

Universität Paderborn

AG Kleine Büning

Betreuer: Dr. Benno Stein, Sven Meyer zu Eissen
e-mail: {stein, smze}@uni-paderborn.de

Thema: Genreklassifikation von Webseiten

12. Februar 2004

Name: Roman Deimann
Adresse: Friedrich-von-Spee Str. 10
33098 Paderborn
E-mail: rdeimann@uni-paderborn.de
Telefon: 05251/877978
Matrikelnummer: 3691270
Studiengang: Diplom-Informatik

Inhaltsverzeichnis

1	Einleitung	5
1.1	Suche im Internet	5
1.2	Möglichkeiten zur Filterung relevanter Dokumente	5
1.2.1	(a-search)	5
1.2.2	Genre-Suche	6
2	Textklassifikation	7
2.1	C4.5	7
2.2	Neuronale Netzwerke	9
3	Genres	12
4	Indikatoren und ihre Berechnung	17
4.1	Linguistische Indikatoren	17
4.1.1	Verwendete Wortklassen	17
4.1.2	Durchschnittliche Satzlänge	21
4.1.3	Durchschnittliche Wortlänge	22
4.1.4	Anzahl von Sätzen	22
4.1.5	Komplexität von Sätzen	22
4.1.6	Anzahl von Abschnitten	22
4.1.7	Verwendete Zeitform	22
4.1.8	Satzzeichen	22
4.1.9	Nominalphrase (Noun Phrases)	22
4.2	Lexikalische Indikatoren	26
4.2.1	Wortfrequenzen	26
4.2.2	Wortfrequenzen bestimmter abgeschlossener Wortmengen	26
4.2.3	Frequenzen von Stop-Words	26
4.2.4	Anzahl von "it"s	26
4.2.5	Frequenzen von Schlüsselwörtern	27
4.2.6	Anzahl von nicht alpha-numerischen Zeichen	27
4.2.7	Anzahl von Zahlen	27

4.3	Gestaltungstechnische Indikatoren	27
4.3.1	Zeilenabstand	27
4.3.2	Anzahl von Aufzählungen	27
4.3.3	Anzahl von Tabellen	27
4.3.4	Anzahl von Grafiken	27
4.3.5	Größe und Format von Grafiken	28
4.3.6	Verhältnis Text zu Bild	28
4.3.7	Schriftgröße, Anzahl der verwendeten Schriften und Farben	28
4.4	HTML spezifische Indikatoren	28
4.4.1	Anzahl von Hyperlinks	28
4.4.2	HTML Metainformationen	29
4.4.3	Anzahl von Tabellen	29
4.4.4	Technische HTML Tags	30
5	Eigenes Vorgehen	32
5.1	Korpus	32
5.1.1	Korpora anderer Autoren	32
5.1.2	Mein Korpus	33
5.2	Programm	35
5.2.1	Die Bookmark-Verwaltung im Browser	36
5.2.2	Nescape Bookmark-Datei	37
5.2.3	Korpusdateiformat	37
5.2.4	Update Corpus	38
5.2.5	Generate Bookmark-File	38
5.2.6	Create Feature-Files	38
5.2.7	Classify	39
5.3	Experimente	40
5.4	Zukünftige Verbesserungsmöglichkeiten	40
6	Anhang	45
6.1	Bookmark-Datei	45
6.2	Korpusdatei	47
6.3	Feature-Datei einer Korpusdatei	48

Abbildungsverzeichnis

1	Ein einfacher Entscheidungsbaum	8
2	Textdarstellung des einfachen Entscheidungsbaumes aus Abbildung 1.	8
3	Ein kleines neuronales Netzwerk	10
4	Beispiele für Genre-Typen	14
5	Beispiele für Genre-Typen	16
6	Beispiel eines stark vereinfachten Entscheidungsbaumes aus [22, 23]	20
7	Beispiel eines Suffixbaumes der Länge 3 des Taggers aus [22, 23]	21
8	Allgemeine Struktur von Nominalphrasen [13]	24
9	Einfaches Beispiel einer Phrasenstruktur-Grammatik in Englisch [13]	25
10	Verwaltung der Bookmarks im Internetbrowser Mozilla	36
11	Sequenzdiagramm der Funktion Update Corpus	42
12	Sequenzdiagramm der Funktion Create Feature-Files	43

Tabellenverzeichnis

1	Kategorien des Korpus von Brown und benutzte "Genres" von [14]	32
2	Der Korpus des Wall Street Journal (WSJ) mit den Genres von [25]	33
3	Genres und Anzahl der HTML-Steiten im erstellten Korpus	34
4	Konfusionsmatrix der Klassifikation aller 8 Genres mit dem SNNS.	41
5	Konfusionsmatrix der Klassifikation aller 8 Genres mit C45	41
6	Konfusionsmatrix der Klassifikation aller 8 Genres mit NaiveBayes	44

1 Einleitung

Das umfangreiche Angebot im World Wide Web erschwert die gezielte Suche nach Informationen. Zur Zeit wird angenommen, dass das Internet mehr als neun Milliarden Webseiten mit einer Gesamtgröße von 700 TByte umfasst.

Um diese enorme Masse an Informationen jedoch bestmöglich nutzen zu können, erleichtern Suchmaschinen die Arbeit.

1.1 Suche im Internet

Bei Internet-Suchmaschinen gibt ein Benutzer ein oder mehrere Suchbegriffe ein und diese werden im Index nachgeschlagen. Der Index ist, vereinfacht dargestellt, eine Datenbank. In dieser wird zu jedem Suchbegriff ein Datensatz angelegt, der auf die Seiten, in denen dieser Suchbegriff vorkommt, verweist.

Die Suchmaschinentypen unterscheiden sich hauptsächlich in der Art und Weise, wie dieser Index erstellt wird. Man unterscheidet zwischen zwei grundsätzlichen Suchmaschinentypen, den Webkatalogen und den Webcrawlern, auch Robots oder Spider genannt.

Der Index eines Webkatalogs wird manuell erstellt und überprüft. Im Gegensatz zu Webkatalogen wird der Index bei Webcrawlern vollautomatisch erstellt. Der Crawler durchsucht das Internet, beginnend bei den angemeldeten Seiten und folgt allen auf diesen Seiten vorgefundenen Hyperlinks. Die auf diese Art erreichten Seiten werden im Index eingetragen.

1.2 Möglichkeiten zur Filterung relevanter Dokumente

Ein Nachteil des automatisch erstellten Index einer Suchmaschine ist dessen Größe. Diese macht sich bei einer Suchanfrage bemerkbar, indem die Anfrage oft mehr als tausend Ergebnisse liefert, die ein Mensch nicht alle auf Relevanz überprüfen kann. Bisher werden alle Ergebnisse einer Suchanfrage einfach auf mehreren Seiten untereinander aufgelistet. Die Dokumente sind zwar absteigend nach ihrer Relevanz zur Suchanfrage sortiert, aber die Relevanz eines Dokumentes für den Suchenden ist abhängig von seinem Problem. Wenn z. B. ein Lehrer und ein Physikstudent etwas zum Thema Schallmauer suchen, dann ist der Lehrer an einer einfachen und anschaulichen Beschreibung des Phänomens für seine Schüler interessiert. Der Student hingegen benötigt technische Informationen zum gleichen Thema. Die Relevanz eines Dokumentes für den Suchenden ist nicht nur abhängig von seiner Suchanfrage, sondern insbesondere von seiner Problemstellung.

Es gibt verschiedene Möglichkeiten, die dem Benutzer helfen können, die für ihn relevanten Dokumente aus der großen Menge der Ergebnisse heraus zu filtern.

1.2.1 (a-search)

„a-search“ ist ein inhaltlich orientierter Ansatz.

1.2.2 Genre-Suche

Ein weiter Ansatz betrachtet nicht den Inhalt einer Webseite, sondern dessen "Genre".

Der Begriff "Genre" ist französisch und abgeleitet von dem lateinischen Wort "genus", was soviel bedeutet wie Art oder Gattung. Der Begriff Genre wird im Deutschen besonders in der Kunst verwendet. Typische Genres in diesem Bereich sind Fiktion und Krimi. Aber auch in der Musik spricht man von Genre, hier sind u. a. Rock und Pop zu nennen.

Aus der Erläuterung des Begriffs Genre geht hervor, dass bei diesem Ansatz nicht der Fokus auf den Inhalt von Dokumenten gelegt wird, sondern auf deren Art und Aufbau.

Hat der Benutzer z. B. Probleme mit einem bestimmten Produkt, sucht er auf die herkömmliche Art und Weise nach dem Produkt und gibt zusätzlich das Genre "FAQ / Q&A" mit an. Als Ergebnis bekommt er nur "FAQ / Q&A"-Seiten, die sich auf das gesuchte Produkt beziehen. Die Wahrscheinlichkeit, auf diesen Seiten die Lösung des Problems zu bekommen, ist im Verhältnis zu den Seiten ohne Angabe des Genre stark gestiegen.

Zur Umsetzung dieses Ansatzes ist es wichtig, Internetseiten zu Genres zuzuordnen zu können. Deshalb beschäftige ich mich in dieser Ausarbeitung mit der "Genreklassifikation von Webseiten".

Dabei ist nicht das Ziel, eine lokale Dokument Library zu kategorisieren, sondern Internetseiten, die als Ergebnis auf eine Anfrage an eine Suchmaschine geliefert wurden, zuzuordnen. Um diese Aufgabe schnell und effizient zu lösen, ist es notwendig, das Datenvolumen bei der Anfrage der Webseiten gering zu halten. Hiermit fällt die Möglichkeit des Ladens von Bildern und anderen Multimediaobjekten weg. Die einzige Information, die man in einer angemessenen Zeit bekommen kann, ist der reine HTML-Code. Zur Erstellung dieser Ausarbeitung beschränke ich mich darauf, eine lokale Dokument Library unter Beachtung dieser Vorgaben zu kategorisieren. Die Erstellung dieser Library wird ab Kapitel 5.1 beschrieben.

Da Englisch die Hauptsprache im Internet ist, werde ich mich auf englische Texte beschränken.

Im folgenden Kapitel 2 erkläre ich den Begriff Textklassifikation. Im Kapitel 3 gebe ich eine Auswahl verschiedener Genres, die von Autoren im Zusammenhang mit Texten und Internetseiten genannt werden, oder die ich für erwähnenswert halte. Anschliessend beschreibe ich im Kapitel 4 eine Auswahl möglicher Indikatoren (Features) für einen Klassifizierer und untersuche sie in bezug auf meine Anforderungen. Im letzten Kapitel schildere ich mein eigenes Vorgehen.

2 Textklassifikation

Das Ziel von Textklassifikation ist die Einordnung von Texten oder Dokumenten in eine fest vorgegebene Menge von Klassen oder Kategorien. Jedes Dokument wird hierbei genau einer Kategorie zugeordnet. Die Zuordnung der Dokumente geschieht mit Hilfe maschineller Lernverfahren. Diese Verfahren werden Klassifizierer genannt.

Der Vorteil der Textklassifikation besteht darin, dass Dokumente schneller und gezielter aufgefunden werden können, weil Dokumente mit thematisch gleichem Inhalt oder gleichem Genre sich in derselben Kategorie befinden.

Methoden zur Klassifikation

Die meisten Programme mit künstlicher Intelligenz basieren auf einem Modell, das von einem menschlichen Experten an Hand seines Wissens erzeugt wurde. Dieser Ansatz wurde in den frühen 80ern zuerst vorgestellt.

Heutzutage werden Verfahren benutzt, die auf maschinellem Lernen basieren. Sie bestehen im allgemeinen aus drei Teilen:

- Trainingsmenge
- Testmenge
- Algorithmus

Die Trainingsmenge und die Testmenge bilden den Korpus und beinhalten Instanzen (Dokumente, Texte, ...), die bereits von Hand zu Kategorien zugeordnet wurden. Im 1. Schritt untersucht der Algorithmus die Trainingsmenge mittels fest vorgegebener Eigenschaften (Features). Hierbei erstellt er Regeln, mit denen er später entscheiden kann, zu welcher Kategorie eine Instanz gehört. Im 2. Schritt wird das erlernte "Wissen" mit Hilfe der Testmenge überprüft. Hierbei entscheidet der Algorithmus mit seinen Regeln, zu welcher Kategorie eine Instanz gehört und vergleicht dann seine Entscheidung mit der Vorgabe. Im 3. Schritt kommt der Algorithmus auf bisher unbekanntem Daten zum Einsatz.

Im weiteren möchte ich "C4.5" und Neuronale Netzwerke vorstellen.

2.1 C4.5

C4.5 basiert auf dem ursprünglich 1978 vorgestellten "Hunt's Concept Learning Systems by way of ID3". Laut [20] ist C4.5 aber nicht nur ein Klassifizierer, der auf einem Entscheidungsbaum beruht, sondern C4.5 ist der Name für ein komplettes System, wovon der Klassifizierer nur ein Teil ist.

Für den Entscheidungsbaum gilt:

- ein Blatt identifiziert eine Klasse (Kategorie)

- ein Entscheidungsknoten definiert einen Test eines Features, der für jedes mögliche Ergebnis einen Nachfolger besitzt.

Die Klassifizierung einer Instanz beginnt an der Wurzel des Baumes. Von dort aus wird an jedem Entscheidungsknoten ein Test wie beschrieben durchgeführt, bis ein Blatt erreicht ist. Der auf diese Weise gefundene Pfad durch den Baum ist endgültig. Das erreichte Blatt identifiziert die zur Instanz gehörige Klasse.

Abbildung 1 und 2 zeigen einen einfachen Entscheidungsbaum.

