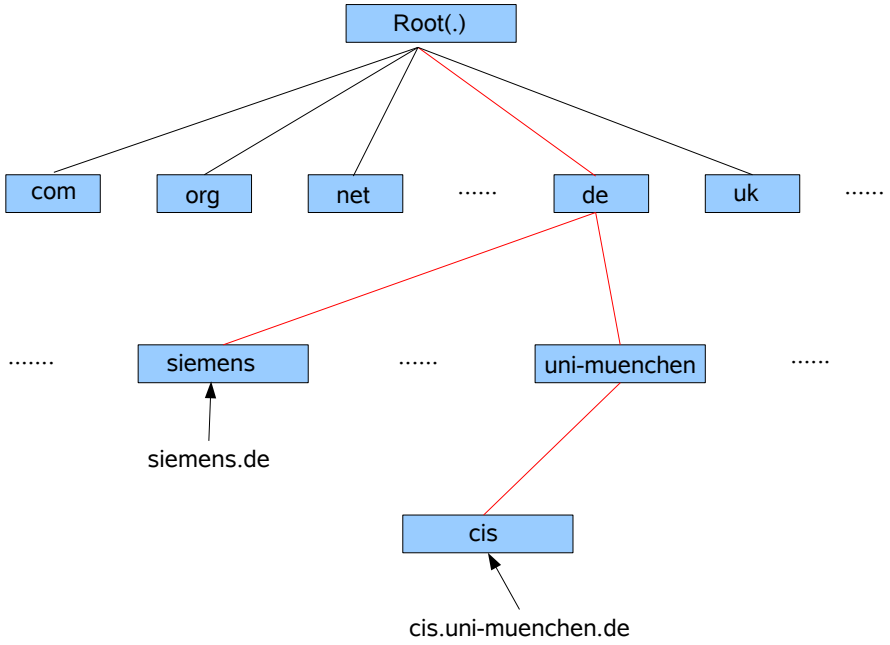


Abbildung 2.1: Schematische Darstellung der Domain-Namen-Hierarchie

aero	Reserviert für Mitglieder der Luft-Transport-Industrien
biz	Eingeschränkt für Business und verwaltet durch *NeuLevel, Inc*
cat	Reserviert für Katalanische linguistische und kulturelle Gemeinde
com	Verwaltet durch *VeriSign Global Registry Services*
coop	Reserviert für kooperative Gesellschaft
info	Verwaltet durch *Afilial Limited*
jobs	Reserviert für Human-Resource-Manager
mobi	Reserviert für mobile Produkte and Dienste
museum	Reserviert für Museen
name	Reserviert für Individuen
net	Verwaltet durch *VeriSign Global Registry Services*
org	Verwaltet durch *Public Interest Registry*
pro	Eingeschränkt für Kreditbranche und verwaltet durch *RegistryPro*
travel	Reserviert für Reisebranche
gov	Ausschließlich reserviert für die Regierung der USA
edu	Reserviert für Bildungsorgane
mil	Ausschließlich reserviert für das Militär der USA
int	Gebraucht für Registry-Organisation

Tabelle 2.1: Generische Top-Level-Domain (gTLD)

neben der länderspezifischen ccTLD „de“ auch die generische gTLD „com“ reserviert.

2.1.1 Website und URI

Primär kann eine Website als Domain-Name angesehen werden. Er soll aber eindeutig im Internet identifizierbar sein. Dafür sorgt der URI[6].

Ein URI (Uniform Resource Identifier) identifiziert die Informationsquelle unabhängig vom Kontext eindeutig und besteht aus verschiedenen Komponenten. Die URL (Uniform Resource Locator) beschreibt die Informationsquelle durch den primären Zugriffsmechanismus[7] und ist eine Untermenge

[6]Die URI-Syntax stützt sich wesentlich auf den Artikel von Berners-Lee (2005) [10].

[7]URIs dienen zur Identifizierung einer abstrakten oder physischen Ressource, während URLs eine Ressource über das verwendete Netzwerkprotokoll und den Ort der Ressource in Netzwerken identifizieren. Als Untermenge einer URI gibt es neben URL noch den URN,

der URIs.

URIs sind hierarchisch identifizierbar: So ist der Doppelpunkt („:“) der Schema-Delimiter. Nach ihm können Schrägstrich („/“), Fragezeichen („?“) und Raute-Zeichen („#“) als Haupt-Delimiter vorkommen.

Die folgenden Zeichen sind als Delimiter reserviert und dürfen nicht als URI-Datenzeichen verwendet werden.

Allgemeine Delimiter:	„:“, „/“, „?“, „#“, „[“, „]“, „@“
Sub-Delimiter:	„!“, „$“, „&“, „ ’ “, „(“, „)“, „*“, „+“, „,“, „=“

Die reservierte Zeichenmenge liefert verschiedene Abgrenzungsmöglichkeiten, damit die Daten innerhalb des URI von den anderen unterschieden werden. Da die URI-Syntax nur US-ASCII kodierte Zeichen erlaubt, werden die Zeichen außerhalb des ASCII-Codes oder Delimiter innerhalb der Komponenten durch das folgende Schema kodiert.

URI-Escape:	„%“ HEXDIGIT HEXDIGIT

Als URI-Datenzeichen können die folgenden und die mit „%“ kodierte Escape-Zeichen verwendet werden.

Nicht-reserviert:	ALPHA, DIGIT, und „-“, „.“, „_“, „~“
ALPHA:	A-Za-z
DIGIT:	0-9

Insgesamt sieht eine absolute vollständige URI-Syntax beispielsweise wie folgt aus[8]:

```
foo://example.com:8042/over/there?name=ferret#nose
\_/   \______________/ \________/ \_________/ \__/
 |           |              |           |       |
scheme     authority        path       query   fragment
```

der eine Ressource über ihren Namen identifiziert. Siehe auch `http://de.wikipedia.org/wiki/URI` und `http://de.wikipedia.org/wiki/URL`.

[8] Berners-Lee (2005) [10].

2.1.1.1 Authority-Teil

Der Authority-Teil besteht aus folgenden Subkomponenten: (Die Benutzerinfo ist optional.)

Authority:	[userinfo „@“] host [„:“ port]

Userinfo: Die Subkomponente der Benutzerinformation kann aus dem Benutzernamen und schema-spezifischen Informationen bestehen, wie der Ressourcenzugang autorisiert werden kann. Falls die Benutzerinfo vorhanden ist, wird sie durch das Zeichen „@“ vom Host getrennt. Für die Benutzerinfo können die folgenden Zeichen verwendet werden:

Userinfo:	Nicht-reservierte Zeichen Durch das Escape-Zeichen „%“ kodierte Zeichen Sub-Delimiter und „:“

Host: Die Subkomponente des Hosts ist mit dem Domain-Namen vergleichbar. Der Host-Teil wird mit der folgenden Syntax konstruiert.

Host:	IP-Literal oder IPv4-Adresse oder Reg-Name
IPv4address:	dec-octet „.“ dec-octet „.“ dec-octet „.“ dec-octet
IP-Literal:	„[“ IPv6-Adresse „]“

Da IPv4 nur 32Bits zur Verfügung hat und immer mehr IP-Adressen vergeben werden, wurde IPv6 mit 128 verfügbaren Bits eingeführt.

Ein IP-Literal, das nur bei Version 6 vorkommt, wird durch die eckige Klammer gekennzeichnet und hexadezimal angegeben.

Diese mit Zahlen kodierten IPv4-Adressen und IP-Literale sind schwer zu merken, und Menschen sind besser mit Namen vertraut. Eine gegebene IP-Adresse kann auch in einen normalen Namen konvertiert werden. Die folgenden drei Adressen sind z.B. gleich:

IPv4-Adresse:	129.187.148.72
IPv6-Literal:	[2001:4C10:4F01::5]
Host-Name:	www.cis.uni-muenchen.de

An den drei Versionen ist ersichtlich, dass die in Buchstaben kodierte Adresse viel einfacher zu merken ist.

Zu bemerken ist, dass trotz der erweiterten IP-Adressen von IPv4 auf IPv6 der zugelassene Zeichensatz für die registrierten Namen unverändert bleibt. Das bedeutet, dass man weiter auch nur mit ASCII-Zeichen arbeitet. Die nur mit ASCII-Zeichen kodierte IP-Adresse kommt der Analyse des Host-Namens zu Gute und wird für die Klassifikation der Websites verwendet.

Auch interessant in Bezug auf Informationsextraktion ist der registrierte Name, weil er oftmals Aufschluss über den Inhaber eines Domain-Namens gibt.

Der Host-Name ist von der DNS-Definition durch einen Punkt („.“) getrennt und jede Domain-Ebene beginnt und endet mit einem alphanumerischen Zeichen inklusive des Bindestrichs („-“).

Die folgenden Zeichen werden für den registrierten Host-Namen verwendet:

Reg-Name:	unreserved oder pct-encoded oder sub-delimits

Die Portnummer besteht aus Ziffern. Falls diese nicht vorhanden ist, dann wird die Default-Portnummer „80“ angenommen.

2.1.1.2 Pfad

Die Pfadkomponente des URI ist hierarchisch strukturiert, was aus der Sicht der Website-Schöpfer leicht verständlich ist, da diese eine Übersicht über die Dokumente schaffen sollen. Diese Charakteristik lässt sich aber auch beim Information-Retrieval und bei der Informationsextraktion ausnutzen.

Die Pfadtiefe wird durch das Zählen der Schrägstriche („/“) im Pfadteil berechnet.

Da der hierarchisch gesehen untere Pfad im Normalfall alle Eigenschaften des oberen Pfades vererbt bekommt, kann diese Menge an Dokumenten durch die Analyse dieses Pfades[9] auffindbar sein.

Falls ein Pfad vorhanden ist, werden für ihn die folgenden Zeichen verwendet:

Path:	unreserved, pct-encoded, sub-delimits, „:“ oder „@“

[9]Wir nehmen an, dass der Pfadinhalt für den Crawler zugänglich ist.

2.1.1.3 Query und Fragment

Die Komponente „Query“ wird bei der Informationsextraktion im Normalfall nur wenig gewichtet, weil sie eine dynamisch erzeugte Webseite darstellt und nicht unmittelbar für den Crawler sichtbar ist. Allerdings wird bei vielen Webseiten durch die Navigation über Menüs eine Query-Form erzeugt und diese zum Browser geschickt.

Das Fragment verweist auf eine bestimmte Stelle eines Dokumentes und zielt auf die Aufmerksamkeit des Besuchers und ist somit für die Informationsextraktion nutzbar, da es einen Begriff oder einen Term kennzeichnen kann.

2.1.2 URI und regulärer Ausdruck

Ist eine URI gegeben, sollten die verschiedenen Komponenten identifizierbar und zerlegbar sein, um daraus die möglichen Informationen ziehen zu können. Dies ermöglicht der folgende reguläre Ausdruck formuliert in der Syntax der Programmiersprache „PERL[10]“:

```
^(([^:/?#]+):)?(//([^/?#]*))?([^?#]*)(\?([^#]*))?(#(.*))?
 12            3  4          5       6  7        8 9
```

Ist z.B. eine URI „`http://www.ics.uci.edu/pub/ietf/uri/#Related`“ gegeben, dann werden die Teile durch den oberen regulären Ausdruck wie folgt identifiziert.

[10] Berners-Lee (2005) [10].

```
$1 = http:
$2 = http
$3 = //www.ics.uci.edu
$4 = www.ics.uci.edu
$5 = /pub/ietf/uri/
$6 = <undefined>
$7 = <undefined>
$8 = #Related
$9 = Related

Schema = $2
Authority = $4
Pfad = $5
Query = $7
Fragment = $9
```

In diesem Beispiel ist der Host-Teil identisch mit dem Authority-Teil, da weder die Benutzerinfo noch der Port vorhanden sind.

2.2 Definition einer Website

Nachdem die Relation zwischen Website und Domain-Namen-System (DNS) veranschaulicht wurde, muss jetzt der Begriff der Website definiert werden.

Grob gesagt kann eine Website einem registrierten Domain-Namen zugeordnet werden. Diese Definition ist aber zu eng gefasst. Wenn nur ein registrierter Domain-Name einer Website entspräche, dann hätten viele Institute einer Universität, die einen eigenen Webserver betreiben, verwalten und warten, keine Website. Der registrierte Domain-Name „uni-muenchen“ hat z.B. eine Subdomain „cis“, die auf einen eigenen Webserver verweist. Diese Subdomains mit der registrierten SLD sollen auch als Website angesehen werden. Falls dem nicht so wäre, dann würde die Subdomain „de“ mit der SLD „yahoo“ nicht als Website betrachtet werden.

Jedoch können nicht alle Subdomains einer SLD als Website betrachtet werden. Z.B. kann die Subdomain „impressum“ mit der SLD „adeos“ in der Regel keine eigene Website darstellen (URL: `http://impressum.adeos.de`). Diese Subdomain ist nur eine Webseite von vielen Webseiten der SLD „adeos“.

Da wir nach den Informationen über den Betreiber einer Website suchen, darf die Subdomain „impressum“ mit der SLD „adeos“ nicht als eigene Website erkannt werden.

In Amento et al.(2000, S. 297) [3] ist eine Website wie folgt definiert:

> „A site (multimedia document) is an organized collection of pages on a specific topic maintained by a single person or group. Sites have structure, with pages that play certain roles (front door, table-of-contents, index). A site is not the same thing as a domain: for example, thousends of sites are hosted on `www.geocities.com`.“

Diese Definition ist aber zu breit gefasst und nicht präzise genug. Ein Webauftritt kann viele unterschiedliche Themen mit strukturierten Seiten haben (Z.B. Yahoo! Verzeichnis-Service).

Andererseits muss eine Website minimale Information über den Betreiber angeben. In Deutschland ist diese Pflicht in einer **Anbieterkennung** durch Gesetze festgelegt. Die relevanten Gesetze dafür sind das Teledienstegesetz und der Mediendienste-Staatsvertrag. Laut dieser Gesetze soll eine Website mindestens die folgenden Informationen enthalten: Den Namen und die Anschrift des Betreibers, bei juristischen Personen zusätzlich den Vertretungsberechtigten, Angaben für die elektronische Kontaktaufnahme; E-Mail-Adresse.

Für die Defintion einer Website sollte diese minimale Information mit einbezogen werden. Ohne Angabe über den Betreiber ist eine Website den Besuchern gegenüber anonym.

Zusammenfassend wird eine Website wie folgt definiert:

- Definition einer Website
  1. ein registrierter Domain-Name, der aktiv ist, oder eine aktive Subdomain davon, die als Webserver operiert
  2. sie enthält minimale Informationen über den Betreiber

Unter Websites fallen alle Webpräsenzen, die diese Definition erfüllen. So steht der Domain-Name „siemens“ ebenso für eine Website, wie die Subdomain „cis“ der SLD „uni-muenchen“.

Die beiden enthalten jedoch völlig unterschiedliche Informationen. Während die SLD „siemens“ über viele Informationen von Kunden- und Investor-Relationen, Produkten und Service, Jobs- und Stellenangeboten usw. verfügt, konzentriert sich die Subdomain „cis“ auf Studienpläne, Lehrveranstaltungen, Lehrkräfte und Forschung. Das heißt, obwohl beide eine Website sind, zeichnen sie sich durch ganz andere Charakteristiken aus.

Inwiefern diese verschiedenen Charakteristiken für eine Website-Kategorisierung eine Rolle spielen, und welche verschiedene Charakteristiken überhaupt zu finden sind, wird im nächsten Abschnitt diskutiert.

2.3 Website-Kategorien

Im Anschluss an das Domain-Namen-System und die regulären Ausdrücke wird in diesem Abschnitt auf die Website-Kategorien eingegangen.

Websites können nach verschiedenen Kriterien kategorisiert werden, etwa nach gTLD oder ccTLD. Wenn sie nach gTLD kategorisiert sind, dann entsprechen sie mehr oder weniger den Tätigkeitsbereichen, während sie dem Territorium entsprechen werden, wenn sie nach ccTLD kategorisiert werden.

Oder sie können nach Themen oder Funktion kategorisiert werden. Die meisten Webverzeichnisse sind zum Beispiel hierarchisch nach Themen organisiert.

Es wird hier nicht versucht, Websites nach Themen zu klassifizieren. Das ist ein zu großes Projekt für diese Arbeit.

Websites nach ccTLD oder gTLD zu kategorisieren ist trivial. Dies erfolgt einfach aus der DNS-Struktur.

Des Weiteren können Websites nach ihrer Funktion kategorisiert werden, so dass sie mehr oder weniger den Tätigkeitsbereichen des Betreibers entsprechen. Da das deutsche Domain-Namen-System über keine vorgegebene SLD verfügt, ist die Kategorisierung der deutschen Websites nach ihrer Funktion eine Herausforderung.

Andererseits stellt sich die Frage, wie viele Kategorien einzuführen sind, da es keine expliziten Kriterien dafür gibt.

Ausgehend von den verschiedenen Kategorienbeispielen der gTLD und SLD von „uk“ wird eine Übersicht über die Kategorien in der Literatur gegeben. Danach werden die möglichen Kategorien der ccTLD „de“ festgelegt.

2.3.1 gTLD und Kategorien

Wie im Abschnitt 2.1 gezeigt wurde, sind gTLDs nach ihrem Zweck gegliedert. Während die gTLD „org“ für eine Organisation gedacht ist, ist die gTLD „edu“ für Bildungsorgane zuständig. Das Problem liegt jedoch darin, dass die gTLD „com“ nicht nach dem eigentlichen Zweck, sondern ganz allgemein genutzt wird. Die Domain-Namen werden nach dem „First come, first serve“-Prinzip vergeben. Unter der gTLD „com“ können daher nicht nur kommerzielle, sondern auch andere Bereiche, wie Organisationen, registriert werden. Nichtsdestotrotz können die gTLDs den ersten Hinweis für die Klassifikation von Websites geben.

Abgesehen von der gTLD „com“ und speziellen gTLDs, können wir Websites nach der gTLD wie folgt aufteilen: *Business, Organisation, Bildungsorgane, Netzwerkdienst, Information, Regierung, Privat.*

Obwohl diese Einteilung nicht immer gemäß dem eigentlichen Zweck der gTLD aufrechterhalten wird, kann sie einen ersten Einblick in die Klassifikation von Websites geben. Die gTLD „com“ wird als ein Sammelbecken betrachtet, weil unter ihr alle möglichen Domains registriert werden können.

2.3.2 SLD und Kategorien

Die gTLD allein ist noch nicht in der Lage, als Website zu agieren. Unter der gTLD muss ein eigentlicher Domain-Name registriert werden, um als Website fungieren zu können. Dieser Domain-Name wird SLD (Secondary-Level-Domain) genannt. In der ccTLD „de“ ist die SLD ohne Weiteres als Website vertreten, was die Klassifikation von deutschen Websites besonders schwierig macht, da in der URL kein Anzeichen für die Kategorie zu finden ist. Anders als die ccTLD „de“ werden in vielen Ländern mit z.B. der ccTLD „uk“ oder „kr“ die SLDs vorgegeben. Diese SLDs dienen der ersten Kategorisierung der Websites. Wir betrachten zuerst die vorgegebenen SLDs anhand des Beispiels der ccTLD „uk“.

2.3.2.1 SLD der ccTLD „uk“

Neben der ccTLD „uk“ gibt es im UK auch die ccTLD „gb“ für *Great Britain* und die ccTLD „sco“ für *Scotland*. Üblicherweise wird aber die ccTLD „uk“ bevorzugt.

Im Gegensatz zur ccTLD „de“ werden bei der ccTLD „uk“ die SLDs auf verschiedene Kategorien vergeben. So ist die Vergabe der Domain-Namen anders als in Deutschland. Folgende Tabelle listet die SLDs in der ccTLD „uk“ auf. Die ccTLD „uk“ wird ebenfalls angegeben.

ac.uk	Akademie
co.uk	Kommerziell / Allgemein
gov.uk	Regierung
ltd.uk	Firma (Ltd)
me.uk	Individuen
mod.uk	Ministerium für Abwehr
net.uk	ISPs und Netzwerkfirma
nic.uk	Netzwerk
nhs.uk	Institution für Gesundheit
org.uk	Nicht-Gewinn-orientierte Organisation
plc.uk	Firma (public limited companies)
police.uk	Polizei
sch.uk	Schule

Außerdem werden die folgenden SLDs der ccTLD „uk“ vor der Einführung von **Nominet**[11] vergeben und bleiben unangetastet. Unter „uk“ sind rund 5.5 Mio. Domain-Namen registriert.

bl.uk	Britische Bibliothek
british-library.uk	Britische Bibliothek
icnet.uk	Imperial Cancer Research Fund
jet.uk	Joint European Torus
nel.uk	National Engineering Laboratory
nls.uk	Nationale Bibliothek von Schottland
national-library-scotland.uk	Nationale Bibliothek von Schottland
parliament.uk	Parlament des „United Kingdom“

[11] `http://www.nic.uk`.

Ursprünglich war die Domain „co.uk“ für kommerzielle Websites gedacht, aber sie wird, wie die gTLD „com“, allgemein verwendet, während „ltd.uk“ und „plc.uk“ streng den Bestimmungen der Registrierung unterliegen. Eine Besonderheit stellt auch die „sch.uk“ dar. Der dritte Domain-Name für die SLD „sch.uk“ wurde an Lokale Autoritäten vergeben. So sind die eigentlichen Domain-Namen für „sch.uk“ erst in vierter Stufe (z.B. `tiffin.kingston.sch.uk`) registriert.

Abgesehen von den speziellen SLDs können bei der ccTLD „uk“ die folgenden Kategorien an der vorgegebenen SLDs erkannt werden: *Akademische, Kommerzielle, Organisation, Regierung, Netzwerkdienst, Gesundheitswesen, Privat, Schule.*

2.3.3 Kategorien nach Website-Funktionen

Wie sich bei der Einteilung der Kategorien von gTLD und SLD bei der ccTLD „uk“ gezeigt hat, werden die gTLD „com“ und die SLD „co.uk“ allgemein gebraucht. Websites unter diesen Domain-Namen können nicht nach dem jeweiligen DNS klassifiziert werden. Daher wird hier versucht, Websites nach ihrer Funktion zu klassifizieren.

Bei Amitay et al. (2003) [4] und Lindemann & Littig (2006, 2007) [85, 86] wurden Websites nach ihrer Funktion kategorisiert. Dabei wurden die Systeme unabhängig vom Inhalt entwickelt. Mit ihren Systemen können Websites ohne Betrachtung der Inhalte klassifiziert werden. Die Kategorisierung erfolgt nur aufgrund struktureller Merkmale.

2.3.3.1 Kategorien bei Amitay et al. (2003)

Bei Amitay et al. (2003) [4] wurden die folgenden acht Kategorien aufgrund ihrer Funktionalität eingeführt: *Firmen-Websites, Inhalts- und Medien-Websites, Suchmaschinen, Webhierarchien und -verzeichnisse, Portale, E-Shops, Virtuelle Host-Dienste, Universitäten.*

Die Intuition hinter dieser Einteilung ist die folgende:

1. Eine reine Suchmaschinen-Site zeichnet sich durch geringe Webinhalte bestehend aus Suchseite, Firmeninfo, fortgeschrittenen Suchoptionen

etc. aus. Sie kann über Links von vielen anderen Websites erreicht werden, während sie selbst nur wenige Links enthält.

2. Webhierarchien und -verzeichnisse kategorisieren Websites in Taxonomien und enthalten viele Links zu diesen klassifizierten Websites. Insbesondere werden Webverzeichnisse sehr viele (tausende) ausgehende Links haben und auch diese Links sind strukturiert.

3. Große Firmen-Websites haben sehr viele (tausende) Webseiten und sie sind in einer Form von Verzeichnis organisiert. Sie zeichnen sich durch gut strukturierte interne Links aus, weil die Seiten durch eine Art Schablone generiert werden. Sie haben meistens ein Navigationsfeld und tendenziell wenig ausgehende Links.

4. Websites von „virtuellen Host"-Diensten haben oft nicht gut strukturierte interne Links. In der Regel gibt es keine Links von einer Firmenseite zu gehosteten individuellen Seiten.

5. Websites von Universitäten haben einen hybriden Charakter, der zwischen Firmen- und Host-Websites anzusiedeln ist. Während sie von der Administration her Websites einer Firmen-Website ähneln, sind die individuellen Homepages eher Host-Websites.

Amitay et al. (2003) haben bei der Klassifikation insgesamt 73 strukturelle Merkmale in Betracht gezogen. Im Wesentlichen sind es die Links (ein- und ausgehende), die Expansionsrate der Top-Level-Seite und die durchschnittliche Seitenpfade. Um alle Verbindbarkeiten zu prüfen, untersuchten sie 500 Mio. Webseiten von AltaVista.

2.3.3.2 Kategorien bei Lindemann & Littig (2006, 2007)

Wie Amitay et al. (2003) verwendeten Lindemann & Littig (2006, 2007) [85, 86] nur strukturelle Merkmale zur Klassifikation von Websites. Sie unterteilten die deutschen Websites nach ihrer Funktion in die folgenden acht Kategorien:

Akademische	Websites von Universitäten und Forschungsinstituten
Blog	Weblogs als eine beliebte Repräsentation einer Gemeinschafts-Website
Community	(ohne Definition)
Firmen	Webpräsenz eines Unternehmens
Information	(ohne Definition)
Organisation	(ihre Bezeichnung „Nonprofit“ ohne Definition)
Privat	private individuelle Homepages
Shop	Online Shops und Auktionsportale

Sie haben dabei 30 Merkmale zusammengestellt: Hauptmerkmale sind z.B. Größe der Website, Organisation der Website, URL-Bestandteile, Technische Realisierung und Linkstruktur. Sie mussten große Mengen von Webseiten sammeln, um sie in eine von acht Kategorien einordnen zu können. Für 1 461 Websites mussten sie ca. 7 Millionen Webseiten crawlen. Außerdem wurden ihnen zum Trainieren einzelner Verbindbarkeit ca. 47 Millionen bekannte Webseiten zur Verfügung gestellt.

2.3.4 Kategorien der ccTLD „de“

Die ccTLD „uk“ hat acht Kategorien, wenn man von den anderen SLDs der ccTLD „uk“ absieht. Die anderen Versuche haben eine gemeinsam: acht Kategorien wurden bei der Klassifikation von Websites angewendet.

Welche Kategorien aufgenommen werden, ist je nach Zweck und Methode variabel. Die meisten Kategorien sind intuitiv klar, während einige Kategorien wie Information und Netzwerkdienst oder Host-Dienst nicht eindeutig sind. Für die Klassifikation der ccTLD „de“ werden zuerst die intuitiv eindeutigen Kategorien aufgenommen. Die Kategorien sind in Tabelle 2.2 aufgelistet.

Die anderen, im Web als eine Kategorie angesehenen üblichen Websites wie Suchmaschinen oder Webverzeichnisse werden unter der Kategorie „Information“ subsumiert, weil sie dem Besucher zum Auffinden von gesuchten Informationen dienen.

Unter „Information“ fallen zum Beispiel nicht nur Portale, sondern auch Inhaltsinformation oder die auf andere hinweisenden Informations-Sites wie Webverzeichnisse oder Suchmaschinen. Gemeinde-Websites gehören auch zu dieser Kategorie.

Akademie	Universitäten und Forschungsinstitute
Firmen	Firmen und Einzelunternehmen
Organisation	Vereine, nicht-gewinnorientierte Organe
E-Shop	Online-Shops und Auktionsportale
Privat, Blog und Forum	private individuelle Homepages, Blogs und Foren
Gesundheitswesen	Apotheken, Krankenhäuser und Arztpraxen
Schule	Schulen
Information	Websites, die die Besucher informieren wollen

Tabelle 2.2: Website-Kategorien

Obwohl auch die Sites anderer Kategorien versuchen, den Besuchern Informationen zu vermitteln, beziehen sich diese Informationen meist primär auf den Betreiber selbst.

E-Shops versuchen, Gewinn zu erzielen. Blogs und Foren kann man zusammen als Internetgemeinschaft betrachten.

Es muss betont werden, dass die Kategorien nach der Funktion eingeteilt werden. Für unseren Zweck würden die oberen acht Kategorien genügen, da wir primär nach den Firmeninformationen suchen.

Außer der in Tabelle 2.2 aufgelisteten Kategorien werden zusätzlich zwei Kategorien mit den strukturellen Merkmalen klassifiziert: „Nicht-aktiv“ und „Erotik-Site“.

2.4 Website-Klassifikation

Nachdem die Kategorien festgelegt wurden, müssen nun die Merkmale definiert werden. Die Merkmale können nach dem Zweck ausgewählt werden.

Da wir Websites nicht nach dem Thema, sondern nach ihrer Funktion kategorisieren wollen, kommen die Ansätze von Amitay et al. (2003) und Lindemann & Littig (2006, 2007) eher in Frage als andere. Sie mussten für die Klassifikation alle Verbindbarkeiten zwischen Webseiten prüfen. Dafür crawlten sie alle Webseiten der betroffenen Website, wodurch der Zeitaufwand enorm wird.

Trotz der großen Menge an Webseiten liegt die Präzision von Amitay et al. (2003) bei 59% und der F1-Score von Lindemann & Littig (2006, 2007) bei

80%.

Aufgrund des Zeitaufwands sind ihre Methoden nicht für unseren Zweck einsetzbar. Wir wollen nicht alle Webseiten crawlen, um entscheiden zu können, zu welcher Kategorie eine Website gehört. Die Klassifizierung ist für unseren Zweck wie ein Filter für die Informationsextraktion. Es wird zuerst entschieden, ob die betroffene Website für die Informationsextraktion weiter verarbeitet werden soll.

Bsiri (2007) hat gezeigt, wie die Einstiegsseite einer Webpräsenz für die binäre Kategorisierung genutzt werden kann. Sie hat sich dabei hauptsächlich auf Anchor-Texte konzentriert.

Die Menge der Anchor-Texte ist z.B. eines von vielen Merkmalen von Einstiegsseiten einer Webpräsenz. Wir wollen alle möglichen Merkmale auf der Einstiegsseite nutzen, um eine Entscheidung treffen zu können.

Daher sind wir der Ansicht, dass die Einstiegsseite einen Webauftritt am besten charakterisiert. Die meisten Einstiegsseiten haben einen „Titel“, viele verfügen über „Meta-Informationen“ und eine Sitemap, die alle für die Website wichtig sind.

2.4.1 Auswahl der Merkmale

Es werden sowohl strukturelle als auch textuelle Merkmale verwendet. Während einige Kategorien durch ihre besonderen strukturellen Eigenschaften bestimmt werden können, kann bei anderen Klassen aufgrund der strukturellen Eigenschaften noch keine Entscheidung getroffen werden. Bei der Klassifikation basierend auf strukturellen Merkmalen werden einfache Heuristiken verwendet. Bei der textuellen Kategorisierung wird die *Naive Bayes'sche Klassifikation*smethode eingesetzt.

2.4.1.1 Strukturelle Merkmale

Jede Website-Kategorie hat eigene Eigenschaften. Dazu gehören auch die strukturelle Eigenschaften. Es können sehr viele strukturrelevante Merkmale aus der Einstiegsseite gefunden werden. Die Merkmale sollen aber aussagekräftig sein. Für die Klassifikation der Websites werden für unseren Zweck die folgenden strukturellen Merkmale betrachtet:

- Strukturelle Merkmale
  - Anzahl der Subdomains der Links
  - Anzahl der internen Links
  - Anzahl der externen Links
  - Durchschnittliche Pfadanzahl von internen Links
  - Länge der internen Anchor-Texte
  - Länge der jeweiligen Meta-Daten der Meta-Keywords und -Beschreibung, des Meta-Copyrights, -Authors und -Publishers
  - Länge des Body-Textes
  - Verhältnis der Anzahl von internen und externen Links
  - Verhältnis von Bildern und internen Links
  - Verhältnis der Länge der internen Anchor-Texte zur Textlänge

Ist eine URL gegeben, dann wird zuerst nach der Einstiegsseite, d.h. Homepage, gesucht. Dort werden alle strukturellen Merkmale bestimmt.

Subdomains werden durch die Analyse der internen Links, die auf der Einstiegsseite zu finden sind, erkannt. Dabei wird der im Abschnitt 2.1.1 etablierte reguläre Ausdruck genutzt. Er zerlegt die URLs in Teile mit semantischem Gehalt. Nachdem der Host-Teil einer URL identifiziert wurde, wird er durch das DNS-System zerlegt. Die Subdomains sind somit die Teile, welche links der SLD vorkommen.

Akademische Websites zeichnen sich oft durch viele Subdomains aus, während Firmen-Websites kaum oder sehr wenige Subdomains haben. Von den Subdomains auf Firmen-Websites ist oft nur die Informationsseite relevant, wie die Subdomain „impressum“ der SLD „adeos“. Akademische Subdomains verweisen oft auf separate Webserver.

Links und Abbildungen werden aus den entsprechenden HTML-Tags extrahiert. Informations-Sites haben oft sehr viele externe Links (hinweisende Informations-Sites) oder sehr lange Texte (inhaltliche Informations-Sites).

Shopsites sind oft mit vielen Abbildungen ausgestattet und das Verhältnis der Abbildungen zu den internen Links ist sehr hoch.

Meta-Daten sind oft bei einer Firma zu finden, während sie bei einer Privat-Site selten angegeben werden. Die Länge der Meta-Daten einer Informations-Site oder einer Shopping-Site ist oft sehr groß.

2.4.1.2 SLD als erster Hinweis

Die URL spielt bei verschiedenen Web-Mining-Aufgaben eine wichtige Rolle. Kavalec & Svátek (2002) [73] haben z.B. URLs für die Erstellung einer Ontologie benutzt. Dabei haben sie die Pfadstruktur analysiert. Falls die niedrigen Pfade den gleichen Elternpfad haben, dann gehören sie zu derselben Klasse.

Devi & Selvakuberan (2005) [33] haben URLs zur Webseitenkategorisierung verwendet. Sie haben dafür URLs in ihre inhaltstragenden Teile zerlegt. Obwohl sie nur die drei Kategorien „*Student, Project, Faculty*“ der Uni-Webseiten in Betracht gezogen haben, könnten sie zeigen, dass die Analyse der URLs auf bestimmten Bereichen bei der Kategorisierung von Webseiten konkurrenzfähig ist.

Auch bei unserer Klassifizierung der Websites wird dieser Hinweis genutzt. Aus der Sicht des Domain-Inhabers ist es wahrscheinlich, dass er einen möglichst zutreffenden Namen für sich selbst aussuchen wird.

Dies ist auch für eine Branche der Fall. Z.B. haben viele Universitäten „uni“ in ihren SLDs; „*uni-muenchen, uni-ulm, uni-dortmund, ...*“. Nachdem die URL in ihre inhaltstragenden Teile segmentiert wurde, wird nach der möglichen Branchenkennzeichnung gesucht.

Diese kann jedoch nicht als ein fester Beweis für eine Kategorie angesehen werden, da etwa mit der Silbe „uni“ auch andere Namen zusammengesetzt werden können, z.B.: `http://www.uni-sex.info/`.

Eine gut segmentierte SLD kann einen starken Hinweis für eine Kategorie liefern. Aber dieser Hinweis muss mit den anderen Merkmalen kombiniert werden, um eine endgültige Entscheidung zu treffen.

2.4.1.3 Textuelle Merkmale

Pierre (2001) hat betont, dass Meta-Informationen gute Indizien für die Klassifikation von Websites liefern können. Auch Golub & Ardö (2005) [56] und Fathi et al. (2004) [45] betrachten die Meta-Daten bei der Webdokumentklassifikation als wichtigen Faktor.

Meta-Daten wie „Keywords“ oder „Beschreibung“ („description“) enthalten oft die relevantesten Informationen zu Domain-Namen. Aus diesen Gründen sind Meta-Ddaten ein gutes Indiz für die Kategorisierung einer Website.

Auch der „Titel“ spielt eine wichtige Rolle bei einem Webauftritt. Oft beinhaltet der Titel einer Website den Namen des Betreibers. Zwischen Domain-Namen und Titel verbirgt sich oft eine Kongruenz. So ist die Relation zwischen dem Titel „BMW Deutschland“ und der Website „`http://www.bmw.de/`“ sichtbar.

„Anchor-Texte“ sind nicht nur bei der binären Klassifikation einer Website wie bei Bsiri (2007), sondern auch in vielen anderen Bereichen, wie Suchmaschinentechniken, gewichtet. Sie sind meistens stichwortartig und daher für den Charakter einer Website besser geeignet. Selbst Amitay et al. (2003) erwähnen, dass die Klassifikation über Hypertexte wie „shopping cart“ für Shopping-Sites bessere Ergebnisse liefern kann.

Insgesamt werden die folgenden textuellen Merkmale für die Naive Bayes'sche Klassifikation verwendet:

- Textuelle Merkmale
  - Titel
  - Meta-Keywords
  - Meta-Beschreibung
  - Meta-Copyright, -Author und -Publisher
  - Anchor-Texte

Bei den textuellen Merkmalen muss eine Stoppwortliste verwendet werden, um eine reine Textmenge zu bilden. Stoppwörter wie „*hier, weiter, home, up, oben, ...*“ sind wenig relevant, um eine Website zu charakterisieren. Aus diesem Grund wurde eine Stoppwortliste aus den Trainingsdaten zusammengestellt.

2.4.2 Algorithmus zur Klassifikation von Websites

Nachdem alle benötigten Merkmale zur Klassifikation zusammengestellt sind, werden die Websites stufenweise klassifiziert. Zuerst werden die strukturellen Merkmale auf die Klassifikation angewandt. Falls eine Website auf Basis der strukturellen Merkmalen nicht zu klassifizieren ist, wird die Naive Bayes'sche Klassifikation angewandt.

Pseudo-Algorithmus
der Website-Klassifikation (SLD, SM, TM, Kategorien)

```
for each Kategorie_i von Kategorien
    if SLD enthält einen hinweisenden Teil für Kategorie_i
      und
      SM (= strukturelle Merkmale) <= Schwellenwert gegen Kategorie_i
      dann weise SLD Kategorie_i zu;
    else SM >= Schwellenwert für Kategorie_i
      dann weise SLD Kategorie_i zu;
    endif
endfor

if SLD ist keine Kategorie_i zugewiesen
  Entferne Stopwörter aus TM (= textuelle Merkmale);
  Wende die Naive Bayes'sche Klassifikation auf TM an;
endif
```

Abbildung 2.2: Algorithmus zur Klassifikation von Websites

2.4.2.1 Grundlegender Algorithmus

Der allgemeine Algorithmus zur Klassifikation von Websites ist in Abbildung 2.2 dargestellt.

Im Algorithmus wird der Schwellenwert mit dem markanten Teil der SLD gegen dieselbe Kategorie berechnet. Denn der hinweisende Teil einer SLD ist ein starkes Indiz für die betroffene Kategorie.

Der Schwellenwert für eine Klasse wurde heuristisch festgelegt.

Die Vorgehensweise bei der Klassifikation der Websites wird in einem Beispiel der Kategorie „Akademie“ erläutert. Diese Kategorie zeichnet sich oft durch eine große Anzahl von Subdomains aus. Insbesondere beinhalten „uni“-Sites den SLD-Teil „uni“. Die Kategorie (z.B. „Uni-Site“) wird aus den strukturellen Merkmalen wie folgt bestimmt:

```
if ein Teil von SLD eq "uni" oder "tu" oder "fh"
   und Anzahl interner Links > Schwellenwert (z.B. 7)
   und Anzahl externer Links / Anzahl interner Links < 4 / 5
   dann klassifiziere die SLD als "Akademie"
else
   if Anzahl interner Links > Schwellenwert (z.B. 7)
       und Anzahl externer Links / Anzahl interner Links < 4 / 5
       und Anzahl von Subdomains > 0
       und Anzahl von Subdomains / Anzahl interner Links > 2 / 5
       und Länge der Meta-Daten > 30)
       dann klassifiziere die SLD als "Akademie"
   endif
endif
```

Im Beispiel der Kategorie „Akademie“ wird ersichtlich, dass die Gewichtung als „Teil der SLD“ sehr hoch ist. Wenn eine SLD einen hinweisenden Teil beinhaltet, wird nur gecheckt, ob die Anzahl interner Links einen Schwellenwert erreicht hat, und die Anzahl interner Links gegenüber der Anzahl externer Links wesentlich größer ist. Ansonsten werden noch die Anzahl der Subdomains und die Länge der Meta-Daten miteinbezogen.

Die Gewichtung der verschiedenen Merkmale variiert je nach Kategorie. Bei Firmen-Websites z.B. werden die Meta-Daten von „*Copyright, Publisher und Author*“ mehr gewichtet als die Meta-Daten von „*Keyword und Description*“.

Es wird für jede Kategorie versucht, die Kategorisierung anhand des hinweisenden Segments aus der SLD und struktureller Merkmalen durchzuführen. Falls hiermit keine Entscheidung getroffen werden kann, wird die „Naive Bayes’sche Klassifikation“ angewandt. Dafür wurden manuell 500 Websites bearbeitet, und die Stoppwörter wurden entfernt.

2.4.2.2 Naive Bayes’sche Klassifikation

Die Naive Bayes’sche Klassifikation ist eine der am meisten verwendeten Klassifikationsmethoden. Sie ist statistisch basiert, einfach zu implementieren und erzielt trotzdem eine hohe Präzision.

Ist ein Testdokument d gegeben, so kann die Wahrscheinlichkeit berechnet

werden, dass dieses Dokument einer bestimmten Kategorie c_j entstammt[12].

$$Pr(C = c_j|d) \tag{2.1}$$

Wir können dann berechnen, welche Klasse c_j die Wahrscheinlichste ist und diese Kategorie dann d zuweisen.

Nehmen wir an, dass A_i, wo $i = 1, 2, ..., n$ ist, jede diskrete Zeichenfolge in den Trainingsdaten D ist, und C die Klasse mit den Werten $c_1, c_2, ..., c_m$ ist. Ist ein Testdokument d mit den beobachten Attributen („Zeichenfolge“ oder „Wort“) A_1 bis A_n gegeben, dann ist d

$$d =< A_1, A_2, ..., A_n > .$$

Die Voraussage ist die Klasse c_j, so dass $Pr(C = c_j|A_i, ..., A_n)$ maximal wird. c_j wird die *maximale a posteriori Hypothese* genannt.

Durch die Bayes'sche Regel wird dann die Formel 2.1 wie folgt umgeschrieben.

$$\begin{aligned} & Pr(C = c_j|A_1, ..., A_n) \\ &= \frac{Pr(A_1, ..., A_n|C = c_j)Pr(C = c_j)}{Pr(A_1, ..., A_n)} \\ &= \frac{Pr(A_1, ..., A_n|C = c_j)Pr(C = c_j)}{\sum_{k=1}^{|C|} Pr(A_1, ..., A_n|C = c_k)Pr(C = c_k)} \end{aligned} \tag{2.2}$$

$Pr(C = c_j)$ ist die *a priori* Wahrscheinlichkeit der Klasse c_j, die aus den Trainingsdaten D berechnet werden kann. Das ist einfach die Dokumentanzahl mit der Klasse c_j in der Trainingsdaten D.

Da wir uns nur für die Klassifikation interessieren, ist $Pr(A_1, ...A_n)$ irrelevant. Denn es ist konstant für jede Klasse. So brauchen wir nur $Pr(A_1 \wedge ... \wedge A_n|C = c_j)$ zu berechnen, was wie folgt umgeschrieben werden kann.

$$\begin{aligned} & Pr(A_1, ..., A_n|C = c_j) \\ &= Pr(A_1|A_2, ..., A_n, C = c_j) * Pr(A_2, ..., A_n|C = c_j) \end{aligned} \tag{2.3}$$

[12]Die Ableitung stützt sich wesentlich auf Liu(2007) [87].

Weiter kann die Formel $Pr(A_2, ..., A_n|C = c_j)$ in $Pr(A_2|A_3, ..., A_n, C = c_j) * Pr(A_3, ..., A_n|C = c_j)$ umgeschrieben werden. Dieser Prozess wird rekursiv angewandt.

Wie das Wort „Naive“ andeutet, wird bei der Naive Bayes'schen Klassifikation angenommen, dass die Auftretenswahrscheinlichkeiten aller Wörter statistisch unabhängig sind. Unter dieser Annahme ist folgende Formel gültig:

$$Pr(A_1|A_2, ..., A_n, C = c_j) = Pr(A_1|C = c_j) \tag{2.4}$$

Diese Bedingung gilt auch für A_2, A_3, usw. So erhalten wir die folgende Formel.

$$Pr(A_1, ..., A_n|C = c_j) = \prod_{i=1}^{|A|} Pr(A_i|C = c_j) \tag{2.5}$$

Zusammenfassend können wir dann Formel 2.2 wie folgt umschreiben

$$Pr(C = c_j|A_1, ..., A_n) = \frac{Pr(C = c_j) \prod_{i=1}^{|A|} Pr(A_i|C = c_j)}{\sum_{k=1}^{|C|} \prod_{i=1}^{|A|} Pr(A_i|C = c_k)}. \tag{2.6}$$

Dann erhalten wir die folgende Formel. c wird der wahrscheinlichsten Klasse zugewiesen.

$$c = \underset{c_j}{\operatorname{argmax}}\, Pr(C = c_j) \prod_{i=1}^{|A|} Pr(A_i|C = c_j) \tag{2.7}$$

Neue Attribute, die nicht in den Trainingsdaten vorgekommen sind, können die Formel 2.7 außer Kraft setzen, weil die Wahrscheinlichkeit eines neuen Attributs 0 ist, und sich dadurch auch 0 als Gesamtwahrscheinlichkeit ergeben wird. Dieses Problem kann durch die Einführung einer normalisierten Form vermieden werden. In der Literatur wird die revidierte Formel der Naiven Bayes'schen Klassifikation eingeführt. Danach wird die Formel $Pr(A_i|C = c_j)$ wie folgt umgeschrieben:

$$Pr(A_i|C = c_j) = \frac{n_{ij} + \lambda}{n_j + \lambda m_i}$$

wobei n_{ij} die Anzahl der Dokumente aus c_j ist, die A_i enthalten, n_j die gesamte Anzahl der Dokumente mit $C = c_j$ und m_i die Häufigkeit des Attributes A_i ist. λ ist ein multiplikativer Faktor und wird oft angesetzt als $\lambda = 1/n$, wobei n die gesamte Anzahl der Dokumente ist.

Bei der Implementierung der Naive Bayes'schen Klassifikation wird dieses Problem durch ein Hash-Verfahren gelöst. Die bei der Hash-Tabelle nicht vorgekommenen Attribute werden einfach ignoriert und nicht mitgezählt.

Nicht vorgekommene Attribute werden sowohl in der Trainingsphase als auch bei der Klassifizierung ignoriert.

Mehrmals vorgekommene Attribute in einem Dokument werden nur einmal gezählt. In Web existieren Dokumente, die die gleichen Wörter mehrfach als Meta-Keywords oder in die Meta-Beschreibung einbringen. Falls die mehrmals auftretetenden Wörter mehrfach gezählt werden, dann wird die Wahrscheinlichkeit für eine bestimmte Klasse erhöht.

Die Grundannahme der statistischen Unabhängigkeit des Auftretens von Wörtern in einem Dokument kann die Qualität der Klassifikation abwerten, weil viele Wörter in einem Dokument mehr oder weniger relevant sind. Jedoch ist die Überprüfung der Relevanz von Wörtern nicht einfach. Es muss dafür Distribution über die Wörter berechnet werden, z.B. mit der Poisson-Überschätzung. Dies würde die Implementierung bei der Naive Bayes'schen Klassifikation erschweren.

Die Reihenfolge der Wörter in einem Dokument wird nicht berücksichtigt. Das bedeutet, dass die Wahrscheinlichkeit eines Wortes unabhängig von dessen Position im Dokument berechnet wird. Wenn man die Reihenfolge der Wörter in einem Dokument mitzählen würde, wird die Aufgabe sehr kompliziert. Das Ignorieren der Reihenfolge entspricht auch den gewählten Attributklassen der textuellen Merkmale.

N-Gramme werden bei vielen Web-Minings-Aufgaben genutzt. N-Gramme können auch auf die Naive Bayes'sche Klassifikation angewandt werden. Die ausgewählten textuellen Merkmale sind jedoch weniger fortlaufende Texte,

sondern bestehen großenteils aus Stichworten und Auflistungen. Lediglich einige Anchor-Texte wie „*Über uns*“ oder „*Rechtliche Hinweise*“ und Meta-Beschreibungen können eine Rolle spielen. Trotz dieser sinnvollen N-Gramm-Ausdrücke kann die Bildung von N-Grammen aus allen textuellen Merkmalen sehr viele irrsinnige Ausdrücke produzieren. Deshalb wird die N-Gramm-Strategie hier nicht weiter verfolgt.

Morphologische Merkmale wie Plural oder Kasus spielen bei den gewählten textuellen Merkmalen nur eine untergeordnete Rolle. Eine mögliche Optimierung wäre die Normalisierung morphologischer Varianten wie „*Kontakt*“ und „*Kontakte*“, die als Anchor-Texte gleichbedeutend sind. Die Normalisierung solcher Variationen können die Wahrscheinlichkeit für eine Klasse erhöhen. Diese Arbeit wurde nicht gemacht, weil dafür linguistische Mittel benötigt werden.