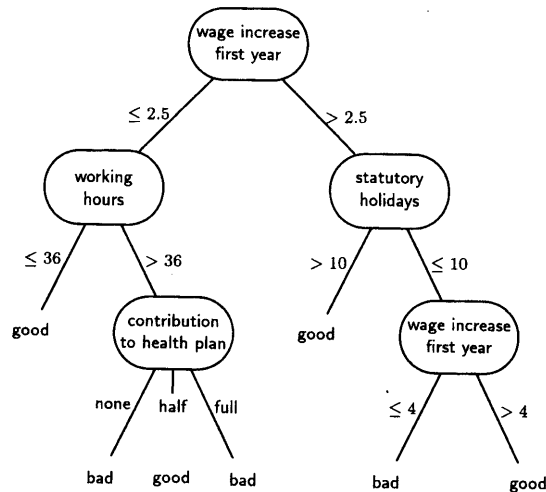


Abbildung 1: Ein einfacher Entscheidungsbaum.

```

if wage increase first year <= 2.5 then
  if working hours <=36 then class good
  else if working hours > 36 then
    if contribution to health plan is none then class bad
    else if contribution to health plan is half then class good
    else if contribution to health plan is full then class bad
else if wage increase first year >2.5 then
  if statutory holidays >10 then class good
  else if statutory holidays <=10 then
    if wage increase first year <=4 then class bad
    else if wage increase first year >4 then class good

```

Abbildung 2: Textdarstellung des einfachen Entscheidungsbaumes aus Abbildung 1.

Zum C4.5 System gehört ebenfalls ein Programm, das den von der Trainingsmenge erstellten Baum vereinfacht, denn dieser Baum ist oft sehr komplex und

überladen. Er ist 100%ig auf die Trainingsmenge abgestimmt. Mas sagt auch, er ist überangepasst an die Trainingsmenge. Der englische Begriff hierfür lautet "overfitting". Die Trainingsmenge sollte eine möglichst gute Zusammenstellung der real zu erwartenden Instanzen sein, was aber nicht immer gelingt. Aus diesem Grund muß der erzeugte Entscheidungsbaum verallgemeinert werden. Die Verallgemeinerung und Vereinfachung von Klassifikatoren wird allgemein als "pruning" bezeichnet. Ein weiterer Vorteil des vereinfachten Baumes besteht darin, dass die Klassifikation schneller geht, weil die Pfade von der Wurzel zu den Blättern gekürzt werden.

2.2 Neuronale Netzwerke

Die hohe Performance des menschlichen Gehirns bei der Erkennung von Mustern in Bildern war die nach [29] die Anregung für Forscher, das komplexe Gehirn nachzubilden. Dabei hat man versucht, die Struktur von untereinander verbundenen Neuronen auf ein technisches System, bestehend aus Units (Einheiten) und Links (Verbindungen), abzubilden. Dieses Model wird "Neuronales Netz" genannt.

Die aktuellen neuronalen Netzwerke versuchen nicht einfach die Biologie nachzubilden, sondern sind vielmehr als paralleler Algorithmus zu verstehen. In diesen Modellen wird das Wissen in der Struktur (Topologie) und den Kantengewichten gespeichert.

Wie bereits beschrieben, besteht ein Netzwerk aus Units (Knoten) und Links (Kanten). Die Links sind dabei gerichtet und mit Gewichten versehen. In Analogie zur Aktivierung der Neuronen im menschlichen Gehirn wird dabei die Eingabe eines Units aus den gewichteten Ausgaben der Vorgängerunits berechnet, die einen Link zum aktuellen Unit besitzen.

Die Berechnung der eingehenden und ausgehenden Werte kann durch eine Aktivierungsfunktion und eine Ausgabefunktion geschehen. Die Aktivierungsfunktion berechnet dabei zunächst den Aktivierungswert, der dann von der Ausgabefunktion benutzt wird, um den Ausgabewert zu berechnen. Sowohl die Aktivierungsfunktion als auch die Ausgabefunktion können für jedes Unit unterschiedlich sein.

Abbildung 3 zeigt ein kleines neuronales Netzwerk.

Units

Abhängig von der Funktion im Netzwerk werden drei Typen von Units unterschieden: Die Eingabeunits (input-units) nehmen die Eingaben im Netzwerk entgegen. Die Ausgabeunits (output-units) beinhalten die Ausgaben des Netz-

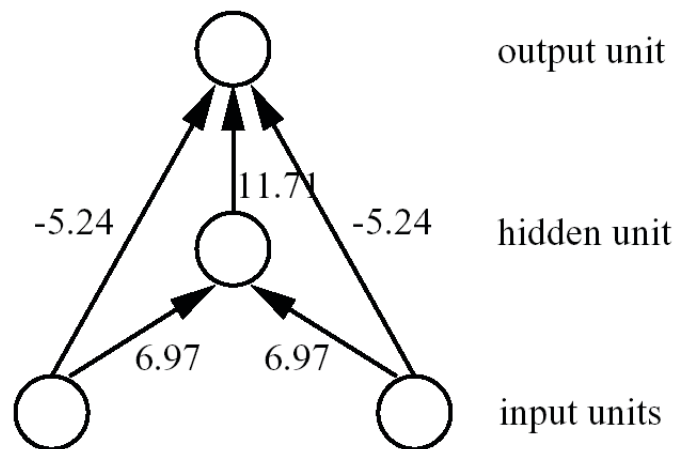


Abbildung 3: Ein kleines neuronales Netzwerk.

werkes, und die restlichen Units werden versteckte Units (hidden-units) genannt. Die versteckten Units sind ausserhalb des Netzwerkes nicht sichtbar. Abbildung 3 beinhaltet Units der 3 verschiedenen Typen.

Units besitzen neben den genannten Eigenschaften noch weitere. Es folgt nun eine Auswahl der wichtigsten Eigenschaften von Units. Die komplette Liste der Eigenschaften ist in [29] aufgeführt.

- **name**
Der Name eines Units
- **io-type**
Der "io-type" definiert die Funktion, die ein Unit im Netzwerk übernimmt. Mögliche Werte sind wie oben beschrieben input, output, hidden.
- **Aktivierungswert**
Der von der Aktivierungsfunktion berechnete Wert.
- **initialer Aktivierungswert**
Diese Variable beinhaltet den initialen Aktivierungswert.
- **bias**
Schwellenwert für die Aktivierungsfunktion
- **Aktivierungsfunktion**
Die Aktivierungsfunktion berechnet den neuen Aktivierungswert für das Unit. In die Berechnung des neuen Wertes fliesst der alte Wert und die Werte der Vorgänger wie oben beschrieben ein. Die folgenden Formeln sind ein Beispiel für eine generelle Aktivierungsfunktion.

$$net_j(t) = \sum_i w_{ij} a_i(t) \quad (1)$$

$$a_j(t+1) = f_{act}(net_j(t), a_j(t), \theta_j) \quad (2)$$

$a_j(t)$ Aktivierungswert des Unit j im Schritt t
 $net_j(t)$ Aktivierungswert des Unit j im Schritt t
 θ_j Schwellenwert (bias) des Unit j

- **Ausgabewert**

Der von der Ausgabefunktion berechnete Wert.

- **Ausgabefunktion**

Die Ausgabefunktion berechnet den Ausgabewert des Unit aus dem aktuellen Aktivierungswert des Unit.

$$o_j(t) = f_{out}(a_j(t)) \quad (3)$$

$a_j(t)$ Aktivierungswert des Unit j im Schritt t
 $o_j(t)$ Ausgabewertwert des Unit j im Schritt t

Entwicklung und anlernen von neuronalen Netzwerken

Die Entwicklung (Konstruktion) eines neuronalen Netzwerkes wird von einem Experten vorgenommen. Dieser legt die Struktur des Netzwerkes fest. Erlern werden in neuronalen Netzwerken die Kantengewichte, die für die Klassifikation verantwortlich sind. Das Erlernen dieser geschieht wie bei anderen Verfahren mit Hilfe von Trainings- und Testmengen. Die Instanzen werden als Input in das Netzwerk gegeben. Die Ausgabe des Netzwerkes wird anschliessend mit der Vorgabe verglichen und daraufhin werden die Gewichtungen der Links angepasst. Dieses wird nacheinander für jede Instanz vorgenommen.

3 Genres

Die Literatur nennt die verschiedensten Genres im Zusammenhang mit Dokumenten. Einige Autoren definieren Genres, die relativ vergleichbar sind zu den bekannten inhaltlichen Kategorien wie Sport, Politik, usw. Andere Autoren unterscheiden nur, ob der Inhalt eines Dokumentes positiv oder negativ ist. Für mich und verschiedene Autoren unterscheiden sich Genres von Dokumenten nicht durch den Inhalt des Textes, sondern durch dessen Aufbau, die Gestaltung und den Schreibstil. Die herkömmliche Textklassifizierung, die nur den Inhalt betrachtet, fasst Interviews, kurze Nachrichten und Berichte zu einem Thema in einer Kategorie zusammen. Die Genreklassifikation fasst Interviews, kurze Nachrichten und Berichte in unterschiedlichen Kategorien zusammen. Dabei kommen z. B. Interviews zu verschiedenen Themen in eine Kategorie. Im weiteren möchte ich verschiedene Genres nennen und, wenn möglich, mit Beispielen verdeutlichen.

Beginnen möchte ich mit Genres, die den Schreibstil betrachten.

- **Kurze Nachrichten**, z. B. Artikel aus einer Tageszeitung.
Beispiel aus der Welt vom 19.07.2003 www.welt.de:
TSCHETSCHENIEN Ärzte ohne Grenzen beklagen humanitäre Misere Die Hilfsorganisation Ärzte ohne Grenzen ist besorgt über die Lage der Menschen im Kaukasus. Tschetschenien sei weltweit das einzige Konfliktgebiet, in dem humanitäre Hilfswerke fast keinen Zugang zur Zivilbevölkerung hätten, klagte die Geschäftsführerin der Deutschen Sektion, Ulrike von Pilar, bei der Vorstellung des Jahresberichtes in Berlin. dpa
- **Anzeigen**, z. B. Anzeigen aus einer Tageszeitung, in denen Häuser angeboten oder gesucht werden.
Beispiel aus <http://www.reviermarkt.de/immo/index.htm> am 20.07.2003:
Bochum-Werne Wohn- und Geschäftshaus in zentraler Lage, 2 Ladenlokale mit Hinterräumen, 2 große Wohnungen, 3 Garagen mit 4 Stellplätzen, gr. Terrasse, Mieteinnahmen pro Jahr 26.000,-EUR, für 295.000,-EUR zu verkaufen, Tel.: 0234-261464.
- **Berichte**, z. B. längere Berichte über ein Thema aus einem wöchentlichen Magazin (Fokus, Spiegel, ...).
- **Geschichten**, z. B. Romane und Erlebnisberichte
- **Interviews**, z. B. Interviews zwischen einem Reporter und einer Person.
Beispiel aus <http://www.spiegel.de/unispiegel/studium/0,1518,202725,00.html>:
"Interview mit Bundesbildungsministerin Bulmahn

Im Notfall auch ohne die Länder

Edelgard Bulmahn ist entschlossen, die Bildungspolitik in Deutschland zu reformieren. Dabei hofft sie auf die Kooperation der Bundesländer. Andernfalls könne sie sich auch einen Zuständigkeitswechsel zum Bund vorstellen, erläutert die Bundesbildungsministerin im Interview mit SPIEGEL ONLINE.

DPA

Bundesbildungsministerin Bulmahn: Vorbilder Finnland und Kanada

SPIEGEL ONLINE: Frau Bulmahn, seit wann sind Ihnen die Ergebnisse der Pisa-E-Studie bekannt?

Edelgard Bulmahn: Die Pisa-Studie ist mir im Detail seit Ende der letzten Woche bekannt. Ich habe sie am Samstag und am Sonntag gelesen und kenne die Studie vom Anfang bis zum Ende.

SPIEGEL ONLINE: Was waren Ihre ersten Gedanken, als Sie die Studie gelesen haben?

Bulmahn: Viele Mängel und Schwächen, die unser Bildungssystem hat, waren mir schon 1999 bekannt, als ich das Forum Bildung" geschaffen habe. Aber der Pisa-Ländervergleich hat mich nicht völlig überrascht. Er zeigt klar, dass wir sehr große Unterschiede zwischen den Bundesländern haben.

..."

- **Werbung**

Beispiel ein kuzer Werbetext von www.audi.de:

Der Audi A4 spricht seine eigene Sprache. Er zeigt, wie aus innovativer Technologie und avanciertem Design eine äußerst attraktive neue Fahrzeug-Generation entstand. Sie folgt nicht den Trends, die der Zeitgeist vorschreibt. Der Audi A4 definiert sich selbst völlig neu: Er weist den Weg in die Zukunft. Erleben Sie den A4 in seinen vielen Varianten. Aktuell mit dem Ambition Paket für Limousine und Avant.

- ...

Wie bereits erwähnt, sind nicht nur Genres denkbar, die den Schreibstil betrachten, sondern auch Genres die den Aufbau oder die Gestaltung berücksichtigen:

- **kurze Texte** sind z. B. Paper zu einem Thema aber auch Artikel in einer Tageszeitung.
- **lange Texte** sind z. B. Diplom- und Doktorarbeiten, aber auch Romane .

- **mehrspaltige Texte** - Am verbreitetsten sind mehrspaltige Texte in Tageszeitungen. (Beispiel in Abbildung 1)
- **Texte mit Bildern** (Beispiel in Abbildung 1)
- **Texte mit Tabellen** (Beispiel in Abbildung 1)
- **Texte mit Aufzählungen**
- ...