2.4.3 Experimentelle Evaluation

Für die Evaluation des Klassifikationssytems wurde eine Testdatei von 924 URLs genommen. Davon waren 313 URLs, die keine ccTLD „de“ aufwiesen, ausgenommen. Von den übrigen 611 URLs wurde jede 6. URL, insgesamt 102, für den Test des Systems übernommen. Jede Website wurde dann manuell mit dem „Firefox“ besucht.

2.4.3.1 Klassifikation mit strukturellen Merkmalen

Zuerst wurde die Klassifikation auf Basis von ausschließlich strukturellen Merkmalen getestet. 42 von 102 URLs wurden keiner Kategorie zugeordnet. So blieben 60 URLs zur Evaluation übrig.

Drei URLs waren „nicht aktiv“ oder eine Domain-Parking-Seite[13]. Sie wurden vom Klassifikator nur auf Basis struktureller Merkmalen erkannt, welche zusätzlich gewichtbar waren. Ein Beispiel für die strukturellen Merkmale ist unten aufgelistet. Von links her sind sie: *Länge der SLD, Länge des Titels, Länge der Meta-Keywords, Länge der Meta-Beschreibung, Länge von*

[13]Die Kategorie „Nicht aktiv“ ist nicht in der Kategorienliste aufgeführt. Sie wurde aber ausschließlich aus den strukturellen Merkmalen heraus berechnet.

Meta-Copyright, Länge von Meta-Author, Länge von Meta-Publisher, Anzahl interner Links, Anzahl externer Links, Länge des Texts, Länge interner Anchor-Texte, Länge externe Anchor-Texte, Anzahl von Img-Tags und Anzahl von Subdomains.

`http://www.heltech.de`
7, 39, 0, 0, 0, 0, 0, 1, 0, 10, 10, 0, 1, 0

Wie das Beipiel zeigt, gibt es keine Meta-Daten für eine nicht aktive Website. Es gibt nur einen internen Link. Die Website ist zur Zeit nur für eine Internet-Präsenz reserviert.

Eine Website wurde falsch als „Privat-Site“ klassifiziert. Die strukturellen Merkmale sind unten aufgelistet.

`http://www.doctorand.de`
9, 9, 0, 0, 0, 0, 0, 0, 1, 149, 0, 37, 1, 0

Von den Merkmalen her ist die Site eine typische Privat-Site. Sie hat keine Meta-Daten, nur einen externen Link und einen Image-Tag. Selbst bei manueller Inspektion ist sie nicht leicht zu klassifizieren. Sie könnte entweder als „Informations-Site“ oder „Forum“, d.h. „Privat-Site“, klassifiziert werden.

Zwei Websites aus „Organisation“ stammen von politischen Parteien. Sie werden hauptsächlich durch die SLD-Analyse klassifiziert. Ein Beispiel:

`http://www.cdu-walluf.de`
10, 38, 43, 60, 26, 12, 26, 3, 0, 48, 26, 0, 0, 0

Während des Experiments ergab sich, dass „Organisations-Sites“ viele Merkmale mit „Firmen-Websites“ teilen. Sie haben Meta-Daten, gewisse interne Links und Anchor-Texte. Ohne die Analyse der SLD ist die Entscheidung für „Organisation“ sehr schwer gefallen. Zur Analyse der SLD für die deutsche „Organisation“ wurden bislang 15 typische Organisationsbezeichnungen aufgenommen. Es ist aber zu bemerken, dass aufgrund der Analyse der SLD keine Entscheidung für die Klasse getroffen werden kann. Es müssen gewisse strukturelle Merkmale für eine Klasse berechnet werden, um sie als solche klassifizieren zu können.

Fünf Websites wurden als „Informations-Site“ klassifiziert. Eine davon war falsch. Ihre Merkmale sind unten angegeben.

`http://www.schaper-apartment.de`
17, 58, 1954, 309, 0, 17, 17, 74, 2, 8624, 2163, 48, 2, 0

Die Merkmale sind typisch für eine Informations-Site: Viele Meta-Keywords, ein langer Body-Text, sehr viele interne Links und sehr viele Anchor-Texte. Bei manueller Inspektion ergab sich, dass die Site eine Sammelangebots-Site für Apartment-Hotels ist.

Zwei Websites wurden als „Shopping-Site“ klassifziert, davon war eine falsch. Die Merkmale sind unten zu sehen.

`http://www.atv-touren.de`
10, 14, 0, 0, 0, 0, 0, 69, 0, 4890, 1075, 0, 231, 0

Wie die Daten zeigen, hat die Site viele interne Links und Anchortexte. Außerdem hat die Anzahl der Image-Tags bei der Entscheidung eine große Rolle gespielt, so dass die Site eher eine Informations-Site ist.

Alle andere 47 Sites wurden als „Firmen-Site“ klassifiziert. Davon waren vier Sites falsch klassifiziert. Drei Websites sind eine „Organisation“ und eine Site eine „Informations-Site“. Wie schon erwähnt, ist die Entscheidung für „Organisation“ nicht einfach, weil diese Sites oft fast die gleichen strukturellen Merkmale wie „Firmen-Websites“ aufweisen. Auch die Merkmale der Informations-Site waren sehr ähnlich zu jenen der „Firmen-Site“. Hierfür werden drei Beispiele angegeben. Das erste ist eine Firmen-Site, das zweite ist vom Typ „Organisation“ und das letzte ist „Information“.

`http://www.abbruch-hipp.de`
12, 54, 581, 44, 12, 18, 24, 4, 0, 155, 27, 0, 0, 0

`http://www.astronomie-in-berlin.de`
20, 23, 691, 54, 13, 13, 13, 5, 0, 96, 34, 0, 0, 0

`http://www.zentrum-pfaelzerwald.de`
20, 115, 236, 195, 0, 37, 37, 4, 0, 562, 27, 0, 0, 0

Wie diese Beispiele zeigen, weisen alle drei Sites fast die gleichen strukturellen Merkmale auf: Sie haben alle Meta-Daten, gewisse Interne Links und Anchor-Texte. So wurden sie alle als „Firmen-Website“ klassifiziert.

Nachdem alle Fälle der Klassifikation durch die strukturellen Merkmale veranschaulicht wurden, muss eingeräumt werden, dass die Trainingsdaten nicht objektiv gewählt wurden. Dies liegt daran, dass viele zugängliche Websites offizielle Websites sind. Daher wurden nur wenige Privat-Websites aufgenommen. Dadurch wurde die strukturelle Evaluation der Privat-Sites nicht gerecht durchgeführt. Dies gilt auch für die „Gesundheits-Sites“.

Die Evaluation des Klassifikationssystems mit strukurellen Merkmalen wird durch Präzision und Recall durchgeführt. Es muss angenommen werden, dass der Recall nicht als entscheidender Faktor für das System bewertet werden soll. Eher ist die Präzision ein Entscheidungsfaktor. Der Sinn der strukturellen Merkmale liegt für das System darin, schnell eine hohe Präzision zu erzielen.

Von allen Websites waren sechs Sites falsch. Das sind 10%. Die Präzision liegt damit bei 90%. Der Recall liegt dann bei $100 * 54/102 = 52,94\%$.

2.4.3.2 Klassifikation mit textuellen Merkmalen

Das System mit den strukturellen Merkmalen klassifizierte 42 Websites nicht. Diese wurden aber durch die „Naive Bayes'sche Klassifikation“ klassifiziert.

Zwei Websites wurden korrekt als „Organisation“ klassifiziert.

Sechs Websites wurden als „Information“ klassifiziert. Davon waren drei Websites ein Gemeindeverein. Die falsche Klassifizierung ist auf die Gemeindeinformationen zurückzuführen. Eine Website war eine „Firmen-Website“. Sie hat auf der Startseite sehr wenige Meta-Daten und Anchor-Text und war die Site für ein Musikgeschäft.

Alle anderen Websites wurden als „Firmen-Site“ klassifiziert. Davon waren drei Websites falsch klassifiziert. Zwei Sites waren eine „Organisation“ und eine war eine „Information“.

Somit liegt die Präzision der „Naive Bayes's Klassifikation“ bei $100 * 36/42 = 85\%$.

2.4.3.3 Gesamte Bewertung des Systems

Insgesamt wird das System anhand 102 Websites durch die Präzision bewertet, wie es bei der Klassifizierung üblich ist. Die Präzision liegt bei $100 * (54 + 36)/102 = 88,23\%$.

Das System hat gezeigt, dass eine Einstiegsseite einer Website alle möglichen Merkmale für die Website-Klassifikation liefern kann.

Zum Schluss muss allerdings eingeräumt werden, dass das System aufgrund der gewählten Trainings- und Test-Daten, die überwiegend aus Firmen-Websites bestehen, nicht uneingeschränkt auf die andere Test-Daten übertragen werden kann.

Aber gerade diese Tatsache lässt auch vermuten, dass das System mit erweiterten Trainingsdaten besser übertragbar sein wird. Insbesondere liefern die strukturellen Merkmale auf der Homepage einer Site wertvolle Indizien für die Charakterisierung einer Website.

Kapitel 3

Firmen-Homepages

Eine Firmen-Homepage zeichnet sich durch syntaktische und semantische Besonderheiten aus. Eine gut strukturierte Firmen-Homepage verfügt über einen erkennbaren Aufbaustil und spezielle semantische Eigenschaften.

Websites und -seiten können von verschiedenen Personen oder Gruppen erstellt werden. Danach kann jede Webseite verschieden organisiert werden. Der Zweck des Webauftritts ist jedoch die Präsentaion über sich selbst. Das ist besonders bei einer Firmenwebseite der Fall.

Eine Firma, die sich für einen Webauftritt entschieden hat, überlegt sich auch, wie sie sich selbst am besten präsentieren kann. Diese Überlegungen haben dazu beigetragen, dass eine Firmen-Homepage strukturell und fachsemantisch weitgehend ähnlich gestaltet wird.

In der Regel bekommt eine Firmen-Homepage einen besonderen Status, da sie die Einstiegsseite eines Webauftritts ist. Deshalb müssen die wesentlichen Informationen und weiterführende Hinweise auf alle weiteren Webseiten sichtbar gemacht werden, um den Besuchern einen möglichst guten Eindruck vermitteln zu können.

Des Weiteren bringen viele Firmen Meta-Informationen wie Meta-Keywords oder Meta-Beschreibungen häufiger in die Homepage als in die anderen Webseiten wie Kontakt- oder Impressumseite ein. Die Statistik in Abbildung 3.1 zeigt die allgemeine Verteilung der Meta-Daten, die in Firmen-Homepages zu finden sind.

Die Informationen von den Meta-Daten in Abbildung 3.1 sind insbesondere für die Website-Klassifikation und Firmennamenerkennung relevant. Meta-

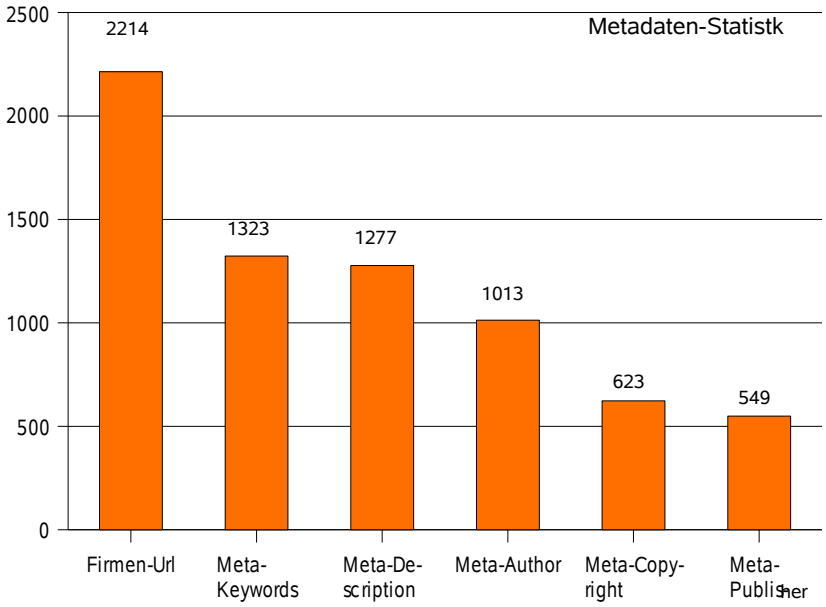


Abbildung 3.1: Statistik zur Verteilung der Metadaten auf Homepages

Keywords und -Beschreibungen werden bei der Naiven Bayes'schen Klassifikation genutzt, während „Meta-Copyright, -Author, -Publisher“ für die Erkennung des Firmennamens relevant sind.

Abgesehen von Werbe- und firmenexternen Texten wie Pressemitteilungen lassen sich die Daten auf einer Firmen-Homepage in vier Abschnitte gruppieren.

- Daten auf Firmen-Homepage
  - **Titel**
  - **Meta-Informationen**: Schlüsselwörter, Beschreibung, Urheberrecht, Autor, Herausgeber, ...
  - **Anchor-Texte**: Firmenprofil, Produkte, Stellenangebote, Kontaktinformation, Rechtliche Hinweise, Referenzen, Kundenservice, ...
  - **Body-Text**: Willkommenstexte, Firmenprofil, ...

3.1 Navigationsmöglichkeiten

Von den Daten auf einer Firmen-Homepage haben die Anchor-Texte eine navigierende Funktion. Über sie kann ein Benutzer leicht auf den verschiedenen Webseiten der Firma stöbern.

Zum größten Teil sind diese Anchor-Texte weitgehend als Fachvokabular etabliert. Ein Computer kann von diesem Fachvokabular profitieren, um automatisch auf die gewünschte Seite zu gelangen. Besonders in Bezug auf die Informationsseiten einer Firmenwebsite ist dies von großem Nutzen, da diese durch die bekannten Anchor-Texte schnell gefunden werden.

3.1.1 Allgemeine Bereiche einer Firmen-Homepage

Eine Homepage ist anders aufgebaut als die Startseite[1] einer Website. Eine Homepage ist die Einstiegsseite, auf der man eine Übersicht über die ganzen Webauftritt bekommen kann. Sie ist kompakt und übersichtlich.

Abbildung 3.2 stellt eine typische Firmen-Homepage dar. Im Bereich **A**, dem Navigationsbalken, sind verschiedene verlinkte Seiten zu finden: *Firmenprofil, Produkte, Referenzen, Kontaktmöglichkeit.* Im Bereich **B** sind die verlinkten Seiten für *Öffentliche, Kundenorientierte und Rechtliche* angeordnet und in der Mitte steht ein kurzer Profiltext. Im Bereich **C** ist die Firmenadresse mit der Telefon- und Faxnummer zu sehen. Ganz oben sind Firmenname und -logo platziert.

Die Positionen der verschiedenen Bereiche und der konkrete Gestaltungsstil können je nach Web-Designer oder nach Branche unterschiedlich sein. Eins ist jedoch auf allen Homepages gleich: Sie verfügen über eine allgemeine Navigationsmöglichkeit, welche hauptsächlich durch Anchor-Texte[2] gekennzeichnet ist.

[1] Es gibt Startseiten, die nur Intro-Animationen oder Bilder zeigen, man aber keine Übersicht über den Webauftritt bekommen kann.

[2] Hier wurde nicht berücksichtigt, dass Anchor-Texte nur durch Images oder Flash-Animation realisiert werden können, was die automatische Navigation erheblich erschwert.

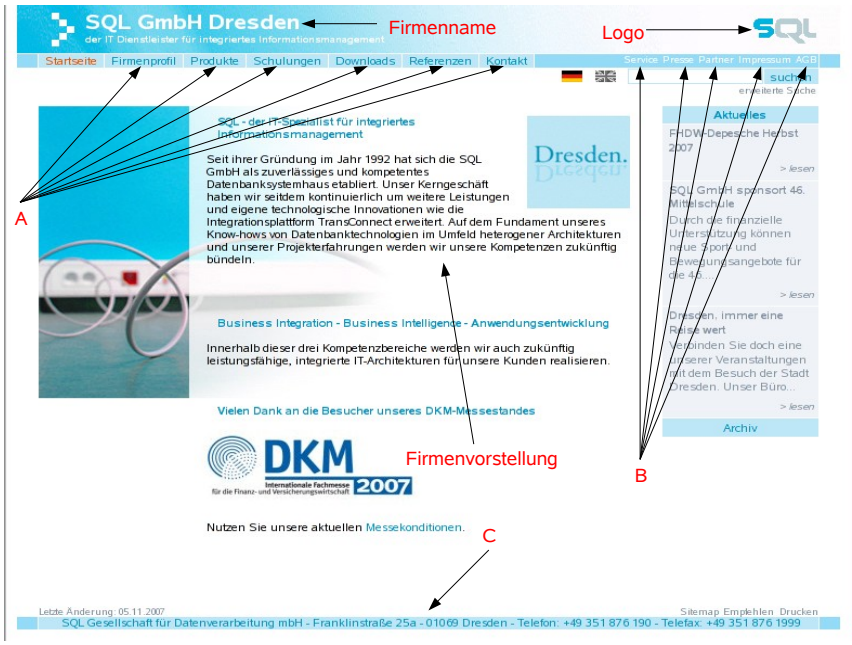


Abbildung 3.2: Beispiel der SQL GmbH Dresden

3.1.2 Anchor-Texte und Navigationsmöglichkeiten

Aus Abbildung 3.2 wird ersichtlich, dass Anchor-Texte eine gute Navigationsmöglichkeit bieten, da sie stichwortartig und wegweisend sind. Diese Charakteristiken wurden bei vielen Webaufgaben genutzt.

Cohen (2003) [28] stellte fest, dass ein Klassifikationssystem mithilfe von Anchor-Texten und Linkanalysen verbessert werden kann. Auch Fujie et al. (2005) [50] nutzen Anchor-Texte für navigationelle Web-Retrieval-Aufgaben aus. Dabei haben sie Synonyme aus Anchor-Texten gebildet und für die Query-Expansion verwendet. Eiron & McCurley (2003) [41] konnten durch die Analyse der Anchor-Texte die Qualität der Websuche verbessern.

Anchor-Texte verhalten sich wie Anfragen und Titel. Durch die Analyse der Anchor-Texte – Verfolgung der Links, Erkennen von Zusammenhängen – wird ein gutes Suchergebnis erzielt. Craswell et al. (2001) [30] fanden heraus, dass Ranking mit Anchor-Texten zwei Mal besser als Ranking basierend auf Dokumentinhalten abschneidet, wenn es darum geht, eine spezifische Website zu finden. All diese Ansätze haben die Effektivität von Anchor-Texten bei einer Vielzahl von Webaufgaben gezeigt.

Diesbezüglich fungiert eine Firmen-Homepage als eine Hubseite[3], die eine Menge an navigierenden Links enthält. Hierbei werden Anchor-Texte auf der Homepage als Quelltexte gesehen, die zu den relevanten Dokumenten führen.

Typische Anchor-Texte auf einer Firmen-Homepage sind, semantisch klassifiziert, die folgenden[4]:

- Klassen von Anchortexten auf einer Firmen-Homepage
  - **Firmenprofil**: *Über uns, About us, Unternehmen, Wir über uns, ...*
  - **Stellenangebot**: *Stellenangebot, Stellen- & Jobangebot, Karriere, Jobs, ...*
  - **Produkt**: *Produkte, ...*
  - **Kundenservice**: *Service, Kundenservice, ...*
  - **Adresse und Kontakt**: *Kontakt, Sie finden uns, Adresse, Wo wir sind, ...*
  - **Rechtliches**: *Impressum, AGB, Rechtliche Hinweise, ...*

Von Homepage zu Homepage können die Anchor-Texte variieren. Dies gilt auch für Branchen. Eine Rechtsanwaltskanzlei hat z.B. kaum einen Anchor-Text für die Klasse „**Produkt**“, während eine Maschinenfabrik oder Software-Firma sicher einen Anchor-Text für die Klasse „**Produkt**“ haben wird.

Von den Klassen der Anchor-Texte sind das „**Stellenangebot**“ und der „**Kundenservice**“ an die Bedürfnisse der Bewerber und Kunden orientiert, während die Klasse „**Produkt**“ zur Vorstellung und Werbung für eigene Produkte genutzt wird.

Die anderen drei Klassen haben mehr oder weniger direkt mit der firmeninternen Informationen zu tun, obwohl die Klasse „**Kontakt**“ hauptsächlich für Kunden und Interessierte gepflegt wird.

Die Profilseite stellt die Firma vor: Dazu gehören Gründungs- und Geschäftsgeschichte, Ambition usw.

[3]Chakrabarti et al. (1999, S. 547) [21]: „pages containing large number of relevant resource links, called *hubs*“.

[4]Die hier aufgelisteten Klassen sind nicht vollständig. Man kann auch Investorrelation oder Management als wichtige Klassen aufnehmen, wenn man sich für M & A interessiert.

Bei der Klasse „**Rechtliches**“ handelt es sich um rechtliche Angaben: *AGB, Impressum usw.* Insbesondere hat sich der Anchor-Text „*Impressum*“ für die dem Gesetz entsprechende Angabe durchgesetzt, und hat sich als Fachvokabur etabliert[5].

Von diesen sechs Klassen interessieren wir uns besonders für die Informationsklassen, in denen die firmenrelevanten Informationen wie *Adresse, Telefon- & Faxnummer, USt-IdNr. usw.* zu finden sind[6].

3.2 Informationsseiten einer Firmen-Website

Firmenrelevante Informationen wie Adresse und Kontaktperson können auf der verschiedenen Seiten vorkommen. Die Einstiegsseite ist eine der geeignetsten Seiten, grundlegende Informationen wie Adresse und Telefonnummer darzustellen, was in Abbildung 3.2 zu sehen ist.

Als Informationsseiten lassen sich die folgenden vier Seiten einschließlich der Einstiegsseite zusammenfassen.

- Informationsseiten
    1. Einstiegsseite
    2. Profilseite
    3. Kontaktseite
    4. Impressumseite

Die Abbildung 3.3 zeigt die verschiedenen Informationsseiten einer Firmen-Website[7].

In Abbildung 3.3 ist der unterschiedliche Charakter der Seiten zu sehen: Die Profilseite hat sich nur auf das Firmenprofil wie Gründungsgeschichte, Partnerschaft und Ambition konzentriert. Kontaktdaten sind über jede Seite auf dem unteren Adressenbalken eingebracht. Die Kontaktseite hat nur Fenster, in die der Besucher eine Anfrage oder Kommentar an die Firma mit seiner

[5] Für die Statistik, s. Abschnitt 5.2 des Kapitels 5.
[6] Für die Extraktion firmenrelevanter Information, s. Kap. 5, Abschnitt 5.7.
[7] Die Kontaktseite wird aufgrund des begrenzten Platzes nicht vollständig abgebildet.

Abbildung 3.3: Beispiel: Informationsseiten einer Firma

Kontaktinformation eingeben kann. Die Impressumseite hat viele verschiedene Informationen über *Kontaktdaten, Geschäftsführer, USt-IdNr. etc.*

Wie die Abbildungen 3.2 und 3.3 andeuten, wird diese Eigenschaft der jeweiligen Informationseite bei der Link-Verfolgung genutzt. Zuerst wird nach der Impressumseite gesucht, da sie die meisten Informationen bietet. Die Informationen werden danach auf der Kontakt-, Profil- und Einstiegsseite gesucht. Die genaue Statistik bezüglich der Informationsseiten ist in Abschnitt 5.2 des Kapitels 5 angegeben.

3.2.1 Einstiegsseite

Die Einstiegsseite fungiert als „Hubseite“. Sie enthält viele Links, über die die nächsten Seiten besucht werden können, und stellt die Firma vor. Was die Informationen betrifft, enthält sie oft die grundlegende Adresse mit Kontaktdaten wie Telefon- & Faxnummer.

Des Weiteren kommt oft die Öffnungs- und Bürozeiten vor. In den Trainingsdaten hatten ca. 5% der untersuchten Websites die Öffnungs- und Bürozeiten angegeben, wovon die meisten auf der Einstiegsseite zu finden waren. Aus diesem Grund werden die Öffnungszeiten primär auf der Einstiegsseite gesucht.

3.2.2 Profilseite

Die Profilseite hat am meisten mit der Branchenidentifikation zu tun. Da sie die Gründungsgeschichte, Tätigkeit, Ambitionen etc. beschreibt, ist sie gut geeignet, um die Branche zu erraten, was aber in dieser Arbeit nicht weiter verfolgt wird. Sie enthält hierfür oft die grundlegenden Informationen wie Adresse und Kontaktdaten.

3.2.3 Kontaktseite

Sie enthält im Normalfall viele firmenrelevante Informationen. Dies ist besonders dann der Fall, falls keine Impressumseite gegeben ist. Einige Firmen geben die Kontaktdaten von Telefon- & Faxnummer und Emailadresse getrennt an. Wenn dies der Fall ist, sind sie meistens auf der Kontaktseite zu finden. Die Kontaktseite kann durchaus nur Kontaktformulare aus Fenstern beinhalten.

3.2.4 Impressumseite

Wie die Abbildung 3.3 verdeutlicht, kommen die meisten Informationen auf der Impressumseite vor. Sie enthält die Adresse mit Kontaktdaten, Anbieter der Site, Geschäftsführer, Verantwortliche(r), Steuer- & USt-IdNr. etc. Hier sind auch verschiedene Äußerungen zu finden, die für den Infomationssuchenden wenig relevant sind.

So enthält sie oft Informationen über Web-Designer bzw. -programmierer. Dieser Bereich sollte aber bei der Informationssuche nicht eingeschlossen werden, sonst werden zwei verschiedene Informationen kombiniert. Für diesen Zweck wurden bei der Trainingsphase insgesamt 78 webdesignerrelevante Überschriften zusammengestellt. Einige davon sind unten aufgelistet.

Design & Programmierung	*Gesamtkonzeption, Design, Realisierung*
Design & technische Umsetzung	*Design, Realisierung und Administration*
Design & Umsetzung	*Gestaltung & barrierefreie Umsetzung*
Gestaltung und Programmierung	*Konzept, Design und Realisierung*
Layout and Graphics	*Konzept, Gestaltung und Produktion der Website*
Konzept, Layout und Webdesign	*Konzept, Design, Layout & Programmierung*
Layout und Gestaltung	*Layout, Design & Programmierung*
Programmierung + Design	*Programming and Realization*
Technische Realisierung	*Verantwortlich für Gestaltung & Realisation*
Technische Umsetzung	*Verantwortlich für Technik und Design*
Webdesign & Programmierung	*Webdesign, Grafik und Navigation*

Kapitel 4

Das System: ACIET

Das hierfür entwickelte System *ACIET* (Automatic Companies Information Extraction Tool) besteht aus drei Teilen (**A, B, C** siehe Abb. 4.1).

Für eine gegebene URL holt der Crawler die Homepage und ermittelt ihre Merkmale, welche dem Klassifikator übergeben werden, welcher sie den entsprechenden Kategorien zuweist (Systemblock **A**, siehe 4.1).

Wenn die Website als Unternehmen klassifiziert wurde, wird sie an die zweite Systemkomponente, d.h. den Info-Seiten-Analysator übergeben, welcher bestimmte Informationen extrahiert und diese in einer Firmendatenbank ablegt (Systemblock **B**, siehe 4.1).

Der dritte Teil wurde als Kontrollinstanz der extrahierten Daten konzipiert. So können die Informationsseite und extrahierten Informationen vom Anwender über eine benutzerfreundliche Oberfläche überwacht werden, und eventuell deren Daten noch korrigiert werden. Dabei neu gefundene Attribute können in die entsprechende Kontextdatei gespeichert werden. Dadurch werden die Kontextdaten immer vollständiger (Systemblock **C**, siehe 4.1).

4.1 Systemübersicht

Eine Systemübersicht wird in Abbildung 4.1 dargeboten.

Das System *ACIET* ist modular aufgebaut. Dadurch kann das System leicht modifiziert und erweitert werden. Z.B. werden alle internen und externen

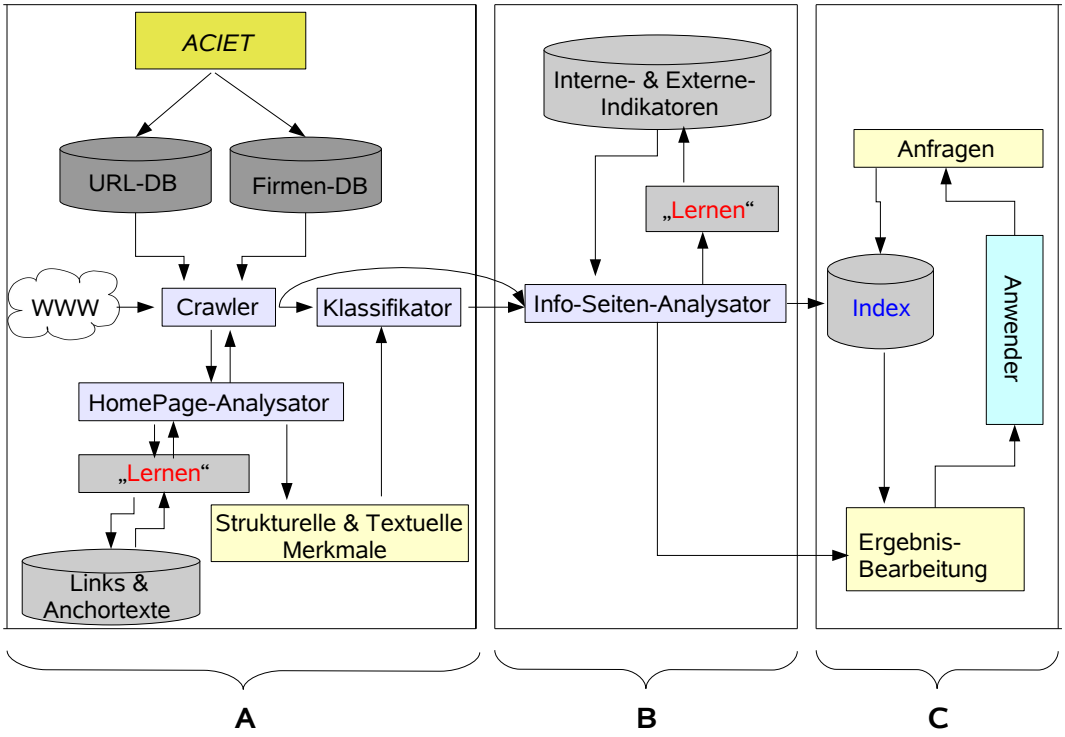


Abbildung 4.1: Systemübersicht des *ACIET*

Indikatorklassen getrennt verwaltet. Auf diese Weise kann man schnell einen Eintrag hinzufügen oder entfernen.

ACIET besteht hauptsächlich aus vier Komponenten. Jede Komponente kann einzeln korrigiert und erweitert werden.

- Komponenten von *ACIET*
  1. Crawler
  2. Klassifikator
  3. Info-Seite-Analysator
  4. Post-Processing

4.1.1 Crawler

Der eingesetzte Crawler ist eine Art fokussierter Crawler[1], der gezielt die Links verfolgt und dadurch die Bandbreite und Speicherbedarf erheblich reduziert.

Die gegebene URL wird auf der URL-Datenbank abgeglichen. Falls diese als neu eingestuft wird, dann sucht der Crawler zuerst nach der Einstiegsseite der Webpräsenz. Während bei den meisten Webauftritten die Startseite die Einstiegsseite (Homepage) ist, kann es bei einigen Webauftritten vorkommen, dass die Startseite nur eine Intro-Animation enthält. In solchen Fällen kann die Einstiegsseite über Links erreicht werden.

Nachdem die Einstiegsseite gefunden wurde, werden Merkmale für die Website-Klassifikation und Info-Seiten-Analyse extrahiert. Für Einstiegsseiten mit Frames werden die Frame-Sourcen geholt, um ihre Merkmale vollständig extrahieren zu können.

Diese Merkmale werden an den Klassifikator übergeben, welcher die Website mit den vorgegebenen strukturellen und textuellen Merkmalen kategorisiert, wie es in Kapitel 2 beschrieben wurde.

Nur für die Kategorie „Firmen“ wird nach der Informationsseite gesucht, und die gefundene Seite wird an den Info-Seiten-Analysator weitergegeben.

Die URLs in der Firmendaten-DB werden regelmäßig gecrawlt, und so auf ihre Aktivität und Aktualität hin geprüft.

[1] Begriffserklärung, siehe Abschnitt 5.2 des Kapitels 5.

4.1.2 Klassifikator

Der Klassifikator ist ausführlich im Kapitel 2 beschrieben. Die Klassifikation erfolgt über die Analyse von strukturellen und textuellen Merkmalen. Jedes strukturelle Merkmal ist nach Kategorien gewichtet. Auf die textuellen Merkmale wird die „Naive Bayes'sche Klassifikation“ angewandt.

4.1.3 Info-Seiten-Analysator

Der Info-Seiten-Annalysator, der Kern des Systems, vorverarbeitet die HTML-Seite, bildet die Baumstruktur, bestimmt den Infomationsbereich, wendet das Attribut-Wert-Verfahren an, extrahiert die Informationen und speichert sie in die Firmendatenbank.

Er ist in Kapitel 5 beschrieben.

4.1.4 Post-Processing

Die extrahierten Informationen können vom Anwender überprüft werden. Dafür wurde ein CGI-Programm geschrieben, bei dem der Anwender URLs eingeben und das Ergebnis des Systems kontrollieren kann. Er kann fehlerhafte Ergebnisse korrigieren, und neu gefundene Attribute in die entsprechende Kontextdateien speichern. Auf diese Weise werden die Firmen- und Kontextdaten immer vollständiger.

4.2 Programmiersprache: PERL

Das hier vorgestellte System wurde ausschließlich mit *PERL* und *UnixTools* geschrieben. Um ein effektives System zu entwickeln, müssen verschiedene Faktoren bedacht werden. Unter anderem gehören dazu die folgenden:

- Systemfaktoren
  - *Effizienz*
  - *Portabilität*
  - *Robustheit*

- *Skalierbarkeit*
- *Verständlichkeit*
- *Kompaktheit*

Effizienz bezieht sich hier auf zwei Ebenen: Systemaufbau und Systemanwendung. Bei der Aufbauphase des Systems muss kein großer Zeitaufwand in Anspruch genommen werden. In Neumanns Experiment[2] zeigt sich z.B., dass PERL am effektivsten aufzubauen ist. Es können auch leicht die vorhandenen Unix-Tools eingebunden werden, da PERL als eine der integrierbaren Sprachen entwickelt wurde.

Des Weiteren verfügt das Open-Source orientierte PERL über sehr große Ressourcen, die leicht verständlich sind, und dadurch ohne große Schwierigkeiten angewandt werden können.

PERL ist eigentlich für die Systemverwaltung entwickelt. Dies bezieht sich hauptsächlich auf Unix. Trotz dieser Geschichte ist PERL mit wenigen Ausnahmen plattformunabhängig. Mit standardisierten Codes ist ein PERL-Programm plattformunabhängig interpretierbar und lauffähig. Damit verfügt PERL über eine weite „Portabilität“.

Ein System muss über Dateiverwaltung *skalierbar* sein. PERL als eine integrierte Sprache kann sehr große Datenmengen verwalten, insbesondere wenn sie mit Unix-Tools kombiniert wird. Dies zeigt Neumanns Experiment in den Abbildungen 4.2 und 4.3[3]. Wie dort zu sehen ist, ist PERL mit Unix-Tools ein mächtiges Werkzeug und sehr effizient bezüglich der Rechenzeit und des Speicherbedarfs. Im Vergleich mit Kompilier- und Interpretationssprachen schneidet PERL zusammen mit den Unix-Tools sehr gut ab.

PERL-Code ist leicht zu verstehen. Durch die unterschiedlich eingebauten Variablentypen macht die Interpretation eines PERL-Programms keine Schwierigkeiten.

Durch viele eingebauten Funktionen ist ein PERL-Programm „kompakt“ zu schreiben. Durch PERLs Objekt-Orientiertheit lässt sich PERL-Code immer wieder verwenden.

[2] `http://www.cis.uni-muenchen.de/~andi/CL_cookbook/themen/frq/auswertung.html`.

[3] Details und Programm-Codes der verschiedenen Sprachen, die in die Auswertung miteinbezogen wurden, sind auf seiner Homepage einzusehen: `http://www.cis.uni-muenchen.de/~andi/CL_cookbook/themen/frq/auswertung.html`.

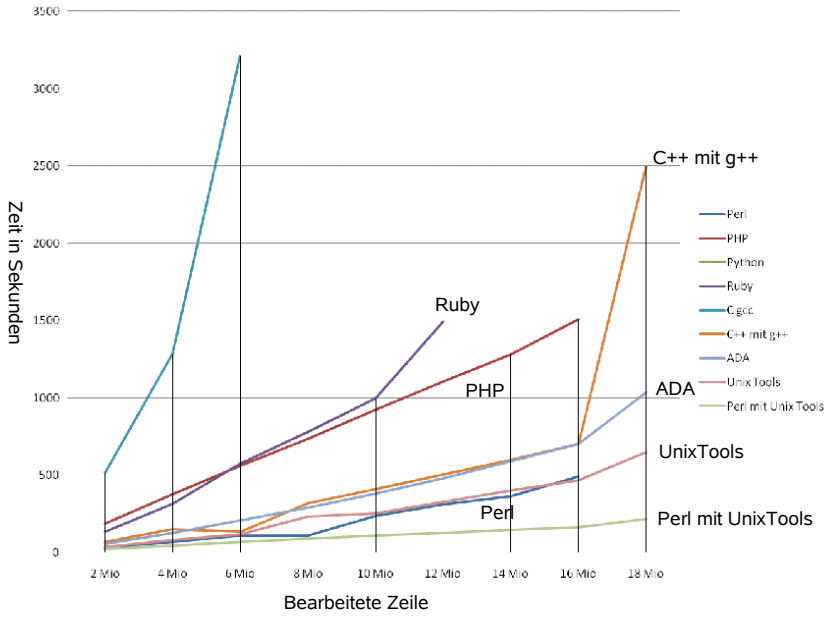


Abbildung 4.2: Zeitvergleich der verschiedenen Programmiersprachen

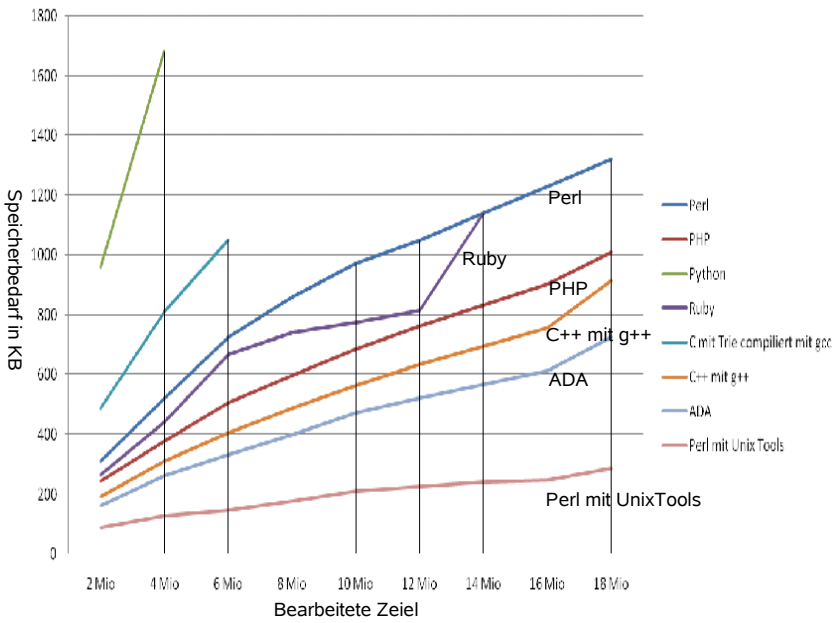


Abbildung 4.3: Speicherbedarf der verschiedenen Programmiersprachen

All diese Vorteile stellen nur wenige Programmiersprachen zur Verfügung.

Des Weiteren verfügt PERL über starke reguläre Ausdrücke. PERLs reguläre Ausdrücke verwenden NFAs (Non-Deterministic Finite-State Automata). Dadurch ist die Rechenzeit etwas länger als bei DFAs (Deterministic Finite-State Automata). Dafür sind aber die Konstruktionen und Anwendungsmöglichkeiten viel flexibler und vielfältiger.

Kapitel 5

Die Extraktionsmethode

In diesem Kapitel wird die Extraktionsmethode beschrieben. Aus den gegebenen URLs werden domainnamenrelevante Informationen extrahiert und in einer Datenbank gespeichert.

Für das Extrahieren von Dateneinheiten aus einer (beliebigen) Website wurden verschiedene Techniken entwickelt, welche je nach Textart und Aufgabe zu differenzieren sind. Für „Plain"-Texte, wie z.B. Nachrichtenartikel, können textbasierte Extraktionstechniken verwendet werden, während für Datensätze, die aus einer Datenbank generiert wurden, eher DOM[1]-basierte Techniken geeignet sind.

Da eine Webseite im Normalfall nicht vollständig auf Text, wie es bei Printmedien der Fall ist, oder auf der Struktur, wie bei XML, basiert, wird sie als semi-strukturiert betrachtet. Die Strukturiertheit eines Dokuments wird nach Hsu und Dung (1998) [70] wie folgt definiert[2]:

- Definition: Strukturiertheit
  - Eine Webseite ist *strukturiert*, falls jedes Attribut korrekt in ein Tupel extrahiert werden kann. Dabei geben einheitliche syntaktische Strukturen (z.B. Delimiter, Reihenfolge der Attribute) darüber Aufschluss, wie die Attribute ausgewählt und abgegrenzt werden können.

[1] `http://www.w3.org/DOM`.

[2] Hsu und Dung (1998) [70], S. 535ff. Bemerkung: Mit *Attribut* ist hier der auszufüllende Wert eines Attributs gemeint, falls es alleine vorkommt.

- Eine Webseite ist *unstrukturiert*, falls linguistisches Wissen notwendig ist, um Attribute korrekt extrahieren zu können.
- Eine Webseite ist *semi-strukturiert*, falls sie nicht unstrukturiert ist. Sie kann Tupel mit einem fehlenden Attribut, ein Attribut mit einem Multi-Wert, Permutationsvarianten der Attribute, Ausnahmen oder Druckfehler enthalten.

Für unstrukturierte Texte ist eine natürlichsprachliche Verarbeitung notwendig. Nach abgeschlossener linguistischer Analyse des Textmaterials wird das Pattern lokalisiert. Dabei liegt der Fokus im Auffinden des grammatikalischen und kontextuellen Musters.

Abgesehen von der Lokalisierung der zu suchenden Informationen ist die Hauptaufgabe der Informationsextraktion aus strukturierten Texten auf die Strukturerkennung zurückzuführen. Da die wichtigsten Fragmente der Informationen auf der Webseite oft in Tabellen oder Listen dargestellt werden, wurden in den letzten Jahrzehnten sehr viele Methoden für die Strukturerkennung entwickelt. Bei diesen Techniken wurde die traditionelle Vorgehensweise der Informationsextraktion so vereinfacht, dass nur Werkzeuge für den Benutzer angeboten wurden. Ihm wird nun überlassen, was er damit extrahieren möchte. Dabei soll er lediglich das für ihn interessante Pattern angeben, so dass das Muster anschließend auf die Struktur abgebildet werden kann. Das Extrahieren erfolgt durch den Strukturabgleich (Struktur-Matching).

Bei den neueren Techniken wird dem automatischen Pattern-Auffinden mehr Bedeutung beigemessen. Für eine Menge von Webseiten wird hierbei iterativ die Struktur automatisch identifiziert und extrahiert, ohne dabei eine Annotation zuzuordnen.

Das automatische Zuordnen des Attribut-Wert-Paares wird in einigen Ansätzen, u.a. bei Yoshida et al. (2003) [133], Guo & Stent (2006) [64] und Zhu et al. (2006) [137], beschrieben. Hierfür erfolgt das automatische Markieren der Attribute mithilfe von Ähnlichkeitsmaßen des Attribut-Wert-Musters. Um diesen Prozess zu ermöglichen, sind bei dieser Methode aber vergleichbar große Datenmengen nötig.

Eine Webseite mit Firmeninformationen ist nach der Definition auf Seite 59 typischerweise semi-strukturiert. Jede Informationsseite hat ihren eigenen Stil, um die zu suchenden Informationen darzustellen. Es können einige Attribute fehlen oder Multi-Werte für ein Attribut existieren. Einerseits können

sie aus Tabellen- oder Listenelementen extrahiert werden, und andererseits basiert das Auffinden von Attributen auf der gezielten Suche nach Mustern auf dem Text.

Da für jede Domain ein einziger Datensatz extrahiert wird, und der Präsentationsstil von Webseite zu Webseite teilweise gravierend unterschiedlich ist, müssen beide Methoden – sowohl das Lokalisieren von textuellen Pattern, als auch die Strukturerkennung – parallel zum Einsatz kommen. Während auf Webseiten ohne Tabellen- und Listenelemente die textuelle patternbasierte Methode angewendet wird, soll für Tabellen- und Listenelemente eher die strukturbasierte Technik eingesetzt werden.

Beim Extrahieren von Informationen aus dem Web gibt es drei verschiedene Stufen[3]:

- Extraktionsebenen:
    1. Datensatz-Ebene (Rekord-Ebene)
    2. Webseiten-Ebene
    3. Website-Ebene

Datensatzbasierte Informationsextraktion findet die Datensatzgrenzen und weist sie den einzelnen Attributen zu, während webseitenbasierte Systeme alle Daten aus der Webseite extrahieren. Dagegen werden bei der websitebasierten Informationsextraktion die Daten auf allen Webseiten einer Website gesucht.

Die domainnamenrelevante Informationsextraktion kann über diese drei Stufen verteilt werden. Während bei einigen Webseiten alle Informationen innerhalb eines Bereichs stehen, können bei anderen die zu suchenden Informationen aus der ganzen Webseite gewonnen werden. In einigen Fällen sind die Informationen über mehrere Webseiten verteilt.

Hinsichtlich dieser Charakteristiken liegt die Schwierigkeit der domainnamenrelevanten Informationsextraktionen darin, dass einerseits alle relevanten Webseiten geholt werden sollen, und andererseits gleichzeitig die Struktur des Textes und sein Inhalt berücksichtigt werden müssen, um die patternbasierte Methode zur Anwendung zu bringen.

[3]Chang et al. (2007), S. 3f [22].

Die bislang entwickelten Wrapper und automatischen Extraktionsmethoden waren meist für große Websites konzipiert, welche die Datensätze durch iterative und strukturierte Muster aus der Backend-Datenbank generieren.

Das Extrahieren von Informationen aus einer Website wird bezüglich ihrer domainnamenrelevanten Informationen wie folgt definiert:

- Definition: Informationsextraktion aus einer Website

  Wird eine Start-URL und eine Menge von Attributen vorgegeben, so wird der Wert für jede Attributklasse extrahiert. Hierfür sollte das Extraktionssystem über einen Crawler verfügen. Mit ihm wird nach der Informationsseite auf der Webseite der vorgegeben URL gesucht. Wird die entsprechende Informationsseite gefunden, dann muss das System in der Lage sein, den Wert für jede vorgegebene Attributklasse zu extrahieren. Die extrahierten Informationen können nun in Form einer relationalen Datenbank strukturiert werden.

Nach dieser Definition sollte ein „Extraktor“ einen Crawler und eine vorklassifizierte Attributklasse haben. Falls eine URL und die vorklassifizierte Attributklasse gegeben sind, kann er aus der Website die Werte für die Attribute extrahieren.

In dieser Arbeit werden die Firmeninformationen aus der Firmenwebsite extrahiert. Bislang wird über Extraktionssysteme für Firmeninformationen nur wenig berichtet.

Bei Krötzsch & Rösner (2002) [74] wurde beispielsweise ein Firmenprofil für die „Gießerei-Branche“ extrahiert. Ihr Ansatz sieht unter anderem die Tabellenverarbeitung für strukturierte Daten und Phrasenmuster für unstrukturierte Texte vor. Bei ihnen beinhaltet das Firmenprofil hauptsächlich Informationen über die Produktionsstätten und -prozesse, die Qualitätszertifikate, das Produktionsmaterial, die Produktpalette und ihren Produktionsumfang.

Bei Svátek et al. (2003) [129] wurden die Firmeninformationen in verschiedenen Stufen extrahiert. So wurden einerseits die Meta-Tags, aber auch andererseits der „Plain“-Text und der strukturierte Text berücksichtigt. Ihr Ziel war aber nicht das Extrahieren der konkreten Firmeninformationen, sondern

es stand die methodische Überlegung[4] im Vordergrund.

Bei Labský & Svátek (2006) [78] wurde die Präsentationsontologie für die Extraktion von Firmeninformationen, wie z.B. Kontaktdetails und Produktinformationen, ansatzweise vorgestellt.

All diese Arbeiten enthalten aber keine explizite Definition eines Firmenprofils, oder geben Aufschluss darüber, was mit einer „Firmeninformation“ konkret gemeint ist.

Da der Begriff „Firmeninformationen“ unterschiedlich zu verstehen ist, wird er zunächst definiert:

- Definition: Firmeninformation

  Firmeninformationen sind die relevanten, relationalen Fakten bezüglich eines Domain-Namens, inklusive des Domain-Inhabers oder -Betreibers.

Zwischen dem jeweiligen Domain-Namen und den Firmeninformationen soll nun eine Beziehung hergestellt werden. Dafür werden nur relevante Informationen in Bezug auf den Domain-Namen extrahiert.

5.1 Vorgehensweise

Die Vorgehensweise ist in Abbildung 5.1 schematisch dargestellt.

Ist eine URL gegeben, dann wird nach der „Einstiegsseite“ gesucht, dabei werden unter gewissen Umständen Frames und Java-Script berücksichtigt. Über die gezielte Verfolgung von Links wird dann eine Informationsseite geholt, die auf ihre Baumstruktur abgebildet wird. Durch „Depth-First-Traverse“ (Tiefensuche) werden Abschnitte, die der Navigation dienen, sowie Bereiche, welche für Werbung vorgesehen sind, automatisch übersprungen. Die eventuell vorhandenen negativen Kontaktdaten werden mithilfe der gesammelten negativen Indikatoren abgeschnitten.

Nachdem der minimale Datenbereich bestimmt wurde, wird darauf das Attribut-Wert-Verfahren angewendet. Durch das Verfahren werden die eventuell

[4] In Svátek et al. (2003) [129] wurden als Referenz 50 Websites aus dem „Business“-Verzeichnis von *Open Directory* entnommen, um herauszufinden, wo genau die gesuchten Informationen zu finden sind.

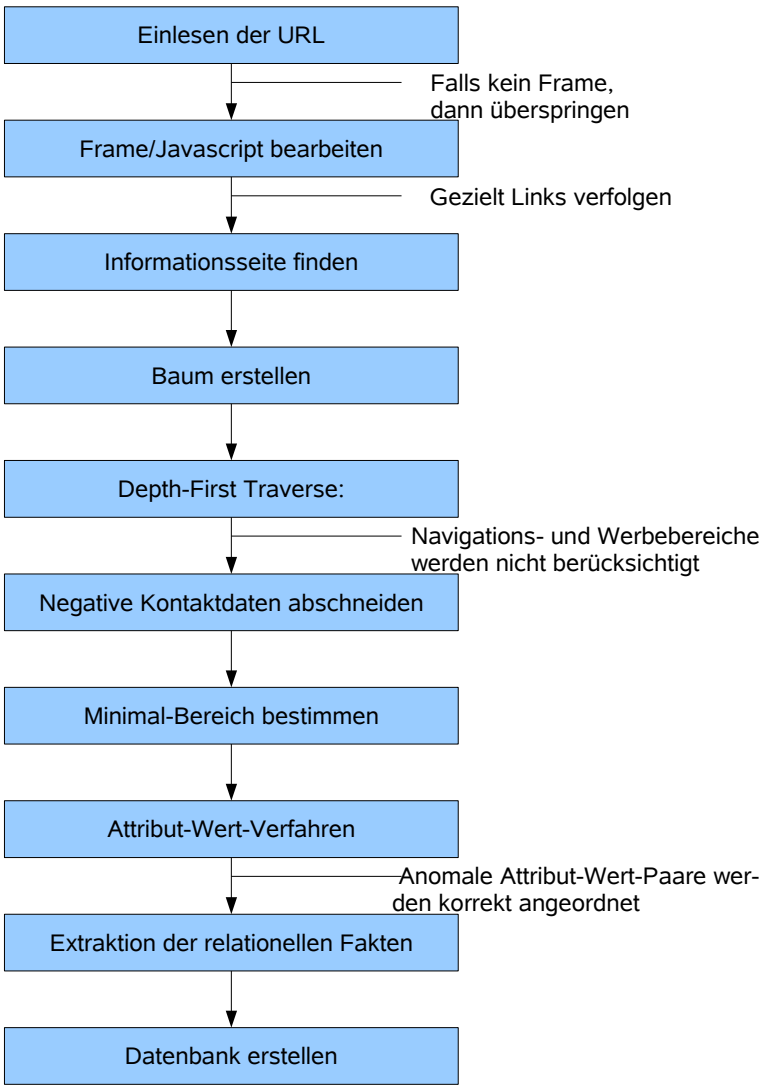


Abbildung 5.1: Fluss-Diagramm zur Vorgehensweise bei der Extraktion

falsch geordneten Attribut-Wert-Paare korrekt sortiert. Danach werden alle relationalen Fakten bezüglich des Domain-Namens extrahiert.

5.2 Statistische Bewertung der Link-Struktur und Anchor-Texte

Das World Wide Web ist über Hyperlinks vernetzt. Seit der Einführung der Suchmaschine „Google“ sind Anchor-Texte ein wichtiger Hinweis, um zur gewünschten Webseite zu gelangen. Brin & Page (1998) [13] beschreiben die Bedeutung der Anchor-Texte wie folgt:

> „... anchors often provide more accurate descriptions of web pages than the pages themselves ... This idea of propagating anchor text to the page it refers to was implemented in the World Wide Web Worm [McBryan 94] especially because it ... expands the search coverage with fewer downloaded documents. We use anchor propagation mostly because anchor text can help provide better quality results.“

Andererseits werden die Anchor-Texte (Links) beim fokussierten Crawler eingesetzt. Chakrabarti et al. (1999) [21] beschreiben den fokussierten Crawler wie folgt:

> „The goal of a focused crawler is to selectively seek out pages that are relevant to a pre-defined set of *topics*. The topics are specified not using keywords, but using exemplary documents. Rather than collecting and indexing all accessible Web documents to be able to answer all possible ad-hoc queries, a focused crawler analyzes its crawl boundary to find the links that are likely to be most relevant for the crawl, and avoids irrelevant regions. This leads to significant savings in hardware and network resources, and helps keep the crawl more up-to-date[5].“

[5] s. auch [34, 20, 127, 83, 6].

Anchor-Texte können sowohl für die Beschreibung der betroffenen Webseite als auch für den fokussierten Crawler der Webseiten benutzt werden.

In dieser Arbeit macht man sich die besondere Charakteristik der Anchor-Texte zu Nutze. Hierfür werden die einzelnen Links gezielt verfolgt, um auf die gewünschte Informationsseite zu gelangen. Da die gesuchten Informationen oft nur auf bestimmten Seiten zu finden sind, ist es unnötig und ineffektiv die ganze Bandbreite einer Website vollständig zu crawlen, und somit sehr viel Zeit zu vergeuden.

Für die statistische Bewertung der Anchor-Texte und Links, welche zur Informationsseite führen, wurde zuerst eine Menge von URLs stichprobenartig ausgewählt und manuell bearbeitet. Dabei ergab sich die folgende Reihenfolge der Anchor-Texte, welche nun zur Informationsseite führen: „*Impressum, Kontakt, Über uns, ...*“[6]. In Abbildung 5.2 wurde nun dieses Ergebnis zusammengefasst.

Unter „**Andere**“ fallen Anchor-Texte wie „*Adresse, Sie finden uns, usw.*“.

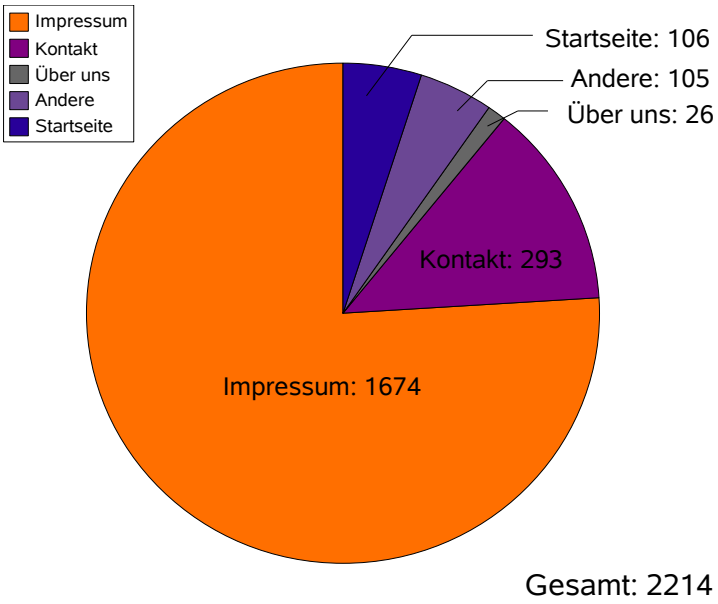


Abbildung 5.2: Anchor-Text-Verteilung

[6]In Svátek et al. (2003) wurden 4 Strings für die Firmenprofilseite benutzt: „*about, company, overview, profile*“.

Abgesehen von der Unterscheidung zwischen Klein- und Großschreibung fallen unter „**Impressum**“ verschiedene Variationen: Einige davon sind „*Impress, Imprint, Impressum & Kontakt, Impressum / Kontakt, usw.*“. Falls „Kontakt“ mit „Impressum“ zusammen vorkommt, dann wird „Kontakt“ nicht zum „**Kontakt**“ gezählt. Dadurch wird die exklusive Verfolgung der Links gesichert.

Beim Fehlen des Anchor-Textes aufgrund des „IMG“-Anchors ohne den entsprechenden „ALT“-Text wird die entsprechende Link-Struktur statistisch ermittelt. Die Statistiken in Abbildungen 5.3 und 5.4 zeigen die entsprechenden Link-Texte zusammen mit ihren Positionen (Stellen) im Pfad, welche möglichst direkt zur Informationsseite führen.

Mithilfe dieser Statistik kann eine Webseite, welche die entsprechenden Firmeninformationen enthält, exklusiv ermittelt werden: Falls nun der Anchor-Text „**Impressum**“ auf der gegebenen Website zu finden ist, wird die Impressumseite geholt und nach den Firmeninformationen durchsucht. Wenn kein Anchor-Text für „**Impressum**“ zu ermitteln ist, dann wird nach dem Anchor-Text „**Kontakt**“ gesucht, und anschließend die „**Über uns**“-Seite, usw. In diesem Sinne ist der hier beschriebene Crawler als fokussiert anzusehen.

5.3 HTML und Baumstruktur

Da HTML in erster Linie nicht für die Informationsextraktion, sondern für die Datenrepräsentation konzipiert wurde, ist die textbasierte Extraktionsmethode nicht einwandfrei anwendbar. Sie setzt grammatikalische Sätze, auf der eine morphologische und syntaktische Analyse durchgeführt wird, voraus.

HTML ist jedoch (semi-)strukturiert, da es in einer Art von Baumstruktur dargestellt werden kann. Jeder HTML-Baum ist ein geordneter Baum, in dem jeder Knoten entweder ein Elementknoten oder ein Textknoten ist. Ein Elementknoten hat eine geordnete Liste von Null oder mehr Tochterknoten und enthält einen HTML-Tag. Ein Textknoten hat keinen Nachfolger (Tochterknoten) und enthält dafür Text.

Der Zusammenhang zwischen HTML-Code und Baumstruktur wird anhand eines Beispiels in Abbildung 5.5 ersichtlich.

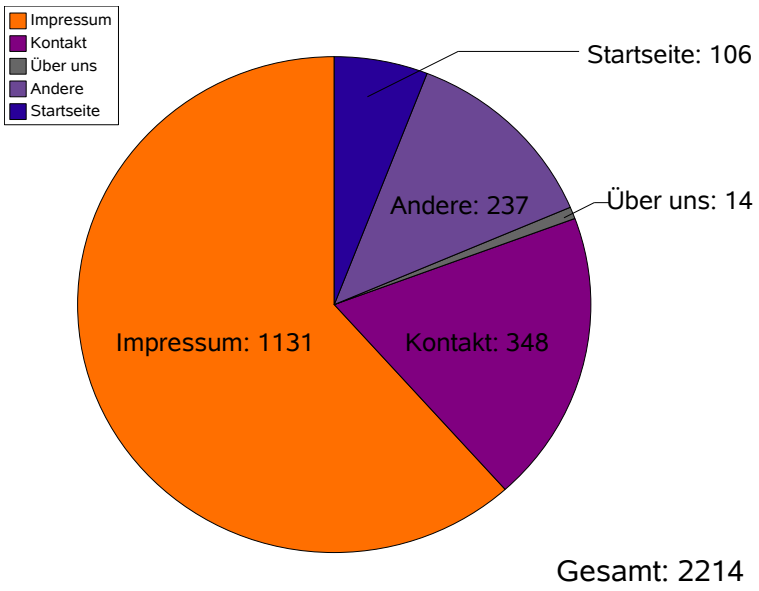


Abbildung 5.3: Verteilung der gesuchten Texte in Bezug auf die „Source-URLs“

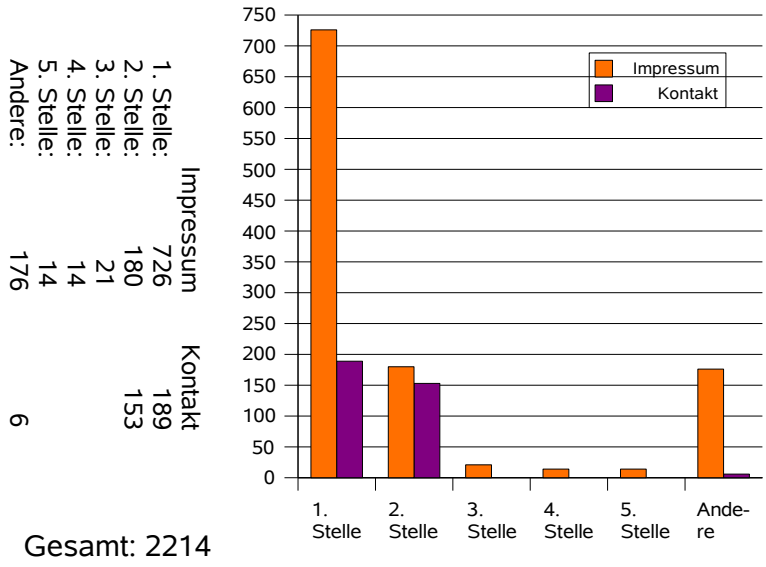
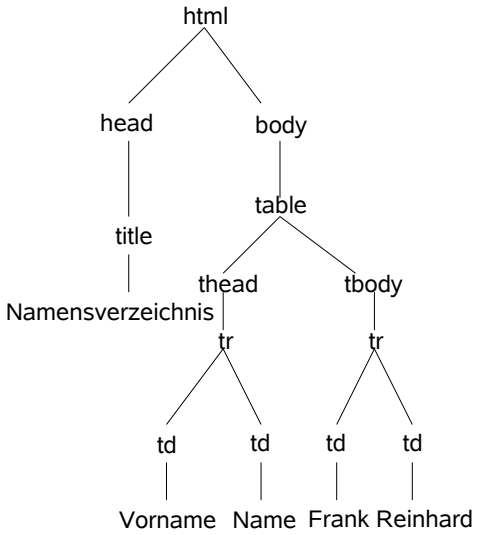


Abbildung 5.4: Position der gesuchten Texte im Source-URL-Pfad

```
<html>
 <head>
  <title>
    Namensverzeichnis
  </titel>
 </head>
 <body>
  <table>
   <thead>
    <tr>
     <th>Vorname</th>
     <th>Name</th>
    </tr>
   </thead>
   <tbody>
    <tr>
     <td>Frank</td>
     <td>Reinhard</td>
    </tr>
   </tbody>
  </table>
 </body>
</html>
```

(a) HTML-Code



(b) Baumstruktur

Abbildung 5.5: Beispiel für eine Baumstruktur

Diese Charakteristik der HTML-Seite hat viele dazu bewegt, dass sie die zu suchenden Informationen durch einen baumbasierten Wrapper – was ein hochstrukturiertes und iteratives Muster voraussetzt – extrahieren wollten. Dieser Versuch stößt aber auch an seine Grenzen, wenn ein strukturbasierter Wrapper auf die verschieden konstruierten Multi-Webseiten angewendet werden soll.

Selbst wenn HTML-Seiten augenscheinlich gleich für den Besucher aussehen, kann ihnen eine unterschiedliche Elementordnung zu Grunde liegen, wie es z.B. bei Cohen et al. (2003) [29] gezeigt wurde[7].

Die in der letzten Zeit entwickelten flexiblen Wrapper-Techniken wie *Tree-Edit-Messungen* oder *interaktive Wrapper-Entwicklungen* haben mehr Flexibilität und breitere Anwendungsmöglichkeiten gezeigt. Trotzdem sind diese Techniken leider nicht für die Informationsextraktion aus Multi-Webseiten geeignet. Denn jede Webseite hat einen eigenen Aufbaustil, was die Anwendungsmöglichkeiten dieser Methoden erheblich begrenzt.

Aufgrund dessen wird angenommen, dass jede Website für sich eine eigene Baumstruktur hat. Nur kann dann das Vergleichen mit anderen Webseiten zum Auffinden des iterativen Musters schwerwiegende Fehlberechnungen mit sich bringen.

5.3.1 Gewichtung der HTML-Elementknoten

Wie die Beispiele <*i*><*b*>foo<*b*><*i*> und <*b*><*i*>bar<*i*><*b*> zeigen, können unterschiedlich geordnete HTML-Tags auf dem Browser gleich formatiert präsentiert werden.

Um dieses Problem zu umgehen, wird jedem HTML-Tag aufgrund seiner Funktion eine Gewichtung zugewiesen. Je nach Zweck werden die HTML-Tags zuerst nach *Zeichen-, Block-, Listen-, Tabellen-, Image-, und Zeilen-*Elementen aufgeteilt. Bei dieser Unterteilung bleiben nicht-gewichtete Elemente unbeachtet.

So sind Zeichen- und Image-Elemente innerhalb von Tabellen- oder Listen-Elementen nicht gewichtet. Der Text unter diesen Elementen wird somit un-

[7] <i><b>foo<b><i> und <b><i>bar<i><b> werden im Browser gleich formatiert dargestellt, sind aber im HTML-Baum anders angeordnet, was das Auffinden des iterativen Musters stark erschwert.

mittelbar an den jeweiligen Elternknoten angefügt. Die simple vollständige Abbildung des HTML-Codes auf die Baumstruktur wird das oben erwähnte Problem nicht umgehen. Dafür taucht dieses Problem bei der Elementgewichtung nicht auf.

5.3.2 Minimaler Datenbereich und Firmeninformationen

Da nicht nur die zu suchenden Informationen, sondern auch andere unerwünschte Informationen, wie z.B. Werbung, eine Navigationsleiste, sowie Kontaktdaten von Web-Hosting und Web-Designern, vorkommen, ist es sinnvoll, von Anfang an diese auszuschließen. Ohne diese Informationen kann somit nur der Bereich, welcher die firmenrelevante Information enthält, lokalisiert werden.

Um diesen Prozess zu veranschaulichen, wird in Abbildung 5.6 eine reale HTML-Seite mit der klassifizierten Information gezeigt.

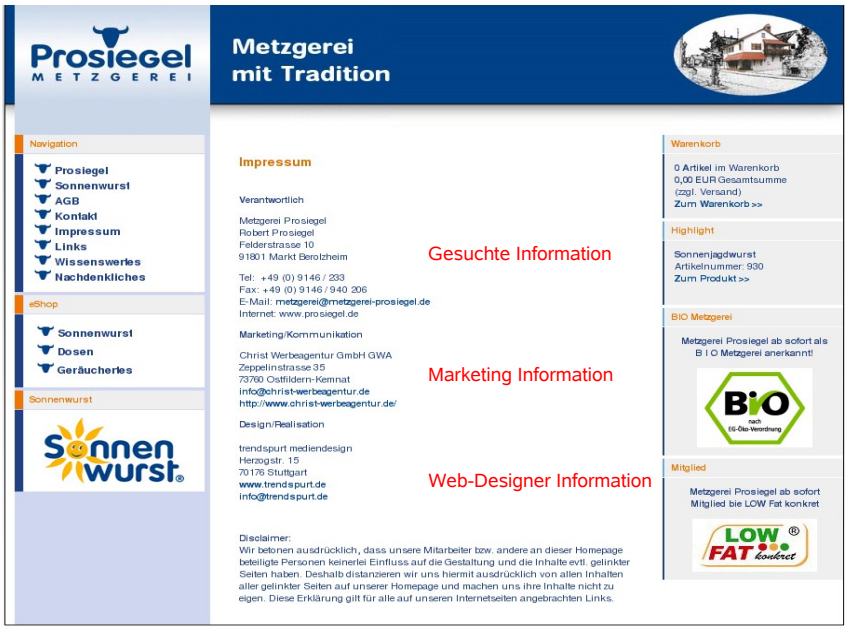


Abbildung 5.6: Impressum-Seite des Domain-Namens „prosiegel“

In der in Abbildung 5.6 präsentierten HTML-Seite sind verschiedene Datenbereiche zu finden. So gibt es den Navigationsbereich auf der linken Sei-

te, sowie einen Werbebereich auf der rechten Seite. Die eigentlich interessanten Informationen sind aber in der Mitte zu finden. Angenommen, dass Navigations- und Werbebereich schon abgeschnitten wären, so blieben trotzdem die 3 verschiedenen Kontaktdaten, von denen nur eine von Interesse ist. Um nur die gewünschten Informationen extrahieren zu können, müssen die beiden anderen Datensätze ausgeschlossen werden.

Es wurde mehrfach beobachtet, dass die Informationseinheiten geschlossen innerhalb eines bestimmten Bereichs vorkommen [88, 67, 137, 135, 90].

In Liu et al. (2003) [88] werden diese Beobachtungen wie folgt beschrieben:

> „A group of data records that contains descriptions of a set of similar objects are typically presented in a particular region of a page... Such a region called a *data region*".

Bei Hiremath et al. (2005) [67] wurde diese Eigenschaft wie folgt definiert:

> „A *data region* is defined as the most relevant portion of a web page. e.g. A region on the product-related web-site that contains a list of products forms the *data region*".

Anschließend an diesen Beobachtungen wird der Datenbereich in Bezug auf die domainnamenrelevanten Informationen wie folgt definiert:

- Definition: Datenbereich bzgl. domainnamenrelevanter Information

  Der Datenbereich bezüglich der domainnamenrelevanten Informationen ist der domainnamenrelevanteste Abschnitt einer HTML-Seite.

Um den entsprechenden Datenbereich zu bestimmen und die gewünschten Informationen zu extrahieren, wurden verschiedene Methoden vorgeschlagen. Bei Liu et al. (2003) wurde zuerst ein HTML-Tag-Baum erstellt, um anschließend durch einen „Top-Down-Traverse" (Top-Down-Suche) mit parallelem „Depth-First-Traverse" (Tiefensuche), sowie durch den String-Vergleich zwischen den vermeintlichen Dateneinheiten den maximalen Bereich an relevanten Informationen zu bestimmen.

Da sich HTML einerseits in stetiger Entwicklungsphase befindet und aufgrund des Missbrauchs nicht immer korrekt abgebildet werden kann, wurden in Hiremath et al. (2005), Zhai & Liu (2005), sowie Liu et al. (2006) [67, 135, 90] visuelle Methoden vorgeschlagen.

Durch diese visuelle Repräsentation von HTML (HTML-Rendering-Technik) können nun die kleinen Abweichungen der HTML-Tags ohne dieses schwerwiegende Problem korrekt abgebildet werden.

Trotz dieser Techniken gilt wie bei allen visuellen Methoden auch hier die Grundannahme, dass die Dateneinheiten („Records“) sich auf der Webseite in demselben HTML-Eltern-Knoten befinden.

Andererseits wurde für das Verfahren angenommen, dass die zu suchenden Daten durch eine Art von der Backend-Datenbank generiert wurden, was zu einer hochstrukturierten HTML-Seite führt.

Bei all diesen Methoden ist der Informationsbereich ein maximaler Datenbereich, welcher alle möglichst relevanten Daten enthalten soll. Wird eine Baumstruktur aus einer HTML-Seite mit Informationsgehalt erzeugt, so wird zunächst der maximale Datenbereich bestimmt, und der irrelevante Datenbereich, wie z.B. die Navigationsleiste oder Werbung, abgeschnitten.

Diese Vorgehensweise kann aber nicht 1:1 auf eine Webseite mit Firmeninformationen übertragen werden, wie sie in Abbildung 5.6 dargestellt ist. Denn wie aus Abbildung 5.6 ersichtlich wird, ist nur der erste Informationsbereich in der Mitte des Datenbereichs als interessant zu werten. Die beiden anderen unten stehenden Informationen sind für unsere Zwecke irrelevant.

Aus diesem Grund wird beim Bestimmen des Datenbereichs für die domainnamenrelevanten Informationen ein „Depth-First-Traverse“ (Tiefensuche) angewendet. Dadurch werden alle anderen Informationen, wie Werbetexte, Kontaktdaten für Web-Hosting oder Web-Designer, automatisch ausgeschlossen.

Aufgrund dessen muss ein minimaler Datenbereich definiert werden:

- Definition: Minimaler Datenbereich

  Ein *minimaler Datenbereich* bezüglich der domainnamenrelevanten Informationen auf einer Webseite ist der kleinste HTML-Tag-Bereich, der alle domainnamenrelevanten Informationen enthält.

Da der minimale Datenbereich, welcher die gesuchten Informationen enthält, von Webseite zu Webseite unterschiedlich sein kann, ist das Auffinden des iterativen Musters, welches von vielen Wrapper-Generatoren und Web-Data-Mining-Techniken vorgeschlagen wird, weniger relevant.

Jede einzelne Webseite, welche gewisse Firmeninformationen enthält, hat eine eigene Baumstruktur. Nach der jeweiligen Abbildung einer Webseite auf ihre Baumstruktur wird der Informationsbereich durch einige Attribut-Wert-Paare bestimmt. Auf diese Weise werden alle anderen Kontaktdaten, wie diejenigen für Web-Designer und Web-Hosting, entfernt.

Da der gesuchte Datenbereich andere Kontaktdaten enthalten kann, wird zuerst versucht, die negativen Datenbereiche auszuschließen. Aus diesem Grund wurden auch Attribute für die negativen Kontaktdaten bei der Trainingsphase zusammengestellt[8].

Zunächst werden die negativen HTML-Tag-Bereiche des Strukturbaumes ermittelt. Falls sie erfolgreich lokalisiert werden konnten, werden sie anschließend aus dem Baum entfernt. So bleibt letztendlich allein der Bereich für die Firmeninformationen zurück.

Leider kann nicht garantiert werden, dass alle negativen Attribute gefunden wurden. Deshalb wurden auch positiven Attribute für den zu suchenden Informationsbereich gesammelt. Falls ein HTML-Bereich von einem dieser positiven Attribute eingeleitet wird, so ist dieser HTML-Tag-Bereich der gesuchte Informationsbereich.

Die Navigations- und Werbebereiche werden aufgrund des „Depth-First-Traverse“ (Tiefensuche) und dem Attribut-Wert-Verfahren automatisch ausgeschlossen.

Beim Bestimmen dieses Bereichs werden die minimalen und maximalen Attribut-Wert-Paare angenommen. Die hier verwendeten Attribut-Wert-Paare sind die *Postleitzahl, Telefonnummer* und *USt-IdNr. (Umsatzsteuer-Identifikationsnummer).*

Falls eine Webseite all diese 3 Paare enthält, wird versucht, den Bereich zu bestimmen, der diese Paare beinhaltet. Falls dies nicht der Fall sein sollte, wird zuerst die Postleitzahl und Telefonnummer, danach die Postleitzahl und USt-IdNr., und zuletzt die Postleitzahl als alleiniges Kriterium herangezogen.

[8] siehe Kap. 3.

Die Postleitzahl ist als Hauptmerkmal anzusehen. Bei den verwendeten Trainingsdaten war die Postleitzahl bei 91,35% der Fälle stets vorhanden, sofern die HTML-Seite die gesuchten Firmeninformationen enthielt. Nur der Ortsname trat mit 91,90% noch häufiger auf.

Für den Fall, dass alle 3 Attribut-Wert-Paare in einem HTML-Tag-Bereich vorkommen und sich die anderen firmenrelevanten Informationen sich in einem eigenen Tag-Bereich befinden, wird zusätzlich die Textlänge eines minimalen HTML-Tag-Bereiches berechnet. Für die minimale Textlänge werden in Bezug auf die gefundenen Attribut-Wert-Paare verschiedene Schwellenwerte angenommen.

- Definition: Minimale Textlänge

  Der minimale Datenbereich soll je nach den gefundenen Attribut-Wert-Paaren eine *bestimmte Textlänge* haben.

Der gesamte Vorgang zur Bestimmung des minimalen Datenbereichs wird als Algorithmus (siehe Abbildung 5.7) geschildert.

5.3.3 Positive und negative Phrasen zur Bestimmung des minimalen Bereiches

Wie bereits Abbildung 5.6 zeigte, können jederzeit irrelevante zusammen mit relevanten Informationen im Text einer HTML-Seite auftreten. Einige negativen Phrasen sind in Abschnitt 3.2 des Kapitels 3 angegeben.

Unter positiven Phrasen sind beispielsweise die Folgenden zu verstehen:

Anbieter
Impressum
Betreiber
Herausgeber
Anbieter-Kennung
Betreiber-Impressum
Anbieter-Kennzeichnung

```
MinimalerBereich
1. plz = Wert-String der Postleitzahl
2. u = Attribut-Klasse der Umsatzsteuernummer
3. t = Attribut-Klasse der Telefonnummer
4. p = positive Klasse der Kontaktdaten
5. n = negative Klasse der Kontaktdaten

6. Begin Depth-First Traverse(HTML-Baum)
7.  überspringe if Knoten Element von n ist
8.  if gefunden plz, u und t auf HTML-Seite und length von Knoten größer als Schwellenwert
9.     if Knoten plz, u und Wert-String von u hat
10.       MinimalerBereich = Knoten
11.  else if gefunden plz und u auf HTML-Seite  und length von Knoten größer als Schwellenwert
12.    if Knoten plz und u hat
13.       MinimalerBereich = Knoten
14.  else if gefunden plz und t auf HTML-Seite  und length von Knoten größer als Schwellenwert
15.    if Knoten plz und t hat
16.       MininmalerBereich = Knoten
17.  else if gefunden plz und t auf HTML-Seite  und length von Knoten größer als Schwellenwert
18.    if Knoten plz hat
19.       MinimalerBereich = Knoten
20.  else if Textbeginn von Knoten p entspricht und length von rechtem Knoten größer
             als Schwellenwert
21.       MinimalerBereich = Knoten
22.  else
23.       next
24.  endif
25. endTraverse
```

Abbildung 5.7: Algorithmus zur Bestimmung des minimalen Bereichs

5.3.4 HTML-Tabellen und das Attribut-Wert-Verfahren

Obwohl die meisten Informationen in Webseiten als Text vorliegen, sind die bedeutendsten Fragmente der Informationen meist strukturiert dargestellt. Ist dies der Fall, so sind diese Daten oft in Tabellen formatiert.

Bei der Datenwiedergabe in Form von Tabellen dürfen zwei Faktoren nicht vernachlässigt werden: die Erkennung der Tabellentypen, sowie die Zuordnung der Werte zu ihren jeweiligen Attributen.

Yoshida et al. (2003, S. 185) [133] haben hierfür 9 unterschiedliche Arten von Tabellen ermitteln können, welche in Abbildung 5.8 gezeigt werden.

Außerdem haben Yoshida et al. (2003) versucht, mithilfe des EM-Algorithmus Attribute mit ihren zugehörigen Werten automatisch zu extrahieren. Ihre Methode benötigt dafür eine große Menge an Tabellen, um die Attribut-Wert-Paare korrekt zuordnen zu können. Jedoch werden bei den Tabellentypen die

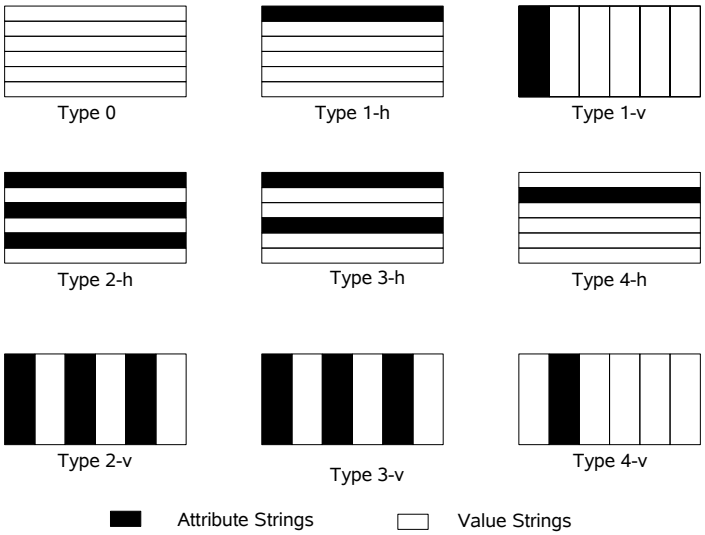


Abbildung 5.8: Tabellentypen nach Yoshida et al. (2003)

verschachtelten Tabellen bislang nicht berücksichtigt.

Des Weiteren ist die angenommene 1:1-Abbildung von Strings auf die einzelnen Zellen der Tabellen nicht stets umsetzbar[9].

Bei Gao et al. (2003) [52] wurden die Extraktionsmuster aus einer einzigen Trainingsseite ohne irgendeine vorherige manuelle Alignierung ermittelt. Jedoch kann dieses System nur mit den Zeilen der Tabellen arbeiten.

Während viele Wrapper-Generatoren mit dem DOM[10] oder in einem visuellen eindimensionalen Raum arbeiten, führen Gatterbauer et al. (2007) [53] einen zweidimensionalen Rahmen ein.

Trotz ihrer richtigen Argumentation, dass die DOM-Struktur durch den Missbrauch der HTML-Elemente und die Einführung der erweiterbaren HTML-Elemente nicht korrekt abgebildet werden könne, und die visuelle Abbildung der Webseite über mehr Flexibilität verfügen würde, liegen die korrekten Erkennungs- und Interpretationsraten jeweils nur bei 68% und 48%, was die

[9] Nach Gatterbauer et al. (2007) [53] liegen die nicht-komplett alignierten und durch Delimiter getrennte Tabellen bei rund 5%. (siehe S. 75).

[10] `http://www.w3.org/DOM`.

Qualität des Systems stark beeinträchtigt.

Insbesondere führt ihre Bemerkung (S.76) „*domain-independant table interpretation cannot result in unambiguously structured information because of existing inherent domain-specific ambiguities that can sometimes not even be resolved by human*“ dazu, dass die Möglichkeiten der domainunabhängigen Extraktion erheblich in Frage gestellt werden.

Da Tabellen die Hauptdarstellungsmethoden von Informationen innerhalb von HTML-Seiten sind, bestanden die Datenbereiche der Trainingsdaten zu ca. 70% aus Tabellen, welche die gesuchte Information enthielten.

Die verwendeten Tabellenstrukturen waren sehr unterschiedlich. Während eine Tabelle hauptsächlich für den Vergleich von Werten konzipiert ist, wird eine Tabelle auf der Webseite mit den Firmeninformationen – der so genannten Informationsseite – jeweils nur mit einem Wert dargestellt. Dies führt zu sehr komplizierten Tabellenstrukturen.

Aufgrund der spezifischen Darstellungsmöglichkeiten wird für Zuordnung des Tabelleninhalts das Attribut-Wert-Verfahren verwendet[11].

Zur Erläuterung des Verfahrens wird ein Beispiel aus einer realen Webseite (Abbildung 5.9) herangezogen.

Abgesehen von der irrelevanten Bereichen und HTML-Elementen, sieht der Quellcode für den Informationsbereich aus Abbildung 5.9 wie in Abbildung 5.10 aus:

Im <TR>-Element aus Abbildung 5.10 sind zwei <TD>-Elemente zu finden. Vom Typ her kann man annehmen, dass das erste <TD>-Element ein Attribut, und das letztere <TD>-Element einen Wert darstellt. Aber im vorderen <TD>-Element wurden durch
 mehrere Attribute, wie Tel., Fax, E-Mail und Umsatzsteuer-Nr., angegeben, was die 1:1-Abbildung beim

[11] Vgl. „presentation ontology“ von Labský & Svátek (2006): “This contrast however becomes less sharp when considering semi-structured web content in the form of lists, tables or forms, and possibly even images and other multimedia objects. Ontologies directly usable for analysis of web structures are likely to borrow a lot from 'customer-service' ontologies, since the fragments of HTML code will often directly map on ontology classes, attributes/relations and instances. We will call them *presentation ontologies*, since their universe of discourse is that of objects as *presented* on the web or similar medium (e.g. computer offers encoded in HTML) rather than of real-world objects (real computers).”

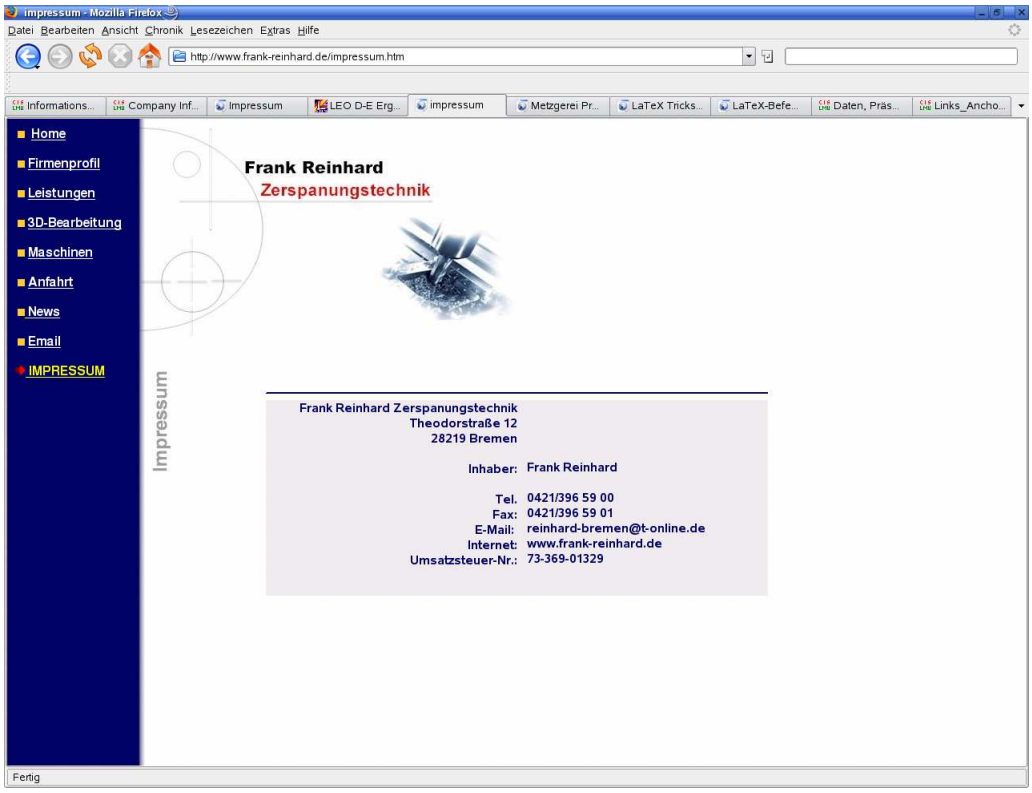


Abbildung 5.9: Beispiel einer Tabelle der SLD „frank-reinhard“

Attribut-Wert-Verfahren nicht möglich macht[12]. Nichtsdestotrotz ist die Anzahl von
s in beiden Zellen nicht gleich, was das Zählen des Tags
 sinnlos macht.

Durch das Attribut-Wert-Verfahren können die gesuchten Attribut-Wert-Paare in Abbildung 5.10 korrekt ermittelt werden.

Hierfür wird zunächst der erste String, der durch
 abgetrennt ist, analysiert. Der String „Frank Reinhard Zerspanungstechnik“ entspricht keinem der zu suchenden Attributen und wird einfach so wiedergegeben. Die nächsten beiden Strings „Theodorstraße 12“ und „28219 Bremen“ passen auch zu keinem der Attribute und werden genauso belassen. Zeilen, welche

[12] Ähnliche Beobachtungen wurden auch bei Krötzsch & Rösner(2002) [74] gemacht: Thereby it must be considered that a list is possible in a table item, which is separated through <Ul>-, <LI>-,
-tags or a comma. siehe auch Gatterbauer et al. (2007) [53].

```
<tr>
 <td>
  Frank Reinhard Zerspanungstechnik<br>
  Theodorstraße 12<br>
  28219 Bremen<br>
  <br>
  Inhaber:<br>
  <br>
  Tel.<br>
  Fax:<br>
  E-Mail: <br>
  Internet:<br>
  Umsatzsteuer-Nr.:
 </td>
 <td>
  <p></p>
  <p>
  Frank Reinhard<br>
  <br>
  0421/396 59 00<br>
  0421/396 59 01<br>
  reinhard-bremen@t-online.de<br>
  www.frank-reinhard.de<br>
  73-369-01329
  </p>
  <p></p>
 </td>
</tr>
```

Abbildung 5.10: Source-Code von Abbildung 5.9

nur aus einer Folge von Leerzeichen bestehen, werden einfach übersprungen. Der nächste String „Inhaber“ entspricht dann einem „**Inhaber**“-Attribut, und somit muss sein Wert gefunden werden. Falls er lokalisiert werden kann, wird das Attribut-Wert-Paar für den „**Inhaber**“ extrahiert, und dann das nächste Attribut-Wert-Paar, usw. Für das Verfahren müssen kanonische reguläre Ausdrücke für jedes Attribut-Wert-Paar gebildet werden.

Für die verschachtelten Tabellen funktioniert das Attribut-Wert-Verfahren ohne jegliche Beschränkung, da das Verfahren auf alle Zellen der Tabelle angewendet wird. Allerdings muss angenommen werden, dass die Attribut-Werte in benachbarten Tabellenzellen vorkommen[13].

Bei der Informationsextraktion aus Tabellen stellen sich drei Probleme[14]:

- Erkennung der Tabellenstruktur
- Tabellen-Clustering
- Attribute-Clustering

Abgesehen vom Erkennungsproblem der Tabellenstruktur sind Tabellen- und Attribute-Clustering beim Attribut-Wert-Verfahren trivial. Weil die Attribute vorklassifiziert und aufgrund einer gewissen Ähnlichkeit erweiterbar sind, tritt das Problem nicht auf.

Das Erkennungsproblem der Tabellenstruktur ist beim Attribut-Wert-Verfahren auf das Zuordnungsproblem der Attribute zu ihren Werten zurückzuführen. Nachdem die korrekten Daten für die Werte gefunden wurden, erfolgt automatisch die Zuordnung des Wertes auf das vorklassifizierte Attribut. Es muss lediglich der korrekte Delimiter erkannt werden.

Beim Attribut-Wert-Verfahren sind nun die in Abbildung 5.11 genannten Typen von Tabellenstrukturen zu berücksichtigen.

In Abbildung 5.11 ist der Typ 1 trivial. Es wird ein zweidimensionales Array aus der Baumstruktur gebildet, dann wird durch das Attribut-Wert-Verfahren Spalte für Spalte abgeprüft und anschließend werden die Attribut-Wert-Paare extrahiert.

[13] Da die Firmeninformationen auf der Informationsseite nur ein Mal vorkommen, ist es unplausibel, dass einem Attribut mehrere Werte zugeordnet werden sollen. Die in unseren Trainingsdaten untersuchten Tabellen haben nur dann mehrere Werte, wenn diese durch einen Delimiter wie
 getrennt sind.

[14] Yoshida et al. (2003) [133].

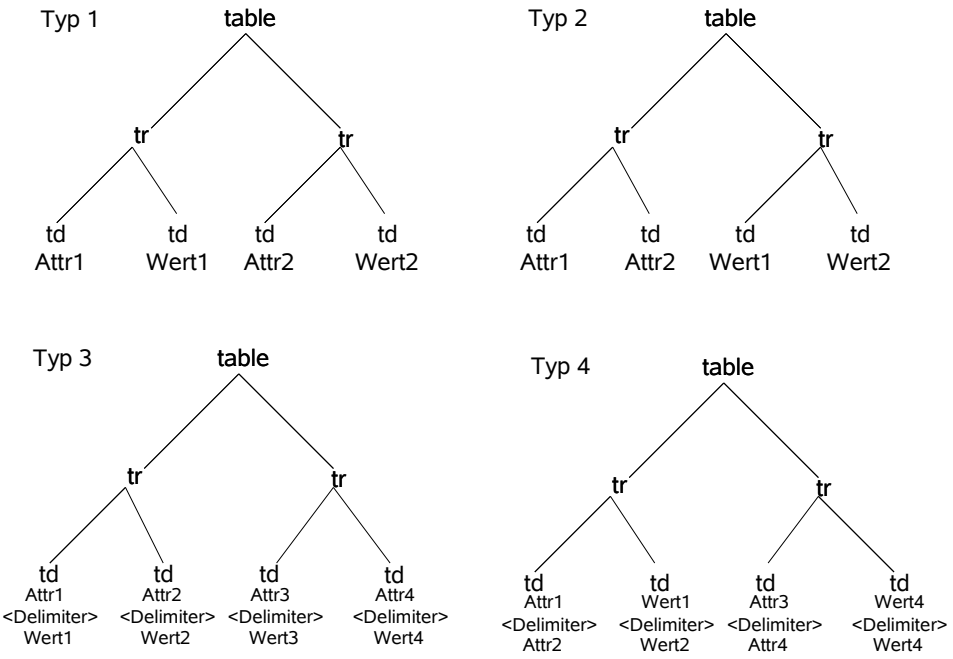


Abbildung 5.11: Baumstruktur der in Betracht gezogenen Tabellentypen

Der Typ 3 stellt kein Problem dar; denn es wird zuerst innerhalb einer Tabellenspalte überprüft, ob der String für den Wert zu einem gefundenen Attribut existiert, sofern eine Spalte überhaupt durch einen Delimiter getrennt wurde.

Die Schwierigkeiten liegen vorwiegend bei den Typen 2 und 4. Beim Typ 2 muss das Attribut-Wert-Verfahren über das <TR>-Element hinaus erweitert werden. Da dieses Verfahren nicht <TR>- sondern nur <TD>-Elemente berücksichtigt, ist das Problem durch ein Array lösbar. Nachdem das erste Attribut lokalisiert wurde, wird die nächste Spalte für den Wert oder für ein anderes Attribut überprüft. Dieser Prozess wird solange rekursiv angewendet, bis der Wert für das erste Attribut gefunden wird. Auf diese Weise wird das erste Attribut-Wert-Paar extrahiert. Der Prozess wird dann auf das zweite Attribut analog angewendet – *ad infinitum* – bis das letzte Attribut-Wert-Paar extrahiert wurde. Was den Typ 4 betrifft, muss eine andere Vorgehensweise erarbeitet werden. Nachdem das Array aus den Tabellenspalten gebildet wurde, muss hierfür noch ein zusätzlicher Verarbeitungsschritt eingeführt werden. An dieser Stelle werden <Delimiter>-Tags in Betracht gezogen, so dass ein zweidimensionales Array entsteht.

Der Algorithmus des Attribut-Wert-Verfahrens für den Typ 4 ist in Abbildung 5.12 dargestellt.

Das Multi-Werte-Problem aus dem Algorithmus in Abbildung 5.12 wird momentan noch nicht beachtet und die Problemlösung somit auf später vertagt[15].

5.3.5 Ähnlichkeitsprozess und automatische Zuordnung unbekannter Attribute

Da nicht alle Attribute vollständig ermittelt werden, können Ähnlichkeitsmaße in Relation zu den existierenden Attributen eingesetzt werden, falls ein String weder eines der bekannten Attribute noch Daten für den Wert aufweist. Dabei wird die Ähnlichkeit wird über Substring-Matching und die Levenshtein-Edit-Distanz berechnet.

[15]Bei den Multi-Werten handelt es sich um die Erkennung des Delimiters und eines anderen Attributs, welches die Wert-Datensätze voneinander abgrenzen kann.

```
TabellenProzess für Typ 4
1. tds = Finde tds; // Array von Spalten
2. anz_tds = Anzahl von tds; // Anzahl von Spalten
3. td_delimiters = Zerlege jede Spalte durch <Delimiter>; // Array von Delimitern der Spalten
4. for ( i = 0; i <anz_tds - 1; i++ ) // Für jede Spalte außer letzter Spalte
5.     anz_delimiters_td = Anzahl der Delimiter von td_delimiters[i];
                                                    // Anzahl von Delimitern der betroffenen Spalte
6.     for ( j = 0; j < anz_delimiters_td; j++ ) // Für jedes Textfeld
7.         next unless length td_delimiters[i][j]; // Überspringe, falls leer ist
8.         td_text = td_delimiters[i][j];
9.         for each Attribute_klasse von vorklassifizierten Attributklassen
10.            if td_text Element von betroffener Attribute_klasse ist
11.              anz_delimiters_next_td = Anzahl der Delimiter von td_delimiters[i+1];
                                                    // Anzahl von Delimitern der nächsten Spalte
12.              for ( k = 0; k <anz_delimiters_next_td; k++ ) // Für die nächste Spalte
13.                 next unless length td_delimiters[i+1][j];
14.                 if td_delimiters[i+1][k] Wert_text der betroffenen Attributklasse ist
15.                   Extrahiere (td_text, td_delimiters[i+1][k]);
16.                   Lösche (td_text, td_delimiters[i+1][k]);
17.                   Break für innerste for-Schleife
18.               endif
19.             endfor
20.           endif
21.       endfor
22.   endfor
23. endfor
```

Abbildung 5.12: Algorithmus des Attribut-Wert-Verfahrens

5.4 Lokale Kontexte und Firmeninformationen

Tabellen und -elemente sind nicht der einzige minimale HTML-Tag-Bereich. Rund 70% aller Informationsseiten haben eine Tabellenstruktur, durch deren Analyse die gesuchte Information zu finden ist. Die anderen 30% sind nicht über Tabellen aufgebaut, oder die Informationen sind nicht innerhalb einer Tabelle.