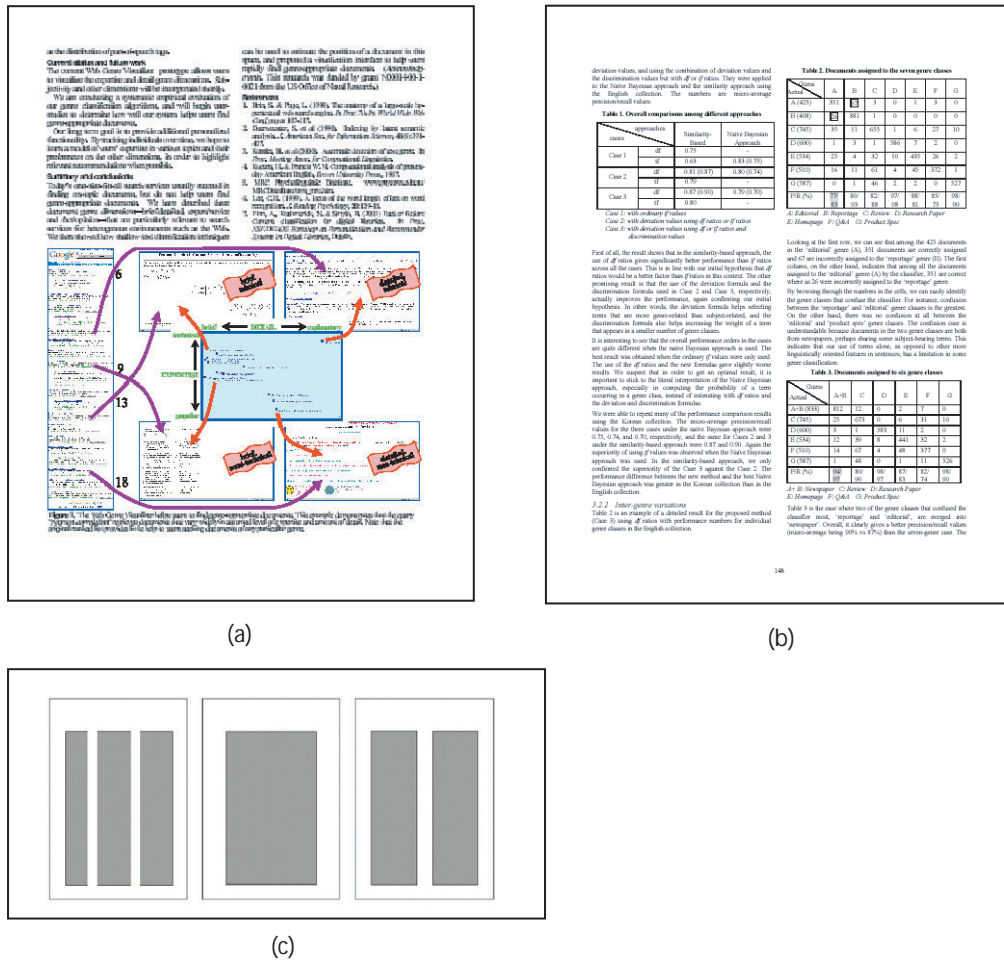


Abbildung 4: Beispiele für Genre-Typen (a) Text mit Grafiken, (b) Text mit Tabellen, (c) mehrspaltige Texte.

In [8] ist das Genre eines Textes abhängig davon, wie detailliert der Text ist. Speziell für Internetseiten sind weitere verschiedene Genres denkbar:

- **kommerzielle Seiten** sind Internetseiten, z. B. von Firmen, die dort sich und ihre Produkte präsentieren. Kommerzielle Seiten sind aber Seiten von Dienstleistern. Das bedeutet, unter kommerziellen Seiten versteht man alle Seiten, mit denen direkt oder indirekt Geld verdient werden soll.
- **eCommerce** Seiten sind ein Teil der kommerziellen Seiten. Auf diesen Seiten werden Produkte oder Dienstleistungen direkt über das Internet angeboten und verkauft. Beispiele hierfür sind Shops im Internet.
- **private (nicht kommerzielle) Seiten** sind .z B. Seiten von Personen, die sich oder ihre Hobbys präsentieren wollen. Die Internetseiten von Vereinen fallen ebenfalls in diesen Bereich. Ein Problem dieser Kategorisierung bereiten Seiten von Städten und Kommunen.
- **Newsgroups** (Beispiel in Abbildung 2)
- **Werbung** (Beispiel in Abbildung 2)
- **FAQ / Q&A** (Beispiel in Abbildung 2)
- **Produkt Reviews** (Beispiel in Abbildung 2)
- **Weblogs**
- **technische Seiten**
- **Foren**
- ...

4 Indikatoren und ihre Berechnung

Das Ziel dieser Arbeit ist u. a. die Zuordnung von Webseiten zu Genres. Hierfür werden Indikatoren benötigt, mit denen man z.B. Textklassifizierer bauen kann. Ich möchte nun verschiedene Indikatoren nennen und sie in Gruppen aufteilen.

Mögliche Gruppen zur Unterteilung der Indikatoren sind:

- linguistische Indikatoren (z.B. Satzlänge)
- lexikalische Indikatoren
- gestaltungstechnische Indikatoren (Tabellen)
- HTML spezifische Indikatoren (z.B. Hyperlinks)

Neben den "einfachen" Indikatoren gibt es verschiedene Kennziffern, die meistens aus mehreren einfachen Indikatoren berechnet werden: Flesch Metric [7, 21], Kincaid [21], Coleman-Liau [21], Wheeler-Smith Index[21]

Hinweis: Die Literaturverweise geben an, in welcher Literatur u. a. der jeweilige Indikator erwähnt wird.

4.1 Linguistische Indikatoren

Die Linguistik¹ ist die Wissenschaft von der menschlichen Sprache. Sie beschäftigt sich mit den Eigenschaften der Sprachen, die von Menschen gesprochen werden, wie z.B. Deutsch, Englisch, etc.. Diese Sprachen bezeichnet man in der Linguistik als natürliche Sprachen, im Gegensatz zu den künstlichen Sprachen, wie etwa Programmiersprachen, die in der Informatik und Mathematik untersucht werden. Das Ziel der Linguistik ist, die psychologischen und biologischen Grundlagen der menschlichen Sprachfähigkeit zu ermitteln; d.h. was befähigt den Menschen, natürliche Sprachen zu erlernen, zu verstehen, zu produzieren oder auch zu vergessen?

4.1.1 Verwendete Wortklassen

Wortklassen sind: Nomen, Verben, Adjektive, Adverben, Die Wortklasse eines Wortes ist nicht eindeutig, das Wort "store" ist im Englischen sowohl ein Nomen als auch ein Verb. Zu welcher Klasse ein Wort gehört, ist abhängig von seinem Kontext.

Die am meisten benutzte Möglichkeit, die Wörtern eines Textes Wortklassen zuzuordnen, sind "**Part-of-Speech Tagger**" (POS).

In der Ausarbeitung von Cutting et al. [6] wird ein Part-of-Speech Tagger wie folgt beschrieben: Ein Part-of-Speech Tagger ist ein System, das den Kontext benutzt, um Wortklassen zu Wörtern zuzuordnen.

¹Vergleiche dazu auch www.phil.uni-passau.de/linguistik/studium/

Der im weiteren Verlauf des Kapitels vorgestellte Part-of-Speech Tagger erzeugt mit dem Satz: "Further, it's fairly likely that the "new" technique that is being proposed has already been evaluated, and found to be inadequate/inappropriate for the kernel." die folgende Ausgabe:

```

reading parameters ...
tagging ...
Further RBR      Further
'          '      '
it        PP      it
's       VBZ      be
fairly   RB      fairly
likely   JJ      likely
that     IN      that
the      DT      the
"        ``      "
new      JJ      new
"        ''      "
technique NN      technique
that     WDT     that
is       VBZ     be
being    NN      being
proposed VVN      propose
has      VHZ     have
already  RB      already
been     VBN     be
evaluated VVN     evaluate
'        '      '
and      CC      and
found    VVD     find
to       TO      to
be       VB      be
inadequate/inappropriate JJ      <unknown>
for      IN      for
the      DT      the
kernel   NN      kernel
.        SENT    .
done .

```

Es existieren Part-of-Speech Tagger mit verschiedenen Ansätzen. Die zuerst entwickelten Tagger hatten regelbasierte Ansätze. [6] entwickelte einen Tagger mit einem statistischem Ansatz. Die Grundlage hierfür bildet das "hidden Markov model"[1]. Dieser Tagger muß im Gegensatz zu den regelbasierten Taggern mit Hilfe eines entsprechenden Korpus trainiert werden. Ich möchte nun auf den Tagger von [22, 23] eingehen. Dafür benötige ich eine Definition.

Folgende Definition stammt aus [1].

Definition Statistische Modelle, bei denen die Wahrscheinlichkeit für das k -te Wort nur von den $n-1$ vorherigen Wörtern abhängt, bezeichnet man als n -gramm-Modelle:

$$\text{Für } N \geq k \geq n \text{ gilt: } P(w_k | w_{k-n+1}, \dots, w_N) = P(w_k | w_{k-n+1}, \dots, w_{k-1}) \quad (4)$$

Im folgenden werden n -gramm-Modelle benutzt, bei denen $n = 3$ ist. Sie werden Trigramm-Modelle genannt. Es handelt sich also um statistische Modelle, in denen die Wahrscheinlichkeit eines Wortes nur von den zwei vorherigen Wörtern abhängt:

$$P(w_k | w_1, \dots, w_{n-1}) = P(w_k | w_{n-2}, w_{n-1}) \quad (5)$$

Somit erhält das statistische Sprach-Modell die einfache Form:

$$P(w_{1,n}) = P(w_1, w_2, \dots, w_n) \quad (6)$$

$$= P(w_1)P(w_2 | w_1)P(w_3 | w_1, w_2) \dots P(w_n | w_1, \dots, w_{n-1}) \quad (7)$$

$$= P(w_1)P(w_2 | w_1)P(w_3 | w_1, w_2) \dots P(w_n | w_{n-2}, w_{n-1}) \quad (8)$$

$$= P(w_1)P(w_2 | w_1) \prod_{i=3}^n P(w_i | w_{i-2}, w_{i-1}) \quad (9)$$

Der Tagger von [22, 23] ist ein "TreeTagger", denn im Gegensatz zu n -gramm-Taggern schätzt dieser Tagger die Wahrscheinlichkeit mit einem binären Entscheidungsbaum ab. Abbildung 6 zeigt ein Beispiel eines stark vereinfachten Entscheidungsbaumes. Der Entscheidungsbaum des TreeTagger ist vergleichbar mit dem des C45-Klassifikators aus Kapitel 2.1.

Die Wahrscheinlichkeit für ein gegebenes Trigramm ergibt sich aus dem entsprechenden Pfad durch den Baum zu einem Blatt. Als Beispiel suchen wir die Wahrscheinlichkeit für ein Nomen, dem ein Determinator (Bestimmungswort) und ein Adjektiv vorausgegangen ist. Die Wahrscheinlichkeit ist: $P(NN|DET, ADJ)$. Wir beginnen mit der Suche an der Wurzel des Baumes und entscheiden uns für den "Ja"-Zweig. Den Test am nächsten Knoten beantworten wir ebenfalls mit "Ja" und gelangen so zu einem Blatt, in dessen Tabelle wir die Wahrscheinlichkeit für ein Nomen (NN) nachschauen können. Der Entscheidungsbaum wird gebildet aus einer Trainingsmenge von Trigrammen. Im Anschluss daran wird der Entscheidungsbaum vereinfacht.

Weiterhin verfügt der TreeTagger noch über ein Lexikon, welches eine mutmaßliche Tag-Wahrscheinlichkeit für jedes Wort besitzt. Das Lexikon ist vergleichbar mit dem von [6] verwendeten. Es besteht aus 3 Teilen: dem Wortlexikon, dem Suffixlexikon und einem Standardeintrag. Beim Nachschlagen eines Wortes im Lexikon wird zuerst das Wortlexikon durchsucht. Wird das Wort nicht gefunden, werden alle Großbuchstaben in Kleinbuchstaben umgewandelt und erneut versucht,

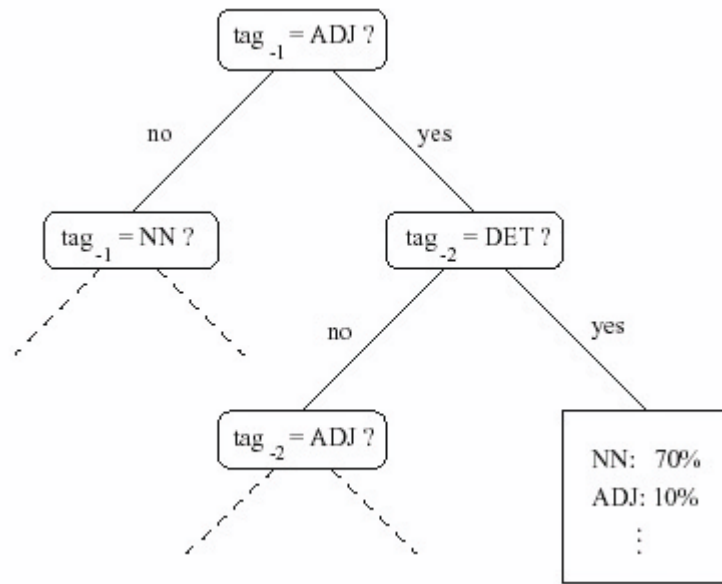


Abbildung 6: Beispiel eines stark vereinfachten Entscheidungsbaumes aus [?].

das Wort zu finden. Schlägt auch dieser Versuch fehl, wird das Suffixlexikon durchsucht. Falls hier ebenfalls kein Eintrag gefunden wird, wird der Standardeintrag verwendet. Das Wortlexikon wird vom Trainingskorporus, dem Penn Reebank Corpus mit 2 Millionen Worten, gebildet. Das Suffixlexikon wird als Baum verwaltet, bei dem jeder Knoten mit einem Buchstaben gelabelt ist, ausser der Wurzel. In den Blättern sind die Tag-Wahrscheinlichkeiten gespeichert. Ein Beispiel für einen einfachen Suffixbaum ist in Abbildung 7 dargestellt.

Beim Durchsuchen des Suffixbaumes beginnt man an der Wurzel mit dem letzten Buchstaben des Wortes und in jedem weiteren Schritt nimmt man einen weiteren Buchstaben vom Ende des Wortes, bis ein Blatt erreicht ist. Der Suffixbaum wird wie auch das Wortlexikon aus der Trainingsmenge gebildet. Hierfür werden alle Suffixes der Länge 5 von "open class part-of-speech"² Wörtern benutzt. Das Suffixlexikon wird genauso wie der Entscheidungsbaum nach der Erstellung automatisch vereinfacht.

Die Berechnung des Standardeintrages erfolgt aus den Tag-Wahrscheinlichkeiten des vereinfachten Entscheidungsbaumes.

Eine weitere Möglichkeit, die Wortklassen von Wörtern zu bestimmen, wird in [7] beschrieben. [7] benutzt zum Erkennen den Porter Stammering Algorithmus [19]. Der Algorithmus analysiert die Wortmorphologie (Wordstruktur). Er redu-

²???

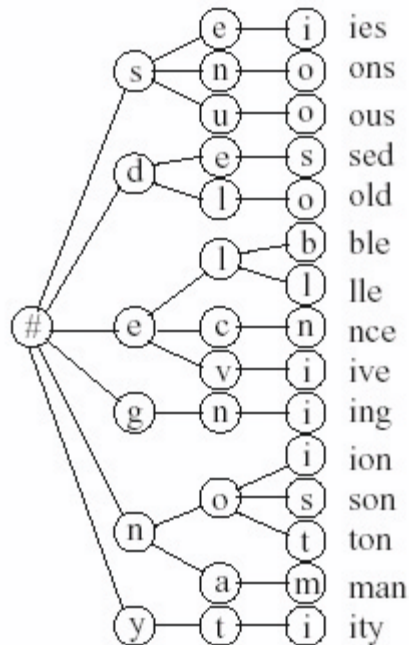


Abbildung 7: Beispiel eines Suffixbaumes der Länge 3 des Taggers aus [?].

ziert Wörter auf ihren Stamm. Die Wörter

- CONNECT
- CONNECTED
- CONNECTING
- CONNECTION
- CONNECTIONS

werden auf CONNECT reduziert, indem er die Endungen jedes Wortes in mehreren Schritten verändert, z. B. wird im ersten Schritt die Endung SSES in SS umgewandelt. Anhand dieser Veränderungen lässt sich unterscheiden, ob das Wort ein Nomen, Verb, Adjektiv oder Adverb ist. Dabei wird das ursprüngliche Wort aber nicht auf seinen Stamm reduziert. Erweitert mit Tabellen von bestimmten Wortklassen, unter anderem Stop-Words, lassen sich auch Pronomen, Konjunktionen und Artikel identifizieren.

4.1.2 Durchschnittliche Satzlänge

Als Maß für die Satzlänge haben sich die Anzahl der Wörter pro Satz durchgesetzt.

4.1.3 Durchschnittliche Wortlänge

Die Länge eines Wortes wird logischerweise als Anzahl der Zeichen pro Wort gemessen.

4.1.4 Anzahl von Sätzen

Ein Problem in HTML ist die Interpunktion. Die Erkennung von Sätzen ist im Gegensatz zu einem einfachen Text komplizierter. In HTML können auch Tags Sätze logisch beenden, ohne dass ein Satzzeichen angegeben ist. Beispiele für solche Tags sind: `<center>`, `<h1>`, usw. Ebenfalls ist nicht klar, wie mit dem Inhalt von Tabellenzellen und Aufzählungen umzugehen ist.

4.1.5 Komplexität von Sätzen

[7] definiert die Komplexität von Sätzen als eine Verbindung von Satzlänge und Wortlänge. Der Durchschnitt der Wortlänge wird dividiert durch den Durchschnitt der Satzlänge.

4.1.6 Anzahl von Abschnitten

Abschnitte sind Textblöcke, die durch Lehrzeilen voneinander getrennt sind. Für die Erkennung von Abschnitten gilt ähnliches wie für Sätze. Abschnitte werden in HTML nicht nur durch Lehrzeilen, sondern auch durch Tags oder horizontale Linien begrenzt.

4.1.7 Verwendete Zeitform

[7] unterscheidet bei den verwendeten Zeitformen nur zwischen Vergangenheit, Gegenwart und Zukunft. Weiterhin wird der Anteil von Zeitwechseln von einem zum nächsten Verb analysiert. Wenn z. B. ein Verb in der Vergangenheit steht und das nächste in der Gegenwart, dann ist das ein "past-to-present" Wechsel. Die Berechnung der Zeitform geschieht bei der Berechnung der Wortformen mit einfachen von Hand gemachten Regeln.

4.1.8 Satzzeichen

(Punkt, Komma, Doppelpunkt, Semikolon, Klammern, Bindestrich und Fragezeichen) Fragezeichen können z.B. hilfreich sein zur Erkennung von FAQs.

4.1.9 Nominalphrase (Noun Phrases)

Eine Phrase ist die nächst höhere syntaktische Einheit nach dem Wort, die normalerweise aus mehr als einem Wort besteht. Die Wörter sind semantisch zusammen-

hängend und bilden eine syntaktische Konstituente.³ Eine Phrase kann aber auch aus Wörtern und Phrasen zusammengesetzt sein. Die Phrase ist ein Objekt. Ihre Art wird durch den Kopf bestimmt, welcher ein Wort ist. Der Kopf wird alternativ auch als "head" oder Kern bezeichnet. Diese Eigenschaft von Phrasen nennt man egozentrisch. Man unterscheidet zwischen 4 verschiedenen Arten von Phrasen:

- Nominalphrase; Kopf ist ein Nomen: das alte **Haus**
- Verbalphrase; Kopf ist ein Verb: **schreibt** einen Brief
- Adjektivphrase; Kopf ist ein Adjektiv: ziemlich **kalt**
- Präpositionalphrase; Kopf ist eine Präposition: **vor** dem Haus

Den Kopf einer Nominalphrase (abgekürzt NP) kann sowohl ein Nomen, Pronomen oder Eigennamen bilden. Der Kopf kann u. a. mit Hilfe von Adjektiven und Appositionen erweitert werden.

Semantisch lassen sich Nominalphrasen in zwei Gruppen aufteilen. Die erste Gruppe bilden die Nominalphrasen, die durch einen Artikel bestimmt werden. In der zweiten Gruppe sind alle Nominalphrasen, die durch einen unbestimmten Artikel gekennzeichnet werden oder keinen Artikel besitzen. Eine Nominalphrase kann sowohl als Subjekt als auch Objekt in einem Satz agieren.

Beispiele für NPs sind:

- das alte Haus
- a very fast car
- a big banana
- the old house
- some angry people

Abbildung 8 zeigt die Elemente, mit denen Nominalphrasen gebildet werden können.

Erkennung von Nominalphrasen

Die Verfahren zur Erkennung von Nominalphrasen lassen sich laut [24] in zwei Gruppen (Ansätze) aufteilen. Die erste Gruppe bilden die regelbasierten Verfahren. Sie basieren auf Grammatiken. In der zweiten Gruppe werden alle Verfahren zusammengefasst, die auf maschinellem Lernen basieren.

³Konstituente := Wortgruppe oder Satzteil von relativer Selbständigkeit
vergl. <http://www.ifi.unizh.ch/cl/Glossar/Phrase.html>

NP					
D	(AP)		N	(PP)	
	(Adv)	A		P	NP
					(D)

A	Adjektiv
ADV	Adverb
AP	Adverbialphrase
D	Artikel
N	Nominativ
NP	Nominalphrase
P	Präposition
PP	Präpositionalphrase

Abbildung 8: Allgemeine Struktur von Nominalphrasen [13].

Die Grundlage für beide Ansätze bildet die Bearbeitung des Textes durch einen Part-of-Speech Tagger POS⁴ oder ein vergleichbares Programm. Der Part-of-Speech Tagger weist jedem Wort des Textes die entsprechende Wortklasse zu, basierend auf dem Kontext des Wortes.

Die beiden Ansätze unterscheiden sich im weiteren Vorgehen. Das Chunking-Verfahren parst die Ausgabe des POS mit Hilfe von Grammatiken und findet so Nominal- und Verbalphrasen. Beim maschinellen Lernen wird die Ausgabe basierend auf erlerntem "Wissen" untersucht und die Nominalphrasen erkannt.

Chunking-Verfahren

Das Chunking-Verfahren läuft in zwei Schritten ab. Als erstes wird der Text mit einem Part-of-Speech Tagger bearbeitet, dabei wird jedes Wort zu einer Wortklasse zugeordnet. Im zweiten Schritt, dem Chunking, wird die Ausgabe des POS mit Grammatiken analysiert, und die gefundenen Phrasen werden markiert.

Die beim Chunking benutzten Grammatiken werden Phrasenstruktur-Grammatiken (PS-Grammatik bzw. PSG) genannt. Ein einfaches Beispiel für eine Phrasenstruktur-Grammatik in Englisch ist in Abbildung 9 dargestellt.

Genau genommen zeigt Abbildung 9 keine Grammatik, sondern nur die Produktionen (Ersetzungsregeln) einer Grammatik. Eine Grammatik G besteht aus:

⁴Vergl. auch Abschnitt 4.1.1

S	→	NP VP
VP	→	V NP
NP	→	D (AP) N (PP)
AP	→	(Adv) A
PP	→	P NP
D	→	the
N	→	boy, dog ...
V	→	petted, kicked, chased, ...
A	→	big, small, tall, brown,...
Adv	→	very, extremely, terribly,
...		
P	→	in, on, under, ...

Abbildung 9: Einfaches Beispiel einer Phrasenstruktur-Grammatik in Englisch [13]

- einer endlichen Menge von Terminalen T in Abb. 9 sind das: boy, dog, ..., petted, kicked, chased, ...
- einer endlichen Menge von Variablen V im Bsp: S, NP, VP, D, AP, N, PP,
- einem Startsymbol S aus der Menge der Variablen in diesem Fall S
- einer endlichen Menge von Produktionen (Ersetzungsregeln) P mit $P \subseteq ((V \cup T)^+ \setminus T^* \times (V \cup T)^*)$ die in Abb. 9 dargestellt sind.

Eine Grammatik heisst kontextsensitiv, falls für jede Regel $u \rightarrow v$ gilt: $|u| \leq |v|$ sind.

Eine Grammatik heisst kontextfrei, falls alle Regeln von der Art $u \rightarrow v$ mit $u \in V$ sind.

Im Gegensatz zu formalen Sprachen, wie Programmiersprachen, lässt sich die menschliche Sprache nicht durch eine kontextfreie Grammatik ausdrücken. Kontextfreie Grammatiken erlauben die Erzeugung grammatisch unkorrekter Sätze und zweitens erlauben sie nicht, alle korrekten Sätze zu erzeugen. Die menschliche Sprache ist kontextsensitiv.

Der Chunker, "LT CHUNK"⁵ der Teil des Part-of-Speech Tagger "LT POS"⁶ ist, benutzt zum Erkennen der Nominalphrasen eine kontextsensitive Grammatik.

Maschinelle Lernverfahren

⁵<http://www.ltg.ed.ac.uk/software/chunk/index.html>

⁶<http://www.ltg.ed.ac.uk/software/pos/>

Beim Chunking-Verfahren werden die Nominalphrasen nach fest vorgegebenen Regeln erkannt. Im Gegensatz hierzu steht das maschinelle Lernen. Hierbei werden die "Regeln" beim Trainieren des Systems gelernt. Es lernt die syntaktischen Strukturen einer Sprache und wendet das gelernte Wissen bei der Erkennung von Nominalphrasen an.

Das Trainieren des Systems geschieht mit Hilfe eines Trainingskorpus. Dieser Korpus enthält eine Sammlung von Texten mit Nominalphrasen. An Hand dieser Phrasen kann das System die Syntax lernen und sein Wissen aufbauen. Für das maschinelle Lernen von Nominalphrasen werden verschiedene Lernverfahren eingesetzt: Entscheidungsbäume, Hidden-Markov-Modelle, ähnlichkeitsbasiertes Lernen und andere.

4.2 Lexikalische Indikatoren

4.2.1 Wortfrequenzen

Die Frequenz eines Wortes ist die Anzahl, wie häufig ein Wort in einem Dokument vorkommt. Es gibt aber neben der Wortfrequenz noch weitere Frequenzen:

- Termfrequenz (Wortfrequenz) $tf_{ij} :=$ Häufigkeit von Term i im Dokument j
- Dokumentfrequenz $df_i :=$ #Dokumente, die Term i enthalten
- Inverse Dokumentenfrequenz $idf_i := \log(n/df_i)$

4.2.2 Wortfrequenzen bestimmter abgeschlossener Wortmengen

Beispiele für solche Wortmengen sind z. B. Wochentage, Monate, Sternzeichen usw. Einige dieser Gruppen sind allgemein, andere sind speziell für Genre. Der Begriff "Löwe" steht nicht allein für das Genre Horoskop. Im Zusammenhang mit einem weiteren Begriff aus dem Bereich Astrologie kann dies ein gutes Anzeichen sein.