Eine Besonderheit stellt auch eine Webseite dar, welche die gewünschten Informationen durch Kombination einer Tabelle mit Nicht-Tabellen-Elementen visualisiert. In diesem Fall sollten die Daten über die Tabellen-Analyse hinaus gesucht werden. Hierfür kommt wieder das Attribut-Wert-Verfahren zum Einsatz. Falls die gesuchten Informationen nicht durch das Attribut-Wert-Verfahren zu finden sind, werden die lokalen Kontexte[16] miteinbezogen.

5.5 Lexika und Firmeninformationen

Die Baumstruktur einer Informationsseite und das Attribut-Wert-Verfahren sind wichtige Hilfsmittel, um den minimalen Datenbereich zu bestimmen, und die Zuordnung zwischen Attributen und Werten zu gewährleisten. Bis auf wenige Ausnahmen werden die zu suchenden Werte über sie gefunden.

Zu den Ausnahmen gehören u.a. Postleitzahlen, sowie Straßen- und Ortsnamen, da sie häufig ohne das hinweisende Attribut auftreten. Für die Zuordnung der Werte ohne Attribute werden die Positionen der gefundenen Strings und die entsprechenden Lexika genutzt, falls sie nicht ohnehin einem der kanonischen Wert-Ausdrücke zuzuordnen sind. So ist beispielsweise der String nach der Postleitzahl ein guter Kandidat für den Ortsnamen. Wenn ein solcher String gefunden wird, muss dieser zunächst normalisiert und im vorhandenen Lexikon nachgeschlagen werden. Falls der String im Wörterbuch enthalten ist, kann er als Ortsnamen identifiziert werden.

Die Telefonnummer tritt meist ohne das hinweisende Attribut auf, wenn keine Fax- und Mobilfunknummer angegeben sind. Falls eine Zahlenkombination

[16]Lokale Kontexte sind folgendermaßen zu verstehen: Das zu suchende Attribut muss innerhalb eines Body-Textes vorhanden sein, wie z.B. „Sie können uns telefonisch unter der folgenden Nr. erreichen“.

ohne ein Attribut dem kanonischen regulären Ausdruck für die Telefonnummer zuzuordnen ist, wird diese ebenfalls normalisiert und mit dem Vorwahl-Lexikon abgeglichen. Für diesen Zweck wurde das Wörterbuch der Vorwahlen und der Postleitzahlen mit ihren zugehörigen Ortsnamen erstellt. Diese Lexika können auch zur Überprüfung auf Relevanz von PLZ, Ortsnamen und Vorwahlen benutzt werden.

5.6 Integration der gefundenen Informationen

Obwohl der minimale Datenbereich die meisten Informationen enthält, sind einige außerhalb des minimalen Datenbereichs auf derselben Webseite oder auf einer separaten Webseite zu finden: Öffnungszeiten und Kontaktdaten, wie Telefonnummer und E-Mail-Adresse, kommen nicht immer innerhalb des minimalen Datenbereichs vor. Die Öffnungszeiten kommen oft auf der „Einstiegsseite“ vor, während die Kontaktdaten auf der „**Kontakt**“-Seite auftauchen. In diesem Fall wird versucht, die separat gefundenen Daten zusammenzuführen, wofür eine Relevanzprüfung zwischen Telefonnummer und Ortsnamen unerlässlich ist.

5.7 Template für Firmeninformationen

Die folgende Schablone sollte stets für die meistem Firmeninformationen aufgefüllt werden:

Start-URL:
Source-URL für Daten:
Firmenname:
Straße mit Hausnummer:
Postleitzahl:
Ort:
Telefon:
Mobilfunk:
Fax:
E-Mail:
Geschäftsführer:
Inhaber:
Kontaktperson:
Vorsitzender:
Vorstand:
Vorsitzender des Aufsichtsrates:
Umsatzsteuer-Identifikationsnummer:
Steuernummer:
Registernummer:
Amtsgericht:
Finanzamt:
Öffnungszeiten:

In unseren Trainingsdaten wurden, wie erwartet, nicht alle Felder gefunden[17]. Durchschnittlich konnten 10,89 aller auszufüllenden Einträge gefunden werden. Somit liegt der Prozentsatz der automatisch erstellten Formulareinträge bei knapp 50%. Am häufigsten war die Klasse „Ortsname“ mit 91,90% vertreten, gefolgt von „Postleitzahl“ mit 91,35% und „Straße/Hausnummer“ mit 89,75%. Dabei fallen „Telefonnummer“ mit 85,55%, „E-Mail“ mit 80,55% und „Fax“ mit 73,40% auch stark ins Gewicht. Die nächste Klasse „USt-IdNr.“

[17]Wir bedanken uns bei Herrn U. Real und Frau M. Gerritsen für ihre evaluierenden Betrachtungen bzgl. der Trainingsdaten.

fällt mit ihren 34,05% auf. Ihr folgen noch die „Registernummer“ (28,85%), der „Geschäftsführer“ (25,80%), das „Amtsgericht“ (25,25%), die „Kontaktperson“ (21,40%), die „Steuernummer“ (17%) und der „Inhaber“ (16,65%). Die anderen Klassen liegen bei unter 10%.

5.8 Exkurs: Schreib- und Lesekonventionen & Tabellenstrukturen der Informationsseite

Nach Embley et al. (2006) [42] ist die Tabelle eine der gefährdeten Arten bei der Darstellung von Informationen im Web. So geht der Trend beim traditionellen Fahrplan inzwischen immer mehr dazu, von Frage-Antwort-Systemen (Q/A-Systemen) abgelöst zu werden.

Trotz dieser Tatsache sind Tabellen eine der beliebtesten Arten, strukturierte Daten, sowohl in gedruckten Artikeln als auch im Web, darzustellen. Durch sie sind Daten übersichtlicher zu visualisieren und zu vergleichen. Je mehr Daten in Form von Tabellen dargestellt werden, desto dringlicher wird das Bedürfnis der automatischen Erkennung der Tabellenstrukturen. Laut Zanibbi et al. (2003) [134] und Embley et al. (2006) wurden in den letzten Jahrzehnten über 200 Forschungsarbeiten über die Tabellenerkennung veröffentlicht.

Die meisten Arbeiten auf diesem Gebiet versuchen, Tabellen in einem zweidimensionalen Array über die Tabellenzellen darzustellen. Danach wird versucht, die Koordinaten und Inhalte der Zellen zu bestimmen.

Tabellen können leider mit den verschiedensten Formatierungen konstruiert werden. In Abbildung 5.8 wurde beispielsweise allgemeiner, nicht komplizierter Tabellentyp gezeigt. Jedoch können auch sehr komplizierte Tabellenarten verwendet werden: eine davon ist der „Multi-Zeilen-Zellen-Typ“. Dieser wurde bei der hier vorgestellten Arbeit intensiv untersucht (siehe Abbildungen 5.9 und 5.10).

Wie bereits in Abbildung 5.11 veranschaulicht wurde, können Daten mit verschiedenen Formatierungen dargestellt werden. Jedoch muss hier angemerkt werden, dass unter den Trainingsdaten kein Tabellentyp „2“ von Abbildung

5.11 auftrat.[18]

Man hätte z.B. die Daten von Telefonnummer, Faxnummer und E-Mail genauso darstellen können, wie es die folgenden Beispiele zeigen:

Tel.:	Fax:	E-Mail:
0421/396 59 00	0421/396 59 01	reinhard-bremen@t-online.de

Stattdessen hat der Web-Designer der SLD „`frank-reinhard.de`“ eine sehr komplizierte Form der Tabelle konstruiert, wie es in Abbildungen 5.9 und 5.10 zu sehen ist.

Warum hat er so einen komplizierten Tabellentyp konstruiert? War es sein Ziel, die automatisierte Datenextraktion zu verhindern?

Andererseits können Texte aber auch von oben nach unten, und dann von rechts nach links, oder von rechts nach links, und dann von oben nach unten geschrieben und gelesen werden, was in alten Texten des Chinesischen und Japanischen, sowie im heutigen Arabisch praktiziert wird.

Im europäischen Raum hat sich eine Schreib- und Lesekonvention von links nach rechts, und dann von oben nach unten etabliert. Es ist anzunehmen, dass diese Konvention stark die Art des Tabellenaufbaus beeinflusst hat.

Dazu kommt noch ein anderer Faktor – die Browser-Breite – die wahrscheinlich nicht entscheidend ist. Da jeder Browser über eine andere Breite verfügt, kann es vorkommen, dass die zu präsentierenden Daten nicht in ein Browser-Fenster passen werden. Allerdings muss auch eingeräumt werden, dass die darzustellenden Daten auf der Informationsseite nicht für den Vergleich von mehreren Werten gedacht sind. Diese Tatsache und die vorwiegend europäische Schreib- und Leserichtung haben verhindert, dass der Tabellentyp „2“ nicht auf der Informationsseite auftrat.

[18] Es kommt natürlich vor, dass eine Zelle gleichzeitig das Attribut und den Wert durch
 getrennt enthält, was aber nicht Gegenstand dieser Diskussion ist.

Kapitel 6

Extraktion von Firmeninformationen

Beim Auffinden von Firmeninformationen geht es überwiegend um die Erkennung von Eigennamen[1].

Eigennamen sind laut Bußmann (1990, S. 204) [16] eine „semantisch definierte Klasse von Substantiven, die Objekte und Sachverhalte im Kontext eindeutig identifizieren“ und sind somit keine grammatikalischen Einheiten[2].

Aufgrund dessen kann die Identifikation eines Eigennamens nicht über die grammatikalische Analyse des Textes erfolgen. Um nun Eigennamen im Text automatisch zu erkennen, müssen die linken und rechten Kontexte der Eigennamen untersucht werden.

6.1 Zu extrahierende Klassen

Üblicherweise werden Eigennamen in die folgenden Kategorien eingeteilt[3]:

[1] siehe Kap. 2 von Rössler (2006) [114] zur Diskussion bzgl. der Unterscheidung von „Eigennamenerkennung“ und „Named Entity Recognition (NER)“.

[2] Kühnlein (2003, S. II-42) [77].

[3] Aufgabenbeschreibung der MUC 6 (Message Understanding Conference, 1995) für Eigennamen: `http://www.cs.nyu.edu/cs/faculty/grishman/NEtask20.book$_$3.html#HEADING4`.

- Klassen von Eigennamen
  - Named Entities (ENAMEX): ORGANIZATION, PERSON, LOCATION
  - Temporal Expressions (TIMEX): DATE, TIME
  - Number Expressions (NUMEX): MONEY, PERCENT

Von den eben genannten Kategorien sind die ersten beiden Hauptkategorien von Interesse, weil die dritte Informationen behandelt, welche in dieser Arbeit nicht von vorrangigem Interesse sind.

So handelt es sich bei mit **Organisation** klassifizierten Begriffen hauptsächlich um Firmennamen, während es sich bei mit **Person** bezeichneten Wörtern um Leute dreht, die verschiedenen Führungspositionen innerhalb der Firmenorganisation besetzen.

Als **Ortsangaben** werden Firmenadressen betrachtet, worunter Straßen- und Ortsnamen, sowie Postleitzahlen fallen. Jedoch werden E-Mail-Adressen, sowie Telefon-, Mobilfunk- und Faxnummern als Kontaktdaten gesondert gespeichert (vgl. mit dem Template aus Abschnitt 5.7 des Kapitels 5).

Dagegen fallen unter **Temporalia** Zeitangaben, wie Öffnungszeiten, Wochentage und Uhrzeiten.

6.2 Allgemeine Web-IE-Methoden und Informationsseiten

Bei der Informationsextraktion sind in Bezug auf die Eigennamenerkennung vier Ansätze zu unterscheiden[4]:

- Web-IE-Methoden
  - manuell aufgebautes System
  - überwachtes System
  - unüberwachtes System

[4] vgl. Kühnlein (2003, S. I-10) [77] und Kap. 4 von Rössler (2006) [114].

- semi-überwachtes System

Bei den MUC-Konferenzen und auch noch später – durch sie motiviert – wurden anfänglich die meisten Systeme manuell implementiert. Dabei waren die beiden Hauptaufgaben die „Lexikonerstellung“ und „Mustererkennung“. Die Pattern wurden von Fachleuten definiert und anschließend in eine reguläre Sprache, wie z.B. PERL oder in andere eigene Tools übersetzt. Dabei erfolgte die Extraktion über diese regulären Sprachen und die jeweiligen Lexika.

Jedoch stellte man sehr schnell fest, dass die manuell aufgestellten Extraktionsregeln selbst bei kleinen Änderungen an den Korpora oder bei analog erweiterten Korpora nicht wieder anwendbar waren. Da die Kosten der Systementwicklung und -wartung dadurch sehr hoch waren, mussten die Spezialisten über das entwickelte System sowie über gewisse domainspezifische Kenntnisse verfügen. Allerdings ist auch die Erstellung eines Lexikons und der notwendigen Regeln eine langwierige Arbeit, und meist sprachabhängig wie die Eigennamen selbst. Darum ist es kaum realisierbar, ein vollständiges Lexikon aufzubauen, denn es werden immer neue Eigennamen eingeführt werden, und die jeweilige Sprache wird sich weiterentwickeln.

Überwachte Systeme, so genannte maschinelle Lernsysteme, benötigen ein großes, manuell annotiertes Korpus, woraus die Extraktionsregeln gewonnen werden. Je größer das annotierte Korpus ist, desto besser werden die Regeln gelernt. Natürlich ist es keine leichte Aufgabe ein großes Korpus zu erstellen, doch können mit einem kleinen Korpus die Extraktionsregeln viel schlechter trainiert werden.

Insbesondere erschwert die Heterogenität der Informationsseiten dem System das Lernen von Regeln, wenn kein großes manuell annotiertes Korpus zur Verfügung steht. Ähnlich wie die manuell konzipierten Systeme sind auch die überwachten Systeme domainabhängig, so dass sie für neue Domains neue Trainingsdaten benötigen, um sich diesen anzupassen. In Anbetracht dessen machen maschinelle Lernsysteme nichts anderes, als Regeln mittels Training auf einem annotierten Korpus zu erstellen.

Auch geht die Tendenz dazu, dass immer mehr Webdokumente durch „Templates“ (Schablonen) aus Datenbanken generiert werden. Dies ist in der Regel dann der Fall, wenn so genannte „Mehrwertdienste“ (Value Added Services), wie z.B. Preisvergleiche oder speziell zusammengestellte Jobangebote, ange-

boten werden. Normalerweise wird bei diesen Webdokumenten eine iterative HTML-Struktur verwendet.

Bei unüberwachten Systemen steht diese iterative Struktur im Mittelpunkt. Dafür müssen nun kleinere Abweichungen von dieser Form durch verschiedene Ähnlichkeitsberechnungen kompensiert werden. Jedoch ist es kaum möglich, diese Methode analog auf die so unterschiedlich generierten Informationsseiten zu übertragen, da jede Website ihren ganz eigenen Stil hat, Daten zu präsentieren. Das erschwert nun erheblich den Struktur-Vergleich zwischen den jeweiligen Informationsseiten.

Deshalb musste eine andere Art der unüberwachten Systeme auf natürlichsprachliche Texte angewendet werden, welche beispielsweise bei Etzioni et al. (2005) [43] genutzt wird, um mit 8 domainunabhängigen Extraktionsmustern, wie z.B. *NP1 „such as“ NP2*, die entsprechenden Kandidaten zu erzeugen. Im Folgenden wird hierfür die entsprechende Regel angegeben:

> Predicate: Class1
> Pattern: NP1 „such as“ NPList2
> Constraint: head (NP1) = plural (label (Class)) & properNoun (head (each (NPList2)))
> Bindings: Class1 (head (each (NPList2)))

Falls es sich bei **Class1** um „City“ handelt, dann lautet die Regel *„cities such as“*, und somit werden die Köpfe derjenigen Eigennamen extrahiert, welche von dieser Phrase als potenzielle Städtenamen eingeleitet werden. Jedoch ist die Anwendung dieser Methode auf den entsprechenden Informationsseiten einer Firmen-Website nicht trivial, denn diese Daten werden häufig nicht über den Text wiedergegeben, sondern über die graphische Darstellung realisiert.

Silva et al. (2004) [121] und Downey et al. (2007) [35] verwenden N-Gramm-Statistiken, um Eigennamen bestehend aus mehreren Wörtern zu extrahieren. Dabei kam die PMI (= Pointwise Mutual Information) und SCP (= Symetric Conditional Probability) zum Einsatz, um die Relevanz und die Bindungsstärke (Assoziationsmaß) zwischen Wörtern zu berechnen. Diese Methode benötigt relativ große Textmengen, in denen die gleichen Wörter mehrmals vorkommen, um die Bindungsstärke von N-Grammen – speziell von Eigennamen – zu überprüfen. Jedoch ist dieses Kriterium nicht auf Informationsseiten zutreffend, da jede Informationsseite die gesuchten Daten

– ausgenommen von Attributen oder Ortsangaben, die mit der begrenzter Anzahl mehrmals auftreten können – nur ein Mal enthält.

Während es sich bei den unüberwachten Systemen häufig Clustering-Verfahren eingesetzt werden, macht man sich bei semi-überwachten Systemen die Bootstrapping-Methode zu Nutze. Dabei wird eine relativ kurze Liste mit einigen der gesuchten Instanzen vorgegeben. Dann sucht das System zuerst nach den Vorkommen in der gegebenen Dokumenten und sammelt deren Kontexte, in denen die Schlüsselbegriffe auftreten. Anschließend werden diese gemeinsam mit ihren Kontexten archiviert. Diese neu gefundenen Instanzen können nun zum Lokalisieren neuer Kontexte genutzt werden. Dieses augenscheinlich simple Verfahren gewinnt in letzter Zeit zusehends an Beliebtheit.

Während unüberwachte Systeme den gefundenen Instanzen keine semantische Annotation zuweisen können, sind semi-überwachte Systeme mithilfe der verwendeten, vorklassifizierten Liste von Instanzen in der Lage, die neu extrahierten Daten semantisch zu kategorisieren.

Diese Methode benötigt aber ähnlich wie unüberwachte Systeme eine ziemlich große Menge an Daten, um die einzelnen Vorkommen innerhalb der Dokumente zu identifizieren, und somit das Bootstrapping zu ermöglichen. Leider kann man dies nicht von Informationsseiten erwarten, da ein Firmenname auch mal nicht auf den verschiedenen Firmen-Websites vorkommt. So tritt jeder Eigenname nur ein Mal auf der jeweiligen Informationsseite der entsprechenden Website auf.

Informationsseiten einer Firmen-Website werden nur ein Mal präsentiert. Dadurch sind die vorhandenen Methoden schwer umzusetzen.

6.3 Methodische Überlegungen

Von allen vorgestellten Extraktionsmethoden sind es gerade maschinelle Lernsysteme, welche stark von der Struktur der Dokumente abhängen. Nur ist leider keine strukturelle Ähnlichkeit auf den verschiedenen Informationsseiten zu erwarten.

6.3.1 Domainspezifische oder -unabhängige IE

Im Allgemeinen hängt ein Extraktionssystem von einer bestimmten Domain ab. So wird zuerst eine Domain ausgewählt und anschließend auf ihr nach den gewünschten Informationen gesucht.

Dabei kann der Begriff der „Domain“ im Rahmen der Informationsextraktion aus Webseiten auf unterschiedliche Weise betrachtet werden: einerseits als fachgebunden und andererseits als strukturgebunden. So hat die fachgebundene Domain ihre eigene Terminologie – ihr spezielles Fachvokabular, besondere syntaktische Eigenschaften und gewisse semantische Relationen.

Dagegen hängt eine strukturgebundene Domain stark von den jeweiligen Textarten ab. Deshalb kann eine strukturbasierte Extraktionsmethode nicht auf unstrukturierte Texte angewendet werden.

Im Gegensatz zur Strukturiertheit wurde das Kriterium der Domain-Gebundenheit in den letzten Jahren abgeschwächt. Da immer mehr Webseiten strukturiert dargestellt werden, haben sich die Forscher verstärkt auf Strukturanalyse konzentriert. Jedoch wurde auch behauptet, dass die Informationsextraktion über Regeln, welche allgemeine syntaktische Eigenschaften definieren, wie bei Etzioni et al. (2005) [43], oder N-Gramm-Statistiken, wie bei Downey et al. (2007) [35] und Silva et al. (2004) [121], domainunabhängig erfolgen kann.

Die beiden Richtungen der domainunabhängigen Informationsextraktion kann aber nicht auf die Informationsseiten von unterschiedlichen Websites übertragen werden. Denn auch hier gilt, dass jede Informationsseite ihre eigene Struktur hat, und damit kann sie nicht mit anderen Informationsseiten verglichen werden. Denn allgemeine syntaktische Merkmale setzen grammatische Texte voraus, die auf der gesuchten Informationsseite schwer zu finden sind. Auch N-Gramm-Statistiken sind nicht effektiv, weil die gesuchten Informationen oft von den anderen Textabschnitten durch den „Delimiter“ abgegrenzt sind. Das macht eine N-Gramm-Statistik eher überflüssig.

Bei Grishman (2001) [59] wurde die Methode der domainadaptiven Informationsextraktion vorgestellt. Hierbei wird davon ausgegangen, dass die domainadaptive Informationsextraktion durch das Auffinden der jeweiligen domainspezifischen Wortklassen und derer Kombinationen möglich wäre.

Ähnlich wie Harris (1970) [66] ist auch er der Ansicht, dass sich alle komplexen Sätze auf kanonische einfachen Sätze (*kernel sentences* laut Harris (1970,

S. 387f) [66]) reduzieren lassen. Auf diese Weise können die domainspezifischen Charakteristiken der Wortklassen und ihre syntaktischen Regularitäten ermittelt werden. Ebenso ist er der Meinung, dass jede Domain über eine domainspezifische Subsprache[5] verfügt. So ist beispielsweise der Satz „*The polypeptides were washed in hydrochloric acid*“ akzeptabel, während der umgekehrte Satz „*Hydrochloric acid was washed in polypeptides*“ inakzeptabel ist.

Nach seiner Einschätzung soll die domainadaptive Informationsextraktion – im Speziellen aber die Mustererkennung oder -lokalisierung – auf der Basis von Wortfolgen, und nicht mithilfe syntaktischer Analysen durchgeführt werden.

Dass die domainadaptiven Informationsextraktionen nicht einfach ist, wurde in Dredze et al. (2007) [37] wie folgt beschrieben: „*no team was able to significantly improve performance on either test domain beyond that of a state-of-the-art parser*“, nachdem sie das mithilfe der Trainingsdaten entwickelte System auf eine andere Test-Domain übertragen wollten.

Rössler (2006) [114] betont auch, dass für die korpusadaptive[6] Eigennamenerkennung (NER) ein unüberwachtes Lernsystem aus großen Daten notwendig ist[7].

Wie bereits erwähnt wurde, sind unüberwachten Systeme schwer auf Informationsseiten anwendbar. Da die Informationsseite einer Firmen-Website einen informativen, komprimierten Teil enthält, können Ambiguitäten nicht aufgelöst werden, selbst wenn ein unüberwachtes System in der Lage wäre, die gesuchten Daten erfolgreich zu extrahieren.

Aufgrund dessen kommen bei diesem Ansatz keine domainunabhängigen IE-Methoden zum Einsatz. Hier wird zunächst ein Trainingskorpus zusammengestellt und von ihm gelernt, welche Eigenschaften eine Informationsseite hat. Die beobachteten Merkmale werden dann analog auf die Testseiten übertragen.

[5] s. Harris (1968) [65], S. 152ff.

[6] Domain-Adaptivität ist seiner Einschätzung nach unklar definiert. Der Begriff „Domain“ wird für viele Bereiche angewendet und zeichnet sich deshalb durch eine große Unschärfe aus. Siehe Rössler (2006, S. 82ff) [114] zum Begriff der „Korpus-Adaptivität“.

[7] Rössler (2006, S. 95) [114].

6.3.2 Subsprache und Vollständigkeit

Im Allgemeinen wird hierbei davon ausgegangen, dass die domainunabhängige Informationsextraktion nicht umfassend angewendet werden kann. Deshalb werden in dieser Arbeit zuerst die domainrelevanten Wortklassen und phrasenhaften Floskeln – so genannte „Floskelphrasen“ – gesammelt, um später damit nach den entsprechenden Informationen zu suchen. Ähnlich wird auch bei der Textklassifikation vorgegangen, da in Abhängigkeit der Textart gewisse Wortklassen und Floskelphrasen beobachtet werden können. So sind beispielsweise folgende Phrasen für eine Seite mit Stellenangeboten einer deutschen Firma typisch:

Wir bieten Ihnen
Wir suchen ab April
Sie haben Kenntnisse in
Haben wir Ihr Interesse geweckt
Auf Ihre Bewerbung freuen wir uns
Praktische Erfahrung sind von Vorteil
Als Qualifikation erwarten wir von Ihnen

Falls eine HTML-Seite die oben aufgelisteten Floskelphrasen enthält, kann sie aller Wahrscheinlichkeit nach als so genannte „Stellenangebotseite“ klassifiziert werden. Dieses Konzept basiert auf der Charakteristik von Subsprachen nach Harris (1968, S. 152) [65]. Laut ihm ist die Subsprache die spezialisierte Form einer allgemeinen Sprache, welche in einer bestimmten Domäne oder einem spezifischen Fach gebräuchlich ist. Die Subsprache zeichnet sich durch spezialisierte Wortklassen, semantische Relationen und vorwiegend durch eine spezielle Syntax aus.

Nach Luckhardt (1991) [92] kann eine Subsprache anhand der Textarten und Fachgebiete unterschieden werden. Dabei repräsentiert der Texttyp die syntaktisch-syntagmatische Ebene einer Sprache, während das Fachgebiet die lexikalische Ebene einer Subsprache prägt. In Bezug auf die Informationsseite einer Firmen-Website werden die beiden Ebenen einer Subsprache adäquat über die Semi-Strukturiertheit und das lexikalische Fachvokabular repräsentiert.

Zusammenfassend kann eine Subsprache durch folgende Eigenschaften charakterisiert werden[8]:

- Eine Subsprache
  - ist thematisch begrenzt.
  - unterliegt lexikalischen, syntaktischen und semantischen Restriktionen.
  - gleicht in ihren grammatikalischen Eigenschaften nicht der Allgemeinsprache
  - wiederholt gewisse lexikalische Strukturen relativ häufig.
  - ist in sich strukturiert.
  - verwendet eine gewisse Symbolik.

All diese Eigenschaften weist die Informationsseite einer Firmen-Website auf. So kann sich die Informationsextraktion diese Merkmale der jeweiligen Informationsseite zu Nutze machen, da ähnliche Attribute oder Daten nicht auf anderen Webseiten zu erwarten sind.

Für die Informationsextraktion auf einer bestimmten Domain sollten zunächst all diese Charakteristiken der betroffenen Domains ermittelt werden. Dazu gehören unter anderem das Fachvokabular, syntaktische Besonderheiten, und semantische Restriktionen. Deshalb ist es zwingend notwendig, alle Charakteristiken aus der Informationsseite einer Firmen-Website möglichst vollständig zu extrahieren, was während der Lernphase des Systems mithilfe von Test-Informationsseiten geschieht. Dafür wurden ca. 2 000 Informationsseiten während der Lernphase der Informationsextraktion analysiert.

6.3.3 Lokale Kontexte und Bootstrapping

Des Weiteren können die gesuchten Informationen auch durch lokale Kontexte – mit Attribut-Wert-Paaren – lokalisiert werden. Bei den meisten Systemen zur Informationsextraktion ist es üblich, die syntaktische Analyse zu überspringen und dafür die Extraktion mittels einer Wortfolgenanalyse zu bestreiten. Wie bei Grishman (2001) [59] erwähnt wird, soll das System

[8] Geierhos (2006, S. 18) [54].

möglichst syntaxunabhängig sein. Wortfolgenorientierte Systeme können leicht durch einen Automaten implementiert werden, wie es **Maurice Gross** (1997) [61] mit dem Formalismus der Lokalen Grammatiken gezeigt hat.

So repräsentieren Lokale Grammatiken die natürlichen Sprachen durch Wortfolgen, welche wiederum semantische Einheiten bilden. Da eine Sprache hierbei als Aneinanderreihung von Wörtern betrachtet wird, kann sie durch einen endlichen Automaten dargestellt werden, welcher durch einen Graphen visualisiert wird.

Maurice Gross geht davon aus, dass „*A scientist who accepts the theories of electromagnetism and of bubble nucleation will nevetheless search literally millions of images in order to find particles for which he has no thoeory*“ (Gross (1979, S. 880) [60]). Weiterhin ist er der Meinung, dass „*any of these types of grammar can be shown to have its validity restricted with rather simple dependancies holding between them*“ (Gross (1997, S. 330) [61]), und dass „*the model we advocate, and which we call it finite-state for short, is of a strictly* **local nature**“ (Gross (1997, S. 330) [61]). Genau diese lokalen Eigenschaften können nun mithilfe von Graphen visualisiert werden.

In Abbildung 6.1[9] wird deutlich, dass die Bindung zwischen Adverbien und *„speaking“* obligatorisch ist, und wenn das Adverb *„democratically“* allein vorkommt, ist der Satz nicht grammatisch.

(a) *Democratically, Bob is authoritarian
(b) Democratically speaking, Bob is authoritarian

Diese Art der Bindung (engl. „binding“) ist durch die endlichen Automaten in Abbildung 6.1 visualisiert. Dabei sollen die Abhängigkeiten zwischen Wörtern mithilfe der endlichen Automaten explizit konstruiert werden. „*The enormity of the number of dependencies between words is itself a compelling reason to consider the sort of fixed free-slot theory that finite-state local grammars suggest*“ (Gross (1997, S. 352f) [61]). Die auf diese Weise erstellten Lokalen Grammatiken können in verschiedenen Bereichen angewendet werden[10]. Beispielsweise finden Lokale Grammatiken in der Informationsextraktion ihre Verwendung, wie Mallchok (2004, S. 69) [93] aufzeigt: „*A local*

[9] Gross (1997, S. 332) [61].

[10] Für eine detaillierte Übersicht möglicher Anwendungsbereiche Lokaler Grammatiken siehe Mallchok (2004, S. 68ff) [93].

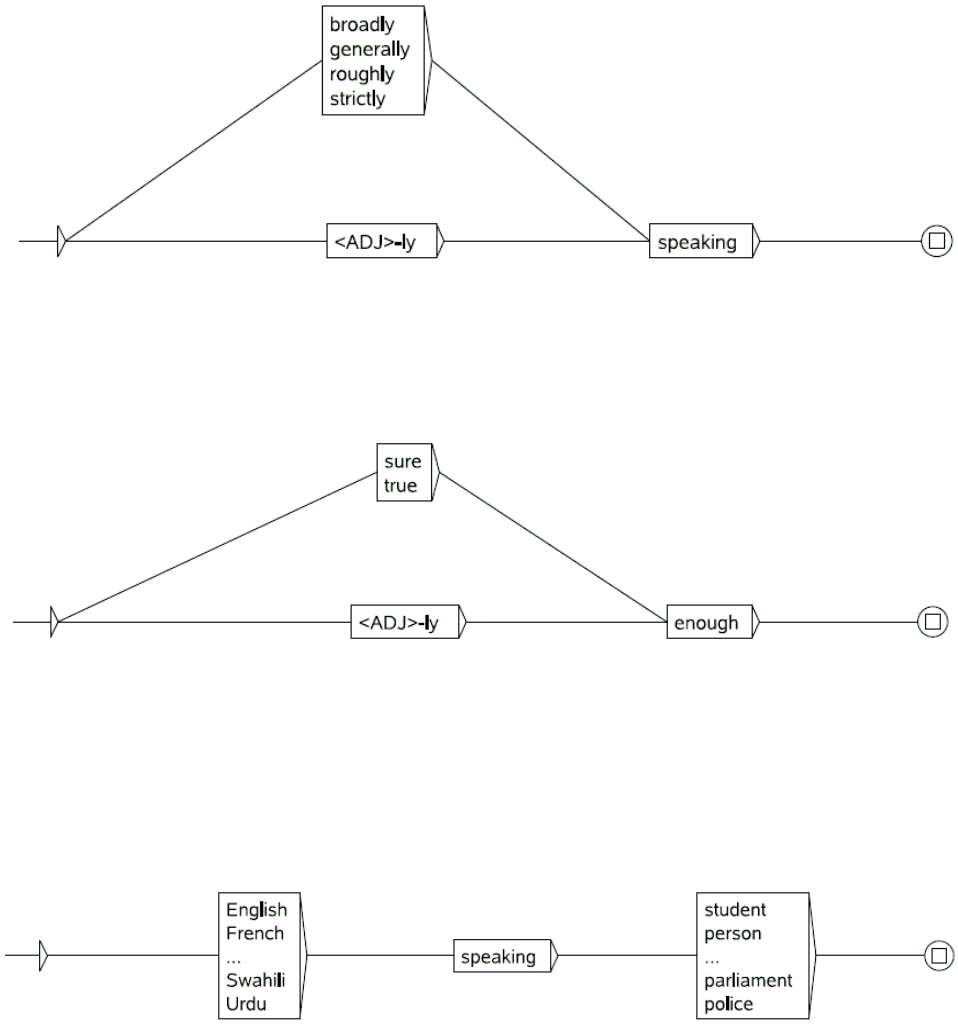


Abbildung 6.1: Beispielgraphen für Lokale Grammatiken

grammar provides the possibility to modulate exactly such context, both internal, representing the structure of the name ifself, and external, defining common surroundings of names.“

In Anbetracht der Perspektiven, welche Lokale Grammatiken bieten, wurde am CIS (Centrum für Informations- und Sprachverarbeitung) in den letzten Jahren verstärkt auf dem Gebiet der Eigennamenerkennung mittels Lokaler Grammatiken geforscht. So konzentrierte sich Mallchok (2004) [93] auf die Erkennung von Organisationsnamen in englischen Wirtschaftsnachrichten, wobei sie beachtliche Ergebnisse erzielte (z.B. ein F-Maß (engl. „F-measure“) von über 92%). Dagegen beschäftigte sich Geierhos (2006) [54] mit der Grammatik der Menschenbezeichner, wobei sie bei der Erkennung von Personennamen ca. 91% Präzision und ca. 85% Recall erreichte. Wiederum versuchte Bsiri (2007) [14], französische Stellenangebote auf Webseiten automatisch zu erkennen. All diese Arbeiten deuten darauf hin, dass Lokale Grammatiken ein adäquates Werkzeug zur Informationsextraktion sind.

Insbesondere macht die Softwareplattform `Unitex`[11] das Arbeiten mit Lokalen Grammatiken durch ihre Visualisierung einfacher, so dass Bootstrapping-Verfahren (Gross (1999) [62]) erfolgreich für die verschiedenen linguistischen Bereiche umgesetzt werden können.

Von allen Arbeiten mit Lokalen Grammatiken am CIS hat speziell Bsiri (2007) [14] gezeigt, dass sich Lokale Grammatiken auch sehr gut für strukturierte Texte einsetzen lassen. Zwar beschränkte sie sich auf Stellenangebotseiten, doch ähnlich wie bei dieser Arbeit waren diese HTML-Seiten stark durch ihre besondere Terminologie und syntaktischen Besonderheiten geprägt. Deshalb war es auch bei ihr notwendig, das vollständige Fachvokabular und alle syntaktischen Besonderheiten auf den betroffenen Webseiten zu ermitteln, wobei sich diese während der Lernphase des Systems durch die Analyse der untersuchten Seiten ergaben. Nachdem möglichst vollständige Stereotypen auf den Webseiten gesammelt wurden, wird das System auf neue Webseiten angewendet. Falls die gesuchte Information nicht gefunden wird, kann eine revidierte Lokale Grammatik zu den vorhandenen Graphen hinzugefügt werden. So werden die Grammatiken immer größer und vollständiger. Allerdings ist es auch möglich die komplizierteren, vollständigen Graphen durch Subgraphen übersichtlicher und verständlicher zu gestalten, um das System wartungsfreundlicher zu machen.

[11] `http://www-igm.univ-mlv.fr/~unitex/`.

Trotz der Anschaulichkeit und Wartungseffektivität von Lokalen Grammatiken wurde aufgrund der Besonderheiten der Daten auf einer Firmen-Website auf den Einsatz von Lokalen Grammatiken verzichtet.

Denn um vom Formalismus der Lokalen Grammatiken zu profitieren, muss die HTML-Seite in einfachen Text konvertiert und anschließend tokenisiert werden. Wenn diese Schritte übersprungen werden, können verschiedene Vorteile, wie der Einsatz von Lexika oder die Satzenderkennung, nicht ausgeschöpft werden. Des Weiteren gibt es auf der Informationsseite einer Firmen-Website in der Regel keine kontextuellen Sätze, da sie hauptsächlich über Attribut-Wert-Paare dargestellt werden. Selbst das Umwandeln einer HTML-Seite in Textform macht es nicht möglich, irgendwelche Kontexteigenschaften zu nutzen.

Jedoch ist das für die hier vorgestellten Zwecke nicht gravierend, da die graphenbasierte Bootstrapping-Methode auch von einem Trie umgesetzt werden kann. Dafür werden alle Kontexte für die jeweilige Klasse in einer Datei gespeichert, so dass auf diese Weise die Repräsentation durch einen Trie ermöglicht wird. Das hat zur Folge, dass der längste Eintrag zuerst gegen den vorgegebenen Kontext gematcht wird. Außerdem profitiert das System von der einfachen Dateiverwaltung, was die Wartung des Systems einfacher und übersichtlicher macht.

Zusammenfassend lässt sich somit in Bezug auf die Extraktionsmethode, welche für die Eigennamenerkennung auf der Informationsseite einer Firmen-Website eingesetzt wird, festhalten, dass folgenden Eigenschaften zum Erfolg der Datenidentifikation beitragen:

- Eigenschaften von Informationsseiten
  - semi-strukturiert
  - informationsintensiv
  - bestehend aus Attribut-Wert-Paaren

6.3.4 Interne und externe Indikatoren

Bei der Informationsextraktion auf Firmen-Websites bedient man sich der Domain-Abhängigkeit und lokalen Kontexte. Dabei werden interne und ex-

terne Indikatoren für die verschiedenen Klassen möglichst vollständig zusammengestellt.

Der Begriff „interne und externe Indikatoren“ wurde von McDonald (1993) [98] eingeführt[12].

> „*Internal evidence* is derived from within the sequence of words that comprise the name. This can be definitive criteria, such as the presence of known incorporation terms (Ltd., G.m.b.H) that indicate compannies; ... By contrast, *external evidence* is the classificatoy criteria provided by the context in which a name appears.“ (Mcdonald (1993, S. 32f) [98]).

Durch die Zusammenstellung von Indikatoren kann man auf ein großes Lexikon oder Ortsregister (engl. „gazetteer“), sowie auf Personennamenlisten verzichten. Deshalb wurden sie bei fast allen Systemen zur Eigennamenerkennung verwendet. So stellten Mikheev et al. (1999) [101], die am MUC-7-Wettbewerb beteiligt waren, fest, dass der Einsatz von umfangreichen Ortsregistern (Gazetteers) die Evaluation mittels Präzision und Recall gegenüber Systemen ohne Ortsregistern nur geringfügig verbessern kann. Deshalb haben sie ein System entwickelt, welches eine regelbasierte Grammatik mit dem maximalen Entropiewert (engl. „Maximum Entropy“) kombiniert. Ihre Regeln basieren hierbei auf internen und externen Indikatoren, sowie auf einem 5-stufigen Erkennungsmodell. Die folgende Tabelle fasst nun ihre Evaluationsergebnisse in Bezug auf die unterschiedlichen Lexikongrößen zusammen[13].

	Full gazetteer		Ltd gazetteer		Some locations		No gazetteers	
	recall	prec'n	recall	prec'n	recall	prec'n	recall	prec'n
organisation	90	93	87	90	87	89	86	85
person	96	98	92	97	90	97	90	95
location	95	94	91	92	85	90	46	59

In dieser Tabelle ist die Erkennungsrate der Ortsangaben bei dem System ohne Ortsregister (Gazetteers) besonders auffällig. Dies ist wohl darauf zurückzuführen, dass für Ortsangaben keine internen und externen Indikatoren vorhanden sind[14].

[12] Sein ursprünglicher Begriff lautet „internal and external evidence“.
[13] Mikheev et al. (1999, S. 7) [101].
[14] Mikheev et al. (1999, S. 7) [101].

Auch Lokale Grammatiken, welche zur Eigennamenerkennung eingesetzt werden, stützen sich nur auf interne und externe Indikatoren, wenn man von Listen absieht. Dabei ist entscheidend, wie vollständig die Merkmale zusammengestellt werden können.

Um die Vollständigkeit der gesammelten Indikatoren zu gewährleisten, wurden statistische und lexikalische Methoden eingesetzt. Dafür wurden ca. 2 000 Informationsseiten analysiert und jeweils interne und externe Indikatoren für die verschiedenen Klassen zusammengestellt. Zur Erstellung einer lexikalischen Grundlage wurden verschiedene Quellen aus der Wikipedia für Branchen- und Berufsbezeichnungen verwendet. Die Statistik in Tabelle 6.1 gibt über die verwendeten externen Indikatoren für die jeweilige Klasse Aufschluss.

	externe Indikatoren	Beispiele
Firmennamen	99	anbieter, firmenbezeichnung
Telefon	25	fon, tel, tel + fax
Fax	7	fax, faxnummer, telefax
Mobilfunk	13	mob, mobil, unterwegs
E-Mail	16	mail, E-Mail, m@il
Geschäftsführer	23	ceo, geschäftsführer, gf
Inhaber	16	inh, inhaber, owner
Kontaktperson	10	ansprechpartner, kontaktperson
Vorsitzender	23	chairman, leiter, vorsitzender
Vorstand	4	vorstand, geschäftsführender vorstand
USt-IdNr.	97	uid, ust-id-nr, umsatzsteueridentnr
Steuernummer	25	st. nr, steuer nr, umsatzsteuer nr
Registernummer	22	handelsr. nr, registernummer
Registergericht	28	ag, amtsgericht, gerichtsstand
Finanzamt	4	fa, finanzamt
Öffnungszeiten	5	bürozeit, geschäftszeiten, geöffnet

Tabelle 6.1: Statistik der externen Indikatoren

Dagegen sind interne Indikatoren für den Wert einer Klasse zuständig, während die externen für das Attribut selbst verantwortlich sind. Bei Mallchok (2004) [93] wurden einige positive, aber auch problematische Kontexte dieser Werte innerhalb der Attribut-Wert-Paare diskutiert. Beispielsweise zählen zu

den positiven Kontexten vor allem die Großschreibung und interne Indikatoren. Wie Rössler (2006) [114] erwähnt, ist die Großschreibung in deutschen Texten aber nicht besonders hilfreich, weil in deutschen Texten alle Nomen groß geschrieben werden. Dazu kommt auch noch die absichtliche Kleinschreibung eines Firmennamens, wie in Abbildung 6.2, was in den unstrukturierten Texten selten vorkommen würde.

```
<strong>
    Anbieter dieses Internet-Angebots im Sinne des TDG bzw. MDStV
    <br /><br />
</strong>
    edel AG<br />
    Neumühlen 17<br />
    D-22763 Hamburg<br />
```

Abbildung 6.2: Abschnitt von SLD „aaliyah“

Von den gesuchten Klassen weisen besonders Firmennamen viele interne Indikatoren auf. Für die Firmennamenerkennung wurden Rechts- und Gesellschaftsformen, sowie organisationstypisches Vokabular zusammengestellt. Andererseits werden auch die Branchen- und Berufsbezeichnungen berücksichtigt. Die internen Indikatoren für den Firmennamen werden im Abschnitt 6.4.1 behandelt.

6.4 Adresse und Kontaktdaten

Unter Adresse ist hier die Firmenadresse zu verstehen, welche den Firmennamen, die Straße, Postleitzahl und den Ort spezifiziert. Telefon-, Mobilfunk- und Faxnummer, sowie die E-Mail-Adresse werden extra in den Kontaktdaten aufgeführt und bilden alle eine Einheit. Diese sollen einen Bezug zum Domain-Namen aufweisen und genauso relevant wie dieser sein.

Bei Maynard et al. (2001) [97] wurden jede einzelne E-Mail, URL, Telefonie und IP unter „Adresse“ zusammengefasst, ohne überhaupt die Relevanz zwischen den Klassen zu überprüfen, weil es bei der Eigennamenerkennung so üblich wäre.

In seiner Magisterarbeit hat Mederle (2004) [100] vor allem folgende Probleme bei der Adressenerkennung erwähnt[15]:

- Fragestellung bei der Adressenerkennung
  - Abgleich von Telefonnummern mit Postleitzahlen
  - Kongruenz zwischen E-Mail und Domain-Namen
  - Approximative Suche einer Klasse
  - Verbesserung der Namenserkennung

Bei der Extraktion der Adressen und Kontaktdaten müssen gerade diese Probleme berücksichtigt werden, welche in dieser Arbeit weitgehend eine Lösung finden.

So erfolgt der Abgleich von Telefonnummern mit Postleitzahlen über das Lexikon. Dafür wurde ein Lexikon mit Postleitzahlen und Vorwahlen für Deutschland erstellt. Wird nun eine Telefonnummer gefunden, wird sie normalisiert, und es wird über den längsten Treffer in der Vorwahlliste die richtige Vorwahl ermittelt, welche anschließend mit dem Ortsnamen abgeglichen wird.

Die Kongruenz zwischen der E-Mail-Adresse und dem Domain-Namen wird über approximatives Matching überprüft. Falls der Hostname der E-Mail-Adresse so allgemein ist, wie „*t-online, web, freenet, ...*“, wird eine Übereinstimmung zwischen dem Benutzernamen und dem Domain-Namen untersucht.

Für die Namenserkennung werden interne und externe Indikatoren verwendet, wobei die externen Indikatoren die Klasse und die internen Indikatoren den Namen bestimmen. Speziell für den Firmennamen ist eine mehrfache Überprüfung mit verschiedenen Quellen notwendig: Falls keine bestimmten externen und internen Indikatoren existieren, wird versucht mit statistischen Mitteln über die kanonische Form des Firmennamens zum Ziel zu gelangen.

[15] Für die Grammatik und Rechtschreibung der deutschen Adressen und der einzelnen Klassen, siehe Mederle (2004), Kap. 1: Eine Grammatik deutscher Adressen.

6.4.1 Firmenname

Der Firmenname ist generell eine Unterkategorie (Subklasse) von Organisation. Nur in dieser Arbeit wird dieser allerdings als Oberbegriff benutzt. So fallen unter Firmennamen nicht nur die eigentlichen kommerziellen Firmennamen, sondern auch alle offiziellen domainnamenrelevanten Unternehmensnamen wie Vereinsnamen (*FC Bayern München*), Geschäfte aus dem Gesundheitswesen (*Apotheke am Rotkreuzplatz*), nicht-gewinnorientierte Stiftungen (*Dr. Marschall Stiftung*), usw. Jedoch werden Privatpersonen, sowie Blogs und Foren gezielt außer Acht gelassen.

Es gibt natürlich verschiedene Möglichkeiten, Firmennamen auf Webseiten zu identifizieren.

Da Firmennamen auf der Informationsseite nicht auf regulärem Text wie Nachrichtenartikel basieren, ist die linguistische Analyse der Seite für die Firmennamenerkennung nur wenig sinnvoll. Die häufig verwendeten Erkennungsmerkmale, wie Großschreibung am Wortanfang, können nicht immer eingesetzt werden. Denn einige Firmen beginnen ihren Namen bewusst mit einem Kleinbuchstaben (z.B. eBay). Andererseits ist der Einsatz eines vollständigen Lexikons nicht möglich, da immer wieder neue Firmen gegründet und neue Namen dafür erfunden werden.

Dennoch ist es möglich, Firmennamen auf der Informationsseite durch gewisse externe Indikatoren vorzuklassifizieren und zu kennzeichnen. Jedoch können die Daten auch ohne einen Indikator vorliegen. Außerdem können zu allgemein gewählte Indikatoren Trugschlüsse verursachen, wie z.B. „Name“ oder „Firma“ auch im Kontaktformular vorkommen. Wie bereits vorgeschlagen wurde, sollten für jede Klasse die korrekten Ausdrücke für den jeweiligen Wert ermittelt werden, um somit eine präzise Extraktion zu ermöglichen. Dafür muss deutlich werden, was zur internen Struktur eines Firmennamens gehört.

6.4.1.1 Grammatik der Firmennamen

Firmennamen haben in der Regel ihre eigene Grammatik. Trotz der Tatsache, dass fast alles ein Firmenname sein kann, können nicht alle Nominalphrasen (NPs) ein Firmenname sein.

Z.B. können *„Organisationsname“* oder *„Extraktionssystem“* selten ein Firmenname sein, während *„Institut für Organisationsnamen“* oder *„Pine Extraktionssystem GmbH“* wahrscheinlich für einen Firmennamen stehen würden. So sind Firmennamen häufig mit der Rechtsformen und weiteren Deskriptoren für Organisationen, wie z.B. *„Institut“* verbunden. Diese können aus verschiedenen Quellen stammen: Personenfirmen haben oft Personennamen im Firmennamen, während Sachfirmen[16] beliebige Firmennamen wie *„Infinion“* haben können.

Bei Mallchok (2004) [93] wurden die internen Merkmale zur Erkennung der Organisationsnamen in positive und problematische Klassen eingeteilt. So stellen die Rechtsformen wie „GmbH, AG, Corp, usw.“ ein positives Merkmal dar. Dagegen sind interne Kontexte, wie die Präpositionen „für, in“ oder Interpunktionszeichen („.“), problematische Merkmale. Außerdem hat sie 300 positive interne Indikatoren für Rechtsformen im englischen Wirtschaftsraum erfasst.

Wittek und Altfeld (1994) [131] haben 20 lexikalische Regeln zusammengestellt und sie in 7 Kategorien eingeteilt – Firmennamen entsprechen einer dieser 7 Kategorien. Sie versuchten dabei, Firmennamen mittels einer kontextfreien Grammatik zu generieren oder mit den entsprechenden Regeln zu rekonstruieren. Diese Vorgehensweise setzt aber ein gut klassifiziertes Lexikon voraus. So gehören bespielsweise *„Müller, Maier, ...“* der Klasse **Name** an, während *„Manfred, Martin, ...“* zur Klasse **Vorname** gehören. Mangels eines solchen Lexikons muss auf die Substitutionsregeln verzichtet werden. Dafür werden Klassen mit Buchstaben, Ziffern und Interpunktionszeichen genutzt.

Das folgende Schema illustriert einen vollständigen kanonischen Firmennamen.

- Schema für Firmennamen
  - Firmenname → **Container** Eigennamen Rechtsform
  - Firmenname → Eigennamen **Container** Rechtsform

[16] Sachfirma ist eine Firma, die nicht aus dem Namen des oder der Inhaber (Personen- oder Personalfirma), sondern aus dem Gegenstand des Unternehmens abgeleitet ist, z.B. Hamburger Holzhandels GmbH.

Unter „**Container**“ werden verschiedene Branchenbezeichnungen oder Betriebsformen fallen, wie sie Abbildung 6.3 zeigt:

Reisebüro Scholl GmbH
Kulmbacher **Brauerei** GmbH
Meisterbetrieb Völler GmbH & Co. KG
Brauerei C. & A. Veltins GmbH & Co. KG
DER *Deutsches* **Reisebüro** GmbH & Co. OHG

Abbildung 6.3: Beispiele für einen vollständigen Firmennamen

Aus Abbildung 6.3 ist ersichtlich, dass **Container** neben den meisten angenommenen Rechtsformen ein guter interner Indikator für einen Firmennamen sind. Dieser kann dem Eigennamen voran- oder nachgestellt werden. Eigennamen selbst können aus mehreren Wörtern zusammengesetzt werden, wie es das Beispiel „*C. & A. Veltins*“ zeigt. „*Kulmbacher*“ in *„Kulmbacher* **Brauerei** *GmbH“* und „*Deutsches*“ in *„DER Deutsches* **Reisebüro** *GmbH & Co. OHG“* wird hierbei nicht als Adjektiv betrachtet.

Um die Wortart berücksichtigen zu können, müssen linguistische Vorverarbeitungsschritte wie Tokenisierung, POS-Tagging und Lexikon-Look-Up miteinbezogen werden, welche im Moment nicht behandelt werden.

Die internen Kontexte wie *„&, ., -“*, welche z.B. von Mallchok (2004, S. 58ff) [93] als problematisch betrachtet wurden, werden hier kaum ein Problem darstellen. Sie kommen zwischen den sehr starken internen Indikatoren „**Container**“ und „**Rechtsform**“ vor und werden als ein Teil des Eigennamens gesehen.

Firmennamen auf Webseiten kommen aber nicht immer mit ihren vollständigen Formen vor. **Container** und **Rechtsform** können wegfallen, oder es ist schwer zwischen **Container** und Eigennamen zu trennen. Bei der *„Bayerische Motoren Werke AG (BMW)“* ist die Trennung zwischen **Container** und Eigennamen nicht einfach. Es kann gelesen werden als *„Bayerische Motoren“* und *„Werke“*, *„Bayerische“* und *„Motoren Werke“* oder einfach *„Bayerische Motoren Werke“*, auch wenn das Wort *„Werke“* im eigentlichen Sinne einen Arbeitsort meint. Bei der *„Gesellschaft für Schwerionenforschung mbH“* kann man vermuten, dass es überhaupt keinen Eigennamenteil gibt, obwohl das Wort *„Gesellschaft“* ein guter interner Indikator ist.

Da wir uns nicht auf die strikten Regeln für die Struktur eines Firmennamens konzentrieren wollen, werden solche Firmennamen als einfache Firmennamen betrachtet, ohne dabei zwischen **Container** und Eigennamen zu unterscheiden.

Ein anderes Problem bezüglich der Erstellung des **Container**-Lexikons stellen die besonderen Wortbildungsregeln der deutschen Sprache dar. Im Deutschen ist die Kompositabildung sehr produktiv. So können z.B. über 2 000 Komposita auf das Wort *„-gesellschaft“* für die Branchen- und Tätigkeitsbezeichnungen enden (*„Ingenieurgesellschaft, Vertriebsgesellschaft, Consultinggesellschaft, ...“*). Es werden daher möglichst allgemeine **Container** zusammengestellt.

Wie schon erwähnt können Firmennamen immer auch ohne **Container** und **Rechtsform** vorkommen. In diesem Falle soll eine allgemeine Eigennamenerkennung vorgenommen werden. So kann ein Firmenname mit der folgenden Grammatik[17] erkannt werden[18].

- Grammatik eines Firmennamens
  - (a) Firmenname → (?:Container)? Eigennamen Rechtsform
  - (b) Firmenname → Eigennamen (?:Container)? Rechtsform
  - (c) Firmenname → Firmenpräfix Eigennamen (?:Rechtsform)?
  - (d) Firmenname → Eigennamen Firmensuffix
  - (e) Firmenname → Berufsbezeichnung Eigennamen (?:Rechtsform)?
  - (f) Firmenname → Eigennamen
  - (g) Container → *Agentur, Apotheke, Betrieb, Büro, Center, Dienst, Firma, Geschäft, Gesellschaft, Handel, Hotel, Institut, Kanzlei, Klinik, Laden, Labor, Meister, Malerei, Praxis, Restaurant, Service, Shop, Studio, Unternehmen, Verlag, Werkstatt, Zentrale, Zentrum, ...*[19]

[17] Verwendete Grammatik und der reguläre Ausdruck entsprechen weitgehend der DCG (Definite Clause Grammaik) und PERL-RE.

[18] *Kursiv* geschriebene Zeichenfolgen sind terminale Symbole, während die anderen nichtterminale Symbole sind. Leerzeichen werden für die Lesbarkeit eingesetzt. Das echte Leerzeichen ist mit dem textvisuellen Leerzeichen („␣“) gekennzeichnet.

[19] Container ohne Schreibvariationen und Unterscheidung zwischen Klein- und Großschreibung.

(h) Rechtsform → *AG, AG & Co. KGaA, GbR, GmbH, GmbH & Co., GmbH & Co. KG, GmbH & Co. KGaA, GmbH & Co. OHG, Ltd, OHG, PartG, e.G., e.K., e.V., mbH, ...*[20]

(i) Firmenpräfix → *Gesellschaft für, Agentur für, Praxis für, Institut für, Kanzlei für, Zentrum für, Akademie für, Center for, Centrum für, ...*

(j) Firmensuffix → *& Sohn, & Kollegen?, & Collegen?, & Partner, & Team, und Sohn, und Kollegen?, und Partner, & Team, ...*

(k) Berufsbezeichnung → *Prof.?, Professor, Professorin, PD, Präsident, Präsidentin, Meister, Meisterin, Architekt, Architektin, Anwalt, Anwältin, Apotheker, Apothekerin, ...*

(l) Eigennamen → Eigenname ((?:LzbBs* (?:(?:DIP|Ziffer) LzbBs*)? Eigenname)+)*

(m) Eigenname → (?:KGZ)+

(n) KGZ → (?:Kb|Gb|Ziffer)

(o) DIP → (?:Det|IntP|Präpk)

(p) Kb → [a-zäöüß]

(q) Gb → [A-ZÄÖÜ]

(r) Ziffer → [0-9]

(s) LzbBs → [-␣]

(t) IntP → (?:[&\.\/\'\+]|und)

(u) PräpK → *für, zur, in*

(v) Det → *de[nmrs]?*

Die obere Grammatik kann sehr viele unsinnige Firmennamen erkennen. Dies wird auf der Informationsseite einer Firmen-Website durch die externen Indikatoren kompensiert. Da die Informationsseite informationsintensiv und semi-strukturiert ist, werden auch die externen Indikatoren stets angegeben.

Der Firmenname wird über eine mehrstufige Erkennungsmethode identifiziert. Für diesen Zweck wurden die externen Indikatoren mehrstufig klassifiziert. Das Fachvokabular, wie „*Firmenname, Anbieter, Betreiber, ...*“ wurde

[20] Rechtsformen ohne Schreibvariationen und Unterscheidung zwischen Klein- und Großschreibung.

als vorrangiger externer Indikator für einen Firmennamen gewählt. Wenn nun solche externe Indikatoren und Rechtsformen gefunden werden, wird die Wortfolge zusammen mit der Rechtsform als ein Firmenname erkannt. Falls zweitrangige externe Indikatoren wie „*Impressum, Anschrift, Adresse, ...*“ in der Wortfolge mit den Rechtsformen zusammen vorkommen, dann wird damit ein Firmenname erkannt. Sind keine unmittelbare externe Indikatoren vorhanden, werden die auf der Informationsseite spezifischen Kontextinformationen wie „*ist als Inhaltsanbieter, übernimmt keine Haftung, ...*“ genutzt.

6.4.1.2 Interne Indikatoren für Firmennamen

Wie im Abschnitt 6.4.1.1 erwähnt wurde, spielen interne Indikatoren eine wichtige Rolle für die Firmennamenerkennung. Interne Indikatoren können somit in drei Klassen unterteilt werden: Rechtsformen, Betriebsformen und typische Affixe für Firmennamen.

Rechtsformen und Firmennamen Rechtsformen, wie *„Gmbh“* , *„e.V.“* und *„e.K.“* kennzeichnen, dass die betroffene Wortfolge auf einen Firmennamen verweist.

Auf der Informationsseite kommen sie entweder in einer strukturierten, separaten Tabellenspalte oder zusammen mit den externen Indikatoren zwischen Delimitern, wie HTML-Tags, vor.

Während der Lernphase wurden möglichst vollständige Rechtsformen mit verschiedenen Schreibvariationen gesammelt (ca. 35 unterschiedliche). Diese werden bei der Erkennung des Firmennamens intensiv genutzt, nachdem ein „minimaler Datenbereich“ korrekt bestimmt wurde. Dann wird die durch Delimiter begrenzte Wortfolge mit Rechtsform als Firmenname akzeptiert. Über 40% aller bei der Lernphase extrahierten Firmennamen beinhalteten eine Rechtsform, wie z.B. „*AG, AG & Co. KGaA, GbR, GmbH, GmbH & Co. KG, KGaA, Ltd, ...* “.

Betriebsformen und Firmennamen Neben Rechtsformen liegt eine Betriebsform beim Firmennamen vor. Wie im Beispiel „**Meisterbetrieb** *Völler GmbH & Co. KG*“ sind diese Betriebsformen ein guter interner Indikator. Ein weiteres Problem stellt sich in deutschen Texten durch die vielen Komposita. So werden in diesem Beispiel zwei einfache Wörter wie *„Meister“* und

„Betrieb“ zusammengesetzt. Es gibt weit mehr als 1 000 Komposita, die im Deutschen auf *„Betrieb“* auslauten. Falls alle Betriebsformen in ein Lexikon aufgenommen werden müssten, würde das Lexikon dadurch sehr groß.

Außerdem können überhaupt nicht alle Komposita in einem Lexikon zusammengestellt werden, da ständig neue Komposita gebildet werden können. Daher ist es sinnvoll, möglichst allgemeine Betriebsformen in ein Lexikon aufzunehmen. Dadurch kann auch ein neues unbekanntes Kompositum erkannt werden. Zu diesem Zweck wurden ca. 130 allgemeine Betriebsformen zusammengestellt. Einige Beispiele davon sind in Tabelle 6.2 zu sehen. Branchenbezeichnungen sind aufgrund der Komplexität unter Betriebsform subsumiert.

Agentur	*Atelier*	*Betrieb*	*Büro*
Café	*Consulting*	*Dienst*	*Fabrik*
Gaststätten	*Geschäft*	*Gesellschaft*	*Hersteller*
Hotel	*Kanzlei*	*Klinik*	*Laden*
Labor	*Praxis*	*Spedition*	*Studio*

Tabelle 6.2: Beispiele von Betriebsformen

Berufsbezeichnungen und Firmennamen Berufsbezeichnungen können ein zuverlässiger interner Indikator sein, wenn es sich um Einzelunternehmen oder um eine kleinere Berufsgruppen wie Rechtsanwälte und Ärzte mit eigener Praxis handelt. So zeigen Beispiele, wie *„Malermeister Detlef Schwarz“* und *„Kfz-Mechanikermeister Günter Donner“*, dass der Firmenname durch das Schlüsselwort *„Meister“* deutlich wird. Zu diesem Zweck wurden insgesamt 400 allgemeine Berufsbezeichnungen gesammelt. Davon sind einige Beispiele in Tabelle 6.3 aufgeführt.

Typische Affixe und Firmennamen Außer beiden Rechts- und Betriebsformen kommen noch firmennamentypische Affixe ins Spiel. So sind die folgenden Affixe typisch für einen Firmennamen: *& Sohn, & Kollege, & Partner, Jun. ....* Diese Affixe werden genau dann, wenn keine andere interne Indikatoren zu finden sind, genutzt.

Agent	*Anwalt*	*Arzt*	*Elektriker*
Elektroniker	*Fotograf*	*Friseur*	*Gärtner*
Händler	*Importeur*	*Ingenieur*	*Installateur*
Kauffrau	*Kaufmann*	*Makler*	*Mechaniker*
Meister	*Optiker*	*Prüfer*	*Sachverständiger*

Tabelle 6.3: Beispiele für Berufsbezeichnungen

6.4.1.3 Relevanz zwischen Firmen- und Domain-Namen

Sicherlich spiegelt der Domain-Name den Firmennamen wider. So ist die SLD *„siemens“* der Domain-Name für das Unternehmen *„Siemens“*. Die Relevanzprüfung zwischen Domain- und Firmennamen kann sehr hilfreich sein, insbesondere wenn kein Anzeichen für den Firmennamen aus der Rechtsform, den jeweiligen Kontexten, den Meta-Informationen, oder Attributen zu vorhanden ist.

Dafür kann die Relevanzprüfung über verschiedene Wege erfolgen, wie z.B. über die Zerlegung des Kompositums. Die unbekannte SLD wird dazu als ein unbekanntes Kompositum angesehen. Um das unbekannte Kompositum nun in die bekannten Worteinheiten zu zerlegen, muss eine Referenzliste verwendet werden. Als Referenzliste können die Worteinheiten aus dem Titel und dem Adressblock genutzt werden, da der Titel oft die wichtigsten Informationen enthält – vor allem den Firmennamen. Natürlich sollte auch der Adressblock den Firmennamen beinhalten.

Dabei erfolgt die Firmennamenerkennung in zwei Schritten: Zuerst muss das unbekannte Kompositum erfolgreich segmentiert werden (*Segmentierung*), danach muss seine ursprüngliche Reihenfolge wieder hergestellt werden (*Wiederherstellung der ursprünglichen Reihenfolge*). Da die SLD nicht immer mit der Reihenfolge des Firmennamens übereinstimmen wird, ist dieser Schritt unerlässlich.

Segmentierung Zur Segmentierung wird das „Maximal-Forward-Match-Verfahren“ verwendet. Diese Methode wurde bei der Segmentierung der Sprachen, die keine natürlichen Leerzeichen zwischen Worteinheiten setzen, erfolgreich eingesetzt. Wird eine unbekannte Zeichenfolge eingegeben, so wird sie durch den längsten Match mithilfe der Referenzliste zerlegt. Die Aufgabe des

längsten Matches ist es, Komposita in möglichst lange Worteinheiten aus der Referenzliste zu zerlegen.

Chi & Ding (1999) [26] und Bsiri (2007, S. 13ff) [14], von denen der Algorithmus der „Segmentierung“ stark beeinflusst wurde, haben diese Methode auf die Segmentierung der SLDs übertragen.

Der Algorithmus startet dafür am ersten Zeichen, für das überprüft wird, ob es mit einem Wort aus der Referenzliste identisch ist. Wenn kein Wort mit ihm übereinstimmt, dann wird nachgeschlagen, ob es ein Substring eines Wortes ist. Dabei sollte der Substring am ersten Zeichen des Wortes beginnen. Wenn es sich um den Substring eines Wortes handelt, wird der Zeiger um ein Zeichen erhöht.

Dieser Vorgang wird so lange wiederholt, bis kein Substring aus der Referenzliste mehr ermittelt werden kann. Wenn keine Substrings mehr übrig sind, dann wird der Zeiger um ein Zeichen zurückgesetzt, und der bisherige Substring als die längste Einheit aus der SLD entfernt. Für den restlichen String der SLD wird der eben beschriebene Vorgang wiederholt.

Wenn kein String der SLD übrig bleibt, ist die Segmentierung erfolgreich beendet. Wenn nicht, dann ist die Segmentierung fehlgeschlagen.

Diese Vorgehensweise ist jedoch, realistisch betrachtet, nicht effektiv genug. Im Normalfall wird die SLD nicht mit den einzelnen Zeichen aus der Referenzliste gebildet. Aber wenn es der Fall sein sollte, kann das Akronym übernommen werden.

Deshalb wird diese Vorgehensweise einfach umgedreht: Zuerst wird der ganze String von der SLD mit den Worteinheiten aus der Referenzliste abgeglichen. Wenn kein Wort mit dem String identisch ist, wird der um ein Zeichen von hinten kürzere String mit den Worteinheiten verglichen. So wird die Anzahl des Look-Ups erheblich reduziert.

Der Algorithmus zur Segmentierung ist in Abbildung 6.4 dargestellt.

Aus dem Algorithmus wird ersichtlich, dass es drei Fälle der Zerlegung gibt. Einerseits wird geprüft, ob der ganze String mit einem Wort identisch ist. Wenn es ein solches Wort in der Referenzliste gibt, dann wird der ganze String zurückgegeben, und der Algorithmus stoppt. Andererseits wird geprüft, ob ein Wort den String als ein Präfix enthält. Wenn er Präfix eines Wort ist, dann wird der String zurückgeliefert, und der Algorithmus wird erfolgreich beendet.

Pseudo-Algorithmus zur Segmentierung(String, RL)

```
WDS = ( ); // Array für gematchte Segmente
i = length von String;
while i > 0
    Substr = substr (String, 0, i);
    for each Wd von Referenzliste
        if wd eq String              // Ein Wort gefunden
            return String;           // Keine Zerlegung nötig
        else if substr(Wd, 0, length(String)) eq String
            return String;           // Keine Zerlegung nötig
        else
            Wd_Substr = substr(Wd, 0, i); // Zerlege Wort bis zur Länge i
            if length(Substr) > 0 and Wd_Substr eq Substr
                push WDS, Wd_Substr;   // Wd in Array-Stack ablegen
                String =~ s/^Wd_Substr//; // Lösche gematchten Wd-Substr
                    i += length(String);       // Die Länge des
                                               // restlichen Strings
                                                // hochzählen
            endif
        endif
    endfor
    i--;
endwhile

if String eq ""  // Wenn String ganz segmentiert ist, dann Array zurückgeben
    return WDS;
else             // Segmentierung fehlgeschlagen; der ursp. String muss
    if Anzahl von WDS > 0                  // wieder hergestellt werden
        SEG = ( ); Array für Segmente
        do
        Segment = pop WDS;       // das letzte Segment vom Stack
        unshift SEG, Segment;    // nehmen und voranstellen
        while length(WDS) > 0;   // so lange bis WDS leer wird
        enddowhile               // Ursp. String ist die Zusammen-
        Ursp-String = join "", SEG.String; //  setzung von SEG
        return Ursp-String                 // und restlichen String
    else                          // Ursp. String zurückgeben
        return String             // Ansonsten wird nicht segmentiert
    endif
endif
```

Abbildung 6.4: Segmentierungsalgorithmus mit Maximal-Forward-Matching

Wenn diese beiden Versuche gescheitert sind, wird eine dritte Möglichkeit in Betracht gezogen. Zuerst wird der maximale Substring mit dem Wortsubstring abgeglichen. Wenn die beiden übereinstimmen, wird der Substring von dem ursprünglichen String getrennt und in den Array-Stack gespeichert, und der Zähler wird um die übrig gebliebene String-Länge hochgezählt. Durch das Inkrementieren wird garantiert, dass die Suche wieder mit dem verbliebenen String beginnt. Dieser Vorgang wird so lange wiederholt bis der String komplett zerlegt ist.

Falls der String nicht komplett zerlegt werden kann, sollte er wieder hergestellt werden. Dieser Vorgang wurde im zweiten Teil des Algorithmus implementiert. Der String wird durch Komposition zwischen dem restlichen Substring und den zerlegten Präfixen neu gebildet. Der Vorgang der Zusammensetzung muss umgekehrt durchgeführt werden, um den ursprünglichen String wiederherzustellen.

Dieser Vorgang soll nun am Beispiel der folgenden URL mit dem entsprechenden Seitentitel veranschaulicht werden:

URL: `http://www.tiho-hannover.de`
TITEL: TIERÄRZTLICHE HOCHSCHULE HANNOVER

Von der URL wird nur die SLD ausgewählt. Dagegen werden aus dem Titel alle Wörter, die durch Leerzeichen oder Bindestrich (-) getrennt sind, in die Referenzliste übernommen. Alle Wörter werden dazu in Kleinschreibung überführt und Umlaute werden in Langzeichen umgeschrieben. Somit liegt dann ein unbekanntes Kompositum, sowie die neue, unsortiert geordnete Referenzliste vor.

SLD: `tiho-hannover`
Referenzliste: *tieraerztliche, hochschule, hannover*

Da die SLD einen Bindestrich hat, wird sie zunächst an ihm zerlegt[21].

Der Algorithmus versucht zuerst den ersten Teil *„tiho“* mit der Referenzliste zu matchen. Da kein Wort mit dem String *„tiho“* identisch ist oder beginnt, wird der nächste, längste Substring *„tih“* genommen. Wenn wieder kein Wort

[21] Dieser Vorgang ist nicht im Algorithmus beschrieben. Wenn eine SLD einen Bindestrich (-) hat, dann wird sie an ihm geteilt.

mit dem String identisch ist oder beginnt, wird der nächste um ein Zeichen kürzere String *„ti“* verwendet. Das Wort *„tieraertzliche“* beginnt mit dem String *„ti“*. Dieser Substring des ersten Teils der SLD wird dann entnommen und auf den Array-Stack gelegt. Der übrig gebliebene String *„ho“* wird mit der Referenzliste abgeglichen. Da *„hochschule“* mit dem String *„ho“* beginnt, wird der String *„ho“* aus dem ersten Teil der SLD ausgewählt und an das Array angehängt. Dann ist der erste Teil der SLD leer und der Vorgang ist erfolgreich abgeschlossen.

Nachdem Algorithmus mit dem ersten Teil fertig ist, wird der zweite Teil der SLD, also *„hannover“*, mit der Referenzliste verglichen. Da das Wort *„hannover“* mit ihm identisch ist, wird der String *„hannover“* extrahiert und in das Array aufgenommen. Der zweite Teil der SLD ist somit auch leer. Der gesamte Vorgang hat nun erfolgreich terminiert, und der Array-Stack enthält jetzt 3 Einheiten: *„ti, ho, hannover“*.

Wiederherstellung der ursprünglichen Reihenfolge Die drei Strings im Array-Stack müssen nun auch noch mit der Referenzliste abgeglichen werden. Der Vorgang ist die umgekehrte Segmentierung, da das Wort, das mit dem String im Array-Stack beginnt, ermittelt werden muss. Das Wort *„tieraertzliche“* hat das Präfix *„ti“*, und kann somit ausgegeben werden. Das Wort *„hochschlue“* beginnt mit dem String *„ho“* und kann gedruckt werden. Der Vorgang wird nun so lange wiederholt bis der Array-Stack leer ist. Dabei werden alle Wörter mit groß geschriebenen ersten Zeichen ausgegeben, so dass der Firmenname dann *„Tierärztliche Hochschule Honnover“* lautet.

Dieses Beispiel lässt vermuten, dass der Schritt der Wiederherstellung redundant ist, weil statt der Substrings der SLD die ursprünglichen Wörter im Array-Stack angehängt wurden. Diese könnten ohne Weiteres ausgegeben werden.

Aber diese Methode wirft ein offensichtliches Problem auf: Es können mehrere Wörter mit dem gleichen String beginnen. Dann gibt es mehrere Wörter, die für einen String ausgegeben werden können. Dieses Problem wurde bei Bsiri (2007) [14] durch mehrere Referenzbasen, die aus mehreren Sequenzfolgen bestehen, gelöst.

Ein anderes Problem bezieht sich darauf, dass nicht alle SLDs die ursprüngliche Reihenfolge widerspiegeln werden. Wir haben z.B. die folgende SLD mit ihrem Titel:

SLD: `sgnh-la`
TITEL: Leichtathletikabteilung der SG Neukirchen-Hülchrath
Referenzliste: *leichtathletikabteilung, der, sg, neukirchen, huelchrath*

Am obigen Beispiel wird deutlich, dass die SLD die umgedrehte Reihenfolge des Titels darstellt. Abgesehen vom Kompositaproblem des Wortes *„Leichtathletikabteilung“* , spiegelt der erste Teil der SLD den zweiten Teil des Titels wider. Ohne den Wiederherstellungsprozess wird der Firmenname als *„SG Neukirchen Hülchrath Leichtathletikabteilung“* ausgegeben[22]. In diesem Fall entspricht die ausgegebene Reihenfolge noch nicht der eigentlichen Reihenfolge.

Um dieses Problem zu umgehen, wurde der Wiederherstellungsprozess eingeführt. Bei diesem Beispiel sind die Strings *„sg, n, h, la“* im Array-Stack. Nun beginnt die Verarbeitung mit dem ersten Wort der geordneten Referenzliste. Wenn die Abkürzung *„la“* des ersten Wortes *„Leichtathletikabteilung“* mit dem String *„la“* deckungsgleich ist, wird das Wort anschließend ausgegeben. Ansonsten ist der Prozess weitgehend identisch – bis auf die Ausnahme des Wortes „der“, was im nächsten Paragraph beschrieben wird.

Miteinbeziehen des Zwischenwortes Im Normalfall werden Stoppwörter wie *„der, des, für, usw.“* nicht in die SLD eingetragen. Wenn solche Wörter zwischen Wörtern auftreten, die Teile eines Firmennamens bilden, können sie einfach mit ausgegeben werden. Im obigen Beispiel ist der ganze Firmenname dann *„Leichtathletikabteilung der SG Neukirchen Hülchrath“*.

Allerdings kann zwischen *„Neukirchen“* und *„Hülchrath“* kein Bindestrich mehr eingesetzt werden, da bei der Bildung der Referenzliste der Bindestrich als ein Trennzeichen des Titels gewertet wurde, und ein Wort mit Bindestrich (-) wird generell in zwei Wörter zerlegt. Ohne diese Trennung kann der erste Teil der SLD nicht ganz zerlegt werden. Die Einheit *„SG“* wird groß geschrieben, was heuristisch gesehen der Normalfall ist, da die meisten Wörter bestehend aus lauter Konsonanten groß geschrieben werden. Deshalb wird diese Konvention ebenfalls bei der Ausgabe der Terme verwendet.

Diese Vorgehensweise kann ebenso auf andere Wortklassen übertragen werden. Dabei ist die einzige Bedingung, dass nur ein Wort zusätzlich zwischen

[22] Angenommen es gäbe eine Abkürzungsliste und für das Kompositum *„Leichtathletikabteilung“* wäre dort *„la“* gespeichert.

den für den Firmennamen gültigen Wörtern auftreten darf. Falls zwei oder mehrere Wörter zwischen den gültigen Wörtern erscheinen, werden sie aus statistischen Gründen nicht beachtet.

Das folgende Beispiel erläutert diese Methode:

> **SLD**: `rst-automation`;
> **TITEL**: RST Industrie Automation GmbH – Home
> **Referenzbasis**: *rst, industrie, automation, gmbh, home*

Die Konsonantenfolge *„rst“* ist identisch mit dem String im Array-Stack *„rst“*, und das Wort *„automation“* stimmt mit dem String *„automation“* überein. Somit wird das dazwischen liegende Wort *„industrie“* nicht übergangen: Der Firmenname lautet dann *„RST Industrie Automation“*[23].

Mehrere Wörter mit dem gleichen Präfix Es kann vorkommen, dass mehrere Wörter das gleichen Präfix haben. Für diesen Fall soll die Distanz zum nächsten Wort, welches mit dem String aus der Stringfolge im Array-Stack beginnt, gemessen werden. Dies wird anhand des folgenden Beispiels erklärt:

> **SLD**: `sah-eschweiler`
> **TITEL**: St.-Antonius-Hospital Eschweiler - Startseite
> **Referenzliste**: *st., antonius, hospital, eschweiler, startseite*

Die Wörter *„St.“* und *„Startseite“* beginnen beide mit dem String *„s“* im Stack. Da der String *„s“* vor dem String *„a“* vorkommt, sollte das Wort, welches *„s“* als Präfix hat, vor dem Wort, das mit dem String *„a“* beginnt, auftauchen. Im obigen Beispiel ist das Wort, das mit dem String *„a“* beginnt, *„antonius“*. Das Wort, das von *„s“* eingeleitet wird, sollte vor dem Wort *„antonius“* vorkommen, wie es auf *„st.“* zutrifft. Da das Wort *„startseite“* nach dem Wort *„antonius“* auftritt, wird es ignoriert.

[23] In diesem Beispiel wird die Rechtsform nicht erwähnt. Tatsächlich wird sie doch ausgegeben, weil die Rechtsform hinter dem erkannten Firmennamen stets genannt wird.

Wenn eine SLD nicht ganz segmentiert wird, wird sie einfach komplett zurückgegeben. Dieser Vorgang ist im zweiten Teil des Algorithmus in Abbildung 6.4 beschrieben. Der vollständig, unbearbeitete, zurückgelieferte String wird dann so ausgegeben, selbst wenn die SLD teilweise zerlegt werden kann, und ein Teil vollständig segmentiert wurde. Statistisch gesehen ist dies für eine Ortsangabe nichts Ungewöhnliches.

Dieser Prozess wird an folgendem Beispiel erläutert:

> **SLD**: `tt-berlin`
> **TITEL**: Jerry über Bord – Tritanitis-Theatergruppe
> **Referenzliste**: *jerry, ueber, bord, tritanitis, theatergruppe*

Nachdem der erste Teil der SLD komplett zerlegt wurde, wird versucht, den zweiten Teil der SLD zu segmentieren. Da das Wort *„bord“* mit dem Präfix *„b“* des Strings *„berlin“* beginnt, wird das Präfix *„b“* vom String entfernt. Nun beginnt kein Wort mit dem übrig gebliebenen String *„erlin“*. Der Algorithmus aus Abbildung 6.4 muss zunächst überprüfen, ob nicht ein leerer String eingelesen wurde. Ist dies aber dennoch der Fall, beginnt der *else*-Teil des Algorithmus.

Dieser nimmt die angehängten Strings wieder heraus, und stellt sie umgekehrt vorn an. Der zuletzt verwendete String wird ganz vorn angefügt. So wird die ursprüngliche Reihenfolge der Strings wieder hergestellt und zurückgegeben. Der zurückgegebene String wird dann einfach an den Array-Stack angehängt[24]. Danach erfolgt die Erkennung des Firmennamens mithilfe der sortierten Referenzliste. So kann beispielsweise *„Tritanitis Theatergruppe“* anhand der Strings *„tt“* im Array-Stack ausgegeben werden.

Es bleibt immer noch ein String im Array-Stack zurück: *„berlin“*. Da kein Wort in der Referenzlist mit *„berlin“* beginnt, wird dieser einfach mit ausgegeben. Dies hat zur Folge, dass der endgültige Firmenname zu *„Tritanitis Theatergruppe Berlin“* zusammengesetzt wird.

Da die Segmentierungsaufgabe eigentlich darin besteht, die unbekannte SLD sinnvoll zu zerlegen, schadet dieser Prozess nicht, sondern erweist sich eher als positiv.

[24]Dieser Vorgang wird nicht im Algorithmus dargestellt.

Wie weit Akronyme bei der Erkennung des Firmennamens gebildet werden sollen, ist nicht einfach abzugrenzen. Im oberen Beispiel kann man Akronyme nach Anzahl der Worte der jeweiligen Teile bilden, und entscheidet sich für das Wahrscheinlichste. Dafür werden zunächst die folgenden Akronyme für Bigrammen gebildet werden:

> Mögliche Akronyme für das Bigramm aus der Referenzliste:
> *ju, ub, bt, tt*

Für den Prozess der „Akronymbildung“ muss jedes Mal die Anzahl der Strings einer SLD berechnet werden. Ohne diese Beschränkung wären viele Akronyme einfach redundant. Für den String *„berlin“* muss z.B. kein Akronym gebildet werden, weil die Länge des Strings 6 beträgt und die maximale Länge für Akronym aus der Referenzliste nur bei 5 liegt. Somit ist hierfür die Akronymbildung sinnlos. Aufgrund dieser Überlegung wird auf die Akronymbildung bei der Erkennung – in Verbindung mit der Segmentierung – verzichtet.

Eine Stoppwortliste ist hierfür eigentlich nicht notwendig. Da das Zwischenwort einfach mit ausgegeben wird, kann auch das Stoppwort automatisch gedruckt werden, falls es zwischen Wörtern auftritt. Jedoch muss eine Stoppwortliste erstellt werden, um den Vorgang zu beschleunigen. Dies ist genau dann sinnvoll, wenn die Anzahl der gematchten Wörter nur um die Anzahl der Stoppwörter größer ist, als die Anzahl der Strings im Array-Stack beträgt.

In diesem Fall wird kein zusätzlicher Schritt benötigt, wenn ein Stoppwort zwischen den gematchten Wörtern vorkommt. Jedoch wird die Gesamtlänge um die Anzahl der Stoppwörter verringert. Falls die Anzahl zwischen den übrig gebliebenen Wörtern und Strings im Array-Stack gleich ist, können ohne Weiteres die ganzen Wörter zusammen mit den Stoppwörtern als ein Firmenname ausgegeben werden. Dabei kann jedes Stoppwort durch einen schnellen Look-Up gefunden werden.

Folgende Stoppwörter treten häufig zwischen Firmennamen auf:

> Stoppwortliste:
> *&, and, der, des, die, für, gegen, in, u., und, zum, zur*

Einschränkung Allerdings muss eingeräumt werden, dass der Algorithmus aufgrund der deutschen Komposita nicht immer funktionieren wird. Z.B. hat die SLD „ra-ksh-kulmbach“ den Titel *„Rechtsanwälte Käss, Schmidt, Hammon in Kulmbach“*. In diesem Fall kann *„ra“* nicht in der Referenzliste gefunden werden, weil das Wort *„Rechtsanwälte“* selbst ein Kompositum ist. Die Lösung dafür kann von zwei Richtungen aus betrachtet werden. Zunächst ließe sich das großes Lexikon der einfachen Lexeme anwenden, um zuerst das Kompositum *„Rechtsanwälte“* in *„Rechts“* und *„anwälte“* zerlegen. Andererseits könnte man die bereits erwähnte Abkürzungsliste erstellen. Für *„Rechtsanwalt“* ist die Abkürzung *„RA“* sehr gebräuchlich. Jedoch liegt derzeit keine solche Liste vor. Bislang wurden nur einige bekannte Abkürzungen zusammengestellt, von denen einige Beispiele in Tabelle 6.4 angegeben wurden.

Komposita	Abkürzungen
Krankenhaus	KH
Ärztekammer	AEK
Handwerkskammer	HWK
Volkshochschule	VHS
Informationstechnologie	IT
Bayerisches Rotes Kreuz	BRK
Rechtsanwalt / Rechtsanwälte	RA
Industrie- und Handelskammer	IHK

Tabelle 6.4: Komposita und ihre Abkürzungen

6.4.2 Wo residiert der Firmenname?

Firmennamen auf der Informationsseite können auf verschiedenen Wegen identifiziert werden. Am häufigsten tritt der Firmenname im Adressblock auf.

Firmennamen können auch gesondert in einem anderen Block vorkommen, oder die Erkennung des Firmennamens kann ohne interne und externe Indikatoren schwierig werden. In diesem Fall wird auf verschiedenen Stellen der Informationsseite nach dem Firmennamen gesucht.

6.4.2.1 Adressblock und Firmenname

Der zu suchende Firmenname befindet sich im Normalfall im Adressblock. Da der Firmenname als Adressat fungiert, kommt er oft vor der Straße vor. Zwischen Straße und Firmenname wird noch ein Delimiter, wie Komma oder
 eingesetzt. Zwischen Firmenname und Straße kann aber auch die Branchen- oder Inhaberinformation des Unternehmens genannt werden, wie es das Beispiel in Abbildung 6.5 zeigt.

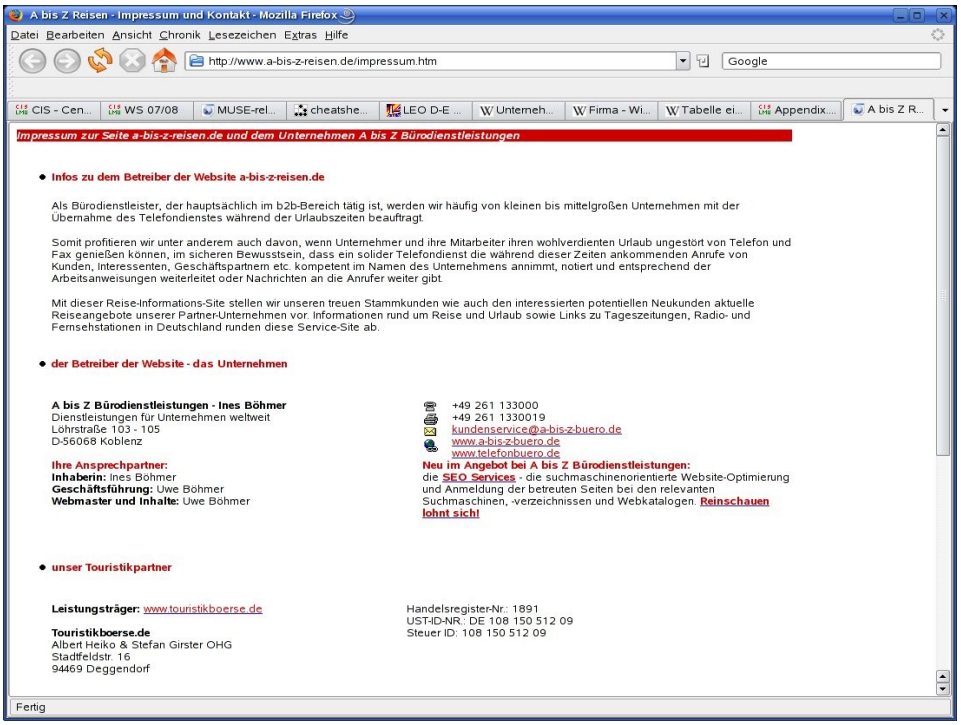


Abbildung 6.5: Beispiel: A bis Z Reisen – Impressum und Kontakt

A bis Z Bürodienstleistungen – Ines Böhmer
Dienstleistungen für Unternehmen weltweit
Löhrstraße 103 – 105
D-56068 Koblenz

Abbildung 6.6: Adressabschnitt aus Abbildung 6.5

Nachdem der Adressblock durch die Depth-First-Traverse (Tiefensuche) und

mittels des Attribut-Wert-Verfahrens bestimmt wurde, wird nach dem Firmennamen für den Domain-Namen gesucht. Bei der oben aufgeführten URL ist die Brancheninformation zwischen dem Firmennamen und der Straße eingefügt. Dabei ist die Brancheninformation nicht ganz irrelevant und hängt mit dem Firmennamen oder zumindest mit der Firmentätigkeit zusammen. Für die korrekten Daten darf aber diese Information nicht dem Firmennamen, sondern der Branche zugeordnet werden[25].

Aber auch die Rechtsform kann von dem Firmennamen durch einen Delimiter getrennt vorkommen. In diesem Fall wird der Delimiter vor der Rechtsform ignoriert, somit kann die Rechtsform an den Firmennamen angeschlossen werden.

6.4.2.2 Titel und Firmenname

Der Titel ist ein wichtiges Merkmal für den Firmennamen. Die wohl strukturierte Website liefert die wichtigsten Informationen, unter denen der Firmenname am markantesten ist, im Titel. In Abbildung 6.5 ist der Firmenname oder ein wichtiger Teil des Firmennamens im Titel zu finden. Abgesehen von der Brancheninformation, in diesem Fall *„Reisen“*, kommt auch der eigentliche Firmenname *„A bis Z“* im Titel vor.

Zur Identifikation des Firmennamens im Titel ist nicht die Impressumseite, sondern die allererste Index-Seite („Einstiegsseite“) wesentlich wichtiger. Leider enthalten die Impressum-Seiten als Titel nicht immer den Firmennamen, sondern den Webseitentitel wie *„Impressum“* oder *„Kontakt“*. Deswegen ist der Titel für eine Website eher auf der Einstiegsseite zu finden.

In Abbildung 6.5 ist der Titel auf der Einstiegsseite *„A bis Z Reisen: Reise- und Urlaubstipps, viele nützliche Reiseinformationen, günstige Flüge und Reisen bei A bis Z Bürodienstleistungen“*. Für den Fall, dass Firmen- und Domain-Namen relevant sind (siehe Abschnitt 6.4.1.3) und von der Brancheninformation aus dem Domain-Namen und Titel abgesehen wird, dann kommt die Zeichenfolge *„A bis Z“* zwei Mal im Titel vor, was höchstwahrscheinlich der Firmenname sein wird. So kann diese Zeichenfolge als Firmennamen extrahiert werden.

[25] Da die Brancheninformation nicht nur aus dem Adressblock heraus erkennbar ist und meistens mit Schlüsselwörtern der betroffenen Website verbunden ist, wird sie in dieser Arbeit nicht berücksichtigt.

6.4.2.3 Meta-Informationen und Firmenname

Es kann der Firmenname auch aus Meta-Informationen extrahiert werden. Diesbezüglich sind Meta-Informationen mit den Attributen „*Copyright, Publisher, Author*“ besonders relevant. Ähnlich wie der Titel auf der Einstiegsseite können ebenfalls die Meta-Informationen aufgespürt werden. Bei der gegebenen URL `http://www.a-bis-z-reisen.de` kommen zwei Meta-Informationen bezüglich der relevanten Attribute vor: *„Copyright“* und *„Publisher“*. Der davon betroffene Abschnitt wird in Abbildung 6.7 gezeigt.

```
<META NAME="copyright" CONTENT="Ines Boehmer -
        A bis Z Buerodienstleistungen; Uwe Boehmer">
<META NAME="publisher" CONTENT="Ines Boehmer -
        A bis Z Buerodienstleistungen">
```

Abbildung 6.7: Beispiel für Meta-Information

Die Werte für diese Attribute werden noch für die statistische Bewertung des Firmennamens aufbewahrt, so wie *„Ines Boehmer - A bis Z Buerodienstleistungen; Uwe Boehmer“* und *„Ines Boehmer - A bis Z Buerodienstleistungen“*. Sowohl der Wert für *„Copyright“* und *„Publisher“* als auch *„Author“* kann von dem echten Domain-Inhaber abweichen, falls die Website-Betreiber für den Inhalt und das Wegdesign Dritte beauftragt haben.

6.4.2.4 Copyright und Firmenname

Die Copyright-Information in der unteren Navigationsleiste enthält oft auch den Firmennamen, welcher den Inhalt der Website liefert. Ähnlich wie bei Meta-Informationen kann dieser Firmenname nicht immer mit dem Domain-Inhaber identisch sein.

Der Firmenname wird dann in der Regel entweder von „©“ oder *„Copyright“* eingeleitet, auf ihn folgt oft eine Jahreszahl und Copyright-Bemerkung wie z.B. *„All Rights reserved“* oder *„Alle Rechte vorbehalten“*. Die Reihenfolge der Elemente kann vertauscht sein. Die Beispiele in Abbildung 6.8 zeigen die verschiedenen Wortstellungen innerhalb des Copyright-Bereichs:

© 2007 edel AG
© Siemens AG 2002-2007
cy.control GmbH © 2007
2006 © Copyright Three X Media Corp
© Copyright Borak Werkzeuge und Maschinen
Copyright: Arnold Multimediacentrum GmbH 2005
Copyright © 2001-2006 Brennstoffhandel Martin Hackenbracht

Abbildung 6.8: Beispiel für Copyright

Nachdem die untere Navigationsleiste mit der Copyright-Information durch den natürlichen Delimiter zerlegt werden kann, ist es auch möglich, den Firmennamen aus dem Copyright-Bereich zu extrahieren.

6.4.2.5 Font-Informationen und Firmenname

Font-Informationen wie <H> oder <B> können auch berücksichtigt werden, falls der Text in diesem Tag-Bereich kein Attribut ist.

6.4.2.6 Voran- & nachgestellte Kontexte und Firmenname

Außer die für Firmennamen charakteristischen Attributklassen können auch die einleitenden und nachfolgenden Kontexte einen Hinweis auf einen Firmennamen liefern. Vorangestellte Kontexte können anhand ihrer semantischen Eigenschaften in drei Kategorien unterteilt werden:

- Kontextarten für Firmennamen
  - *Willkommen*
  - *Anbieter/Betreiber*
  - *Service*

Während *Willkommen* oft im Titel einer Website auftritt, kommen die anderen Kategorien meist auf der Informationsseite vor.

Unter *Willkommen* fallen wiederum folgende Kontexte, die in Tabelle 6.5 aufgeführt werden:

Willkommen beim	*Willkommen bei*	*Willkommen im*
Willkommen auf	*Welcome to the*	*Willkommen auf den Seiten der*

Tabelle 6.5: Beispiele von Willkommenskontexten

Dagegen weichen die Kontexte für *Anbieter/Betreiber* von den Attributen der anderen Firmennamen ab. Bei *Anbieter/Betreiber* sind die Attribute vorrangig, sofern sie im strukturierten Abschnitt oder zwischen gewissen Delimitern vorkommen. Jedoch werden solche Attribute nicht als hinweisende Kontexte aufgefasst, sondern als natürlichsprachliche Kontexte auf der Informationsseite angesehen, was in Tabelle 6.6 veranschaulicht wird.

Herausgeber dieser Webseiten ist die
Dienstanbieter dieser Seiten ist die
Dies ist der gemeinsame Internetauftritt der Firmen
Anbieter dieser Internetpräsenz ist im Rechtssinne der
Diese Webseite wird von
Diese Website wird im Auftrag des

Tabelle 6.6: Beispiele für Anbieter-Kontexte

Unter *Service*-Kontexte fallen nun die Beispiele aus Tabelle 6.7. Sie spezifizieren, dass die betroffene Website ein Dienst oder Angebot von einem Anbieter – von einer Firma – ist. Firmennamen, die von solchen Kontexten eingeleitet werden, müssen auch für die spätere statistische Bewertung archiviert werden.

Diese Dienstleistung ist ein Projekt der
Dieser Dienst ist ein Angebot von
Dieses Internetangebot wird unterhalten von
Diese Internetseite gehört und wird unterhalten von
Dies ist die offizielle Homepage des

Tabelle 6.7: Beispiele für Service-Kontexte

Beispiele mit nachgestellten Kontexten werden in Tabelle 6.8 aufgelistet. Die nachfolgenden Kontexte beziehen sich oft auf die gesetzlichen Bestimmungen,

in denen der Anbieter des Internetauftritts keine Haftung oder Garantie für die fremden Inhalte der Internetpräsenz übernehmen will. In diesem Fall wird häufig das Verb in seiner Singularform verwendet. Ansonsten ist es wahrscheinlich, vor dem Verb ein Pronomen wie z.B. *„Wir“* vorzufinden.

ist als Inhaltsanbieter nach § 6 des Teledienstgesetzes
ist als Inhaltsanbieter
übernimmt keine Garantie für die Vollständigkeit
übernimmt keine Haftung
übernimmt keinerlei Gewähr für
haftet weder für direkte noch indirekte

Tabelle 6.8: Beispiele für nachgestellte Kontexte von Firmennamen

Im Gegensatz zu den voran- und nachgestellten Kontexten können die folgenden Texte aus Tabelle 6.9 für das Attribut-Wert-Verfahren verwendet werden. Auf diese Weise kommen sie in einer strukturierten Umgebung oder zwischen zwei Delimitern vor und sind ein eindeutiger externer Indikator für die Firmennamenerkennung.