4.2.3 Frequenzen von Stop-Words

Stop-Words sind Wörter, die keine Informationen, z. B. zur Suche im Internet bei Google⁷, enthalten. Englische Beispiele für Stop-Words sind: "the", "and", "asked". Diese Wörter werden bei Suchanfragen ignoriert. Die Erkennung von Stopwords geschieht mit Hilfe von Stopword-Listen. Vorteile können Stop-Words bei der Erkennung von Werbung bringen, da diese sehr viel mit Stop-Words arbeitet.

4.2.4 Anzahl von "it"s

wird in [25] als Indikator genannt. Die Anzahl von "it"s ist aber kein eigenständiger Indikator, weil "it" ein Stop-Word ist.

⁷www.google.de

4.2.5 Frequenzen von Schlüsselwörtern

Verschiedene charakteristische Schlüsselwörter, die ein Genre identifizieren. Die Identifizierung von Schlüsselwörtern kann wie bei den Stop-Words mit Hilfe von Listen geschehen. Beispiele für solche Schlüsselwörter sind im Genre Shop z. B.: shop, buy, store, price, online, order, sale. Die Begriffe: Link und Links tauchen im Zusammenhang mit "Link Collections" immer wieder auf.

4.2.6 Anzahl von nicht alpha-numerischen Zeichen

Die wichtigsten nicht alpha-numerischen Zeichen sind die Währungssymbole. Sie sind ein gutes Merkmal für Shops, da die Waren durch Preise ausgezeichnet sind. Die Unterscheidung von Preisen zu anderen Zahlen geschieht am einfachsten durch die Währungssymbole wie: \$, £, EUR, €.

4.2.7 Anzahl von Zahlen

4.3 Gestaltungstechnische Indikatoren

Unter dem Punkt "gestaltungstechnische Indikatoren" fasse ich solche zusammen, die nicht den textuellen Inhalt von Dokumenten betrachten, sondern deren Darstellung und die Aufbereitung des Textes durch Grafiken, Tabellen usw.

4.3.1 Zeilenabstand

4.3.2 Anzahl von Aufzählungen

Die Anzahl von Aufzählungen lässt sich in HTML auf zwei verschiedene Arten messen:

1. durch das Zählen der öffnenden `` Tags und der öffnenden ``. Als Ergebnis erhält man die Anzahl der Listen.
2. durch das Zählen der öffnenden `` Tags. Als Ergebnis erhält man die Anzahl der Listenelemente.

Es sind aber auch Kombinationen denkbar, z. B. Listenpunkte pro Liste.

4.3.3 Anzahl von Tabellen

Siehe 4.4.3.

4.3.4 Anzahl von Grafiken

Die Anzahl der Grafiken lässt sich in HTML leicht durch Zählen der Zeichenfolge `` messen.

4.3.5 Größe und Format von Grafiken

Das Format von Grafiken wäre interessant, um Banner zu erkennen. Leider ist aber der Parameter des `` Tags zur Angabe der Höhe und Breite nur optional, und ein Laden von Grafiken zur Analyse wäre wie bereits beschrieben zu aufwändig.

4.3.6 Verhältnis Text zu Bild

Das Verhältnis Text zu Bild ergibt sich aus der Fläche, die für Bilder und Text belegt ist.

4.3.7 Schriftgröße, Anzahl der verwendeten Schriften und Farben

Diese Eigenschaften können sowohl in der HTML-Datei, als auch in einer externen CSS-Datei gespeichert werden, welche zur Analyse geladen werden müsste.

4.4 HTML spezifische Indikatoren

Die HTML spezifischen Indikatoren grenzen sich von den gestaltungstechnischen Indikatoren ab, indem sie speziell auf HTML spezifische Eigenschaften eingehen.

4.4.1 Anzahl von Hyperlinks

Hyperlinks lassen sich in 5 einfache Kategorien unterteilen:

- Javascript-Links rufen Javascript Methoden auf.
- Mailto-Links verweisen auf Emailadressen
- Anker-Links sind Links, die auf einen Anker innerhalb einer HTML-Seite verweisen. Diese Anker werden definiert durch das `<a>`-Tag.
- Domain-Links sind Verweise innerhalb einer Domäne.
- Internet-Links sind alle anderen Links

Berechnung

Zur Erkennung wird beim Einlesen des HTML-Codes nach der Zeichenfolge `href=` gesucht und der nachfolgende Link analysiert. Beginnt dieser mit `"javascript:"`, handelt es sich um einen Javascript Link. Beginnt er mit `"mailto:"`, ist es ein Verweis auf eine Emailadresse. Beinhaltet er eine `"#"`, ist es ein Anker-Link. Zuletzt wird untersucht, ob er die Zeichenfolge `://"` beinhaltet. Ist dieses nicht der Fall, ist es ein Domain-Link, wenn er aber `://"` beinhaltet, wird die Domäne des Links mit der Domäne der Webseite verglichen. Sind diese identisch, ist es ein Domain-Link; sind sie verschieden, ist es ein Internet-Link. Ich benutze als Domäne nur die Toplevel-Domäne und die direkte Subdomäne. Beispiele für solche Domains sind: `uni-paderborn.de`, `ibm.com`, `linux.org`.

4.4.2 HTML Metainformationen

Im Kopf einer HTML-Datei besteht die Möglichkeit, Informationen über die Datei zu speichern. Dieses geschieht in den Metainformationen.

Die folgenden Zeilen sind ein Auszug aus den Metainformationen der Homepage der Universität Paderborn⁸:

```
<meta name="description"
      content="Universit&auml;t Paderborn - Homepage">
<meta name="keywords" lang="de"
      content="Universit&auml;t, Paderborn, Studium, Institutionen,
      Institute, Einrichtungen, Forschung, Lehre, Wissenschaft">
<meta name="keywords" lang="en"
      content="university, Paderborn, studies, institutes,
      facilities, faculties, research, teaching, science">
```

Selfhtml [18] schreibt über den Verwendungszweck von Metainformationen folgendes: "In Meta-Angaben können Sie verschiedene nützliche Anweisungen für Web-Server, Web-Browser und automatische Suchprogramme im Internet (Robots) notieren. Meta-Angaben können Angaben zum Autor und zum Inhalt der Datei enthalten. Sie können aber auch HTTP-Befehle absetzen, zum Beispiel zum automatischen Weiterleiten des Web-Browsers zu einer anderen Adresse."

Bei der Verwendung der Metainformationen als Indikator für einen Klassifikator bestehen zwei Probleme:

1. HTML Metainformationen müssen nicht angegeben werden. Sie sind nur optional.
2. Sie werden mißbraucht, um einen anderen Inhalt vorzutäuschen. Sexseiten z. B. geben in den Metainformationen häufig angegebene Suchbegriffe an, in der Hoffnung, dadurch bei diesen Suchen genannt zu werden.

Viele Suchmaschinen ignorieren aus diesen Gründen die Metainformationen.

4.4.3 Anzahl von Tabellen

Tabellen haben in HTML das Problem, dass sie sowohl für die Gestaltung (Layout), aber auch für die Aufbereitung von Informationen verwendet werden. Aus diesem Grund hat [26] einen Klassifikator entwickelt, der die Tabellen in einer HTML-Seite unterteilen kann in:

- Tabellen zur Gestaltung (unechte Tabellen) Dieser Typ von Tabellen wird häufig benutzt, um die Verwendung von Frames zu umgehen.
- Tabellen für Informationen (echte Tabellen)

⁸www.upb.de

Wang definiert in [26] "echte" und "unechte" Tabellen. "Echte" sind seiner Meinung nach Teile eines Dokumentes, bei denen ein zweidimensionales Gitter semantisch signifikant den Zusammenhang zwischen den Zellen darstellt. "Unechte"-Tabellen sind Teile eines Dokumentes, bei denen das <table>-Tag benutzt wird, um den Inhalt für die Anzeige zu gruppieren. Wang entwickelt zur Unterscheidung von echten und unechten Tabellen einen Klassifizierer. Als Eingabe für seinen Klassifizierer benutzt er Features aus drei verschiedenen Gruppen.

- Die erste Gruppe bilden die **Layout Features**. Hierfür berechnet Wang zunächst eine Matrix, die ein pseudo Rendering der Tabelle enthält. Diese Matrix dient als Grundlage zur Berechnung der Layout Features, wie z. B. durchschnittliche Anzahl von Spalten, Standardabweichung der Anzahl von Spalten.
- Die zweite Gruppe bilden die **inhaltlichen Features**. Zu dieser Gruppe gehören Features, wie die Anzahl von Bildern, Formen, aber auch Zahlen.
- Die dritte und letzte Gruppe bilden die **Wortgruppen Features**. Zu dieser Gruppe gehören Features, die den Wortvektor jeder neu zu klassifizierenden Tabelle mit den Wortvektoren für echte und unechte Tabellen vergleichen. Die Referenzvektoren berechnet Wang an Hand seines Korpus.

Als Korpus dient ihm eine Sammlung von 1392 HTML Dokumenten, die von 200 Webseiten gesammelt wurden. Diese Sammlung beinhaltet 14609 Tabellen, von denen 11477 keine weiteren Tabellen beinhalten. 1740 dieser 11477 Tabellen sind echte Tabellen. Die Anzahl der Tabellen, die zur Informationsvermittlung benutzt werden, wäre interessant, um Aussagen über den Informationsgehalt einer Seite treffen zu können. Leider benutzt [26] hierfür aber ebenfalls einen Klassifikator, der trainiert werden muss, was den Rahmen dieses Projektes sprengen würde.

4.4.4 Technische HTML Tags

Das Ziel der Interpretation von technischen HTML Tags sind Internetseiten, wie Werbung oder Produktinformationen, von wissenschaftlichen Artikeln zu unterscheiden. In der folgenden Tabelle aus Selfhtml [18] sind Beispiele für solche Tags aufgelistet.

<pre>	Textabschnitt mit präformatiertem Text
<code>	zeichnet einen Text aus mit der Bedeutung "dies ist Quelltext"
<samp>.	zeichnet einen Text aus mit der Bedeutung "Dies ist ein Beispiel"
<kbd>	zeichnet einen Text aus mit der Bedeutung "dies stellt eine Tastatureingabe dar"
<var>	zeichnet einen Text aus mit der Bedeutung "dies ist eine Variable oder ein variabler Name"
<cite>	zeichnet einen Text aus mit der Bedeutung "dies ist ein Zitat von einer anderen Quelle"
<dfn>	zeichnet einen Text aus mit der Bedeutung "dies ist eine Definition".
<acronym>	zeichnet einen Text aus mit der Bedeutung "dies ist eine Abkürzung" (z.B. "z.B.")
<abbr>	zeichnet einen Text aus mit der Bedeutung "dies ist eine abgekürzte Schreibweise" (z.B. "WWW")
<q cite="Quelle">	zeichnet einen Text aus mit der Bedeutung "dies ist ein Zitat mit Quellenangabe"

5 Eigenes Vorgehen

5.1 Korpus

Wie bereits unter Punkt 2 beschrieben, beinhaltet der Korpus Beispielinstanzen zum Trainieren und Testen eines Klassifizierers. In meinem Fall benötige ich HTML-Seiten. Bei der Erstellung des Korpus werden auch die zukünftigen Genres ausgewählt. Daher möchte ich zuerst einen Überblick über die Korpora und benutzten Genres anderer Autoren geben, die sich mit dem Thema der "Genre-Klassifikation" befasst haben.

5.1.1 Korpora anderer Autoren

Karlgren und Cutting benutzen für ihre Arbeit am Text "Recognizing text genres with simple metrics using discriminant analysis" [14] von 1994 den Korpus von Brown. Dieser beinhaltet englische Texte gleicher Länge. In Tabelle 1 sind die Kategorien des Korpus von Brown angegeben und die von [14] benutzten "Genres".

Experiment 1	Experiment 2	Experiment 3
		Brown categories
	Press	Press: reportage
		Press: editorial
		Press: reviews
Informative	Misc	Religion
		Skills and Hobbies
		Popular Lore
		Belles Lettres, etc.
	Non-fiction	Gov. doc. & misc.
		Learned
		General Fiction
		Mystery
Imaginative	Fiction	Science Fiction
		Adv. & Western
		Romance
		Humor

Tabelle 1: Kategorien des Korpus von Brown und benutzte "Genres" von [14]

Man erkennt, dass der Korpus von Brown thematisch unterteilt ist. Seine Kategorien sind z. B. Religion, Mystery oder Humor. Der Begriff "Genre" bedeutet aber, wie bereits unter Punkt 1.2.2 beschrieben, Art oder Gattung. Dieses trifft höchstens auf die Kategorien Press: reportage, Press: editorial, Press: reviews zu. Wegen der thematischen Unterteilung ist der Korpus für mich unbrauchbar.

Brett Kessler et al. benutzten 1997 in [15] ebenfalls den Korpus von Brown.

Yong-Bae Lee und Sung Hyon Myaeng haben sich für [17] im Jahre 2002 einen eigenen Korpus von Webseiten gebaut. Er beinhaltet die Genres: Reportage, Editorial, Research articles, Reviews, Homepage, Q&A, Spec.

Im Paper [7] wird der Genre Korpus der Carnegie Mellon University (CMU) benutzt. Er wurde erstellt von Jaime Carbonell und Fang Liu und beinhaltet die Genres: Advertisement, Bulletin Board, Frequently Asked Questions, Message Board, Radio News, Reuters Newswire, Television News.