Anbieterkennzeichnung gem. § 6 Teledienstgesetz
Anbieter gemäß TDG und verantwortlich gemäß MDStV
Verantwortlich für den Content und Anbieter im Sinne §6 TDG
Betreiberangaben entsprechend §6 TDG
Anbieterkennzeichnung gemäß den Inhalten des Teledienstgesetzes (TDG)
verantwortlich für den Inhalt laut Mediendienste-Staatsvertrag (MDStV)
Anbieter dieses Internet-Angebots im Sinne des TDG bzw. MDStV

Tabelle 6.9: Beispiele für externe Indikatoren von Firmennamen

6.4.3 Straßennamen

Zu einer Straße gehört in der Regel auch eine Hausnummer. Nur in seltenen Fällen kommt ein Straßenname ohne die entsprechende Hausnummer vor. Normalerweise ist dies nur üblich, wenn die Adresse auch ohne eine Hausnummer eindeutig zuzuordnen ist. Außerdem tritt sie meist vor der Postleitzahl

auf, und ein Straßenname hat oft ein Suffix wie „*str, weg, allee, ...*“. Rund 80% aller deutschen Straßennamen können bei der Lernphase durch das Suffix erkannt werden. Die am häufigsten vorkommende Suffixvariante ist „*str*“ mit ihren Variationsmöglichkeiten „*Str., str., Straße, straße, Strasse, strasse, Str, str*“.

Da die Identifizierung der Straßennamen ohne ein Lexikon erfolgt, sollte neben diesen Faktoren auch die interne Struktur des Straßennamens erkannt werden, um eine möglichst hohe Präzision zu erzielen. Nach Mederle (2004) [100] hat ein Straßennamen die folgende Grammatik:

(a) Straße → Postfachbezeichner Postfachnummer

(b) Postfachnummer → (\d|␣){1,8}

(c) Postfachbezeichner → *Postfach, pf., Postf., PF*

(d) Straße → Straßenname Hausnummer

(e) Hausnummer → ^\d{1,3}\D? (-\d{1,3}\D?)? (/.*)?$

(f) Straßenname → Name Straßengrundwort

(g) Straßengrundwort → *Straße, Ring, Alle, ...*

(h) Straßenname → GEO-er Straßengrundwort

(i) Straßenname → Vorname-Nachname-Straßengrundwort

(j) Straßenname → Titel-Vorname-Nachname-Straßengrundwort

(k) Straßenname → Titel-Nachname-Straßengrundname

(l) Straßenname → Straßengrundwort Determinator Nominalphrase

(m) Straßenname → Lokalpräposition Nominalphrase

(n) Straßenname → *Alte, Neue, ...* Nominalphrase

(o) Straßenname → Sonstiger Straßenname

Abgesehen von der fälschlich bezeichneten „Nominalphrase“ [26] und von einigen Schreibvariationen kann diese Grammatik auf die Erkennung von Straßennamen weitgehend übertragen werden. Als einzige Bedingungen bleiben noch ein verfügbares, vorklassifiziertes Lexikon, und dass der Text auf der Informationsseite tokenisiert und getaggt wurde[27].

6.4.3.1 Grammatik der Straßennamen

Da bei der Erkennung der Straßennamen weder Ortsregister (Gazetteers) noch eine linguistische Analyse zu Einsatz kommt, und trotzdem eine hohe Präzision erzielt werden muss, ist es unerlässlich, die verschiedenen Typen der Straßennamen anders zu klassifizieren.

Da das erste Wort eines Straßennamens und alle zum Eigennamen gehörenden Adjektive und Zahlwörter groß geschrieben werden, wird zwischen Eigennamen und anderen Wortarten innerhalb des Straßennamens, die mit Großbuchstaben beginnen, nicht unterschieden. Stattdessen werden alle als Eigennamen betrachtet. Was nun übrig bleibt, sind Präpositionen und Determinatoren, welche zwischen den einzelnen Wörtern klein geschrieben werden. Darunter fallen in der Regel Titel und Zeichenfolgen mit Interpunktion. Jedoch muss die maximale Wortanzahl begrenzt werden[28]. Außerdem wird die Postfachangabe anstelle eines Straßennamens und der Hausnummer unangetastet bleiben.

- Grammatik der Straßen
  (a) Straße → Straßenname Hausnummer
  (b) Straßenname → Eigennamen Straßengrundwort
  (c) Straßenname → Eigennamen LzbBs? Straßengrundwort[29]

[26] Eine Nominalphrase (NP) umfasst in der Regel auch einen Determinator. Ein Verbesserungsvorschlag für diese Grammatik wäre, dass hier der Begriff „Name“ verwendet werden sollte. (vgl. *Platz der Opfer des Nationalsozialismus* oder *Platz der Freiheit*). Außerdem kann eine Nominalphrase auch expandiert werden.

[27] Mederle (2004) verwendet ein Straßennamenlexikon mit 212 000 Einträgen.

[28] Nach Mederle (2004, S. 42) kann ein deutscher Straßenname aus maximal 6 Wörtern bestehen.

[29] Leerzeichen werden nur für die Lesbarkeit eingesetzt.

(d) Straßenname → Gb (?:LzbBs KGPD LzbBs)+ Eigennamen LzbBs? Straßengrundwort

(e) Straßenname → PräpG LzbBs (?:KGPD LzbBs)? Eigennamen (?:LzbBs? Straßengrundwort)?

(f) Straßenname → PräpG LzbBs KGPD LzbBs (?:Ziffer){1,3} LzbBs Eigennamen

(g) Straßenname → Titel LzbBs (?:KGPD LzbBs)? Eigennamen (?:LzbBs? Straßengrundwort)?

(h) Straßenname → Eigenname LzbBs KGPD LzbBs Eigenname (?:LzbBs? Straßengrundwort)?

(i) Straßenname → Straßengrundwort (?:LzbBs KGPD LzbBs)? Eigenname (?:LzbBs KGPD LzbBs Eigenname)?

(j) Straßengrundwort → *Allee, Bach, Berg, Berge, Blick, Brücke, Bühl, Burg, Busch, Chaussee, Damm, Deich, Dorf, Ecke, Feld, Garten, Gasse, Graben, Grund, Hain, Haus, Hausen, Heide, Heim, Höhe, Hof, Hütte, Kamp, Leite, Markt, Mühle, Park, Pfad, Platz, Promenade, Ring, See, Siedlung, Steg, Steig, Steige, Stieg, Straße, Tal, Teich, Twiete, Ufer, Wald, Weg, Wiese, Wiesen, ...*[30]

(k) Eigennamen → Eigenname (?:LzbBs Eigenname){,3}

(l) Eigenname → [A-ZÄÖÜ][A-ZÄÖÜa-zäöüß]{3,}

(m) Kb → [a-zäöüß]\.?

(n) Gb → [A-ZÄÖÜ]\.?

(o) Ziffer → [1-9][0-9]{,2}

(p) LzbBs → [-␣]

(q) PräpG → *Am, An, Auf, Bei, Beim, Hinter, Im, In, La, Nach, Unter, Vor, Von, Zum, Zur*

(r) PräpK → *am, an, auf, bei, beim, hinter, im, in, la, le, nach, unter, vor, von, zum, zur*

(s) Titel → *Bischof, Dr\.?, Kardinal, Prof\.?, Professor, St\.?*

(t) Det → *de[nmrs]?*

[30]Straßengrundwort ohne Schreibvariationen und Unterscheidung zwischen Klein- und Großschreibung.

(u) KGPD → (?:Kb|Gb|PräpK|Det)

Die folgenden Beispiele in Tabelle 6.10 zeigen einige Straßennamen, die mit der obigen Grammatik erkannt werden können:

Maximilianstraße	*Römerplatz*
Münchner Straße	*Potsdamer Str.*
Am Marktplatz	*Am Alten Lindenbaum*
An den 12 Bäumen	*An den 13 Morgen*
Kardinal-Faulhaber-Straße	*Prof.-Huber-Platz*
Platz der Freiheit	*Platz der Opfer des Nationalsozialismus*

Tabelle 6.10: Beispiele für Straßennamen

6.4.3.2 Normalisierung der Straßennamen

Für das Grundwort *„Straße“* kommen generell vier Realisierungen in Frage: *„Straße, Strasse, Str., Str“*. Nur durch Normalisierung lassen sich diese Variationen in eine einheitliche Form überführen. Dies bringt einige Vorteile mit sich, wie z.B.:

- Schnelles Matching
- Einheitliche Verwaltung der Daten

Alle vier Variationen für das Straßengrundwort *„Straße“* werden entweder durch *„oder“* oder mit sinnvoll gebildeten regulären Ausdrücken beim Matching-Prozess abgeprüft. Der Prozess ist im Vergleich zu dem einfachen String-Matching ineffektiv. Aber durch die Normalisierung der Varianten zu einer einheitlichen Form kann die Komplexität des String-Vergleichs auf *O(1)* reduziert werden und ist somit konstant.

Andererseits ist die Verwaltung der gewonnenen Daten aufgrund der Vielfalt an Wortvariationen nicht eindeutig. Denn *„Münchner Straße“* und *„Münchner Str.“* sind keine unterschiedlichen Straßen. Für eine eindeutige Zuordnung der gewonnenen Daten ist die Normalisierung unerlässlich.

Bei Straßennamen kommen viele Variationen vor: *Al.* für *Allee*, *Pl.* für *Platz*, *Wg* für *Weg*, *Bg* für *Berg*, *Siedl* für *Siedlung*, usw. Außer des Straßengrundwortes können Präpositionen, Determinatoren und Titel in einer abgekürzten Form auftreten.

Bei der Normalisierung der Straßennamen müssen zwei Faktoren bedacht werden:

- Vereinfachte Form
- Mögliche Rückführung auf die ursprünglichen (Original-)Straßennamen

Das Finden einer vereinfachten Form ist für das Matching und die Verwaltung der gewonnenen Daten relevant. So gilt für das Matching und die Datenverwaltung, dass das kürzere Wort die effektivsten Ergebnisse liefert. Dabei besteht allerdings weiterhin das Problem, dass der Original-Straßenname meist nicht wiederzuerkennen ist. Nachdem die kürzere Variante *„A d 12 Bäumen“* für *„An den 12 Bäumen“* eingesetzt wurde, kann man schwer den ursprünglichen Straßennamen ermitteln. Denn es könnte auch genauso *„Auf den 12 Bäumen“* heißen. Bei einem abgekürzten Personennamen ist dies noch schwieriger: *F.* in *F.-Joseph-Str.* kann *Frank*, *Franz* oder *Friedrich* bedeuten.

Aus diesem Grund wird die Normalisierung möglichst mit der ausbuchstabierten Variante vollzogen. Sollte dies aus irgendwelchen Gründen nicht möglich sein, wird die abgekürzte Form verwendet. Auch Schreibvarianten werden in eine einheitliche Form überführt.

Einige Beispiele für Schreibvariationen und deren jeweilige normalisierte Formen werden in Tabelle 6.11 dargestellt:

6.4.4 Postleitzahlen und Ortsnamen

Am häufigsten treten Postleitzahlen und Ortsnamen im Adressblock auf, in dem sie in direkter Nachbarschaft vorkommen. Eine deutsche Postleitzahl besteht aus einer 5-stelligen Ziffernfolge, der oft ein Präfix für Deutschland wie „*D-, DE-*“ vorangeht, wobei dieses nicht als Bestandteil der Postleitzahl gesehen wird. Theoretisch kann die Anzahl möglicher Postleitzahlen 100 000 betragen. Jedoch liegt sie in der Praxis bei ca. 29 600. So kann sich eine kleine Ortschaft eine Postleitzahl mit einem anderen Dorf teilen. Wenn man

A → Am	*Str → Straße*	*Pk → Park*
Pl → Platz	*St → Sankt*	*Wg → Weg*
Hf → Hof	*Hs → Haus*	*Pf → Pfad*
Wk → Werk	*Bn → Bahn*	*Ch → Chaussee*
Bk → Brücke	*Rg → Ring*	*Bge → Berge*
Kan → Kanal	*Fabrk → Fabrik*	*Schloßp → Schloßpark*
Prüfungsan → Prüfungsanlage	*Schl → Schloss*	*Stdt → Stadt*
Anlg → Anlage		

Tabelle 6.11: Schreibvariationen bei Straßenangaben

nun die Kombinationen von Postleitzahlen mit Ortschaften berechnet, betragen sie ungefähr 46 000. Dagegen kann eine große Stadt eine Vielzahl an Postleitzahlen haben, was die Zahl der eindeutigen Ortsnamen auf ca. 16 000 reduziert.

In verschiedenen Ländern, wie Frankreich oder in den USA, wird auch eine 5-stellige Postleitzahl verwendet. Um Verwechselungen bei der Zuordnung von Postleitzahlen zu Ortsnamen zu vermeiden, wurde ein Lexikon aus Kombinationen von Postleitzahlen mit ihren entsprechenden Ortsnamen erstellt worden. Dieses liegt in Form einer Hash-Tabelle vor, in der die Postleitzahl als Schlüssel und der Ortsname als Wert fungiert. Deshalb wird beim Matching-Verfahren zuerst nach der Postleitzahl gesucht, und anschließend nach dem benachbarten Ortsnamen. Dabei muss aber der gefundene Ortsname mit dem Lexikoneintrag des Wertes approximativ abgeglichen werden.

6.4.4.1 Postleitzahlen

Bei der Erkennung von Postleitzahlen, welche den ersten Anhaltspunkt für die Informationsextraktion auf der Informationsseite liefern, wird neben dem regulären Ausdruck für Postleitzahlen überwiegend das Lexikon eingesetzt. Ohne Lexikon würden viele, nicht zu den Postleitzahlen gehörige, 5-stellige Ziffernfolgen als solche erkannt werden, und die Informationsextraktion würde fehlschlagen.

6.4.4.2 Ortsnamen

Ortsnamen können aus einem Wort, aus mehreren Ortsnamen (z.B. *„Garmisch-Partenkirchen“*) oder Mehrwortlexemen (z.B. *„Au in der Hallertau“*) bestehen. Eine Grammatik für Ortsangaben sieht beispielsweise wie folgt aus[31]:

- Grammatik des Ortsnamens
  - (a) Ortsangabe → Postleitzahl Ortsname
  - (b) Postleitzahl → (?:DE? ?- ?)?\d{5}
  - (c) Ortsname → Ort-Ort
  - (d) Ortsname → Zusatzbezeichner Ort
  - (e) Zusatzbezeichner → *Bad, Heilbad, Kurort, Luftkurort, Markt, ...*
  - (f) Ortsname → Ort GEO
  - (g) GEO → (GEO)
  - (h) GEO → / (?:OT)? GEO
  - (i) GEO → Präp (?:Det)?GEO
  - (j) OT → *OT, Ot, Ortsteil*
  - (k) Präp → *a., am, an, bei, i., in*
  - (l) Det → *d., der*

Die Normalisierung der Ortsnamen erfolgt hierbei mithilfe des Lexikons. Da der gefundene Ortsname mit dem Lexikon der Tupel aus Postleitzahlen und Ortsnamen approximativ verglichen werden muss, sollte der Ortsname aus diesem Lexikon stammen.

6.4.5 Kontaktdaten

Anders als die kanonische Wohnadresse kommen die Kontaktdaten[32] oft gesondert auf anderen Seiten vor. Unter Kontaktdaten dabei fallen Telefon-, Fax- und Mobilfunknummer, sowie die E-Mail-Adresse[33].

[31] Vgl. Mederle (2004), S. 8ff.

[32] Bei Maynard et al. (2002) [97] werden Telefon, E-Mail, URL und IP der Adresse zugeordnet.

[33] URL und IP sind Startpunkte, so dass alle anderen Klassen explizit für diese gesucht werden müssen.

6.4.5.1 Telefon-, Fax- und Mobilfunknummer

Für die Relevanzprüfung wird das Lexikon in Form einer Hash-Tabelle mit Vorwahlnummern als Schlüssel und Ortsnamen als Werte[34] angelegt. Nachdem schon eine Telefonnummer gefunden werden konnte, wird diese im Vorwahl-Lexikon nachgeschlagen[35].

Allerdings können Faxnummern jederzeit auch ohne die entsprechenden Vorwahlnummern auftreten. Dies ist besonders dann der Fall, wenn diese mit der der Telefonnummer übereinstimmen. Dann muss die Vorwahl der Telefonnummer übernommen werden. Normalerweise sind Vorwahlnummern für Ortskennzahlen 3- bis 6-stellige Ziffernfolgen. Außer der Ortskennzahl gibt es aber auch Vorwahlen mit Sondernummern. In Deutschland existieren momentan 45 Sondervorwahlnummern, wobei es für die Mobiltelefonie derzeit 24 Vorwahlnummern gibt.

Eine Grammatik für Telefon-, Fax- und Mobilfunknummern[36] könnte wie folgt aussehen:

- Grammatik der Telefon-, Fax- und Mobilfunknummer
  - (a) Telefon → Telefonnummernbezeichner Telefonnummer
  - (b) Fax → Faxnummernbezeichner Faxnummer
  - (c) Mobiltelefon → Mobiltelefonnummernbezeichner Mobilfunknummer
  - (d) Mobiltelefonnummernbezeichner → (?:Länderkennzahl)? Ortskennzahl
  - (e) Telefonnummer → (?:Länderkennzahl)? Ortskennzahl Rufnummer
  - (f) Faxnummer → (?:Länderkennzahl)? Ortskennzahl Rufnummer
  - (g) Mobilfunknummer → (?:Länderkennzahl)? Ortskennzahl Rufnummer

[34] Für Sondernummern werden keine Werte angelegt.

[35] In seiner Arbeit erwähnte Mederle (2004) lediglich die Notwendigkeit der Normalisierung der Telefonnummern und der Relevanzprüfung zwischen der Ortskennzahl und dem Ortsnamen.

[36] Vgl. Mederle (2004), S. 13.

(h) Telefonnummernbezeichner → *Telefonnummer, Tel[.:]?, Tel.? \+ Fax.?, ...*[37]

(i) Faxnummernbezeichner → *Faxnummer, Fax[.:]?, Telefax, Tel.? \+ Fax.?, ...*

(j) Mobiltelefonnummernbezeichner → *Mobilfunk, Mobil, Mobil-Tel, unterwegs, ...*

(k) Länderkennzahl → (?:[0\+\(\)␣]{,7}|49)

(l) Ortskennzahl → [0-9␣],9

(m) Telefonnummer → [\d\W]+

(n) Faxnummer → [\d\W]+

(o) Mobilfunknummer → [\d\W]+

Durch den Lexikonabgleich kann die Telefonnummer ohne einen externen Indikator erkannt werden. Ebenfalls wird ohne weitere Prüfung oder weitere Hinweise angenommen, dass es sich bei dieser Ziffernfolge um eine Telefonnummer handelt. Anschließend wird die gefundene Nummer im Lexikon nachgeschlagen. Falls nun eine Vorwahlnummer gefunden wird, und die restliche Ziffernfolge zwischen den minimalen und maximalen Rufnummernbereichen liegt, wird sie der jeweiligen Telefonnummer zugewiesen. Dazu wird ein Beispiel in Abbildung 6.9 angegeben:

```
<BODY>
<H1>Steuernummer: 71/410/20034</H1>

<P>Abakus-IT -
Weserstieg 46 -
21079 Hamburg -
040/30085385 -
admin@abakus-it.de
</BODY>
```

Abbildung 6.9: Abschnitt der SLD „abakus-it“

[37]Es sind insgesamt 32 Einträge für Telefonnummern-, 9 für Faxnummern- und 16 für Mobiltelefonnummernbezeichner verwendet.

Während die anderen Klassen extrahiert werden, kann die Ziffernfolge mit dem Schrägstrich *„040/30085385“* keiner Klasse zugeordnet werden. Darum wird nun anhand des längsten Matches auf der Vorwahlliste des Lexikons überprüft, ob die gefundene Ziffernfolge damit kompatibel ist. Es wird *„040“* aus dieser Liste gematcht, und der Ortsname *„Hamburg“* ist der entsprechend zugeordnete Wert für die Vorwahl *„040“*. Die verbleibende Ziffernfolge bleibt zwischen den minimalen und maximalen Rufnummernbereichen. So kann nun die Ziffernfolge *„040/30085385“* der Telefon-Klasse zugeordnet werden.

Leider muss eingeräumt werden, dass es schwierig ist, die maximale Ziffernfolge für eine Rufnummer zu ermitteln. So variiert die Minimale Rufnummer nach der Vorwahlnummer; große Ortschaften mit 3-stelliger Ortskennzahl *„040“* wie Hamburg haben eine längere Rufnummer als die 6-stellige Ortskennzahl wie z.B. *„039298“* in *„Barby Elbe“*. Bei den Nebenstellenbetreibern wird die Direktwahlnummer willkürlich vergeben. Generell wird angenommen, dass die Rufnummer zwischen 2 und 11 Ziffern hat.

Die Normalisierung der Telefonnummern erfolgt durch den Abgleich zwischen dem Lexikon und der gefundenen Nummer.

- Form der Normalisierung für Telefonnummern

  Telefonnummer → (Vorwahlnummer) Rufnummer

Der Algorithmus für die Normalisierung der Telefonnummer ist in Abbildung 6.10 zu sehen:

Normalisierungsalgorithmus für Telefonnummern

1. Lösche alle nicht-nummerischen Zeichen
2. Lösche die Länderkennzahl
3. Matching mit der längsten Ortskennzahl aus dem Lexikon
4. Trenne die gefundene Telefonnummer
   in Ortskennzahl und Restnummer

Abbildung 6.10: Normalisierungsalgorithmus für Telefonnummern

6.4.5.2 E-Mail-Adresse

Das kanonische Merkmal der E-Mail-Adresse ist das @-Zeichen. Ein einfacher regulärer Ausdruck für die Erkennung einer E-Mail-Adresse wäre:

$$[\hat{}@\lrcorner]+\backslash@[\hat{}\backslash.\lrcorner]+\ (?:\backslash.[\hat{}\backslash.\lrcorner])*\backslash.\mathrm{TLD}^{38}$$

wobei TLD für die Top-Level-Domain steht[39]. Durch diesen einfachen regulären Ausdruck mit einer kleinen Änderung für die gültigen Benutzer- und Domain-Namen können die meisten E-Mail-Adressen extrahiert werden.

Jedoch treten noch ein paar Schwierigkeiten auf, da diesem einfachen regulären Ausdruck einige Adressen entgehen werden, und zudem sollte die gefundene E-Mail-Adresse in Bezug zum Domain-Namen der URL stehen. Für das letztere Problem wurde eine Liste der gängigen Hostnamen wie „*t-online, web, gmx, usw.*“ erstellt. Falls der Hostname der gefundenen E-Mail-Adresse in dieser Liste enthalten ist, wird die Relevanz zwischen dem Hostnamen der gegebenen URL bezüglich des Benutzernamens der gefundenen E-Mail-Adresse approximativ überprüft.

Varianten von E-Mail-Adressen Auf der Informationsseite kommt die E-Mail-Adresse im Normalfall zusammen mit der Anchortext-Umgebung vor, wie es in Abbildung 6.11 zu sehen ist.

Die E-Mail-Adresse in einer Anchortext-Umgebung kann durch die Analyse der Link-Struktur richtig ermittelt werden, sofern sie in einem normalen HTML-Plain-Text eingebettet ist. In Beispiel 2 in Abbildung 6.11 wird die richtige E-Mail-Adresse durch Analyse der Link-Struktur trotz des nicht gültig formatierten Anchortextes extrahiert. Beim Anchortext wird hierbei der String *„(at)“* statt dem Zeichen @ verwendet.

Das große Problem liegt jedoch darin, dass die E-Mail-Adresse entweder nicht in einer Anchortext-Umgebung nach den üblichen Schreibkonventionen vorliegt, oder die Link-Struktur mit JavaScript oder einem Image maskiert wird,

[38] Friedl (2006), S. 71ff [49] erfasst eine gültige E-Mail-Adresse mit folgendem regulärem Ausdruck:
\w[-\.\w]*\@[-a-z0-9]+ (\.[-a-z0-9]+)*\.
(?:com|edu|gov|int|mil|net|org|biz|info|name|museum|coop|aero|[a-z][a-z]).

[39] siehe Kapitel 2 und Fußnote 38. Die TLD ist in gTLD und ccTLD eingeteilt. Für Deutschland ist die ccTLD *„de“*.

Beispiel 1: `http://www.a-bis-z-reisen.de/impressum.htm` kundenservice@a-bis-z-buero.de
Beispiel 2: `http://www.4pfotenparadies.de/impressum.html` n.marschall (at)4pfotenparadies.de

Abbildung 6.11: Einfache Varianten von E-Mail-Adressen

und die Anchortexte somit keine gültige Umgebung für E-Mail-Adressen darstellen, wie es in Abbildung 6.12 gezeigt wird.

Beispiel 1: `http://www.capitol-bingen.de/impressum/`   E-Mail: info [at] capitol-bingen [punkt] de
Beispiel 2: `http://www.elektroanlagen-maechler.de/impressum.htm` E-Mail: Elektro.Maechler (at)t-online.de
Beispiel 3: `http://www.geruestbauinnung.de/impressum.html` info (at)geruestbauhandwerk.de
Beispiel 4: `http://www.autoklein.de/index\_op\_imprint.html` <tr><td>E-Mail:</td><td> <script type=“text/javascript”> <!– var name = “info”; var domain = “autoklein.de”; document.write (’<a href=\“mailto:’ + name + ’@’ + domain + ’\”>’); document.write (name + ’ (at)’ + domain + ’<\/a>’); // –> </script> </td></tr>

Abbildung 6.12: Komplexe Varianten von E-Mail-Adressen

Die Beispiele 1, 2 und 3 aus Abbildung 6.12 zeigen relativ einfache Varianten

von gültigen E-Mail-Adressen. Aus ihnen wird ersichtlich, welche Varianten für das At-Zeichen (@) und den Punkt (.) auftreten können. Anstelle von @ wird häufig auch „at“ verwendet, während für „.“ gerne die ausgeschriebene Variante *„punkt“* oder englisch *„dot“* genommen wird. Außerdem werden für die eckigen Klammern meist runde Klammern eingesetzt und all diese Variationen können mit oder ohne ein Leerzeichen vorkommen. Mit oder ohne den Link-Verweisen in den Beispielen 2 und 3 aus Abbildung 6.12 können die E-Mail-Adressen aus den entsprechenden HTML-Texten extrahiert werden. Dagegen wird in den Beispielen 1 und 2 aus Abbildung 6.12 eindeutig durch das Attribut „E-Mail“ gekennzeichnet, dass es sich um eine E-Mail handelt, während beim Beispiel 3 in Abbildung 6.12 dies durch den Benutzer- und Domain-Namen sowie die TLD ermöglicht wird[40].

Beim Beispiel 4 in Abbildung 6.12 wird das Erkennen der E-Mail-Adresse durch JavaScript erschwert[41].

Obwohl sie als „info(at)autoklein.de“ im Browser zu sehen ist, ist sie im HTML-Quellcode nicht erkennbar. Zur Rekonstruktion solch komplexer E-Mail-Adressen muss zuerst das JavaScript interpretiert werden.

Der erste Hinweis kommt aus dem Link-Verweis: „*document.write ('<href=\" mailto:' + name + '@' + domain + '\" >')*“. Der String „*mailto*“ kennzeichnet, dass die darauf folgende Kombination von Zeichenfolgen eine E-Mail-Adresse betrifft. Lediglich müssen noch die Werte für die Variablen *„name“* und *„domain“* ermittelt werden. Diese erfolgt über *„var name“* und *„var domain“*.

Übliche Benutzernamen E-Mail-Adressen von Firmen-Websites haben in der Regel einen typischen Benutzernamen wie „*info, kontakt, mail, service, usw.*“. Nachdem nun keine E-Mail-Attribute oder gültigen E-Mail-Adressen gefunden wurden, werden diese Benutzernamen für eine mögliche E-Mail-Adresse genutzt. Über diese typischen Benutzernamen können über 70% der vorhandenen E-Mail-Adressen abgedeckt werden. In Abbildung 6.13 wird illustriert, welche Benutzernamen bei Firmen-Websites gebräuchlich sind.

[40] Für eine Statistik der Benutzernamen, siehe Abschnitt 6.4.5.2.

[41] Abgesehen von der @-Variante, wäre es auch möglich, E-Mail-Adressen mit einer visuellen HTML-Rendering-Technik, wie sie z.B. von Gatterbauer et al. (2007) [53] eingesetzt wird, zu identifizieren. In diesem Fall würd die E-Mail-Adresse wie in Beispiel 4 aus Abbildung 6.12 wie im Browser dargestellt werden: info (at)autoklein.de.

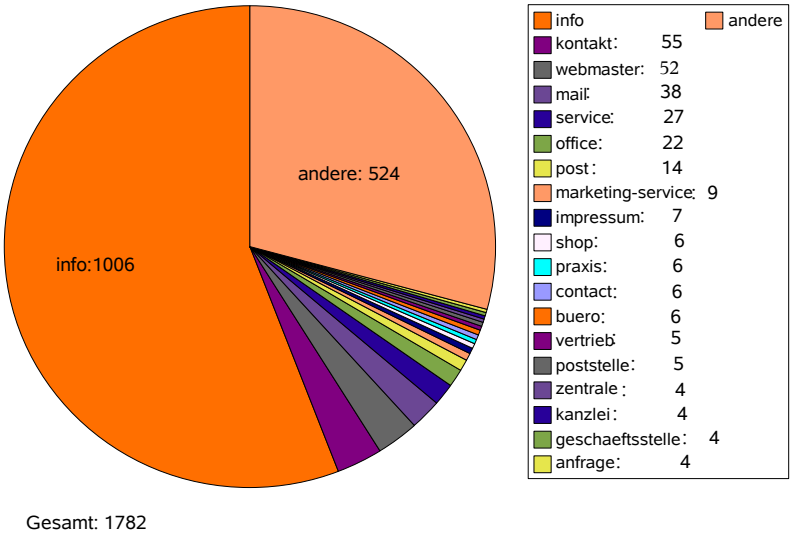


Abbildung 6.13: Typische Benutzernamen einer Firmen-Website

Wie in Abbildung 6.13 weiter gezeigt wird, wird der Benutzername „*info*“ am häufigsten genutzt. Darauf folgen „*kontakt, webmaster, mail, service, usw.*“. Kommt der „*webmaster*“ und zusätzlich noch „*info*“ vor, wird die „*info*“-Variante bevorzugt. Denn „*webmaster*“ wird eher für den Administrator verwendet.

Unter „*andere*“ sind Benutzernamen wie „*team, support, presse, mailbox, kundenservice, information, email, willkommen, welcome, und webshop*“ als typische Benutzernamen anzusehen.

6.5 Personen

Auf der Informationsseite einer Firmen-Website sind Personennamen mit ihrer Funktion in der Firma aufgelistet. Davon treten besonders häufig „*Geschäftsführer, Inhaber, Ansprechpartner, usw.*“ auf. Um die Datenbank der firmenrelevanten Informationen aufzubauen, werden Personennamen zusammen mit ihrer Funktion extrahiert. Dabei ist die Personennamenerkennung

eine der Hauptaufgaben im Bereich der Entitätenerkennung, worunter insbesondere „*Personen, Organisationen und Ortsangaben*“ fallen.

So können Personennamen in den verschiedensten Variationen ausgeschrieben werden: mit den abgekürzten oder vollständigen Formen, mit den akademischen und aristokratischen Titeln oder mit den Funktionstiteln.

Zur Identifizierung von Personennamen können zwei unterschiedliche Methoden angewendet werden: Die statistische Methode benötigt normalerweise ein manuell annotiertes großes Korpus, während die regelbasierte Methoden sich auf ein wissensbasiertes Verfahren stützen.

Leider sind die Informationsseiten einer Firmen-Website meist nicht linguistisch konzipiert, sondern sind eher kompakt für den menschlichen Besucher strukturiert. Jedoch sind statische Methoden oft schwer anzuwenden, da sie entweder auf ein großes annotiertes Korpus angewiesen sind (überwachte Methoden), oder sich auf die HTML-Struktur des Dokumentes (unüberwachte Methoden) verlassen, die auf einigen Websites nicht ganz korrekt vorzufinden ist.

6.5.1 Allgemeine Erkennungsprobleme von Personennamen

Personennamen treten in den verschiedensten Variationen auf, wie in Tabelle 6.12 gezeigt wird.

Die in Abbildung 6.12 aufgeführten Personennamen zeigen die diversen Möglichkeiten für deutsche Personennamen.

Das erste Beispiel ist eine typische Kombination von Vor- und Nachnamen[42], während das zweite und dritte Beispiel eine Kombination von doppeltem Vornamen mit einfachen Nachnamen ist.

Das vierte Beispiel ist ein doppelter Nachname, der durch einen Bindestrich zusammengesetzt ist. Das fünfte und sechste Beispiel illustiert, dass Nachnamen einen adeligen Zusatz beinhalten dürfen. Dagegen geht bei den restlichen

[42] Für die Bildung der deutschen Vor- und Nachnamen, siehe `http://de.wikipedia.org/wiki/Vorname` und `http://de.wikipedia.org/wiki/Nachname` und die darin erwähnten Links. Siehe auch Kap. 5 und 6 von Geierhos (2006) [54].

1. *Martin Schäfer*
2. *Paul A. Stodden*
3. *Karl-Friedrich Linder*
4. *Oliver Thiele-Lorenzen*
5. *Elke Gräfin zu Rantzau*
6. *Ernst Fürst von Waldburg-Zeil*
7. *Dr. Johannes Horstmann*
8. *Dipl.-Ing. Herbert Utz*
9. *Dipl.-Inf. (Univ.) Dimitri Schischkin*
10. *Prof. Dr. rer. nat. Thomas Martinetz*
11. *Dipl.-Ing. (FH) Architekt Hermann-Josef Schäfer*

Tabelle 6.12: Beispiele für Personennamen

Beispielen ein akademischen Titel dem Namen voran[43].

Ein allgemeines Merkmal eines Personennamens ist für Sprachen, die Klein- und Großbuchstaben unterscheiden, dass dieser mit dem Großbuchstaben beginnt. Diese Eigenschaft hilft aber bei der Erkennung von Eigennamen im Deutschen wenig, da alle Nomina mit einem Großbuchstaben beginnen.

Es ist unmöglich, ein vollständiges Lexikon von Vor- und Nachnamen zu erstellen. Abgesehen von den Nachnamen[44] kann das Lexikon der deutschen Vornamen allein durch die Verdoppelung kartesisch vergrößert werden, um alle möglichen Vornamen zu erfassen. Ein anderer Faktor kommt mit der Globalisierung hinzu: Das Zusammenleben verschiedenster Völker hat die Namensgebung viel internationaler gemacht.

Erschwerend kommt noch hinzu, dass auf den Informationsseiten die gesuchten Personennamen in einer strukturierten Umgebung mit gewissen Attributen vorkommen.

Wie Abbildung 6.14 zeigt, können Personennamen durch das vorangestellte

[43]Es ist umstritten, ob ein akademischer Titel oder eine Berufsbezeichnung wie „*Präsident*“ zum Teil des Eigennamens zählt. Nach den Richtlinien von MUC-6 (siehe `http://www.cs.nyu.edu/cs/faculty/grishman/NEtask20.book_9.html#HEADING30`) gehören solche Titel nicht zum Eigennamen, während er bei Maynard et al. (2001) [97] miteinbezogen wird. In dieser Arbeit wird er vorsichtshalber mitgezählt.

[44]`http://www.namepedia.org` hat eine Nachnamendatenbank von ca. 2 Millionen Namen.

```
<td width="50%" align="left" valign="top">
    <font face="Arial, Helvetica, sans-serif"> <b>
      <font size="2" color="#CC0000">Ihre Ansprechpartner:</font></b>
      <font size="2" color="#000000"><br>
        <b>Inhaberin:</b> Ines Böhmer<br>
        <b>Geschäftsführung:</b> Uwe Böhmer<br>

        <b>Webmaster und Inhalte:</b> Uwe Böhmer
      </font>
    </font>
</td>
```

Abbildung 6.14: Beispiel für Personenangaben im „Verantwortlichenblock“ der SLD „a-bis-z-reisen“

Attribut identifiziert werden. Jedoch kann das Attribut auch nachgestellt werden, wie das Beispiel in Abbildung 6.15 illustriert:

```
<p style="margin-top: 0; margin-bottom: 0"><u><b><font size="4">
Vertretungsberechtigter Vorstand</font></b></u>:</p>
<p style="margin-top: 0; margin-bottom: 0"> </p>
<p style="margin-top: 0; margin-bottom: 0">Allein berechtigt:
    Henning Hermanns (Vorsitzender)</p>
<p style="margin-top: 0; margin-bottom: 0"> </p>
```

Abbildung 6.15: Beispiel der SLD „schuetzenverb-bs“

Das Attribut „*Vertretungsberechtigter Vorstand*“ bezieht sich durch den Hinweis von *„Allein berechtigt:“* auf den Personennamen „*Henning Hermanns*“ , der aber auch die Wert-Rolle für das nachgestellte Attribut „*(Vorsitzender)*“ inne hat. Die Person *„Henning Hermanns“* kann in diesem Kontext 3 Rollen zugewiesen bekommen: **Vertreter**, **Vorstandmitglied** und **Vorsitzender**, was in einem Verein nicht selten der Fall ist.

Bei der Erkennung von Personennamen mitsamt ihrer Funktion auf der Informationsseite können die Attributklassen und die personennameninterne

Struktur genutzt werden. Die personennameninterne Struktur wird in Abbildung 6.16 veranschaulicht[45]:

Anrede	Titel	Vorname	Nachname	Zusatz

Abbildung 6.16: Maximal mögliche Bestandteile eines Personennamens

Eine Grammatik von Personennamen für die semi-strukturierten Informationsseiten kann also wie folgt gestaltet werden.

- Grammatik eines Personennamens
  - (a) Personenname → (?:Anrede (?:LzbBs Titel)+ LzbBs)? Vorname LzbBs Nachname (?:LzbBs (?:Zusatz (?:LzbBs (?:Titel|Abschlusszusatz)?)?)?)?
  - (b) Vorname → Eigennamen
  - (c) Nachname → Nachnamenzusatz LzbBs Eigennamen
  - (d) Titel → Titelbezeichnung
  - (e) Titel → Titelbezeichnung LzbBs Fachbezeichnug
  - (f) Titel → Titelbezeichnung LzbBs Fachbezeichnug LzbBs Abschlusszusatz
  - (g) Titel → Berufsbezeichnung
  - (h) Titel → Berufsbezeichnung LzbBs Titelbezeichnung
  - (i) Titel → Berufsbezeichnung LzbBs Titelbezeichnung LzbBs Fachbezeichnung LzbBs Abschlusszusatz
  - (j) Eigennamen → (?:Gbb|Eigenname) (?:LzbBs (?:Gbb|Eigenname)){,3}[46]
  - (k) Eigenname → [A-ZÄÖÜ][A-ZÄÖÜa-zäöüß]{3,}
  - (l) Anrede → *Frau, Herrn?, Fräulein, Fr\.?, Hr\.?, Frl\.?*
  - (m) Titelbezeichnung → *Bischof, Kardinal, Dr\.?, Dipl\.?, Diplom*
  - (n) Fachbezeichnung → *h\. c\., mult\.?, phil\.?, rer\.?, jur\.?, med\.?, dent\.?, Oek\.?, Päd\.?, Chem\.?, Inf\.?, Ing\.?, Kfm\.?, Kfr\.?, Phys\.?, Wirt\.? LzbBs Ing\.?, Agraring\.?, Fischereiing\., Betriebsw\.?, ...*

[45] Vgl. Kühnlein (2003), S. II-56ff [77].

[46] Leerzeichen wurden nur zur Lesbarkeit verwendet.

(o) Berufsbezeichnung → *Prof\.?, Professor, Professorin, PD, Präsident, Präsidentin, Meister, Meisterin, Architekt, Architektin, Anwalt, Anwältin, Apotheker, Apothekerin, ...*

(p) Abschlusszusatz → *FH, FS, TH, Univ\.?*

(q) Nachnamenzusatz → *v\.?, von, van*

(r) Nachnamenzusatz → Adelsbezeichnung LzbBs *v\.?, von, van, zu*

(s) Adelsbezeichnung → *Gräfin, Graf, Prinzessin, Prinz*

(t) Zusatz → *I, II, III, IV, V, VI, VII, VIII, IX, X, XI, jun\.?, sen\.?, ...*

(u) Kb → [a-zääöüß]\.?

(v) Gb → [A-ZÄÖÜ]\.?

(w) LzbBs → [-␣]

(x) Gbb → Gb (?:LzbBs Gb (?:LzbBs Kb)?){,3}

Die in der Grammatik der Personennamen erwähnten regulären Ausdrücke können je nach Zweck noch strikter oder allgemeiner formuliert werden. Da die Personennamen auf der Informationsseite im Regelfall mit den passenden Attributklassen zusammen vorkommen, wird diese Grammatik für die Erkennung von Personennamen für die Informationsseiten ausreichend sein[47].

6.5.2 Titel und Zusätze

Ein Titel und ein Namenszusatz sind wichtige personennameninterne Indikatoren für die Personennamenerkennung. So können Titel in akademische und berufliche Titel eingeteilt werden. Während die akademischen Titel abzählbar sind, können berufliche Titel unbegrenzt je nach Berufsgruppen erweitert werden. Beispielsweise könnte außer dem *Vertriebsmanager* auch ein *Vize-Vertriebsmanager* im Sinne von *Stellvertretender Vertriebsmanager* existieren.

Aufgrund der eben genannten Eigenschaft von beruflichen Titeln wird versucht, möglichst allgemeine Begriffe ins Lexikon aufzunehmen. Da Personennamen auf der Informationsseite meist mit vorangestellter Attributklasse

[47] Bei **Titel** können die akademischen Titelbezeichnungen den Berufsbezeichnungen vorangehen, wie z.B. *Dipl.-Ing. (FH) Architekt Hermann-Josef Schäfer.*

auftreten, kann auf ein großes Lexikon mit Berufstiteln verzichtet werden. Durch den Gebrauch von allgemeinen Begriffen, wie „*Manager, Meister, Leiter, usw.*“ können neue Komposita wie „*Vize-Vertriebsmanager*“ ins Lexikon aufgenommen werden.

Bei „berufe.net“ [48] sind über 10 000 offizielle Berufsbezeichnungen aufgelistet. Dort gibt es z.B. über 100 Berufsbezeichnungen für den generischen Begriff *„Leiter“* . Trotzdem fehlen noch Berufsbezeichnungen, wie „*Mediendienstleiter, Wachdienstleiter, usw.*“.

Durch die verallgemeinerten Begriffe kann das Lexikon sehr kompakt gehalten werden. In Tabelle 6.13 ist ein Auszug aus „berufe.net“ für den berufliche Funktion „*Leiter*“ zu sehen.

Abteilungsleiter	*Altenheimleiter*	*Aufnahmeleiter*
Auftragsleiter	*Ausbildungsleiter*	*Außendienstleiter*
Badebetriebsleiter	*Bandleiter*	*Bankbereichsleiter*
Bankettleiter	*Bankzweigstellenleiter*	*Bauleiter*
Bereichsleiter	*Berufsschulleiter*	*Betriebsleiter*
Bezirksleiter	*Borddienstleiter*	*Bühnenleiter*

Tabelle 6.13: Beispiele für den allgemeinen Beruf *Leiter*

Allgemeine Berufsbezeichnungen, die häufig bei der Extraktion von Personennamen auf der Informationsseite ausgenutzt werden können, sind beispielsweise in Tabelle 6.14 aufgeführt.

Im Gegensatz zu den verallgemeinerten Berufsbezeichnungen sind Komposita von den Begriffen in Tabelle 6.15 nicht besonders produktiv.

Somit hat das Lexikon der **Berufstitel** ca. 400 allgemeine Einträge.

Dazu werden als akademische Titel z.B. „*Dr., Dipl., M.A., Master, Bachelor*“ aufgenommen. Die Anzahl der Fachrichtungen für einen akademischen Titel beträgt knapp über 100 Einträge. Davon sind einige Beispiele in Tabelle 6.16 aufgelistet. Alle Einträge sind auch mit den abgekürzten Varianten versehen, die den ausgeschriebenen Formen entsprechen.

Als Zusätze werden noch Generationskennzeichnungen, wie „*I, II, III, jun., sen.*“, oder Abschlusszusätze, wie „*BA, FH, FS, Ing.H, TH, Univ*“ verwendet.

[48] `http://www.berufe.net`.

Agent	*Anwalt*	*Arzt*
Assistent	*Berater*	*Designer*
Direktor	*Elektroniker*	*Händler*
Ingenieur	*Kaufmann*	*Leiter*
Makler	*Manager*	*Mechaniker*
Meister	*Optiker*	*Pfleger*
Praktiker	*Programmierer*	*Prüfer*
Spezialist	*Sprecher*	*Trainer*
Vertreter	*Verwalter*	*Vorsitzender*
Vorstand	*Wirt*	

Tabelle 6.14: Beispiele für allgemeine Berufe

Apotheker	*Architekt*	*Fotograf*
Geschäftsführer	*Informatiker*	*Professor*

Tabelle 6.15: Beispiele für spezifische Berufstitel

Dr. dent	*Dr. ing*	*Dr. med*
Dipl.-Architekt	*Dipl.-Betriebswirt*	*Dipl.-Biologe*
Dipl.-Chemiker	*Dipl.-Computerlinguist*	*Dipl.-Designer*
Dipl.-Design-Ingenieur	*Dipl.-Finanzwirt*	*Dipl.-Informatiker*
Dipl.-Informationsjurist	*Dipl.-Ingenieur*	*Dipl.-Jurist*
Dipl.-Kaufmann	*Dipl.-Volkswirt*	*MBA*

Tabelle 6.16: Beispiele für akademische Fachbezeichnungen

6.5.3 Extraktion von Personennamen

Folgende Personennamen werden nun aus den Informationsseiten extrahiert: *Geschäftsführer, Inhaber, Vorsitzender, Kontaktperson, Vorstand, Verantwortlicher, Vorsitzender des Aufsichtsrates.*

6.5.3.1 Geschäftsführer

Juristisch betrachtet kann der **Geschäftsführer** nur dann angegeben werden, wenn eine Firma einer Kapitalgesellschaft entspricht. Bei Personengesell-

schaften wird stattdessen der Begriff *„persönlich haftender* **Gesellschafter***“* verwendet[49], wobei in dieser Arbeit nicht auf diesen Unterschied eingegangen wird.

Geschäftsführer ist die am häufigsten angegebene Information in Bezug auf Personennamen innerhalb der Informationsseite. Über 25% aller Firmen-Websites geben ihren Geschäftsführer an, welcher meist durch eines der Attribute in Tabelle 6.17 eingeleitet wird.

CEO	*Direktor*
GF	*Geschäftsführer*
Geschäftsführenden Direktor	*Geschäftsführender Gesellschafter*
Geschäftsführer ist	*Geschäftsführung*
Managing Director	*Vertretungsberechtigter Geschäftsführer*

Tabelle 6.17: Attributklasse für „Geschäftsführer“

Treten diese Attribute nun im Text auf und entspricht der nachfolgende Text dem Personennamen, dann ist der gefundene Personenname der des Geschäftsführers. Manchmal kann es jedoch sein, dass das jeweilige Schlüsselwort hinter dem Personennamen vorkommt, was in Abbildung 6.17 gezeigt wird.

```
<tr>
    <td width=“50%”>vertreten durch:</td>
    <td width=“50%”>Ralf Poloczek (Geschäftsführer)</td>
</tr>
```

Abbildung 6.17: Ausschnitt der SLD „iek“

Aus Abbildung 6.17 geht hervor, dass das Attribut **Geschäftsführer** hinter dem Personennamen steht[50]. Dies ist oft dann der Fall, wenn der **Geschäftsführer** gemeinsam mit dem Attribut **Vorstand** angegeben wird.

[49] `http://de.wikipedia.org/wiki/Handelsregister#Abteilungen`.

[50] Auf die Frage, ob der vorangehende Text „*vertreten durch*“ zu der erweiterten und eindeutigen Attributklasse **Geschäftsführer** gehört, wird hier nicht eingegangen. Juristisch gesehen sind die beiden Begriffe „*Geschäftsführer*“ und „*Vertreter*“ jedoch anders zu sehen.

```
<b>Vertretungsberechtigter Vorstand:</b>
    Wolfgang Göndöcs (Vorsitzender),
    Gerrit Horstmann (stellvertr. Vorsitzender),
    Benjamin Kiefel (Geschäftsführer),
    Michel Hallmann (Kassenwart),
    Frank Willner (Sportwart)
<br>
```

Abbildung 6.18: Ausschnitt der SLD „bfc-fortuna“

6.5.3.2 Inhaber

Der Begriff **Inhaber** ist im Normalfall für den Besitzer eines Handelsgeschäfts oder eines Betriebs gebräuchlich. Dieses Wort wird hier ähnlich weit gefasst wie **Geschäftsführer** betrachtet. So kommen Attribute, wie „*Eigentümer und Betreiber dieser Seite i.S. des Teledienstgesetzes, Owner of the domain and contact*“, oder gemeinsam mit den zur anderen Klasse **Inhaber** gehörenden Attributen „*Inhaber und Leiter, Inhaber / Geschäftsführer*“ vor. Typische Attribute für **Inhaber** sind unter anderem „*Inh . Inhaber(in), Firmeninhaber(in)*“.

6.5.3.3 Vorsitzender

Der Begriff **Vorsitzender** wird für den Leiter eines Vereins, einer Organisation oder ähnlich großen Versammlungen und Gruppierungen verwendet[51]. Im Falle einer Aktiengesellschaft hat sich der Terminus „**Vorstandsvorsitzender**“ oder „**Präsident**“ durchgesetzt. Das Wort **Vorsitzender** kann nahezu für alle leitenden Personen gebraucht werden. Die Beispiele aus Tabelle 6.18 sind mögliche Attribute für die Klasse **Vorsitzender**:

Wie auch beim **Geschäftsführer** kann das Attribut nachgestellt werden (siehe Abschnitt „**Geschäftsführer**“ und die entsprechenden Beispiele dort.).

[51] `http://de.wikipedia.org/wiki/Vorsitzender`.

1. Vorsitzender	*Vorsitzender*	*Vorstandsvorsitzender*
Präsident	*Der Vorsitzende*	*Geschäftsleitung*
Leitung	*Director*	*Direktor*
Geschäftsführenden Direktor	*Leiter*	*Chairman*

Tabelle 6.18: Attributklasse für „Vorsitzender“

6.5.3.4 Kontaktperson

Anders als die Klassen „**Geschäftsführer, Inhaber, Vorsitzender**“ wird dem Begriff **Kontaktperson** kein offizieller Status zugewiesen. Trotzdem wird die „**Kontaktperson**“ neben der Angabe „**Geschäftsführer**“ am häufigsten angegeben. Als **Kontaktperson** können nun mehrere Personen wie im Beispiel 6.14 auf der Seite 147 angegeben werden.

Folgende Beispiele fungieren meist als Attribute für die Klasse **Kontaktperson** (siehe Tabelle 6.19):

Ansprechpartner	*Ihr Ansprechpartner ist*
Tech. Ansprechpartner	*Verantwortliche Ansprechperson*
Kontaktperson	*Ihr Ansprechpartner*

Tabelle 6.19: Attributklasse für „Kontaktperson“

6.5.3.5 Vorstand

In großen Organisationen wie „Aktiengesellschaften, Vereinen, Genossenschaften“ oder „Stiftungen“ fungiert der **Vorstand** als ausführendes Organ, das die Funktion der Organisationsleitung übernommen hat. Die Klasse **Vorsitzender** ist nur ein Mitglied des **Vorstand**es, welcher aus mehreren Personen gebildet wird. Genau das erschwert die Erkennung und Extraktion eines einzelnen Mitglieds des **Vorstand**es. Das Beispiel in Abbildung 6.18 auf der Seite 153 illustriert diese Schwierigkeiten. Aus diesem Grund basiert die Extraktion der Vorstandsschaft auf dem gleichen Delimiter, den HTML-Elemente gemeinsam haben. In Abbildung 6.18 auf der Seite 153 wird der Delimiter „*Komma (,)*“ verwendet. Für das hier gewählte Beispiel „**Vor-**

stand“ wird die Begrenzung durch das HTML-Element <**br**> am Ende der anderen Klassen gewährleistet.

Häufig verwendete Attribute für **Vorstand** sind „*1. Vorstand, geschäftsführender Vorstand, vertretungsberechtigter Vorstand usw.*“.

6.5.3.6 Verantwortlicher

Ca. 19% aller Firmen-Websites geben die verantwortliche Person im Sinne des Teledienstgesetzes und Mediendienste-Staatsvertrages (MDStV) an. Im Vergleich zu den anderen Attributklassen für Personennamen sind die Attribute für **Verantwortlicher** relativ lang und extrem unterschiedlich, obwohl sie fast immer das Wort „*verantwortlich*“ enthalten. Dazu werden einige Beispiele in Tabelle 6.20 angegeben.

V.i.S.d.P.
Verantwortlich
Verantwortl. i.S.d. StV.
Verantwortlich i. S. d. P.
Verantwortlich für Inhalte
Verantwortlich für den Inhalt
Verantwortlich für die Website
Verantwortlich i.S.d. § 6 MDStV & 6 TDG
Verantwortlich für den Inhalt der Website
Verantwortlich im Sinne des Teledienstgesetzes
Verantwortlich für den Inhalt dieser Internetseiten
Verantwortlich im Sinne des §6 des TDG und §10 des Medienstaatsvertrages

Tabelle 6.20: Attributklasse für „Verantwortlicher“

Hierbei können derzeit ca. 90 Variationen von Attributen verzeichnet werden.

6.5.3.7 Vorsitzender des Aufsichtsrates

Dagegen tritt der Vorsitzende des Aufsichtsrates viel seltener in diesen Kontexten auf. Ca. 1% der Websites aus den Trainingsdaten haben diese Information angegeben. Im Gegensatz dazu werden eher Attribute, wie „*AR Vorsitz,*

Aufsichtsrat, Aufsichtsratsvorsitzender, Vorsitzender des Aufsichtsrats“, angegeben.

6.6 Rechtliches

Neben Kontaktdaten und Personennamen kommen auf der Informationsseite auch rechtliche Hinweise vor. Dabei sind Daten, welche das Handelsregister oder die Steuer betreffen, vorwiegend in diesen Passagen zu finden. Außer dieser Informationen kann beispielsweise noch die zuständige Kammer erwähnt werden, was hier aber ignoriert wird.

Zu den steuerbezogenen Daten gehören die (Umsatz-)Steuernummer, die jedem vom Finanzamt zugewiesen wird, und die Umsatzsteueridentifikationsnummer (USt-IdNr.), die EU-weit gültig ist.

6.6.1 Registernummer und Registergericht

Die Häufigkeit der Bekanntmachung der Registernummer und des Registergerichts einer Firmen-Website liegt bei zwischen 25% und 28%. Während das Registergericht nicht nach der Firmenform variiert, gibt es bei der Registernummer verschiedene Variationen je nach der Art der Firmen. Allein schon bei der Handelsregisternummer wird nach der Gesellschaftsform in die Abteilungen *A* und *B* eingeteilt. Außer der **Handelsregisternummer** existieren noch **Vereins-, Genossenschafts-** und **Partnerschaftsnummern**. In Einzelfällen treten zusätzlich **Verkehrsnummern** für Verlage und **Betriebsnummern** für Einzelunternehmen auf.

6.6.1.1 Registergericht und Finanzamt

Beim Registergericht werden **Handels-, Vereins-, Genossenschafts-** und **Partnerschaftsregister** geführt. Als Registergericht fungiert in der Regel das Amtsgericht.

Die folgenden Attribute in Tabelle 6.21 sind häufige Kennzeichen für die Nennung eines **Registergerichts**. Diese Attribute enthalten oft Terme, wie

Amtsgericht
Gerichtsstand
Registergericht
Eingetragen beim
Eingetragen beim Amtsgericht
Eingetragen beim Handelsregister Amtsgericht
Eingetragen im Handelsregister beim Amtsgericht
Erfüllungsort und ausschließlicher Gerichtsstand ist

Tabelle 6.21: Attributklasse für „Registergericht“

„*AG, Amtsgericht, Gerichtsstand, Registergericht*“, als ein Haupttextteil. Insgesamt 30 Attribute konnten für das Registergericht zusammengestellt werden.

Das Finanzamt dagegen ist für die Steuer zuständig und ist vor allem eine ortsgebundene Behörde. Für das Finanzamt konnten Attribute wie „*FA, Finanzamt*“ ermittelt werden.

Aufgrund des lokalen Bezuges der Finanzämter und Registergerichte werden die Werte der Attribut-Wert-Paare mit Ortsnamen belegt, die auf Seite 135 im Abschnitt 6.4.4 aufgeführt werden.

6.6.1.2 Registernummer

Natürlich kommen die Registernummern in der Regel gemeinsam mit dem Amtsgericht vor. Als Identifikationsmuster für **Registernummern** werden gerne Kürzel wie *HRB, VR* oder ausgeschriebene Wörter wie *Handelsregister, Vereinsregister* genommen.

Die Kürzel werden dem Wert hinzugefügt, da sonst die Art des Registers nicht eindeutig ersichtlich ist. Außerdem folgen der Abkürzung Ziffern und anschließend optional maximal 2 Buchstaben.

Beim Handelsregister muss zusätzlich noch die Abteilung unterschieden werden. So sind Personengesellschaften bei der Abteilung „**A**“ registriert und ihre Rechtsformen sind hauptsächlich „*GbR, KG, OHG*“. Falls ein Handelsregistereintrag ohne Abteilungsinformation vorliegt, muss die Abteilung über

die Rechtsform des Firmennamens entschieden werden. Kapitalgesellschaften wie „*GmbH, AG*“ sind bei der Abteilung „**B**“ verzeichnet[52].

6.6.2 Steuer- und Umsatzsteueridentifikationsnummer

Für die hier vorgestellte Arbeit sind Steuernummern und Umsatzsteueridentifikationsnummern (USt-IdNr.) zwei irrelevante Ziffernfolgen. Die erste wird jedem Steuerpflichtigen vom Finanzamt zugewiesen, während die zweite EU-weit gültig ist. Das Standardschema der Steuernummer wird von jeder Landesregierung anders geregelt.

Die Tabelle 6.22 zeigt die Standardschemata der Steuernummer in Deutschland[53]. Zusätzlich wurde auch ein bundeseinheitliches Schema eingeführt, welches aus 13 Ziffern aufgebaut ist.

Im Vergleich zu Steuernummern beginnt die USt-IdNr. mit dem zweistelligen Landespräfix, welches im **ISO-3166-Alpha-2-Code** definiert ist[54]. Das Landespräfix kennzeichnet den Ort, wo das Unternehmen ansässig ist. Die USt-IdNr. eines Unternehmens, das in Deutschland ansässig ist, beginnt mit „**DE**“. Diesem Präfix folgen anschließend 9 Ziffern.

6.6.2.1 Steuernummer und USt-IdNr. in der Praxis

Leider werden auch Steuernummer und USt-IdNr. nicht immer korrekt angegeben, selbst wenn sie gut mit diesem Schema vorgegeben sind. Oft verwechselt man Steuernummer mit der USt-IdNr., wie das Beispiel aus Abbildung 6.19 zeigt.

In Abbildung 6.19 ist die USt-IdNr. eindeutig durch den Text „*Umsatzsteuer-Identifikationsnummer gemäß § 27 a Umsatzsteuergesetz*“ identifizierbar. Jedoch ist offensichtlich, dass die Nummer nicht dem Schema der USt-IdNr. entspricht, sondern dem Schema der Steuernummer einer Landesregierung, sofern der Bindestrich (-) durch den Schrägstrich (/) ersetzt wird. Die falsch

[52] Rechtsformen wie *GmbH & Co. KG, KGaA* sind grundsätzlich „*Kommanditgesellschaften*“ und gehören damit zur Abteilung „**A**“, während die Rechtsform *GmbH* zur Abteilung „**B**“ gehört.

[53] vgl. `http://de.wikipedia.org/wiki/Steuernummer`.

[54] siehe `http://de.wikipedia.org/wiki/ISO-3166-1-Kodierliste`.

Bundesland	Standardschema der Länder
Baden-Württemberg	\d{5}/\d{5}
Bayern	\d{3}/\d{3}/\d{5}
Berlin	\d{2}/\d{3}/\d{5}
Brandenburg	\d{3}/\d{3}/\d{5}
Bremen	\d{5}␣\d{5}
Hamburg	\d{2}/\d{3}/\d{5}
Hessen	\d{3}␣\d{3}␣\d{5}
Mecklenburg-Vorpommern	\d{3}/\d{3}/\d{5}
Niedersachsen	\d{2}/\d{3}/\d{5}
Nordrhein-Westfalen	\d{3}/\d{4}/\d{4}
Rheinland-Pfalz	\d{2}/\d{3}/\d{5}
Saarland	\d{3}/\d{3}/\d{5}
Sachsen	\d{3}/\d{3}/\d{5}
Sachsen-Anhalt	\d{3}/\d{3}/\d{5}
Schleswig-Holstein	\d{2}␣\d{3}␣\d{5}
Thüringen	\d{3}/\d{3}/\d{5}

Tabelle 6.22: Bildungsschemata der Steuernummern

```
<td class="rahmenkoerper" valign="top">
    Umsatzsteuer-Identifikationsnummer
    <br>
    gem&auml;&szlig; &sect; 27 a Umsatzsteuergesetz:
</td>
<td class="rahmenkoerper">
    DE 053-116-00763
</td>
```

Abbildung 6.19: Beispiel für die Steuernummer von „kino-im-ziel“

zugeordnete Steuernummer wird durch die Normalisierung dem Standard korrekt angepasst.

6.6.2.2 Attribute für Steuernummer und USt-IdNr.

Durch den strikten regulären Ausdruck für die Steuernummer und die USt-IdNr. können die Werte korrekt extrahiert werden, wenn alle angegebenen Steuernummern und USt-IdNr. dem Schema entsprechen. Tatsächlich kann man aber nicht zu viel erwarten, wie es das Beispiel in Abbildung 6.19 zeigt.

Auf der Informationsseite können neben der Steuernummer und der USt-IdNr. auch noch die Bankverbindungen und andere Zahlenkombinationen vorkommen. So mussten anfänglich zur korrekten Zuordnungen verschiedener Zahlenkombinationen bei der Lernphase Attributklassen für die Steuernummer und Ust-IdNr. erstellt werden.

Dabei fanden sich weniger Variationen für die Attributklasse „Steuernummer“ als für die USt-IdNr., welche über 100 Einträge verzeichnet.

Hierzu werden einige Beispiele für Steuernummer-Varianten in Tabelle 6.23 angegeben.

STEUER-NR	*St-Nr*	*St. Nr*
Steuer - Nr	*Steuer Nr*	*Steuer Nummer*
UST-Nr	*UST.-Nr*	*UST.NR*
Umsatzsteuer	*Umsatzsteuer Nr*	*Umsatzsteuer Nummer*

Tabelle 6.23: Attributklasse für „Steuernummern“

Für die USt-IdNr. sieht man in Tabelle 6.24 die folgenden Beispiele:

6.7 Öffnungszeiten

Die Öffnungszeiten werden in der Regel zusammen mit dem Wochentag angegeben, welche gern als Folge von 2 Zeichen abgekürzt, aber auch ausgeschrieben werden. Jedoch kann es manchmal der Fall sein, dass Wochentage als Adverb wiedergegeben werden. All diese Variationen werden in einem Lexikon festgehalten. Da aber die vorangehenden Kontexte oft nicht vorkommen und die Angabe normalerweise eindeutig anhand der Formatierung identifizierbar ist, wird kein linker Kontext vorausgesetzt. Meist wird zwischen den Wochentagen noch ein Bindestrich oder eine Präposition „*bis, von ... bis*“ eingesetzt.

Ust-Id Nr	*Ust-Id.Nr*
Ust-IdNr	*Ust-Ident Nr*
Ust-Ident-No	*Ust.-ID*
Ust.-ID-Nr	*Ust.-ID. Nr*
Ust.-Ident.-Nr	*VAT-/USt-ID*
VAT-ID	*VAT.ID-number*
VAT. ID	*Umsatzsteuer ID*
Umsatzsteuer ID-Nummer	*Umsatzsteuer- Identifikationsnummer*
UmsatzsteuerID	*Umsatzsteueridentifikations-Nummer*
Umsatzsteueridentifikationsnummer	

Tabelle 6.24: Attributklasse für „USt-IdNr.“

Beim Zeitformat werden entweder das 24-Stunden- oder 12-Stunden-System verwendet. Wenn das Zeitformat im 12er-System vorkommt, dann wird es mit nachgestellten *„a.m.“* und *„p.m.“* gekennzeichnet. Außerdem wird dann zwischen Stunden und Minuten oft ein Doppelpunkt oder ein einfacher Punkt eingesetzt.

Diese Beispiele werden in Tabelle 6.25 als eine Auswahl an Variationen für Öffnungszeiten aufgeführt.

Mo-Do 10-13 Uhr
Mo. - Sa. 8 -20 Uhr
Mo-Fr 14 bis 16 Uhr
Mo-Fr 09.00 - 12.00 Uhr
Mo-Fr von 9.00-17.30 Uhr
Montag - Freitag 9.00 - 17.00
Montag bis Freitag: 10.00h - 19.00h
montags bis donnerstags: 9.30 - 14.00 Uhr
Montags bis Freitags von 9.00 bis 13.00 Uhr
Montag/Dienstag/Donnerstag: 11.00 - 17.00 Uhr
täglich von 12-14 Uhr und 18-22 Uhr
täglich von 11 bis 14 Uhr und 17 - 24 Uhr

Tabelle 6.25: Beispiele für Öffnungszeiten

Kapitel 7

Evaluation des Systems

Zur Evaluation des Systems werden Präzision, Recall und F-Maß bewertet. Präzision und Recall sind mit folgender Formel berechnet.

$$Präzision = \frac{Korrekt\ extrahierte\ Daten}{Alle\ extrahierten\ Daten}$$

$$Recall = \frac{Korrekt\ extrahierte\ Daten}{Alle\ zu\ extrahierenden\ Daten}$$

Präzision und Recall verhalten sich komplementär. Mit der erhöhten Präzision fällt im Normalfall der Recallwert. Daher wird das System auch an Mittelwert von beiden, an F-Maß, bewertet. F-Maß wird wie folgt berechnet.

$$F_\alpha = \frac{(1+\alpha) * Präzision * Recall}{(\alpha * Präzision) + Recall}$$

Je höher „α“ ist, desto mehr der Recallwert gewichtet. Bei der Evaluation des Systems wird „α“ auf „1“ gesetzt. Damit werden Präzision und Recall gleich gewichtet.

Als Testdaten wurden zuerst 924 SLDs genommen. Daraus wurden 478 SLDs als Kandidaten zur Firmen-Website klassifiziert. Für den Test wurde jede dritte URL genommen, insgesamt 159 URLs. Die URLs wurden vom System verarbeitet, wobei fünf URLs doppelt vorkommen. Damit bleiben 154 URLs für die Evaluation.

Für den Vergleich zwischen den Webdaten und extrahierten Daten wurde jede URL manuell mit dem Browser „Firefox“ besucht. Für vier URLs wurde keine Adresse gefunden. Drei URLs waren eine Domain-Park-Site. Für eine URL wurde keine Adresse gefunden, obwohl sie angegeben war. Der Grund dafür war, dass die URL für die Informationsseite auf eine andere SLD umleitet. Da das System nur die internen Links verfolgt, wird die URL für Informationsseite einfach ignoriert. Daher wird auch diese SLD nicht bei der Evaluation berücksichtigt.

Von den übrigen 150 URLs wird jede Klasse evaluiert. Wie erwartet, waren nicht alle SLDs eine Firmen-Website. Insgesamt 19 SLDs fielen in die Bereiche „*Organisation, Schule, Information, Shop und Privat*“, was bereits bei der Evaluierung des Klassifikators erläutert wurde. Die Präzision des Klassifikators lag bei ca. 88%.

„*Postleitzahl*“ und „*Ortsname*“ wurden alle richtig erkannt. Wenn keine Postleitzahl gefunden wird, bricht das System einfach ab. Daher soll die Postleitzahl immer dabei sein.

Drei Straßennamen wurden entweder teilweise oder gar nicht erkannt. „*Im Bahnhof*“ ohne Hausnummer wurde beispielsweise nicht als „Straße“ erkannt.

„*Mobilfunknummer*“ war 13 Mal angegeben, und alle wurden richtig erkannt. „*Telefon-*“ und „*Faxnummer*“ wurde jeweils 137 Mal und 125 Mal genannt. Drei Telefonnummern wurden entweder falsch oder nicht erkannt, wobei nicht-erkannte Telfonnummern keinen Indikator hatten. Der HTML-Source-Code war eine Abbildung ohne einen alternativen Text, und die falsch erkannte Telefonnummer lag in textueller Form, wie „0700 TEATRON“, vor. „*Faxnummer*“ wurde nur ein Mal aufgrund fehlender Indikatoren nicht erkannt. Die nicht vorhandenen Indikatoren für Telefon- und Faxnummer stammten aus derselben Website.

„*E-Mail-Adresse*“ wurde 126 Mal angegeben. Davon wurden zwei E-Mail-Adressen nicht erkannt, da sie mittels „*Javascript*“ versteckt wurden und sich außerhalb des Adressblocks befanden. Deshalb wurde keine Analyse des „*Javascript*“ durchgeführt.

„*USt-IdNr.*“ kam 73 Mal vor. Davon wurde nur eine nicht erkannt. Es ist aber zu bemerken, dass 13 „*USt-IdNr.*“ eigentlich dem Schema einer „*Steuernummer*“ entsprachen. Erst durch das Postprocessing konnten sie richtig der Klasse „Steuernummer“ zugeordnet werden.

Die Klasse „*Steuernummer*“ trat insgesamt 25 Mal auf, wovon drei Steuernummern nicht erkannt wurden. Nach dem Besuch der Webseite konnte festgestellt werden, dass der Indikator nicht in der Datenbank war. Nach der Aufnahme des fehlenden Indikators wurden sie jedoch erkannt.

„*Finanzamt*“ tauchte vier Mal auf und wurde richtig erkannt. Die Klasse „*Amtsgericht*“ und die Klasse „*Registernummer*“ kommen oft zusammen vor. Sie wurden jeweils 44 und 45 Mal angegeben. Davon wurden jeweils sechs Mal „Amtsgericht“ und sieben Mal „Registernummer“ nicht erkannt. Sie wurden oft durch die Intervenierung der Kürzel verdunkelt, wie das folgende Beipiel unten zeigt.

```
Handelsregisternr.: HH 100042 Hamburg
```

Der externe Indikator erwartet eine gültige Registernummer. Dies wurde durch „HH“ verhindert.

Eine besondere Registernummer, die nicht in den Trainingsdaten vorkam, wird unten gezeigt.

```
Versicherungsvermittlerregister Nr.:D-TZJL-Z0W4W-36
```

Diese Nummer wurde logischerweise nicht erkannt.

In Betracht des „**Personennamens**“ kommt die Klasse „*Geschäftsführer*“ am häufigsten vor. Insgesamt wurde er 39 Mal angegeben, wovon elf Mal nicht erkannt wurde. Es fehlten entweder der „Indikator“ oder der reguläre Ausdruck für den „**Personenname**n“ war zu strikt. Z.B.

```
Dipl.-Ing. Architekt Christian Stanitzeck
Dipl.-Biol. Elek Szabo
```

wurden nicht als „**Personenname**“ erkannt, weil die Erweiterung durch „Architekt“ oder die erweiterte Form „Biol“ nicht in regulärem Ausdruck berücksichtigt wurde.

Dies gilt auch für alle anderen „*Personennamenklassen*“.

„*Veranwortlicher*“ kam 33 Mal vor. Davon wurden neun nicht erkannt. „*Inhaber*“ trat 24 Mal auf, wovon drei nicht identifiziert wurde. „*Vertreter*“ tauchte

12 Mal auf. Einer davon wurde nicht erkannt. Vier „*Vorsitzender*“ und ein „*Vorstand*“ wurden komplett richtig erkannt, während ein „*Aufsichtsratsvorsitzender*“ nicht erkannt wurde. Zwei „*Kontakpersonen*“ wurden erkannt und eine nicht.

Nun wenden wir uns aber der Firmennamenerkennung zu. „*Firmenname*“ wird „per Default“ erkannt, falls eine SLD aktiv und eine Adresse gegeben ist. Das bedeutet, dass jede SLD einen extrahierten Firmennamen bekommt. Deshalb soll nur die Fehlerkennung gewertet werden. Insgesamt wurden 26 Firmennamen nicht richtig erkannt. Davon fielen 5 SLDs auf „*Informaiton*“ oder „*Privat*“. Wenn wir sie abziehen, dann bleiben immer noch 21 falsch oder nicht (d.h. einfache Wiedergabe der SLD) erkannte Firmennamen.

Die Fehlerkennung kann durch „Abbildungen“ oder „Flash-Animationen“ verursacht werden. Bei einigen SLDs ist der Firmenname kaum zu erkennen. SLDs, wie „hardmedia“ und „koerperkult“, sind ein Beipiel dafür.

Zusammenfassend wird in Tabelle 7.1 eine Übersicht für jede einzelne Klasse angegeben (G: *Alle zu extrahierenden Daten*, E: *Alle extrahierten Daten*, C: *Korrekt extrahierte Daten*, F: *Falsch extrahierte Daten*, P: *Präzision*, R: *Recall*, F1: *F1-Maß*).

Es muss eingeräumt werden, dass die Evaluation des Systems mit dem richtig ausgefüllten Template durchgeführt werden muss. Sie wurde leider nicht gemacht.

Außer bei der Firmennamenerkennung wurde bei allen Klassen eine hohe Präzision erzielt.

Die Präzision der Firmennamenerkennung kann allerdings erhöht werden, wenn man die wiedergegebenen SLDs als Firmennamen abzieht. Das waren 16, somit bleiben nur fünf falsch erkannte Firmennamen. Wenn man daraus die Präzision neu berechnet, dann liegt sie bei

$$\frac{100 * 129}{150 - 16} = 96,26\%.$$

Dadurch liegt „F1-Maß“ bei

$$\frac{100 * 2 * 0.9626 * 0.86}{0.9626 + 0.86} = 90,84\%.$$

	G	E	C	F	P	R	F1
FN	150	150	129	21	86%	86%	86%
STR	150	149	147	2	98,6%	98%	98,3%
PLZ	150	150	150	0	100%	100%	100%
ORT	150	150	150	0	100%	100%	100%
TEL	137	135	134	1	99,2%	97,8%	98,5%
FAX	125	124	124	0	100%	99,2%	99,5%
MOB	13	13	13	0	100%	100%	100%
EMA	126	124	124	0	100%	98,4%	99,2%
UID	73	72	72	0	100%	98,6%	99,3%
STE	25	22	22	0	100%	88%	93,6%
GF	39	28	28	0	100%	71,7%	83,5%
INH	24	21	21	0	100%	87,5%	93,3%
VST	4	4	4	0	100%	100%	100%
KON	3	2	2	0	100%	66,6%	80%
VAT	33	24	24	0	100%	72,7%	84,2%
VTT	12	11	11	0	100%	91,6%	95,6%
VSD	1	1	1	0	100%	100%	100%
AST	1	0	0	0			
AG	44	38	38	0	100%	86,3%	92,7%
FG	4	4	4	0	100%	100%	100%
HR	45	38	38	0	100%	84,4%	91%

Tabelle 7.1: Evaluation einzelner Klassen

Kapitel 8

Datenbankaufbau und -verwaltung

Die extrahierten Firmendaten werden mithilfe einer Datenbank verwaltet, wodurch ihre Vollständigkeit und Aktualität sicher gestellt wird. Eine manuelle Datenverwaltung würde schnell an ihre Grenzen stoßen, wenn die Daten sehr groß werden. Ebenso ist die Aktualisierung bei der manuallen Verwaltung nachvollziehbar.

Es werden zwei Datenbanken aufgebaut, um die Arbeit effektiv zu verteilen: eine für Domain-Namen, die andere für Firmendaten. Diese kommunizieren interaktiv über das System. Die Abbildung 8.1 skizziert den Zusammenhang zwischen den beiden Datenbanken.

Das System selbst wurde in Abbildung 8.1 weggelassen[1], jedoch prüft es die Aktualität der Daten regelmäßig. Falls die Daten nicht aktuell sind, dann wird es den Domain-Namen überprüfen. Wenn der Domain-Name nicht mehr aktiv ist, dann wird er aus der DB gelöscht. Wenn der Domain-Name aber aktiv ist, dann wird nach der neuen Infomationsseite gesucht.

8.1 Datenbankstruktur

Ausgehend von dem Domain-Namen werden zwei Datenbanken erstellt, wie es aus Abbildung 8.1 ersichtlich wird. Über die Domain-Namen-Datenbank

[1] Für die gesamte Systemübersicht, s. Kapitel 4.

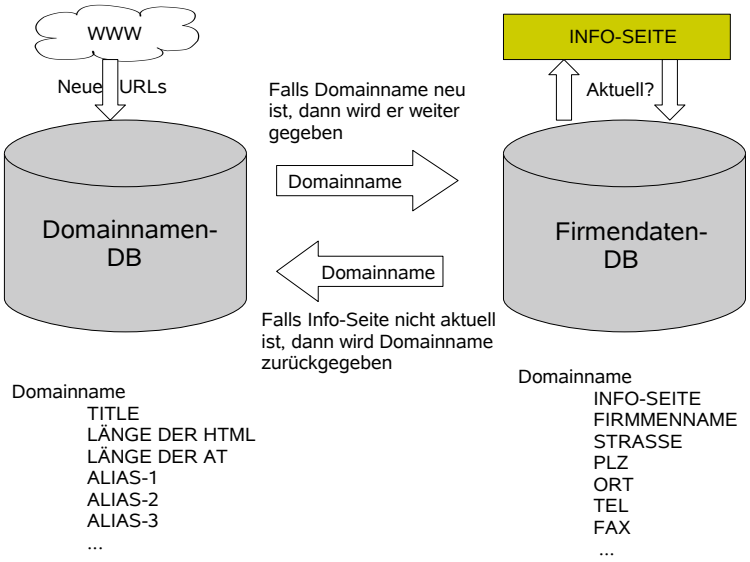


Abbildung 8.1: Zusammenhang zwischen Domain-Namen- und Firmendaten-DB

wird jede Firmen-Website eindeutig identifiziert, und dadurch wird doppeltes Crawlen der gleichen Websites sicher verhindert. Die andere Datenbank verwaltet die Firmeninformationen, und wird regelmäßig aktualisiert, indem neue Firmeninformationen hinzugefügt werden.

8.1.1 Datenbank für Domain-Namen

Die Domain-Namen-Datenbank wurde zur effektiven und persistenten Verwaltung der Domain-Namen aufgebaut. So wird ein vermeintlich neuer Domain-Name schnell überprüft, was durch den schnellen Hash-Lookup realisiert wird. Ebenfalls wird hier auch der Alias-Domain-Name geprüft, wofür die ersten drei Werte im DB-Schema sorgen: Titel, Länge des HTML-Textes, Länge der Anchortexte.

Die Domain-Namen-Datenbank hat folgende Einträge:

1. Schlüssel	2. Schlüssel	Wert
DOMAINNAME	TITLE	Titel
	LÄNGE der HTML	Länge der HTML
	LÄNGE DER AT	Länge der Anchortexte
		Alias-Domain-Name-1
	ALIAS	Alias-Domain-Name-2
		...
		Alias-Domain-Name-n

Tabelle 8.1: Struktur der Domain-Namen-Datenbank

8.1.1.1 Kanonische Form

Die kanonische Form des Domain-Namens ist der Host-Teil der Einstiegsseite. Dieser soll auch der kanonischen Form des Hosts entsprechend umgeschrieben werden, welche klein geschrieben wird. Hierfür werden Escape-Zeichen in große Hexadezimalen umgeschrieben. So ist die kanonische Form des Domain-Namens „sql-gmbh“ mit der Einstiegs-URL „`http://www2.sql-gmbh.de/sqlgmbh2007/`“ „`www2.sql-gmbh.de`“.

Die Einstiegs-URL „`http://www.xba.info`“ bekommt die Kanonische Form „`www.xba.info`“. Ohne „www“ oder „www2“ wird der Domain-Name funktionieren, da „www“ bei den meisten Domain-Namen als Default vorangefügt wird. Jedoch gibt es Beispiele, die ohne „www“ auftreten, wie z.B. „`de.yahoo.com`“. Wenn man hier „www“ voranstellt, wird die Adresse automatisch umgeleitet. Diese Behauptung ist aber nicht ohne weiteres zu halten. Während die URL „`http://de.personello.com/`“ aufrufbar ist, ist diese mit „www“ vorangestellte URL „`http://www.de.personello.com/`“ nicht erreichbar.

8.1.1.2 Verwaltung der Domain-Namen

Ist ein potenziell neuer Domain-Name verfügbar, wird er mit dem Schlüssel verglichen. Wenn dieser bereits existiert, wird er nicht als neu angenommen. Wenn dem nicht so ist, dann wird das Alias-Verfahren angewandt, was über die ersten drei Werte des DB-Schemas realisiert wird. Der Alias-Domain-Name hat den gleichen Titel, und beide HTML-Texte sind gleich lang. Die

meisten Alias-Domain-Namen sind mit diesen beiden Merkmalen zu identifizieren.

Da es aber viele stoppwortartige Titel wie „*Home, Startseite, ...*“ gibt[2], kann es vorkommen, dass die beiden verglichenen Domain-Namen zufällig den gleichen Titel und die gleiche Länge des HTML-Textes haben.

Um damit sicher umzugehen, wird noch ein drittes Merkmal aufgenommen: die Länge der gesamten Anchortexte. Mit diesen drei Merkmalen wurden bei den Trainingsdaten alle Alias-Domain-Namen erkannt. Falls ein als neu angenommener Domain-Name als „Alias“ verifiziert wird, dann muss er als Alias-Domain-Name in die Werteliste aufgenommen werden. Der Domain-Name selbst wird in Domain-Namen-Datenbank eingetragen. Dadurch wird sichergestellt, dass die Datenbank vollständig wird. Aber er wird nicht für den Zweck der Informationsextraktion aufgenommen. Wenn der Domain-Name neu ist, dann wird er mit den Werten „*Titel, Länge des HTML-Textes, Länge der Anchortexte*“ in die Domain-Namen-Datenbank aufgenommen.

8.1.1.3 Alias-Verfahren

Es wird ein Alias-Verfahren angewandt, um untereinander entsprechende Domain-Namen zu filtern. Alias-Domain-Namen kann man nach ihrem Charakter in drei Gruppen einteilen. So sind Alias-Domain-Namen u.a., mit kleinen Variationen registriert, was die folgenden Domain-Namen zeigen[3].

jcfit $\simeq$ *jcf-it*
ideebau $\simeq$ *idee-bau*
petrastock $\simeq$ *petra-stock*
taxineuwied $\simeq$ *taxi-neuwied*
petewyomingbender $\simeq$*: pete-wyoming-bender*
pluemersystemtechnik $\simeq$ *pluemer-systemtechnik*

internationalkinesiologyacademy $\simeq$ *internationalekinesiologieakademie* $\simeq$ *internationale-kinesiologie-akademie* $\simeq$ *international-kinesiology-*

[2]Man kann solche Stoppwörter zusammenstellen, was wenig Erfolg versprechend ist, weil man dazu ein anderes Merkmal für den Vergleich braucht.

[3]Hier werden nur die registrierten Domain-Namen angegeben. In der Datenbank werden sie jedoch mit der kanonischen Form eingetragen.

academy

Die zweite Gruppe der Alias-Domain-Namen ist mit einem Zusatz, wie Branche oder Ortsname, registriert. Oder es wird ein Teil des Domain-Namens durch Synonyme ersetzt.

primavera $\simeq$ *primavera-life*
datatronic $\simeq$ *dek-datatronic*
patenthotline $\simeq$ *patentberatung*
putzke-edv $\simeq$ *putzke-datentechnik*
injektionstechnik $\simeq$ *injektionsschlauch*
intensive-dachbegruenung $\simeq$ *intensivbegruenung*
privilegierte-apotheke $\simeq$ *privilegierte-apotheke-barmstedt*
haldermanns$\simeq$ *haldermann* $\simeq$ *halderman* $\simeq$ *haldermans-stb*
renaplan $\simeq$ *finanzierung-renaplan* $\simeq$ *finanzdienstleistungen-renaplan*
prima-klima-massivbau $\simeq$ *prima-klima-haus* $\simeq$ *prima-klima-fertighaus*
iduna-signal-versicherung $\simeq$ *iduna-signal-gruppe* $\simeq$ *iduna-sachversicherung*

pulvertrockner $\simeq$ *pulverofen* $\simeq$ *pulverlackieren* $\simeq$ *pulverlackieranlage* $\simeq$ *pulverlack* $\simeq$ *pulverkabine* $\simeq$ *pulverbeschichtungsanlage* $\simeq$ *pulverbeschichtung* $\simeq$ *pulverbeschichten* $\simeq$ *importanlage*

In der letzten Gruppe lässt sich die Relevanz der einzelnen Domain-Namen nicht abschätzen.

loepertz $\simeq$ *ibas-software*
vvf-verlag $\simeq$ *utz* $\simeq$ *tuduv*
transglutaminase $\simeq$ *ajitide*
gestaltung-in-stein $\simeq$ *baecht*
landhaus-schiffmann $\simeq$ *heilfasten* $\simeq$ *fasten*
ravensbergerholz $\simeq$ *holzsortiment* $\simeq$ *holzausstellung*

wortanzeige $\simeq$ *weihnachtsangebote* $\simeq$ *thomasvonloeper* $\simeq$ *preisumfrage* $\simeq$ *neuheitendienst* $\simeq$ *lgkatalog* $\simeq$ *briefmarkenzeitung* $\simeq$ *briefmarkenmessen* $\simeq$ *auswahldienst* $\simeq$ *atm-international*

schuttrohr $\simeq$ *schalungen* $\simeq$ *minifoerderband* $\simeq$ *mauertechnik* $\simeq$ *mauertec* $\simeq$ *lerch-von-huennebeck* $\simeq$ *lerch-gmbh* $\simeq$ *lerch-baugeraete* $\simeq$ *holztraeger* $\simeq$ *hfq* $\simeq$ *geruestprofi* $\simeq$ *gebrauchtes-geruest* $\simeq$ *gebrauchte-schalungen* $\simeq$ *gebrauchte-schalung* $\simeq$ *bauzaeune* $\simeq$ *baumaschiene* $\simeq$ *baugeraete-vermietung* $\simeq$ *bau-und-industriebedarf* $\simeq$ *an-verkauf*

Es muss jedoch eingeräumt werden, dass das hier vorgestellte Alias-Verfahren keine sprachliche Umstellung berücksichtigen kann. „`www.siemens.de`“ und „`www.siemens.com`“ werden nicht als Alias erkannt, weil „`www.siemens.com`“ auf Englisch vorliegt, und damit die Länge sowie der Inhalt nicht identisch sein wird.

8.1.1.4 Domain-Namen-Datenbank

Ein Ausschnitt der Domain-Namen-Datenbank wird in Tabelle 8.2 für drei Alias-Host-Namen gezeigt. Diese haben alle den gleichen Titel, die gleiche Länge an HTML-Text und Anchor-Text.

Gewinnung der neuen Domain-Namen: Zu diesem Zweck wurde zuerst der von M. Moricz konzipierte Crawler verwendet. Falls der Domain-Name als neu nachgewiesen werden kann, wird er in die DB aufgenommen.

Als zusätzliches Mittel zur Lokalisierung neuer Domain-Namen können allgemeine Suchmaschinen verwendet werden. Dafür werden gezielte Anfragen an sie geschickt, und aus der Antwort werden nur die URLs extrahiert. Domain-Namen können leicht mit dem in Abschnitt 2.1 des Kapitels 2 etablierten regulären Ausdruck extrahiert werden. Danach werden die extrahierten Domain-Namen mit den Domain-Namen in der Datenbank verglichen. Wenn sie tatsächlich neu sind, dann werden sie auch eingetragen.

8.1.2 Datenbank für Firmendaten

Die extrahierten Firmendaten werden mittels einer Datenbank automatisch verwaltet, da sich Firmendaten ändern. Eine Firma kann umgezogen sein,

DAMAINNAME	ATTRUBUT	WERT
www.utz.de	TITLE	Herbert Utz Verlag GmbH - Fachveröffentlichungen, Dissertationen, Habilitationen
	LÄNGE DER HTML	33628
	LÄNGE DER AT	161
	ALIAS	www.vvf-verlag.de
		www.tuduv.de
		utz.de
		vvf-verlag.de
		tuduv.de
www.vvf-verlag	TITLE	Herbert Utz Verlag GmbH - Fachveröffentlichungen, Dissertationen, Habilitationen
	LÄNGE DER HTML	33628
	LÄNGE DER AT	161
	ALIAS	www.tuduv.de
		www.utz.de
		utz.de
		vvf-verlag.de
		tuduv.de
www.tuduv.de	TITLE	Herbert Utz Verlag GmbH - Fachveröffentlichungen, Dissertationen, Habilitationen
	LÄNGE DER HTML	33628
	LÄNGE DER AT	161
	ALIAS	www.utz.de
		www.vvf-verlag.de
		utz.de
		vvf-verlag.de
		tuduv.de

Tabelle 8.2: Beispieldatenbank für Domain-Namen

oder einen neuen Geschäftsführer haben. Die Aktualisierung der Daten kann nicht manuell erfolgen, außer man verfügt über enorm viel Personal. Durch

den regelmäßigen Besuch der Informationsseiten werden die erneuerten Daten automatisch erkannt und ersetzen die alten Daten. Neue Daten werden auch automatisch hinzugefügt. Durch die automatisierte Datenverwaltung können die Daten immer mehr vervollständigt werden.

Als Datenbasis wird eine einfache Multi-Level-Hash-Tabelle angelegt. Ausgehend von den Domain-Namen werden die Informationsseiten und die daraus extrahierten Daten als Werte gespeichert. Das DB-Schema der Firmendaten wird in Tabelle 8.3 durch ein Beispiel veranschaulicht.

Die Informationsseite wird analog zu der im Kapitel 5 bereits genannten Reihenfolge von "Impresseum, Kontakt, Profil, Einstiegsseite" besucht. Die Öffnungszeiten und Kontaktdaten von Telefon- & Faxnummer und E-Mail-Adresse können nicht immer aus der Impressumseite extrahiert werden. In diesem Fall wird die Kontaktseite für Kontaktdaten und die Einstiegsseite für Öffnungszeit verwendet. Es wird angenommen, dass die Informationsseite eindeutig identifiziert wird.

8.2 Aktualisieren der Daten

Das Update der Daten erfolgt durch regelmäßigen Besuch der Informationsseite. Beim Besuch der Seite wird zuerst nach der modifizierten Zeit gesucht. Wenn sie vorhanden und nicht geändert worden ist, dann werden die Daten ohne Aktualisierung als aktuell betrachtet. Wenn die Seite modifiziert worden ist, dann werden die Daten neu extrahiert. Falls die Seite umgezogen ist, dann wird der Domain-Name besucht, und nach der neuen Seite gesucht, und die Daten werden komplett neu extrahiert. Falls auch der Domain-Name verändert oder nicht mehr aktuell ist, dann werden die Daten aus der Datenbank genommen. Auch der Domain-Name wird aus Domain-Namen-Datenbank genommen. Dadurch werden Domain-Namen- und Firmendaten-Datenbank aktualisiert, und der neueste Stand der Daten wird sichergestellt.

DOMAINNAME	ATTRIBUT	WERT
www2.sql-gmbh.de	INFORMATIONSSEITE	http://www2.sql-gmbh.de /sqlgmbh2007/menue-right/ impressum.html
	KATEGORIE	Firma
	FIRMENNAME	SQL Gesellschaft für Datenverarbeitung mbH
	STRASSE	Franklinstraße 25a
	POSTLEITZAHL	01069
	ORT	Dresden
	TELEFONNUMMER	(0351) 876190
	FAXNUMMER	(0351) 8761999
	MOBIL	
	EMAIL	info@sql-gmbh.de
	USTID	DE140300780
	STEUERNUMMER	
	GESCHÄFTSFÜHRER	Dipl.-Ing.oec. Jürgen Bittner
	INHABER	
	VORSITZENDER	
	KONTAKTPERSON	
	VERANTWORTLICHER	
	VERTRETER	
	VORSTAND	
	AUFSICHTSRATS-VORSITZENDER	
	AMTSGERICHT	Dresden
	FINANZAMT	
	REGISTERNUMMER	HRB 5256
	ÖFFNUNGSZEIT	

Tabelle 8.3: Datenbankstruktur bei den Firmendaten

Kapitel 9

Zusammenfassung und Aussichten

Das entwickelte System *ACIET* extrahiert Firmeninformationen nur aus den Informationsseiten einer Firmen-Website. Informationen werden hierfür auf domainnamenrelevante Informationen eingeschränkt.

Firmen-Websites nehmen in unserer Internet-Gesellschaft einen bedeutenden Stellenwert ein. Einerseits ändern sich häufig Domain-Inhaber und Seiteninhalte. Andererseits werden täglich neue Domains registriert. Dadurch steigt das Bedürfnis nach einem zentralen Firmenverzeichnis. Beim manuellen Erstellen und Verwalten eines solchen Verzeichnisses stößt man schnell an Grenzen.

Das System *ACIET* automatisiert den Prozess, Informationen zu extrahieren und zu verwalten. Es versucht zuerst, den gegebenen Domain-Namen anhand struktureller und textueller Merkmale aus der Einstiegsseite zu kategorisieren. Wenn dieser als Firmen-Website erkannt wurde, wird nach Informationsseiten gesucht. Dabei wird eine gezielte Link-Verfolgung angewandt.

Für die Klassifikation werden sowohl strukturelle als auch textuelle Merkmale aus der Einstiegsseite zusammengestellt. Die Klassifikation erfolgt nur mit diesen Merkmalen. Dadurch konnten die Bandbreite und der Zeitaufwand enorm reduziert werden. Mithilfe der strukturellen Merkmale wurden mehr als die Hälfte der Websites klassifiziert. Auf die textuellen Merkmale wurde eine Naive Bayes'sche Klassifikation angewandt.

Nachdem *ACIET* Informationsseiten gefunden hat, werden Werte für die vorgegebenen Attribut-Felder extrahiert. Dabei arbeitet *ACIET* mit der HTML-Baumstruktur und kontextuellen Indikatoren. Dadurch wird der Vorteil eines HTML-Textes maximal ausgenutzt, und es kann auf ein großes Lexikon verzichtet werden. Durch das Attribut-Wert-Verfahren kann auch das nicht richtig zugeordnete Attribut-Wert-Paar korrekt extrahiert werden. Interne und externe Indikatoren ermöglichen, dass der gesuchte Wert leicht und korrekt lokalisiert wird.

Hohe Präzision und Recall zeichnen *ACIET* aus. Durch seinen Einsatz wird ein Firmenverzeichnis erstellt und aktuell gehalten. Davon profitieren besonders Anwendungen, wie Gelbe Seiten oder Jobsuchmaschinen. Über das automatisierte Vorgehen kann die Aktualität der Daten leichter sichergestellt werden. Vorteilhaft erweist sich hierbei auch die Schnelligkeit von *ACIET*.

Durch die modulare Bauweise von *ACIET* kann das System leicht auf andere Sprachen übertragen werden. Dazu benötigt das System lediglich die kontextuellen Indikatoren und das landesspezifische Adressformat. Das System wurde bereits mit Erfolg auf Firmendaten aus Österreich (AT) und dem Vereinigten Königreich Großbritannien (UK) adaptiert.

ACIET kann auch erweitert werden. Z.B. könnte zusätzlich ein Text-Analyse-Modul integriert werden, womit man u.a. die Brancheninformation aus dem Vorstellungstext der Firmen-Website gewinnen könnte.

Anhang A

Verwendete und referenzierte Open-Source-Produkte

A.1 Unix-Tools und freie Software

Sun Wu & Udi Manber, Agrep, Version 2.04. 1992. http://packages.debian.org/de/source/sarge/agrep.

Lou Montulli, Garrett Blythe, Craig Lavender, Michael Grobe, Charles Rezac und Foteos Macrides, Lynx, Version 2.8.5. 2005. http://lynx.isc.org/lynx2.8.5/index.html

Dave Raggett, HTML Tidy for Linux/x86, released on 12 April 2005. 2005. http://tidy.sourceforge.net/.

Hrvoje Niksic, Wget, Version 1.10.2. 2005. http://www.gnu.org/software/wget/.

Weitere Unix-Tools

A.2 CPAN

Gisle Aas, LWP::UserAgent. 2004. http://search.cpan.org/~gaas/libwww-perl-5.808/lib/LWP.pm.

Gisle Aas, HTML::Entities. 2006. http://search.cpan.org/~gaas/HTML-Parser-3.56/lib/HTML/Entities.pm.

Gisle Aas, Sean Burke und Andy Lester, HTML::TreeBuilder, Verwaltet von Pete Krawczyk. 2006. http://search.cpan.org/search?query=html%3A%3Atreebuilder&mode=all.

Joshua Chamas, MLDBM. 2002. http://search.cpan.org/~chamas/MLDBM-2.01/lib/MLDBM.pm.

Jarkko Hietaniemi, String::Approx. 2006. http://search.cpan.org/~jhi/String-Approx-3.26/Approx.pm.

Dan Kogai, Regexp::Trie. 2006. http://search.cpan.org/~dankogai/Regexp-Trie-0.02/lib/Regexp/Trie.pm.

Dan Kogai, Encode. 2006. http://search.cpan.org/~dankogai/Encode-2.23/Encode.pm.

Ken Williams, AI::Categorizer::Learner::NaiveBayes. 2003. http://search.cpan.org/~kwilliams/AI-Categorizer-0.09/lib/AI/Categorizer/Learner/NaiveBayes.pm.

Ken Williams, AI::Categorizer. 2003. http://search.cpan.org/~kwilliams/AI-Categorizer-0.09/lib/AI/Categorizer.pm.

Ken Williams, Algorithm::NaiveBayes. 2003. http://search.cpan.org/~kwilliams/Algorithm-NaiveBayes-0.04/lib/Algorithm/NaiveBayes.pm.

Weitere CPAN-Module

A.3 PERL Referenzbücher

Randal L. Schwartz und Tom Phoenix, *Learning Perl*, 3rd Edition. O'Reilly. 2001.

Joseph N. Hall mit Randal Schwartz, *Effective perl Programming: Writing Better Programs with Perl*. Addison Wesley. 1998.

Larry Wall, Tom Christiansen und Jon Orwant, *Programming Perl*, 3rd Edition. O'Reilly. 2000.

Tom Christiansen & Nathan Torkington, *Perl Cookbook*, 2nd Edition. O'Reilly. 2003.