Die Autoren von [25] waren mit dem Korpus von Brown ebenfalls nicht zufrieden. Sie meinten, der Korpus sei nicht exklusiv zur Genreerkennung von Texten erzeugt worden. Deswegen seien die Texte in einer Kategorie stilistisch nicht homogen. Aus diesen Gründen benutzten sie den Korpus des Wall Street Journal (WSJ). Die Tabelle 5.1.1 zeigt die im WSJ Korpus vorhandenen Überschriften und die von [25] zugeordneten Genres.

Überschrift im Korpus	Genre
REVIEW & OUTLOOK (Editorial) :	editorial
Letters to the Editor: '	letters to the editor
What's News -	news
Who's News:	
International :	reportage
Marketing & Media:	
Politics & Policy:	
World Markets :	

Tabelle 2: Der Korpus des Wall Street Journal (WSJ) mit den von [25] verwendeten Genres

5.1.2 Mein Korpus

Ich benötige für die Genre-Klassifikation von Webseiten einen Korpus mit HTML-Dateien. Einen vergleichbaren Korpus haben nur die Autoren von [17] benutzt. Ich habe mich ebenfalls entschieden, einen Korpus selber zu erstellen. Begonnen habe ich mit 8 - 10 Genres. Bei der Erstellung bekam ich Probleme mit der Einteilung. Darum habe ich verschiedene Genres zusammengefasst und ihn neu sortiert. In der Tabelle 5.1.2 sind die von mir benutzten Genres aufgelistet und die Anzahl der Dokumente im Korpus angegeben.

- Unter dem Genre **articles** habe ich alle Webseiten, die längere Berichte oder Artikel über ein Thema enthalten, zusammengefasst. Es ist gedacht für Leute, die zu einem Thema ausführliche Informationen suchen.
- Im Genre **linklists** befinden sich Internetseiten von Webkatalogen, sowie die Linksammlungen von Homepages. Dieses Genre ist ebenfalls interessant für

articles	123
help	136
linklists	204
portrait-non_priv	171
portrait-priv	127
shop	169
download	152
discussion	127

Tabelle 3: Genres und Anzahl der HTML-Seiten im erstellten Korpus

Personen, die Informationen zu einem Thema suchen. Es ist auch vorstellbar, dass man dieses Genre bei der Suche im Internet ausschliesst, weil die Seiten zwar auf andere Seiten verweisen, die Informationen beinhalten können, aber selber keine Informationen enthalten.

- Wie der Name des Genres **shop** bereits sagt, befinden sich in diesem Genre Internetseiten, die Produkte oder Dienstleistungen zum Kauf anbieten. Die mögliche Zielgruppe sind Personen, die Produkte oder Dienstleistungen kaufen wollen oder deren Preis erfahren möchten.
- Das Genre **help** beinhaltet Webseiten, auf denen Besucher, die Probleme mit einem Produkt haben, Hilfe finden.
- Das Genre **discussion** beinhaltet Webseiten aus FAQs (Q&A) und Foren. Die Zielgruppe dieses Genre sind ebenfalls Personen, die Probleme mit einem Produkt haben und Hilfe benötigen.
- Internetseiten, auf denen man Treiber oder Programme herunterladen kann, habe ich unter dem Genre **download** zusammengefasst.
- Die letzten beiden Genres **portrait-non_priv** und **portrait-priv** waren in der Auswahl am schwierigsten. Begonnen habe ich mit dem kommerziellen und privaten Portrait. Die Einordnung der Homepage einer Firma in kommerziell und die einer einzelnen Familie in privat ist eindeutig. Die Frage war nur, in welche Gruppe gehört z. B: die Homepage eines Sportvereins. Ein Verein oder eine Stiftung ist nicht kommerziell, aber auch nicht privat. Aus diesem Grund habe ich mich für die Genres "Portrait privat" und "Portrait nicht privat" entschieden. Bei dieser Aufteilung kommen die Homepages einer Firma und eines Vereins in dieselbe Kategorie (Portrait nicht privat). Diese Aufteilung hat sich bei der Erstellung des Korpus als sehr praktisch heraus gestellt.

Bei der Erstellung des Korpus habe ich nicht nur auf die Auswahl der Genres geachtet, sondern auch auf eine möglichst ausgeglichene thematische Zusammenstellung der Webseiten innerhalb eines Genre, z. B. sollte das Genre "help"

nicht nur Seiten aus dem Bereich Informatik (Computer) enthalten und das Genre "shop" nicht nur Internet-Buchhandlungen enthalten.

5.2 Programm

Bevor ich mit der Erstellung des Korpus begonnen habe, habe ich mir überlegt, wie Software mich bei meiner Arbeit unterstützen kann und welche Funktionen sie besitzen muss. Daraufhin habe ich mir folgende Anforderungen und Ziele überlegt:

- schnelle und einfache Erzeugung eines neuen Korpus
- schnelle und einfache Verwaltung eines vorhandenen Korpus
- Möglichkeit zur Anbindung an vorhandene Klassifizierungstools

Um eine schnelle und einfache Erzeugung und Verwaltung des Korpus zu erreichen, habe ich mich entschieden, auf vorhandene Software zurückzugreifen. Meine Wahl fiel dabei auf den Internetbrowser Mozilla⁹. Die Verwaltung des Korpus geschieht dabei mit Hilfe der Bookmark-Verwaltung des Browser. Für jedes neue Dokument, das in den Korpus eingefügt werden soll, wird ein neuer Bookmark erzeugt. Die Zuordnung des Genre geschieht dabei durch Einsortieren des Bookmark in den entsprechenden Ordner. Um eine bereits einsortierte Webseite einem anderen Genre zuzuordnen, wird der dazu gehörige Bookmark einfach in einen anderen Ordner verschoben.

Der Internetbrowser erlaubt mir, Bookmarks zu sammeln und diese zu Genres zuzuordnen. Mozilla speichert seine Bookmark-Sammlung in einer Bookmark-Datei, die ich als Einabe in meinem Programm verwende.

Für die Berechnung der Features, die ein Klassifizierungstool als Eingabe benötigt, brauche ich aber nicht nur den Link (Bookmark) auf eine Webseite, sondern auch deren HTML-Code.

Für die Erzeugung und Verwaltung der HTML-Dateisammlung (den Korpus) auf der Festplatte habe ich ein Programm geschrieben, das mit Hilfe der Bookmark-Datei von Mozilla den Korpus verwalten kann. Es erfüllt vier verschiedene Funktionen:

1. Die Funktion **Update Corpus** bewirkt, dass für neu eingefügte Bookmarks der HTML-Code heruntergeladen und auf der Festplatte gespeichert wird. Bei Bookmarks, die von einem Ordner in einen anderen verschoben wurden, wird einfach das Genre in der zugehörigen Datei geändert.
2. Die Software beinhaltet eine Funktion **Generate Bookmark-File**, die aus einem vorhandenen Korpus die Bookmark-Datei generiert.

⁹<http://www.mozilla.org>

3. Die dritte Funktion **Create Feature-Files** der Software berechnet für den Korpus die verwendeten Indikatoren und speichert diese in mehreren Textdateien ab.
4. Die letzte Funktion **Classify** klassifiziert die Daten im Korpus, basierend auf den Ausgabedateien der Funktion "Create Feature-Files".

Weiterhin gehe ich nun im Einzelnen auf die Verwaltung der Bookmarks im Browser ein und erkläre auch die Funktionen des Programms im Detail.

5.2.1 Die Bookmark-Verwaltung im Browser

Wie bereits beschrieben, benutze ich für die Verwaltung der Bookmarks den Internetbrowser Mozilla oder den vergleichbaren Browser Netscape. Hierfür wird im Browser ein neues Profil (Benutzer) angelegt. Im Verzeichnis des angelegten Profils speichert der Browser die verwendete Bookmark-Datei. Die Abbildung 10 zeigt die Verwaltung einer Bookmark-Sammlung in Mozilla.

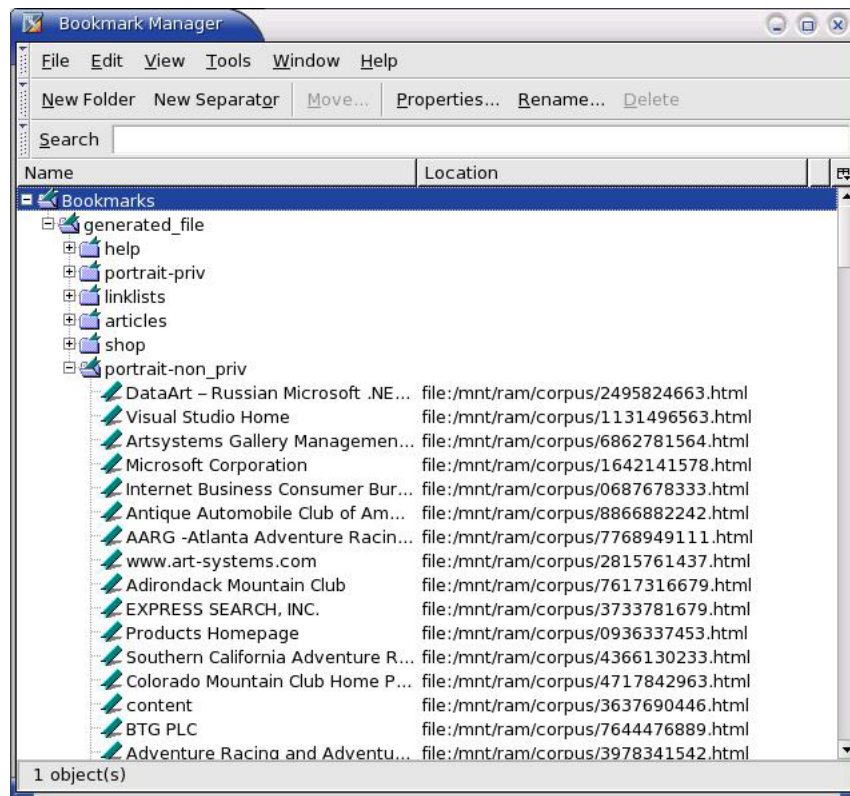


Abbildung 10: Verwaltung der Bookmarks im Internetbrowser Mozilla.

Ein Problem ergibt sich aus der Zweiteilung der Software in den Internetbrowser Mozilla und das Verwaltungsprogramm. Mozilla lädt seine Bookmark-Datei

beim Starten in den Arbeitsspeicher. Die während des Laufens gemachten Änderungen der Bookmarks werden erst beim Beenden von Mozilla auf die Festplatte geschrieben, somit stehen die Änderungen erst dann für das Verwaltungsprogramm zur Verfügung. Änderungen, die ein anderes Programm, also auch das Verwaltungsprogramm, während des Laufens von Mozilla an der Bookmark-Datei vornehmen, gehen beim Beenden des Browsers verloren, weil Mozilla die Datei immer überschreibt, unabhängig davon, ob beim Betrieb Änderungen an den Bookmarks vorgenommen wurden oder ob ein externes Programm die Datei verändert hat. Aus diesem Grund muss Mozilla immer beendet sein, wenn das Verwaltungsprogramm benutzt wird.

5.2.2 Nescape Bookmark-Datei

Die Nescape Bookmark-Datei hat ihr eigenes Dateiformat. Das Format ist vergleichbar mit einer HTML-Datei, jedoch besitzt die Bookmark-Datei keinen "head" und keinen "body". Die Bookmarks werden, wie in HTML die Links, in <A>-Tags gespeichert. Diesen <A>-Tags werden noch <DT>-Tags vorangestellt, die den Link als Bookmark kennzeichnen. Schließende </DT>-Tags sind nicht vorhanden. Die Ordner werden in der Bookmark-Datei durch öffnende und schließende <DL>-Tags realisiert. Ein Beispiel für eine Bookmark-Datei ist im Anhang unter 6.1 abgedruckt.

Zum Parsen dieses Dateiformates in Java benutze ich den von Scott Violet angepassten Swing HTML Parser¹⁰.

5.2.3 Korpusdateiformat

Den aus dem Internet heruntergeladenen HTML-Code habe ich nicht eins zu eins in eine HTML-Datei abgespeichert. Am Anfang der HTML-Datei habe ich einen HTML-Kommentar eingefügt. Dieser beinhaltet folgende Tags: <URL>, <TITLE>, <GENRE>, <CONTENT>, die vom Browser nicht interpretiert werden.

- Im <URL> Tag ist die Quell-URL der Webseite gespeichert.
- Der vom Browser ermittelte Titel der Webseite ist im <TITLE> Tag gespeichert.
- Das Tag <GENRE> beinhaltet das Genre der Webseite.
- <CONTENT> markiert den Beginn des eigentlichen HTML-Codes.

Der Vorteil einer modifizierten HTML-Datei besteht darin, dass diese im Gegensatz z. B. zu einer XML-Datei problemlos im Browser angezeigt werden kann und damit der Benutzer die Einordnung der Webseiten in Genres besser überprüfen kann. Ein Beispiel für eine Korpusdatei ist im Anhang unter 6.2 abgedruckt.

¹⁰<http://java.sun.com/products/jfc/tsc/articles/bookmarks/>

- C45 [?]]
- Weka [12]

Der Ablauf der Funktion Create Feature-Files ist im Sequenzdiagramm in Abbildung 12 beispielhaft für eine Datei aus dem Korpus dargestellt. Hierbei erzeugt die GUI ein neues GenreExperiment, das ein eigenständig laufender Thread ist und somit die GUI nicht blockiert. Das GenreExperiment ließt als erstes mit Hilfe der Methode sortSites in der Klasse SiteSelection alle Korpusdateien in den Speicher ein. Daraufhin wird von der Funktion makeFeatureFiles in der Klasse GenreExperiment geprüft, ob für jede Korpusdatei eine Datei existiert, in der die berechneten Features der Korpusdatei abgespeichert sind. Ein Beispiel für eine solche Feature-Datei ist im Anhang unter 6.3 abgedruckt. Falls dieses nicht der Fall ist, werden die Dateien erzeugt.

Im nächsten Schritt werden von der Funktion removeIrrelevantFeatureFiles alle Korpusdateien aus dem Speicher entfernt, die keine sinnvollen Einträge enthalten. Dieses ist der Fall, wenn die HTML-Seite, die in der Datei abgespeichert ist, z. B. nur ein Frameset enthält.

Als nächstes wird der User mit einem kleinen erzeugten Fenster gefragt, wieviele Dokumente pro Genre benutzt werden sollen und welcher Name für die Ausgabedateien verwendet werden soll. Im Anschluss an die Eingabe der Werte und deren Bestätigung wird als erstes die SPSS Ausgabedatei erzeugt und im Anschluss daran die SNNS Trainings- und Testmenge in Dateien gespeichert. Wenn der Benutzer in den Einstellungen ebenfalls den Export von Dateien im ARFF- und C45-Format ausgewählt hat, werden als letztes in diesen Formaten sowohl die Trainingsmenge als auch die Testmenge exportiert.

5.2.7 Classify

Der Titel meiner Arbeit ist Genreklassifikation von Webseiten. Meine Software unterstützt mich bei der Erstellung und Verwaltung eines Korpus und beim Berechnen der Features für die Webseiten im Korpus. Die Klassifikation von Webseiten ist bisher aber nur mit Hilfe eines externen Programmes, wie dem Stuttgart Neural Network Simulator [29], möglich, das die abgespeicherten Features auswertet. Die Funktion "Classify" implementiert die folgenden Klassifikatoren: NaiveBayes und C45. Diese Klassifikatoren habe ich nicht selber programmiert, sondern greife auf die Implementierungen aus dem WEKA Projekt [12] zurück. Um eine einheitliche Schnittstelle für die Verwendung der Klassifikatoren zu bekommen, habe ich für jeden einzelnen eine Wrapperklasse geschrieben. Als Grundlage für die Klassifikation verwende ich die in der Funktion "Create Feature-Files" im WEKA-Format abgespeicherten Ausgabedateien. Die Ausgabe der Klassifikationsgüte geschieht dabei auf der Standardausgabe (Textausgabe).

5.3 Experimente

In diesem Kapitel berichte ich über meine durchgeführten Experimente und gebe eine Interpretation der Ergebnisse. Das Ziel der Experimente ist, zu überprüfen, in wie weit eine automatische Genreklassifizierung von Webseiten möglich ist. Hierfür vergleiche ich verschiedene Klassifikatoren miteinander.

Für die Tests verwende ich alle von mir berechneten Features. Die Tabelle ?? listet alle diese Features auf und gibt Verweise auf Kapitel 4.

Wie bereits erwähnt, möchte ich untersuchen, welcher Klassifikator sich am besten für die Genreklassifikation von Webseiten eignet und verwende dafür folgende Klassifikatoren:

- C45[20] aus dem Weka-Projekt[12] Ich verwende den C45-Klassifikator, weil dieser von Aidan Finn u. a. in der Ausarbeitung [9] von ihm als ein Klassifikator beschrieben wurde, der sich gut für die Genreklassifikation eignet.
- NaiveBayes aus dem Weka-Projekt[12] Der NaiveBayes-Klassifikator ist ein Klassifikator, mit dem sich Dokumente schnell klassifizieren lassen. Aus diesem Grund wird er häufig zur Erkennung von Spam-Mails eingesetzt. Die bei der Spam-Klassifikation verwendeten Features sind in Teilen mit meinen vergleichbar. Dort werden ebenfalls Features eingesetzt, die von Part-Of-Speech Taggern berechnet werden. Auf Grund dieser Ähnlichkeiten möchte ich überprüfen, ob sich der NaiveBayes-Klassifikator auch für die Genreklassifikation eignet.
- SNNS Stuttgart Neural Network Simulator [29] Ich verwende das SNNS, weil es mir von meinem Betreuer als ein gutes Klassifizierungstool empfohlen wurde.

Die Tabellen 4, 5, 6 beinhalten die Konfusionsmatritzen der Klassifikationen. Man kann erkennen, dass der SNNS Klassifikator mit ca. 65% die meisten Dokumente aller Klassifikatoren korrekt klassifiziert. Im Vergleich dazu hat der C45 nur 52 % Ergebnisse richtig klassifiziert. Der NaiveBayes Klassifikator ist von allen drei Klassifikatoren der mit dem einfachsten Ansatz zur Klassifikation, hat aber auch nur 43 % der Ergebnisse richtig klassifiziert.

5.4 Zukünftige Verbesserungsmöglichkeiten

	portrait non-priv	linklists	shop	discussion	portrait priv	articles	download	help
portrait non-priv	77%	4%	4%	0%	12%	4%	0%	0%
linklists	8%	61%	3%	11%	6%	6%	6%	0%
shop	4%	12%	68%	4%	4%	4%	4%	0%
discussion	15%	23%	3%	45%	3%	5%	5%	3%
portrait priv	4%	4%	11%	11%	54%	4%	14%	0%
articles	8%	4%	0%	4%	8%	73%	0%	4%
download	3%	7%	3%	3%	0%	3%	80%	0%
help	6%	17%	3%	0%	6%	0%	0%	69%

Tabelle 4: Konfusionsmatrix der Klassifikation aller 8 Genres mit dem SNNS. Korrekt klassifiziert wurden: ca. 65%

	portrait non-priv	linklists	shop	discussion	portrait priv	articles	download	help
portrait non-priv	33%	16%	17%	4%	10%	4%	8%	5%
linklists	13%	53%	7%	2%	11%	3%	5%	5%
shop	19%	8%	55%	4%	1%	2%	5%	3%
discussion	7%	6%	2%	65%	6%	1%	6%	6%
portrait priv	8%	19%	4%	6%	48%	4%	2%	5%
articles	6%	10%	3%	3%	9%	53%	1%	12%
download	16%	4%	3%	5%	2%	0%	64%	2%
help	8%	8%	4%	7%	5%	11%	6%	47%

Tabelle 5: Konfusionsmatrix der Klassifikation aller 8 Genres mit C45 aus dem Weka-Projekt. Korrekt klassifiziert wurden: ca. 52 %

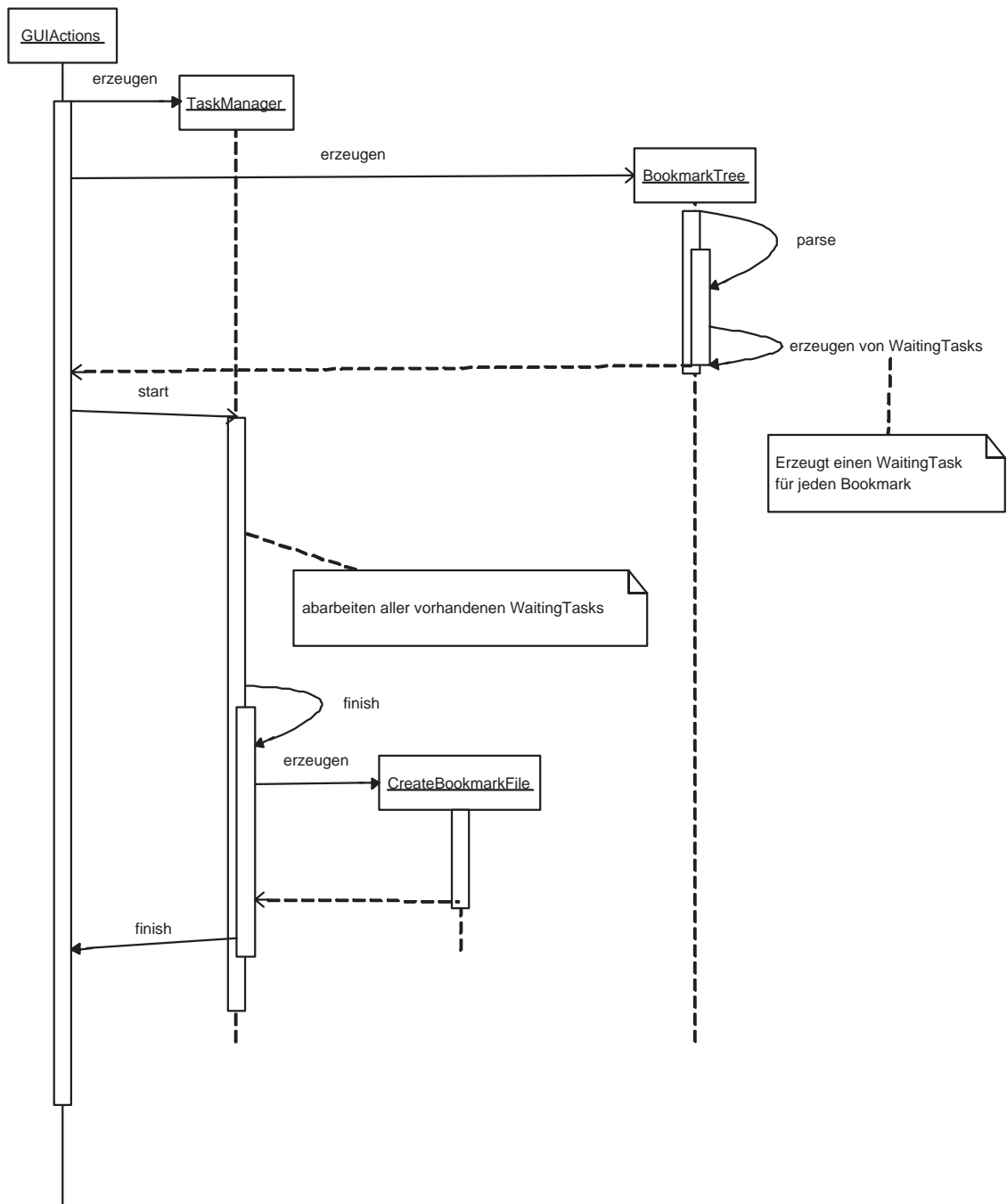
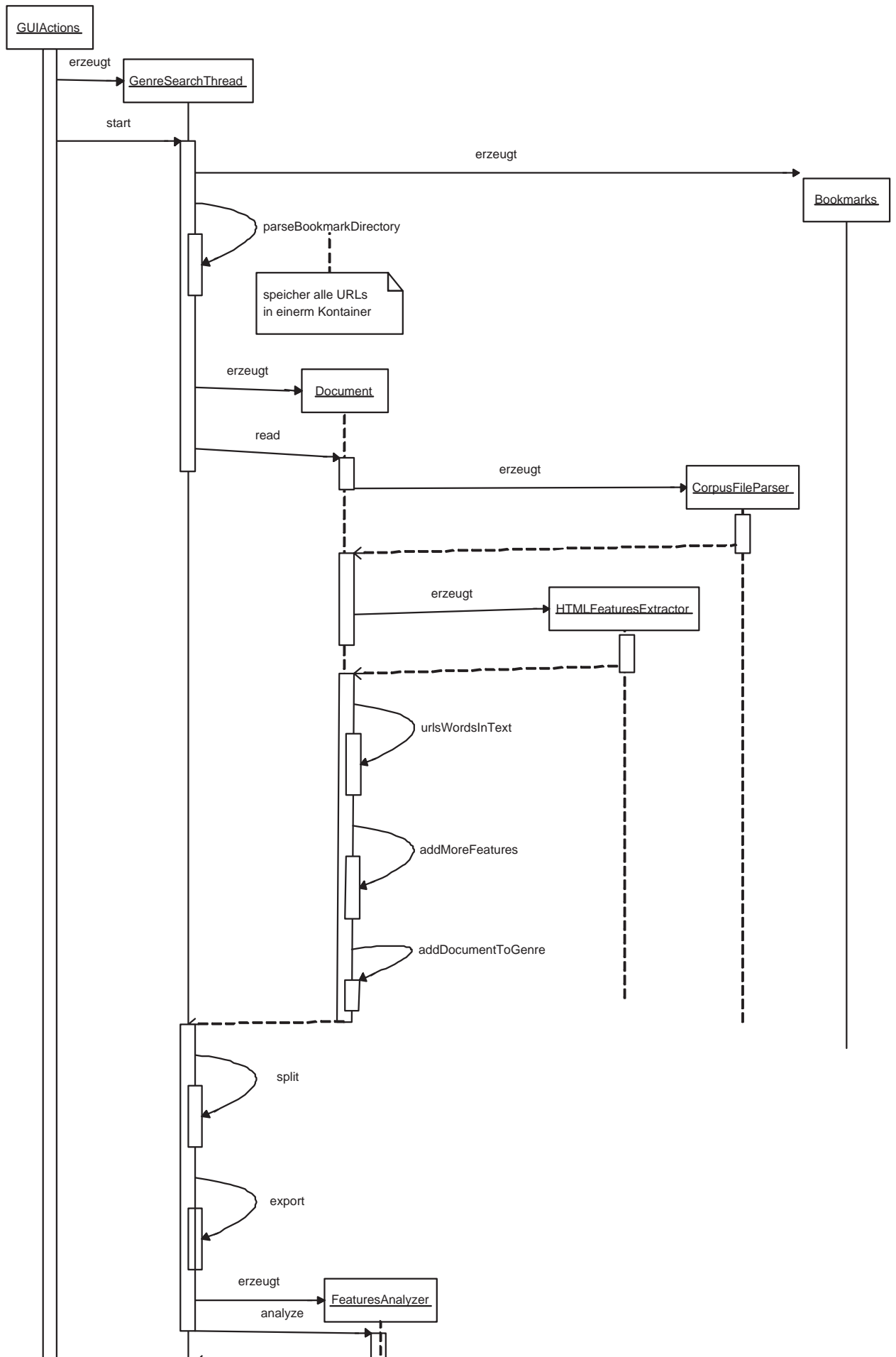