Jon Orwant, Jarkko Hietaniemi und John Macdonald, *Mastering Algorithms with Perl*, 1st Edition. O'Reilly. 1999.

Damian Conway, *Object oriented Perl*. Manning. 1999.

Sean M. Burke, *Perl and LWP*. O'Reilly. 2002.

Anhang B

Erstellte Lexika und Kontextdateien

B.1 Lexika

Lexikon der allgemeinen Firmentypen

Lexikon der allgemeinen Berufsbezeichnungen

Lexikon der Firmensuffixe

Lexikon der Firmenpräfixe

Lexikon der Straßensuffixe

Lexikon der PLZen, Orte und Telefonvorwahl

Lexikon der Fachbezeichnungen

Lexikon der Wochentage

B.2 Kontextdateien

Sichere Kontextdatei für die Firmennamenerkennung

Kontextdatei für die Firmennamenerkennung

Kontextdatei für die Erkennung der Telefonnummer

Kontextdatei für die Erkennung der Faxnummer

Kontextdatei für die Erkennung der Mobilfunknummer

Kontextdatei für die Erkennung der E-Mail-Adresse

Kontextdatei für die Erkennung der USt-IdNr.

Kontextdatei für die Erkennung der Steuernummer

Kontextdatei für die Erkennung des Geschäftsführers

Kontextdatei für die Erkennung des Inhabers

Kontextdatei für die Erkennung der Kontaktperson

Kontextdatei für die Erkennung des Vorsitzenden

Kontextdatei für die Erkennung des Vorstands

Kontextdatei für die Erkennung des Verantwortlichen

Kontextdatei für die Erkennung des Vertreters

Kontextdatei für die Erkennung des Vorsitzenden des Aufsichtsrates

Kontextdatei für die Erkennung des Amtsgerichts

Kontextdatei für die Erkennung des Finanzamtes

Kontextdatei für die Erkennung der Registernummer

B.3 Weitere Listen

Stoppwortliste für die Website-Klassifikation

Stoppwortliste für die Analyse der Informaionsseite

Liste für den Web-Designer

Liste der bekannten Abkürzungen

Anhang C

Auszug aus den verwendeten regulrären Ausdrücken

Firmensuffix:

```
(?-xsm:(?:\&\ (?:Collegen|Kollegen?|Partner|Sohn)|\(\ GmbH\ \&\ Co\ \)\
KG|AG(?:\ \&\ Co\.\ KGaA)?|Co(?:nsulting|rp(?:oration)?)|E(?:WIV|ngineer
ing)|G(?:\.b\.R|BR|bR|mbH(?:\ (?:\&\ Co(?:(?:\ KG|\.(?:\ (?:Handels\ und
\ Service\ KG|KG(?:aA)?|OHG)|KG)|KG))?|i\.G|und\ Co\.\ KG))?)|In(?:vAG|c
)|Jun|K(?:\.(?:d\.ö\.R|G)|G(?:aA)?|OLLEGEN|o(?:llegen|mmunikation))|L(?:
TD|td)|OHG|Part(?:enreederei|G)|S(?:CE|ervice|tille\ Gesellschaft)|V(?:V
aG|erein)|e(?:\.(?:\ V|[GKV])|G)|g(?:AG|GmbH|mbh)|jun|ltd|mbH|oHG|u(?:\.
\ (?:Koll|Partner)|nd\ Partner)))
```

Straßensuffix:

```
(?-xsm:(?:a(?:cker|llee|nger|venue|u)|b(?:a(?:ch|hn|[du])|erge?|lick|oge
n|rücke|u(?:rg|sch)|ühl)|ch(?:aussee|en)|d(?:amm|eich|orf)|e(?:cke?|rei)
|f(?:eld|lur)|g(?:a(?:rten|sse|ß)|r(?:ab(?:en)?|enze|u(?:be|nd))|ä(?:rte
n|ssle))|h(?:a(?:fen|in|ng|us(?:en)?)|ei(?:de|m)|o(?:hl|rst|f)|uette|öhe
|ütte)|k(?:a(?:mp|nal|pelle)|irche|reuz|ämpe)|l(?:ach|ein)|m(?:oo[rs]|üh
l(?:en?)?)|p(?:ark|fad|latz|romenade)|r(?:ing|uine)|s(?:ch(?:acht|loß)|e
e|i(?:edlung|tz)|t(?:e(?:ige?|tten?|g)|ieg|r(?:a(?:sse|ße))?|ätten?))|t(
?:al|eich)|ufer|w(?:al(?:de?|l)|eg|ie(?:sen?|te))|äcker))
```

Kontextdatei für die Erkennung der Telefonnummer:

```
(?-xsm:(?:Fon(?:\ Zentrale)?|Info\-Tel|Phone|Servicerufnummer|T(?:EL(?:E
FON)?|alr|el(?:(?:\ \+\ Fax|\-Nr|\.(?:\ (?:\&\ Fax|\+\ Fax|\/\ Fax)|\-Nr
|\/Fax|\:\ unter)|\/Fax|\:\ Bundesrepublik\ Deutschland|Nr|efon(?:(?:\ (
?:Support\ Deutschland|und\ Fax)|\/Fax|nummer))?))?)|Zentral(?:ruf(?:num
mer)?|e)|direktline|fon|hotline|phone?|s(?:ervice(?:line|nummer)|witchbo
ard)|tel(?:efon(?:nummer)?)?))
```

Literaturverzeichnis

[1] S. P. Algur and P. S. Hiremath. Extraction of Flat and Nested Data Records from Web Pages. In *Conferences in Reasearch and Practice in Information Technology*, volume 61, Sydney, Australia, 2006. Online: `http://crpit.com/confpapers/CRPITV61Algur.pdf`, Last Checked: 2007-12-01.

[2] I. S. Altingövde and Özgür Ulusoy. Exploiting Interclass Rules for Focused Crawling. *IEEE Intelligent Systems*, pages 66–73, 2004. Online: `http://www.cs.bilkent.edu.tr/~oulusoy/ieee_intelligent_systems.pdf`, Last Checked: 2007-12-01.

[3] B. Amento, L. Terveel, and W. Hill. Does "Authority" mean Quality? Predicting Expert Quality Ratings of Web Documents. In *SIGIR 2000*, Athen, Greece, 2000. ACM 1-58113-226.3/00/0007. Online: `http://www-users.cs.umn.edu/~terveen/papers/sigir2000.pdf`, Last Checked: 2007-12-01.

[4] E. Amitay, D. Carmel, A. Darlow, R. Lempel, and A. Soffer. The Connectivity Sonar: Detecting Site Functionality by Structural Patterns. In *HT' 03*, Nottingham, UK, 26-30 Aug. 2003. ACM 1-58113-704-4/03/0008. Online: `http://einat.webir.org/Hypertext_2003_p38-amitay.pdf`, Last Checked: 2007-12-01.

[5] Andrew McCallum and Dayne Freitag and Fernando Pereira. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *Proc. 17th International Conf. on Machine Learning*, pages 591–598, San Francisco, CA, 2000. Morgan-Kaufmann. Online: `http://www.cs.umass.edu/~mccallum/papers/memm-icml2000.ps`, Last Checked: 2007-12-01.

[6] A. Badia, T. Muezzinoglu, and O. Nasraoui. Focused Crawling: Experiences in a Real World Project. In *WWW'06*, Edinburgh, Scotland, 22-26 May 2006. ACM 1-59593-332-9/06/0005. http://louisville.edu/~o0nasr01/Websites/PAPERS/conference/Focused-Crawling-Badia-Moezzinoglu-Nasraoui.pdf, Last Checked: 2007-12-01.

[7] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open Information Extraction from the Web. In *Proceedings of IJCAI*, 2007. Online: http://turing.cs.washington.edu/papers/ijcai07.pdf, Last Checked: 2007-12-01.

[8] R. Baumgartner, S. Flesca, and G. Gottlob. Supervised Wrapper Generation with Lixto. In *Procedings of the 27th VLDB Conference*, Rome, Italy, 2001. Online: http://www.vldb.org/conf/2001/P715.pdf, Last Checked:2007-12-01.

[9] R. Baumgartner, S. Flesca, and G. Gottlob. Visual Web Information Extraction with Lixto. In *Proceedings of the 27th VLDB Conference*, Roma, Italy, 2001. Online: http://www.vldb.org/conf/2001/P119.pdf, Last Checked: 2007-12-01.

[10] T. Berners-Lee. Uniform Resource Identifier (URI): Generic Syxtax, 2005. Online: http://www.ietf.org/rfc/rfc3986.txt, Last Checked: 2007-12-01.

[11] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Link Analysis Ranking: Algorithms, Theory, and Experiments. *ACM Transactions on Internet Technology*, 5(1):231–297, Feb. 2005.

[12] T. M. Breuel. Information Extraction from HTML Documents by Structural Matching, 2003. Online: http://www.csc.liv.ac.uk/~wda2003/Papers/Section_I/Paper_3.pdf, Last Checked: 2007-12-01.

[13] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine, 1998. Online: http://infolab.stanford.edu/~backrub/google.html, Last Checked: 2007-12-01.

[14] S. Bsiri. Infromationsextraktion zur automatisierten Erzeugung einer Datenbank frankophoner Stellenanzeigen: Deutsche Zusamenfassung der auf französisch verfassten Dissertation. Technical report, CIS (Centrum für Informations- und Sprachverarbeitung), München, Germany, 2007. CIS Arbeitsbericht.

[15] R. Burget. *Information Extraction from HTML Documents Based on Logical Document Structure*. PhD thesis, Brno University of Technology, Brno, CZ, 2004. Online: `http://www.fit.vutbr.cz/~burgetr/thesis/burget_thesis.pdf`, Last Checked: 2007-12-01.

[16] H. Bußmann. *Lexikon der Sprachwissenschaft, 2., völlig neu bearbeitete Auflage*. Kröner, Stuttgart, 1990.

[17] M. J. Cafarella, D. Downey, S. Soderland, and O. Etzioni. KnowItNow: Fast, Scalable Information Extraction from the Web. In *HLT/EMNLP*, Vancouver, Oct. 2005. Online: `http://acl.ldc.upenn.edu/H/H05/H05-1071.pdf`, Last Checked:2007-12-01.

[18] S. Chakrabarti. Data Mining for Hypertext: A Tutorial Survey. *SIGKDD: SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining, ACM*, 1, 2000. Online: `http://www.cs.berkeley.edu/~soumen/doc/kddexp2000/survey.ps`, Last Checked:2007-12-01.

[19] S. Chakrabarti. *Mining the Web. Discovering Knowledge from Hypertext Data*. Morgan-Kaufmann, San Francisco, CA, 2003.

[20] S. Chakrabarti, K. Punera, and M. Subramanyam. Accelerated Focused Crawling through Online Relevance Feedback. In *WWW'02*, Honolulu, Hawaii, 7-11 May 2002. Online: `http://www.cse.iitb.ac.in/~soumen/doc/www2002m/p336-chakrabarti.pdf`, Last Checked: 2007-12-01.

[21] S. Chakrabarti, M. van den Berg, and B. Dom. Focused Crawling: a new approach to topic-specific Web resource discovery. In *WWW'99*, pages 545–562, Toronto, Canada, 1999. Elsevier Science B.V. Online: `http://www.cse.iitb.ac.in/~soumen/doc/www1999f/pdf/www1999f.pdf`, Last Checked: 2007-12-01.

[22] C.-H. Chang, M. Kayed, M. R. Girgis, and K. Shaalan. A Survey of Web Information Extraction Systems. *IEEE Transactions on Knowledge and Data Engineering*, 2007. Online: `http://www.csie.ncu.edu.tw/~chia/pub/iesurvey2006.pdf`, Last Checked: 2007-12-01.

[23] C.-H. Chang and S.-C. Lui. IEPAD: Information Extraction based on Pattern Discovery. In *WWW'01*, Hong Kong, 2-5 May 2001. ACM 1-58113-348-0/01/0005. Online: `http://www10.org/cdrom/papers/223/index.html`, Last Checked: 2007-12-01.

[24] K. C.-C. Chang, B. He, C. Li, M. Patel, and Z. Zhang. Structured Databases on the Web: Observations and Implications. Technical Report UIUCDCS-R-2003-2321, Dept. of Computer Science, UIUC, Feb. 2003. Online: `http://eagle.cs.uiuc.edu/pubs/2004/dwsurvey-sigmodrecord-chlpz-aug04.pdf`, Last Checked: 2007-12-01.

[25] Z. Chen, S. Liu, L. Wenyin, G. Pu, and W.-Y. Ma. Building a Web Thesaurus from Web Link Structure. In *SIGIR'03*, Toronto, Canada, 28 Jul. - 1 Aug. 2003. Online: `http://research.microsoft.com/~zhengc/papers/p14325-chen.pdf`, Last Checked: 2007-12-01.

[26] C.-H. Chi and C. Ding. Word Segmentation and Recognition for Web Document Framework, 1999. Online: `http://www.scs.ryerson.ca/~cding/papers/cikm99-wordseg.pdf`, Last Checked: 2007-12-01.

[27] C. Chow and R. Miller. Learning Wrappers Efficiently for Web Information Extraction Using Unlabeled Examples. In *AAAI'05*. AAAI Press, 2005. Online: `http://groups.csail.mit.edu/uid/projects/wrappers/aaai05.pdf`, Last Checked: 2007-12-01.

[28] W. W. Cohen. Improving a Page Classifier with Anchor Extraction and Link Analysis. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1481–1488. MIT Press, Cambridge, MA, 2003. Online: `http://books.nips.cc/papers/files/nips15/AP11.pdf`, Last Checked: 2007-12-01.

[29] W. W. Cohen, M. Hurst, and L. S. Jensen. A Wrapper Induction System for Complex Documents, and its Application to Tabular Data

on the Web. In A. Antonacopoulos and J. Hu, editors, *Web Document Analysis*, pages 155–177. World Scientific, New Jersey, 2003. Online: `http://www.cs.cmu.edu/~wcohen/postscript/ws-chap-2002.pdf`, Last Checked: 2007-12-01.

[30] N. Craswell, D. Hawking, and S. Robertson. Effective Site Finding using Link Anchor Information. In *SIGIR'01*, New Orleans, Lousiana, 9-12 Sep. 2001. Online: `http://research.microsoft.com/users/nickcr/pubs/craswell_sigir01.pdf`, Last Checked: 2007-12-01.

[31] T. C. Cravan. HTML Tags as Extraction Cues for Web Page Description Construction. *Informaing Science Journal*, 6, 2003. Online: `http://inform.nu/Articles/Vol6/v6p001-012.pdf`, Last Checked: 2007-12-01.

[32] V. Crescenzi, G. Mecca, and P. Merialdo. RoadRunner: Towards Automatic Data Extraction from Large Web Sites. In *Proceedings of 27th International Conference on Very Large Data Bases (VLDB'01)*, pages 109–118, 2001. Online: `http://www.dia.uniroma3.it/~vldbproc/015_109.pdf`, Last Checked: 2007-12-01.

[33] M. I. Devi and K. Selvakuberan. Fast web page categorization without the web page, 2005. Online: `https://drtc.isibang.ac.in/bitstream/1849/391/1/p81_selva-kuberan.pdf`, Last Checked: 2007-12-01.

[34] M. Diligenti, F. Coetzee, S. Lawrence, C. L. Giles, and M. Gori. Focused Crawling using Context Graphs. In *26th International Conference on Very Large Databases, VLDB 2000*, pages 527–534, Cairo, Egypt, 10-14 Sep. 2000. Online: `http://www.beyond-media.net/fileadmin/user_upload/quellen/focused_crawling_using.pdf`, Last Checked: 2007-12-01.

[35] D. Downey, M. Broadhead, and O. Etzioni. Locating Complex Named Entities in web Text. In *Proc. of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07)*, Hyderabad, India, 2007. Online: `http://turing.cs.washington.edu/papers/IJCAI-DowneyD1178.pdf`, Last Checked: 2007-12-01.

[36] D. Downey, O. Etzioni, S. Soderland, and D. S. Weld. Learning Text Patterns for Web Information Extraction and Assessment. In *AAAI'04 Workshop*. AAAI Press, 2002. Online: `http://www.cs.washington.edu/research/knowitall/papers/DowneyATEM04.pdf`, Last Checked: 2007-12-01.

[37] M. Dredze, J. Blitzer, P. P. Talukdar, K. Ganchev, J. V. Graça, and F. Pereira. Frustratingly Hard Domain Adaptation for Dependency Parsing. In *Proceedings of the CoNLL Shared Test Session of EMNLP-CoNLL*, pages 1051–1055, Prague, CZ, Jun. 2007. Association for Computational Linguistics. Online: `http://www.seas.upenn.edu/~mdredze/publications/adaptation_conll07.pdf`, Last Checked: 2007-12-01.

[38] M. Ehrig, J. Hartmann, and C. Schmitz. Ontologie-basiertes Web Mining. In P. Dadam and M. Reichert, editors, *Informatik 2004 - Informatik verbindet, Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V., Workshop Semantische Technologien für Informaitonsportale*, volume 2, pages 187–193. Köllen Druck+Verlag, Bonn, Germany, Sep. 2004. Online: `http://www.kde.cs.uni-kassel.de/schmitz/publ/2004-gi-crawler_v3.0.pdf`, Last Checked: 2007-12-01.

[39] M. Ehrig and A. Maedche. Ontology-Focused Crawling of Web Documents. In *SAC 2003*, Melbourne, Florida, 2003. Online: `http://www.aifb.uni-karlsruhe.de/WBS/meh/publications/ehrig03ontology.pdf`, Last Checked: 2007-12-01.

[40] L. Eikvil. Information Extraction from World Wide Web - A Survey -. Technical report, Norwegian Computing Center, 1999. Online: `http://www.nr.no/files/samba/bamg/webIE_rep945.ps`, Last Checked: 2007-12-01.

[41] N. Eiron and K. S. McCurley. Analysis of Anchor Text for Web Search. In *SIGIR'03*, Toronto, Canada, 2003. Online: `http://www.mccurley.org/papers/anchor.pdf`, Last Checked: 2007-12-01.

[42] D. W. Embley, D. Lopresti, and G. Nagy. Notes on Contemporary Table Recognition. In H. Bunke and A. L. Spitz, editors, *DAS 2006*, number

3872 in LNCS, pages 164–175. Springer-Verlag, Berlin Heidelberg, Germany, 2006. Online: `http://www.ecse.rpi.edu/homepages/nagy/PDF_files/Lopresti_Nagy_DAS06.pdf`, Last Checked: 2007-12-01.

[43] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised Named-Entity Extraction from the web: An Experimental Study. *Artificial Intelligence*, 165(1):191–134, 2005. Online: `http://www.cs.washington.edu/homes/etzioni/papers/knowitall-aij.pdf`, Last Checked: 2007-12-01.

[44] O. Etzioni, S. Kok, M. Cafarella, A.-M. Popescu, D. S. Weld, D. Downey, T. Shaked, and A. Yates. Web-scale Information Extraction in KnowItAll (Preliminary Results). In *WWW'04*, New York, NY, 17-20 May 2004. Online: `http://turing.cs.washington.edu/papers/www-paper.pdf`, Last Checked: 2007-12-01.

[45] M. Fathi, N. Adly, and M. Nagi. Web Documents Classification Using Text, Anchor, Title and Metadata Information. In *CSITeA 2004*, pages 445–452, Cairo, Egypt, 2004.

[46] C. G. Figuerola, J. L. A. Berrocal, A. F. Z. Rodriguez, and E. Rodriguez. REINA at the WebCLEF Task: Combining evidences and Link Analysis, 2005. Online: `http://www.clef-campaign.org/2005/working_notes/workingnotes2005/figuerola05.pdf`, Last Checked: 2007-12-01.

[47] A. Finn and N. Kushmerick. Information Extraction by Convergent Boundary Classification. In *Proc. Workshop Adaptive Text Extraction and Mining*, 2004. Online: `http://kushmerick.org/nick/research/download/finn-aaai04-atem.pdf`, Last Checked: 2007-12-01.

[48] S. Flesca, G. Manco, E. Masciari, L. Pontieri, and A. Pugliese. Exploiting Structural Similarity For Effective Web Information Extraction. In F. Neven, T. Schwentick, and D. Suciu, editors, *Foundations of Semistructured Data*, number 05061 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2005. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany. Online: `http://drops.dagstuhl.de/opus/volltexte/2005/`

230/pdf/05061.MasciariElio.Paper.230.pdf, Last Checked: 2007-12-01.

[49] J. E. Friedl. *Mastering Regular Expressions, Third Edition*. O'Reilly, Beijing, 2006.

[50] A. Fujii, K. Itou, T. Akiba, and T. Ishikawa. Exploiting Anchor text for the Navigational Web Retrieval at NTCIR-5. In *Proceedings of NTCIR-5 Workshop Meeting*, Tokyo, Japan, 6-9 De. 2005. Online: http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings5/data/WEB/NTCIR5-WEB-FujiiA.pdf, Last Checked: 2007-12-01.

[51] A. Fukumoto, T. Endo, and K. Shimada. Information Extraction from Specifications on the World Wide Web. In *PACLING'01*, 2001. Online: http://www.afnlp.org/archives/pacling2001/pdf/fukumoto.pdf, Last Checked: 2007-12-01.

[52] X. Gao, M. Zhang, and P. Andreae. Learning Information Extraction Patterns from Tabular Web Pages without Manual Labelling. Technical Report CS-TR-03/3, Victoria University of Wellington: School of Mathematical and Computing Sciences, Mar. 2003.

[53] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krüpl, and B. Pollak. Towards Domain-Independent Information Extraction from Web Tables. In *WWW'07*, Banff, Alberta, 2007.

[54] M. Geierhos. Grammatikder Menschenbezeichner in biographischen Kontexten. Master's thesis, LMU München, München, Germany, 2006. Online: http://www.cis.uni-muenchen.de/~micha/publikationen/magisterarbeit.pdf, Last Checked: 2007-12-01.

[55] L. Getoor and C. P. Diehl. Link Mining: A Survey. In *SIGKDD Explorations*, volume 7, pages 3–12, New York, NY, Dec. 2005. ACM Press. Online: http://www.sigkdd.org/explorations/issues/7-2-2005-12/1-Getoor.pdf, Last Checked: 2007-12-01.

[56] K. Golub and A. Ardö. Importance of HTML Structural Elements and Metadata in Automated Subject Classification. In A. R. et al, editor, *ECDL 2005*, number 3652 in LNCS, pages 368–378. Springer-Verlag,

Berlin Heidelberg, Germany, 2005. Online: `http://www.alvis.info/alvis_docs/ECDL05.pdf`, Last Checked: 2007-12-01.

[57] G. Gottlob and C. Koch. Logic-based Web Information Extraction. *SIGMOD Record*, 34:87–94, 2004. Online: `http://www.sigmod.org/record/issues/0406/DBPrincipleLeonid-gottlob.pdf`, Last Checked: 2007-12-01.

[58] R. Grishman. Information Extraction: Techniques and Challenges. In *SCIE*, pages 10–27, 1997. Online: `http://web.njit.edu/~ql23/teaching/cis634FTF/notes/Information_extraction_paper.pdf`, Last Checked: 2007-12-01.

[59] R. Grishman. Adaptive Information Extraction and Sublanguage Analysis, 2001. Online: `http://www.smi.ucd.ie/ATEM2001/proceedings/grishman-position-atem2001.pdf`, Last Checked: 2007-12-01.

[60] M. Gross. On the Failure of Generative Grammar. *Language*, 6(4):859–885, 1979.

[61] M. Gross. The Construction of Local Grammars. In E. Roche and Y. Schabès, editors, *Finite-State Language Processing: Language, Speech, and Communication*, pages 329–354. The MIT Press, Cambradge, MA, 1997.

[62] M. Gross. A Bootstrap Method for Constructing Local Grammars. In *Contemporary Mathematics: Proceedings of the Symposium*, pages 229–250, University of Belgrad, Belgrad, 1999.

[63] F. Guenthner and P. Maier. Das CISLEX Wörterbuchsystem. Technical Report CIS-Bericht-94-76, CIS (Centrum für Informations- und Sprachverarbeitung), München, Germany, 1994. Online: `http://www.cis.uni-muenchen.de/pub/cis-berichte/cislex.ps`, Last Checked: 2007-12-01.

[64] H. Guo and A. Stent. Taxonomy Based Data Extraction from Multi-item Web Pages. In *ISWC'06*, Athens, Georgia, Nov. 2006. Online: `http://orestes.ii.uam.es/workshop/14.pdf`, Last Checked: 2007-12-01.

[65] Z. S. Harris. *Mathematical Structures of Language*. Interscience, New York, NY, 1968.

[66] Z. S. Harris. *Papers in Structural and Transformational Linguistics*. D. Reidel, Dordrecht, Netheland, 1970.

[67] P. S. Hiremath, S. S. Benchalli, S. P. Algur, and R. V. Udapudi. Mining Data Regions from Web Pages. In *International Conference on Management of Data*, Hyderabad, India, 20-22 Dec. 2005.

[68] T. W. Hong and K. L. Clark. Towards a Universal Web Wrapper. In *FLAIRS Conference*. AAAI Press, 2004. Online: `http://www.cl.cam.ac.uk/~twh25/academic/papers/uwm.pdf`, Last Checked: 2007-12-01.

[69] T. W. Hong and C. K. L. Using Grammatical Inference to Automate Information Extraction from the Web. *LNCS*, 2168, 2001. Online: `http://www.cl.cam.ac.uk/~twh25/academic/papers/ecml.pdf`, Last Checked: 2007-12-01.

[70] C.-N. Hsu and M.-T. Dung. Generating Finite-State Transducers for Semi-Structured Data Extraction from the Web. *Information Systems*, 23(8):521–538, 1998. Online: `http://www.iis.sinica.edu.tw/~chunnan/DOWNLOADS/jis2.ps.gz`, Last Checked: 2007-12-01.

[71] Y. Hu, G. Xin, R. Song, and G. Hu. Title Extraction from Bodies of HTML Documents and its Application to Web Page Retrieval. In *SIGIR'05*, Salvador, Brazil, 15-19 Aug. 2005.

[72] K. Kaiser and S. Miksch. Information extraction: A survey. Technical Report Asgaard-TR-2005-6, Vienna University of Technology: Institute of Software Technology & Interative Systems, May 2005. Online: `http://ieg.ifs.tuwien.ac.at/techreports/Asgaard-TR-2005-6.pdf`, Last Checked: 2007-12-01.

[73] M. Kavalec and V. Svátek. Information Extraction and Ontology Learning Guided by Web Directory. In *ECAI Workshop on NLP and ML for Ontology engineering*, Lyon, France, 2002. Online: `http://nb.vse.cz/~svatek/rainbow/olt02su.ps`, Last Checked: 2007-12-01.

[74] S. Krötzsch and D. Rösner. Ontology based Extraction of Company Profiles. In *Workshop DBFusion*, Kalsruhe, Germany, 2002. Online: `http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-124/02kroetzsch.ps`, Last Checked: 2007-12-01.

[75] N. Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, 118(1-2):15–68, 2000. Special issue on Intelligent Internet Systems. Online: `http://kushmerick.org/nick/research/download/kushmerick-aij2000.pdf`, Last Checked: 2007-12-01.

[76] N. Kushmerick. Finite-state approaches to Web information extraction. In *Proc. Summer Convention on Information Extraction*, 2002. Online: `http://kushmerick.org/nick/research/download/kushmerick-scie2002.pdf`, Last Checked: 2007-12-01.

[77] C. Kühnlein. Eigennamenerkennung über Konzeptsensoren, 2003. Lizentiatsarbeit der Philosophischen Fakultät der Universität Zürich im Bereich Computerlinguistik. Online: `http://www.ifi.unizh.ch/cl/study/lizarbeiten/lizcarolakuenlein.pdf`, Last Checked: 2007-12-01.

[78] M. Labský and V. Svátek. On the Design and Exploration of Presentation Ontologies for Information Extraction. In *ESWC'06 Workhshop on Mastering the Gap: From Information Extraction to Semantic Representation*, Budva, Montenegro, Jun. 2006. Online: `http://nb.vse.cz/~svatek/eswc06ws.pdf`, Last Checked: 2007-12-01.

[79] M. Labský, V. Svátek, P. Praks, and O. Šváb. Information Extraction from HTML product catalogues: coupling quantative and knowledge-based approaches, 2005. Online: `http://rainbow.vse.cz/dags05.pdf`, Last Checked: 2007-12-01.

[80] M. Labský, V. Svátek, and O. Šváb. Types and Roles of Ontologies in Web Information Extraction. In *ECML/PKDD04 Workshop on Knowledge Discovery and Ontologies*, Pisa, Italy, 2004. Online: `http://nb.vse.cz/~svatek/rainbow/kdo04labsky.pdf`, Last Checked: 2007-12-01.

[81] A. H. Laender, B. A. Ribeiro-Neto, A. S. da Silva, and J. S. Teixeira. A Brief Survey of Web Data Extraction Tools. In *SIGMOD Record*, volume 31, No.2, Jun. 2002. Online: `http://www.sigmod.org/sigmod/record/issues/0206/laender-survey.pdf`, Last Checked: 2007-12-01.

[82] A. Lavelli, M.-E. Califf, F. Ciravegna, D. Freitag, C. Giuliano, N. Kushmerick, and L. Romano. IE evaluation: Criticisms and recommendations. In *Proc. Workshop Adaptive Text Extraction and Mining*, 2004. Online: `http://kushmerick.org/nick/research/download/lavelli-aaai04-atem.pdf`, Last Checked: 2007-12-01.

[83] J. Li, K. Furuse, and K. Yamaguchi. Focused Crawling by Expoiting Anchor Text Using Decision Tree. In *WWW'05*, Chiba, Japan, 10-14 May 2005. ACM 1-59593-051-5/05/0005. Online: `http://www2005.org/cdrom/docs/p1190.pdf`, Last Checked: 2007-12-01.

[84] Y. Li, X. Meng, Q. Li, and L. Wang. Hybrid Method for Automated News Content Extraction from the Web. In *Proceedings of 7th International Conference on Web Information Systems Engineering (WISE2006)*, pages 327–338, Wuhan, China, Oct. 2006.

[85] C. Lindemann and L. Littig. Coarse-grained Classification of Web Sites by Their Structural Properties. In *Proceedings of WIDM 2006*, 2006. Online: `http://rvs.informatik.uni-leipzig.de/de/publikationen/papers/WIDM06.pdf`, Last Checked: 2007-12-01.

[86] C. Lindemann and L. Littig. Classifying Web Sites. In *WWW 2007*, Alberta, Canada, 8-12 May 2007. Online: `http://www2007.org/posters/poster876.pdf`, Last Checked: 2007-12-01.

[87] B. Liu. *Web Data Mining. Exploring Hyperlinks, Contents, and Usage Data*. Springer-Verlag, Berlin Heidelberg, 2007.

[88] B. Liu, R. Grossman, and Y. Zhai. Mining Data Records in Web Pages. In *SIGKDD'03*, Washington, DC, 24-27 Aug. 2003. ACM 1-58113-737-0/03/0008. Online: `http://www.cs.uic.edu/~liub/publications/kdd2003-dataRecord.pdf`, Last Checked: 2007-12-01.

[89] L. Liu, W. Han, D. Buttler, C. Pu, and W. Tang. An XML-based Wrapper Generator for Web Information Extraction. In *SIGMOD'99*, pages 540–543, 1999.

[90] W. Liu, X. Meng, and W. Meng. Vision-based Web Data Records Extraction. In *WebDB*, 2006. Online: `http://db.ucsd.edu/webdb2006/camera-ready/paginated/04-144.pdf`, Last Checked: 2007-12-01.

[91] Z. Liu, W. K. Ng, and E. P. Lim. An Automated Algorithm for Extracting Website Skeleton. Technical report, Centre for Advanced Information Systems, Nanyang Technological University, Singapore, 2003. Online: `http://www.cais.ntu.edu.sg/~liuzh/papers/dasfaa04_sew_extended.pdf`, Last Checked: 2007-12-01.

[92] H.-D. Luckhardt. Sublanguages in Machine Translation, 1991. Online: `http://acl.ldc.upenn.edu/E/E91/E91-1054.pdf`, Last Checked: 2007-12-01.

[93] F. Mallchok. *Automatic Recognition of Organization Names in Englich Business News.* PhD thesis, LMU München, München, Germany, 2004. Online: `http://www.cis.uni-muenchen.de/\%7Eschmidt/lg/Complete.pdf`, Last Checked: 2007-12-01.

[94] B. Markines, F. Erdine, and L. Stoilova. Focused Crawlers vs Accelerated Focused Crawlers, 2004. Online: `http://www.informatics.indiana.edu/fil/Class/b659/Projects/S04-g8/finalprojectpaper.pdf`, Last Checked: 2007-12-01.

[95] D. Maurel and F. Guenthner. *Automata and Dictionary.* College Publications, Milton Keynes, UK, 2005.

[96] W. May and G. Lausen. Information Extraction from the Web. Technical Report Technical Report No. 136, Institut für Informatik, Albert Ludwig-Universität, Freiburg, Germany, Mar. 2000. Online: `http://www.informatik.uni-freiburg.de/~dbis/Publications/2K/TR136-InfoExtr.ps`, Last Checked: 2007-12-01.

[97] D. Maynard, V. Tablan, C. Ursu, H. Cunningham, and Y. Wilks. Named Entity Recognition from Diverse Text Types, 2001. Online: `http://gate.ac.uk/sale/ranlp2001/maynard-etal.pdf`, Last Checked: 2007-12-01.

[98] D. D. McDonald. Internal and External Evidence in the Identification and Semantic Categorization of Proper Names. In *Proceedings of the Workshop on Acquisition of Lexical Knowledge from Text*, pages 32–43, Columbus, OH, 1993. Online: `http://acl.ldc.upenn.edu/W/W93/W93-0104.pdf`, Last Checked: 2007-12-01.

[99] L. K. McDowell and M. Cafarella. Ontology-driven Information Extraction with OntoSyphon. In *ISWC'06*, number 4273 in LNCS, Athens, GA, 5-9 Nov. 2006. Online: `http://iswc2006.semanticweb.org/items/McDowell2006fu.pdf`, Last Checked: 2007-12-01.

[100] W. Mederle. Automatische Adreßerkennung - ein Ansatz für deutsche Adressen und seine Implementierung. Master's thesis, LMU München, München, Germany, 2004. Online: `http://www.gate.ac.uk/sale/ranlp2001/maynard-etal.pdf`, Last Checked: 2007-12-01.

[101] A. Mikheev, M. Moens, and C. Grover. Named Entity Recognition without Gazetteers. In *EACL'99*, 1999. Online: `http://www.ltg.ed.ac.uk/np/publications/ltg/papers/Mikheev1999Named.pdf`, Last Checked: 2007-12-01.

[102] I. Muslea. Extraction Patterns for Extraction Tasks: A Survey, 1999. Online: `http://www.ai.sri.com/~muslea/PS/ml4ie-aaai99.pdf`, Last Checked: 2007-12-01.

[103] A. Nakamura, H. Hasegawa, T. Saito, and M. Kudo. Flexible Wrappers for Keyword-Related Information. Technical Report TCS-TR-A-07-24, Hokkaido University: Graduate School of Information Science and Technology, 13 February 2007.

[104] B. Novak. A Survey of Focused Web Crawling Algorithms. In *SIKDD'04*, 2004. Online: `http://eprints.pascal-network.org/archive/00000738/01/BlazNovak-FocusedCrawling.pdf`, Last Checked: 2007-12-01.

[105] A. Passerini, P. Frasconi, and G. Soda. Evaluation Methods for Focused Crawling. *Lecture Notes in Computer Science*, 2175:33–39, 2001. Online: `http://www.dsi.unifi.it/~paolo/ps/aiia01-focused-crawling.pdf`, Last Checked: 2007-12-01.

[106] S. Patwardhan and E. Riloff. Learning Domain-Specific Information Extraction Patterns from the Web. In *Proceedings of the Workshop on Information Extraction Beyond The Document*, pages 66–73, Sydney, Australia, Jul. 2006. Association for Computational Linguistics. Online: `http://acl.ldc.upenn.edu/W/W06/W06-0208.pdf`, Last Checked: 2007-12-01.

[107] G. Petasis, V. Karkaletsis, and C. D. Spyropoulos. Cross-lingual Information Extraction from Web pages: the use of a genaral-purpose Text Engineering Platform. In *Proceedings of the RANLP'2003 Conference*, pages 381–388, Borovets, Bulgaria, 12-13 Sep. 2003.

[108] J. M. Pierre. On the Automated Classification of Web Sites. *Linköping Electronic Articles in Computer and Information Science*, 6, 2001. Online: http://www.sukidog.com/jpierre/etai.pdf. Last Checked: 2007-12-01.

[109] A.-M. Popescu, A. Yates, and O. Etzioni. Class Extraction from the World Wide Web. In *AAAI'04 Workshop on Adaptive Text Extraction and Mining*. AAAI Press, 2004. Online: `http://www.cs.washington.edu/homes/ayates/papers/class-extraction-atem04.pdf`, Last Checked: 2007-12-01.

[110] B. Popov, A. Kiryakov, D. Manov, A. Kirilov, D. Ognyanoff, and M. Goranov. Towards Semantic Web Information Extraction. In *Human Language Technologies Workshop at the 2nd International Semantic Web Conference (ISWS'03)*, Florida, USA, 2003. Online: `http://gate.ac.uk/conferences/iswc2003/proceedings/popov.pdf`, Last Checked: 2007-12-01.

[111] D. Reis, P. Golgher, A. Silva, and A. Laender. Automatic Web News Extraction using Tree Edit Distance. In *WWW'04*, pages 502–511, New York, NY, 2004. Online: `http://www.www2004.org/proceedings/docs/1p502.pdf`, Last Checked: 2007-12-01.

[112] D. Rocco, L. Liu, and T. Critchlow. Focused Crawling of the Deep Web Using Service Class Description. In *International Conference on Service Oriented Computing*, New York, NY, 15-18 Nov. 2004. Online: `https://e-reports-ext.llnl.gov/pdf/308855.pdf`, Last Checked: 2007-12-01.

[113] M. Rössler. Corpus-based Learning of Lexical Resources for German Named Entity Recognition. In *Proceedings of LREC 2004*, Lisboa, Portugal, 2004. Online: `http://www.uni-duisburg-essen.de/imperia/md/content/computerlinguistik/lrec2004.pdf`, Last Checked: 2007-12-01.

[114] M. Rössler. *Korpus-Adaptive Eigennamenerkennung*. PhD thesis, Universität Duisburg-Essen, Duisburg, Deutschland, Dezember 2006. Online: `http://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-16089/diss_final2007_DS.pdf`, Last Checked: 2007-12-01.

[115] M. Rössler and K. Morik. Using Unlabeled Texts for Named-Entity Recognition. In *Proceedings of the Workshop on Learning with Multiple Views, 22. ICML*, Bonn, Germany, 2005. Online: `http://www-ai.cs.uni-dortmund.de/MULTIVIEW2005/MultipleViews.pdf#page=59`, Last Checked: 2007-12-01.

[116] M. Sabou. Learning Web Service Ontologies: an Automatic Extraction Method and its Evaluation. In P. Buitelaar, P. Cimiano, and B. Magnini, editors, *Ontology Learning and Population*. IOS, 2005. Online: `http://kmi.open.ac.uk/people/marta/papers/IOS2005.pdf`, Last Checked: 2007-12-01.

[117] S. Sarawagi and W. W. Cohen. Semi-Markov Conditional Random Fields for Information Extraction. In *ICML'04*, 2004. Online: `http://www.cs.cmu.edu/~wcohen/postscript/semiCRF.pdf`, Last Checked: 2007-12-01.

[118] S. Sekine. Named Entity: History and Future. Technical Report Proteus Project Report, New York University, New York, NY, 2004. Online: `http://cs.nyu.edu/~sekine/papers/NEsurvey200402.pdf`, Last Checked: 2007-12-01.

[119] C. Siefkes. Incremental Information Extraction Using Tree-based Context Representation. In *CICLing-2005*, 2005. Online: `http://www.siefkes.net/papers/incremental-ie.pdf`, Last Checked: 2007-12-01.

[120] G. Sigletos, G. Paliouras, C. D. Spyropoulos, and M. Hatzoppoulos. Mining Web sites using wrapper induction, named entities and post-processing. *LNAI*, 3209:97–112, 2004. Online: `http://books.google.com/books?id=oMhrB9-XHZQC&pg=PA97&lpg=PA97&dq=\%22mining+web+sites+using+wrapper+induction\%22&source=web&ots=IZ1ildkP_q&sig=bMPl-KmJ7wINyUS04_2YTNbua7c#PPA102,M1`, Last Checked: 2007-12-01.

[121] J. Silva, Z. Kozareva, V. Noncheva, and G. Lopes. Extracting Named Entities. A Statistical Approach. In *TALN'04*, pages 347–351, Fez, Marroco, 19-21 Apr. 2004. ATALA. Online: `http://aune.lpl.univ-aix.fr/jep-taln04/proceed/actes/taln2004-Fez/Silva-Kozareva-Lopes.pdf`, Last Checked: 2007-12-01.

[122] K. Simon, T. Hornung, and G. Lausen. Learning Rules to Pre-process Web Data for Automatic Integration. In *RuleML'06*, Athens, Georgia, 9-10 Nov. 2006. Online: `http://www.informatik.uni-freiburg.de/\%7Ehornungt/papers/ruleml06.pdf`, Last Checked: 2007-12-01.

[123] K. Simon and G. Lausen. ViPER: Augmenting Automatic Information Extraction with Visual Perceptions. In *CIKM'05*, Bremen, Germany, 31 Oct. - 5 Nov. 2005. Online: `http://www.informatik.uni-freiburg.de/~ksimon/papers/CIKM-05-ViPER.pdf`, Last Checked: 2007-12-01.

[124] S. Sizov, J. Graupmann, and M. Theobald. From Focused Crawling to Expert Information: an Application Framework for Web Exploration and Portal Generation. In *Proceedings of the 29th VLDB Conference*, Berlin, Germany, 2003. Online: `http://www.mpi-inf.mpg.de/~mtb/pub/vldb2003demo.pdf`, Last Checked: 2007-12-01.

[125] M. Skounakis, M. Cravan, and S. Ray. Hierarchical Hidden Markov Models for Information Extraction. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, Acapulco, Mexico, 2003. Morgan Kaufmann. Online: `http://www.biostat.wisc.edu/~craven/papers/ijcai03.pdf`, Last Checked: 2007-12-01.

[126] S. Soderland. Learning to Extract Text-based Information from World Wide Web. In *Proceedings of Third International Conference on Knowledge Discovery and Data Mining (KDD'97)*, 1997.

Online: http://www-nlp.cs.umass.edu/pubs/Soderland-KDD97.ps, Last Checked: 2007-12-01.

[127] K. Stamatakis, V. Karkaletsis, G. Paliouras, J. Horlock, C. Grover, j. R. Curran, and S. Dingare. Domain-Specific Web Site Identification: The CROSSMARC Focused Web Crawler. In *Proceedings of the Second International Workshop on Web Document Analysis (WDA'03)*, Edinburgh, UK, 2003. Online: http://www.ltg.ed.ac.uk/np/publications/ltg/papers/Stamatakis2003Domain.pdf, Last Checked:2007-12-01.

[128] M. Surdeanu, S. Harabagiu, J. Williams, and P. Aarseth. Using Predicate-Argument Structures for Information Extraction. In *Proceedings of the ACL*, Sapporo, Japan, 2003. Online: http://acl.ldc.upenn.edu/P/P03/P03-1002.pdf, Last Checked: 2007-12-01.

[129] V. Svátek, P. Berka, M. Kavalec, J. Kosek, and V. Vávra. Discovering Company Descriptions on the Web by Multiway Analysis. In *New Trends in Intelligent Information Processing and Web Mining (IIPWM'03)*, Zakopane, 2003. Online: http://keg.vse.cz/_papers/2003/iipwm03.ps, Last Checked: 2007-12-01.

[130] V. Vydiswaran and S. Sarawagi. Learning to extract information from large websites using sequential models. In J. Haritsa and T. M. Vijayaraman, editors, *Advances in Data Management 2005*. CSI, 2005. Online: http://comad2005.persistent.co.in/COMAD2005Proc/pages003-014.pdf, Last Checked: 2007-12-01.

[131] B. R. Wittek and J. Altfeld. Eine kontextfreie Grammatik zur Erkennung von Firmenbezeichnungen. Technical report, CIS (Centrum für Informations- und Sprachverarbeitung), München, Germany, 1994. CIS Arbeitsbericht.

[132] B. Yildiz and S. Miksch. Motivating Ontology-Driven Information Extraction, 2007. Online: http://www.donau-uni.ac.at/imperia/md/content/department/ike/ike_publications/2007/refereedconferenceandworkshoparticles/yildiz_2007_icsd_ontolgy_management.pdf, Last Checked: 2007-12-01.

[133] M. Yoshida, K. Torisawa, and J. Tsujii. Extracting Attributes and their Values from Web Pages. In A. Antonacopoulos and J. Hu, editors,

Web Document Analysis: Challenges and Oppertunities, pages 179–200. World Scientific, London, 2003. Online: `http://www.r.dl.itc.u-tokyo.ac.jp/~mino/PAC.pdf`, Last Checked: 2007-12-01.

[134] R. Zanibbi, D. Blostein, and J. R. Cordy. A Survey of Table Recognition: Models, Observations, Transformations, and Inferences, 2003. Online: `http://www.cs.queensu.ca/~cordy/Papers/IJDAR_Tables.pdf`, Last Checked: 2007-12-01.

[135] Y. Zhai and B. Liu. Web Data Extraction Based on Partial Tree Alignment. In *WWW'05*, Chiba, Japan, 10-14 May 2005. ACM 1-59593-046-9/05/0005. `http://www2005.org/cdrom/docs/p76.pdf`, Last Checked: 2007-12-01.

[136] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu. Fully Automatic Wrapper Generation For Search Engines. In *WWW'05*, Chiba, Japan, 10-14 May 2005.

[137] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang, and W.-Y. Ma. Simultaneous Record Detection and Attribute Labeling in Web Data Extraction. In *KDD'06*, Philadelphia, Pennsylvania, 20-23 Aug. 2006.

[138] Z. Zhuang, R. Wagle, and C. L. Giles. What's There and What's Not? Focused Crawling for Missing Documents in Digital Libriries. In *JCD'05*, Denver, Colorado, 7-11 Jun. 2005. ACM 1-58113-876-8/05/0006. Online: `http://clgiles.ist.psu.edu/papers/JCDL-2005-Focused-Crawling.pdf`, Last Checked: 2007-12-01.

[139] P. Zigoris, D. Eads, and Y. Zhang. Unsupervised Learning of Tree Alignment Models for Information Extraction. In *ICDMW'06: Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*, pages 45–49, Washington, DC, 2006. Online: `https://www.soe.ucsc.edu/~eads/papers/zigoris2006.pdf`, Last Checked: 2007-12-01.

Lebenslauf

Persönliche Daten:

Name:	LEE, Yeong Su
Reisepass Nr.:	FK0013223
Geburtsdatum:	03.02.1964
Geburtsort:	Puan/Chonbuk, Republik Korea
Staatsangehörigkeit:	Koreanisch
Wohnort:	Guido-Schneble-Str. 3, 80689 München
Telefon:	+49 (0)89 3241909

Sprachkenntnis:

Koreanisch(Muttersprache)
Deutsch(verhandlungssicher)
Englisch(fortgeschrittene Kenntnisse)
Chinesisch, Japanisch(Grundkenntnisse)

Schulbildung:

März 1970 – Feb.1976	Shinchuk Grundschule in Puan
März 1976 – Feb.1982	Baeksan Mittel-(Middle School) und Oberschule(High School) in Puan

Studium:

März 1982 – Aug.1988	Studium an der Chonbuk Nationale Universität (Abschluss B.A. im Fach Germanistik) Koreanischer Wehrdienst (Jan.1984 – April 1986)
März 1989 – Feb.1991	Magisterstudium an der Seoul Nationale Universität (Abschluss M.A. im Fach Germanistik)
Okt.1991 – März 1992	Sprachkurs bei der LMU
SS 1992	Immatrikulation an der LMU München
WS 1994/1995	Umschreibung zum Promotionsstudium

WS 2001/2002 – WS 2004/2005 Aufbaustudium
am Centrum für Informations- und Sprachverarbeitung (CIS), LMU München
Abschlussthema: Formenbildung im Koreanischen: Prädikative Kategorien

Praktische Tätigkeiten:

März 1990 – Feb.1991	- tätig als Hilfskraft am Institut für Sprachwissenschaft an der Seoul Nationale Universität - tätig als wissenschaftliche Hilfskraft beim Projekt für die maschinelle Übersetzung zwischen englischen und koreanischen mit und durch Unterstützung von IBM an der SNU
März 1991 – Aug.1991	tätig als Lehrbeauftragter an der Yangchong Oberschule(Highschool) in Seoul
März 1998 – Feb.1999	Software-Übersetzung vom Deutschen ins Koreanische bei RVS-COM
April 2004 – März 2005	tätig als Hilfswissenschaftler am CIS - Koreanische Lexikonerstellung - Bilinguale Lexikonaufbau Patentrecht-relevanter Begriffe (Englisch – Koreanisch) - TVG-Verlag-Projekt - Meta-Suchmaschine - Branchen klassifizieren und identifizieren
Seit April 2005	Wissenschaftlicher Angestellter am CIS

München, den _____ Unterschrift _____