Abbildung 11: Sequenzdiagramm der Funktion Update Corpus.



	portrait non-priv	linklists	shop	discussion	portrait priv	articles	download	help
portrait non-priv	12%	8%	7%	8%	17%	10%	10%	25%
linklists	6%	22%	4%	11%	23%	16%	6%	11%
shop	12%	3%	50%	5%	3%	3%	8%	13%
discussion	2%	6%	4%	54%	4%	10%	3%	16%
portrait priv	2%	2%	0%	3%	50%	27%	7%	8%
articles	0%	1%	1%	3%	2%	85%	2%	3%
download	3%	1%	6%	18%	1%	4%	39%	25%
help	1%	2%	7%	9%	5%	34%	2%	37%

Tabelle 6: Konfusionsmatrix der Klassifikation aller 8 Genres mit NaiveBayes aus dem Weka-Projekt. Korrekt klassifiziert wurden: ca. 43 %

6 Anhang

6.1 Bookmark-Datei

```

<!DOCTYPE NETSCAPE-Bookmark-file-1>
<!-- This is an automatically generated file.
It will be read and overwritten.
Do Not Edit! -->
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=UTF-8">
<TITLE>Bookmarks</TITLE>
<H1>Bookmarks</H1>

<DL><p>
  <DT><H3 ADD_DATE="961099767" LAST_MODIFIED="1065351566"
    ID="NC:BookmarksRoot#$30de080c">genres</H3>
  <DL><p>
    <DT><H3 ADD_DATE="961099767" LAST_MODIFIED="1065351566"
      ID="NC:BookmarksRoot#$aaaaaaa4">help</H3>
    <DL><p>
      <DT><A HREF="file:/corpus/6245781037.html"
        ADD_DATE="1042812142" LAST_VISIT="1060843284"
        LAST_MODIFIED="1042812142" LAST_CHARSET="ISO-8859-1">
        FAQ : HURRICANES, TYPHOONS, AND TROPICAL CYCLONES</A>
      <DT><A HREF="file:/corpus/5786304263.html"
        ADD_DATE="1042812142" LAST_VISIT="1060843284"
        LAST_MODIFIED="1042812142" LAST_CHARSET="ISO-8859-1">
        Address Munging FAQ: Spam-Blocking Your Email Address</A>
      <DT><A HREF="file:/corpus/3188215727.html"
        ADD_DATE="1042812142" LAST_VISIT="1060843284"
        LAST_MODIFIED="1042812142" LAST_CHARSET="ISO-8859-1">
        Sculpture Community - Welcome Center</A>
      <DT><A HREF="file:/corpus/7474651365.html"
        ADD_DATE="1042812142" LAST_VISIT="1060843284"
        LAST_MODIFIED="1042812142" LAST_CHARSET="ISO-8859-1">
        Frequently Asked Questions about CalendarsVersion 2.6</A>
      <DT><A HREF="file:/corpus/5863176369.html"
        ADD_DATE="1042812142" LAST_VISIT="1060843284"
        LAST_MODIFIED="1042812142" LAST_CHARSET="ISO-8859-1">
        Java Programmers FAQ</A>
      <DT><A HREF="file:/corpus/2722264111.html"
        ADD_DATE="1042812142" LAST_VISIT="1060843284"
        LAST_MODIFIED="1042812142" LAST_CHARSET="ISO-8859-1">
        Trusted Computing FAQ TCPA / Palladium / NGCSB / TCG</A>
    </DL><p>
  </DL><p>

```

```

<DT><H3 ADD_DATE="961099767" LAST_MODIFIED="1065351566"
ID="NC:BookmarksRoot#$aaaaaaa1">portrait-priv</H3>
<DL><p>
  <DT><A HREF="file:/corpus/2662342285.html"
  ADD_DATE="1042812142" LAST_VISIT="1060843284"
  LAST_MODIFIED="1042812142" LAST_CHARSET="ISO-8859-1">
  Home page for Brian Foley</A>
  <DT><A HREF="file:/corpus/9071834445.html"
  ADD_DATE="1042812142" LAST_VISIT="1060843284"
  LAST_MODIFIED="1042812142" LAST_CHARSET="ISO-8859-1">
  Eleni's homepage</A>
  <DT><A HREF="file:/corpus/5714525852.html"
  ADD_DATE="1042812142" LAST_VISIT="1060843284"
  LAST_MODIFIED="1042812142" LAST_CHARSET="ISO-8859-1">
  Pierre L'Ecuyer</A>
  <DT><A HREF="file:/corpus/8768863333.html"
  ADD_DATE="1042812142" LAST_VISIT="1060843284"
  LAST_MODIFIED="1042812142" LAST_CHARSET="ISO-8859-1">
  home</A>
  <DT><A HREF="file:/corpus/5534268198.html"
  ADD_DATE="1042812142" LAST_VISIT="1060843284"
  LAST_MODIFIED="1042812142" LAST_CHARSET="ISO-8859-1">
  Ottmar Loos</A>
  <DT><A HREF="file:/corpus/6486327193.html"
  ADD_DATE="1042812142" LAST_VISIT="1060843284"
  LAST_MODIFIED="1042812142" LAST_CHARSET="ISO-8859-1">
  Homepage of Erwin Marsi: Top</A>
</DL><p>
<DT><H3 ADD_DATE="961099767" LAST_MODIFIED="1065351566"
ID="NC:BookmarksRoot#$aaaaaaa5">linklists</H3>
<DL><p>
  <DT><A HREF="file:/corpus/0554067655.html"
  ADD_DATE="1042812142" LAST_VISIT="1060843284"
  LAST_MODIFIED="1042812142" LAST_CHARSET="ISO-8859-1">
  ART HISTORY RESOURCES ON THE WEB: Contents</A>
  <DT><A HREF="file:/corpus/0155123111.html"
  ADD_DATE="1042812142" LAST_VISIT="1060843284"
  LAST_MODIFIED="1042812142" LAST_CHARSET="ISO-8859-1">
  City of Toronto: Toronto links</A>
  <DT><A HREF="file:/corpus/2734888891.html"
  ADD_DATE="1042812142" LAST_VISIT="1060843284"
  LAST_MODIFIED="1042812142" LAST_CHARSET="ISO-8859-1">
  DeskRef: Sources for Quick Answers</A>
  <DT><A HREF="file:/corpus/3061828585.html"

```

```

        ADD_DATE="1042812142" LAST_VISIT="1060843284"
        LAST_MODIFIED="1042812142" LAST_CHARSET="ISO-8859-1">
        LinkOut Journals by Title</A>
    </DL><p>
</DL><p>
</DL><p>

```

6.2 Korpusdatei

```

<!-- <DOCUMENT>
    <FILE>
        /corpus/2420249768.html
    </FILE>
    <URL>
        http://astro.uwaterloo.ca/~lcparker/main.html
    </URL>
    <TITLE>
        LCP Private Homepage
    </TITLE>
    <GENRE>
        portrait-priv
    </GENRE>
    <CONTENT>
-->
<HTML>
<HEAD> <TITLE>LCP Private Homepage</TITLE>
        <LINK REV="made" HREF="mailto:lcparker@astro.uwaterloo.ca">
</HEAD>
<BODY BGCOLOR="WHITE">

<H2>

<BR>
    Laura Parker<BR></H2>
    Department of Physics <BR>
    University of Waterloo <BR>
    Waterloo, Ontario <BR>
    Canada,          N2L 3G1 <BR>
    (519) 888-4567 x5130 <BR>
    Email: <A HREF="mailto:lcparker@astro.uwaterloo.ca">
    lcparker@astro.uwaterloo.ca </A><BR>

<BR>

```

<HR>

</BODY>

</HTML>

6.3 Feature-Datei einer Korpusdatei

```
corpusFilename /corpus/1402113211.html
genre portrait-non_priv
genreId 0
url http://www.ea.com/home/home.jsp
chars 3381
digits 39
letters 1142
capitals 217
questionMarks 1
dots 22
semicolon 11
comma 5
colon 24
exclamation 3
wordCount 252
rare 68
surname 6
firstName 13
misspellings 0
wordClassSum 2473
currency 0
dayOrMonth 6
numeral 10
help 8
shop 8
country 0
ignoreCount 0
download 3
discussion 1
verb 26
noun 117
adjective 15
adverb 6
article 10
pronoun 6
to 5
alphanumeric 10
```

modal 3
preposition 11
copula 7
relativePronoun 0
nrOfPs 0
nrOfBRs 11
nrOfTDs 148
nrOfTRs 70
nrOfFramesets 0
nrOfHrefs_ANCHOR_LINK 1
nrOfHrefs_DOMAIN_LINK 23
nrOfHrefs_INTERNET_LINK 7
nrOfHrefs_JAVASCRIPT_LINK 19
nrOfHrefs_MAILTO_LINK 0
nrOfHrefs_OTHER_LINK 0
nrOfUls 0
nrOfOls 0
nrOfLis 2
nrOfImgs 110
nrOfForms 0
nrOfSelects 0
nrOfInputs 0
nrOfHeadlines 0

Literatur

- [1] Amir Ayazi. Hidden-markov-modelle und die anwendung auf trigramm-modelle und sprachteil-markierung, 1999.
- [2] Ivan Bretan, Johan Dewe, Anders Hallberg, and Niklas Wolkert. Web-specific genre visualization.
- [3] Eric Brill. Some advances in transformation-based part of speech tagging. In *National Conference on Artificial Intelligence*, pages 722–727, 1994.
- [4] K. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the nd A CL Conference on Applied NLP. ACL*, 1988.
- [5] Kevin Crowston and Marie Williams. The effects of linking on genres of web documents. In *HICSS*, 1999.
- [6] Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A practical part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, 1992.
- [7] Nigel Dewdney, Carol VanEss-Dykema, and Richard MacMillan. The form is the substance: Classification of genres in text.
- [8] M. Dimitrova, A. Finn, N. Kushmerick, and B. Smyth. Web genre visualization, 2002.
- [9] A. Finn. Machine learning for genre classification, 2002.
- [10] Aidan Finn and Nicholas Kushmerick. Learning to classify documents according to genre. In *IJCAI-03 Workshop on Computational Approaches to Style Analysis and Synthesis*, 2003.
- [11] Aidan Finn, Nicholas Kushmerick, and Barry Smyth. Genre classification and domain transfer for information filtering. In *Proceedings of the 24th BCS-IRSG European Colloquium on IR Research*, pages 353–362. Springer-Verlag, 2002.
- [12] Eibe Frank Ian H. Witten. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, October 1999.
- [13] Susanne Hackmack Karl Heinz Wagner. Grundkurs sprachwissenschaft, 1998.
- [14] Jussi Karlgren and Douglass Cutting. Recognizing text genres with simple metrics using discriminant analysis. In *Proceedings of the 15th. International Conference on Computational Linguistics (COLING 94)*, volume II, pages 1071 – 1075, Kyoto, Japan, 1994.

- [15] Brett Kessler, Geoffrey Nunberg, and Hinrich Schütze. Automatic detection of text genre. In Philip R. Cohen and Wolfgang Wahlster, editors, *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 32–38, Somerset, New Jersey, 1997. Association for Computational Linguistics.
- [16] Judith Klavans and Min-Yen Kan. Role of verbs in document analysis. In *COLING-ACL*, pages 680–686, 1998.
- [17] Yong-Bae Lee and Sung Hyon Myaeng. Text genre classification with genre-revealing and subject-revealing features. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 145–150. ACM Press, 2002.
- [18] Stefan Münz. Selfhtml: Version 8.0, 2001.
- [19] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
- [20] J. Ross Quinlan. *C4.5: PROGRAMS FOR MACHINE LEARNING*. Morgan Kaufmann Publishers, Inc., 1993.
- [21] Andreas Rauber and Alexander Mglér. Integrating automatic genre analysis into digital libraries. In *ACM/IEEE Joint Conference on Digital Libraries*, pages 1–10, 2001.
- [22] Helmut Schmid. Probabilistic part-of-speech tagging using decision trees, 1994.
- [23] Helmut Schmid. Improvements in part-of-speech tagging with an application to german, 1995.
- [24] Marcus Skowronek. Entwurf, realisierung und evaluierung von linguistischen suchprädikaten für hyrex, 2002.
- [25] E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Text genre detection using common word frequencies.
- [26] Yalin Wang and Jianying Hu. A machine learning based approach for table detection on the web. In *Proceedings of the eleventh international conference on World Wide Web*, pages 242–250. ACM Press, 2002.
- [27] Takeshi Yoshioka and George Herman. Coordinating information using genres, 2000.
- [28] Takeshi Yoshioka, George Herman, JoAnne Yates, and Wanda J. Orlikowski. Genre taxonomy: A knowledge repository of communicative actions. *Information Systems*, 19(4):431–456, 2001.

- [29] Andreas Zell. SnnS stuttgart neural network simulator, usermanual, version 4.2.
- [30] Sven Meyer zu Eissen und Benno Stein. Genre classification of web pages with parsimonious feature sets, 2003.

Erklärung

Ich versichere, daß ich die vorliegende Arbeit selbständig angefertigt und keine anderen als die angegebenen und bei Zitaten kenntlich gemachten Quellen und Hilfsmittel benutzt habe. Diese Arbeit lag in gleicher oder ähnlicher Weise noch keiner Prüfungsbehörde vor und wurde bisher noch nicht veröffentlicht.

Paderborn, 12. Februar 2004

(Roman Deimann)