Bauhaus-Universität Weimar
Faktultät Medien
Web Technology and Information Systems

Diplomarbeit

Genre-Analyse von Web-Dokumenten

Andrea Lahn

22. November 2006

Betreuer:
Sven Meyer zu Eißen
Prof. Dr. Benno Stein (Erstgutachter)
Prof. Dr. Tom Gross (Zweitgutachter)

Danksagung:

Ich möchte mich an dieser Stelle bei Herrn Prof. Dr. Benno Stein für die Überlassung dieses Diplomarbeitsthemas bedanken. Ein großer Dank geht auch an meinen wissenschaftlichen Betreuer Herrn Dipl.-Inf. Sven Meyer zu Eißen, der mir durch seine Unterstützung diese Arbeit erst ermöglichte und mir stets beratend sowie motivierend zur Seite stand.

Erklärung:

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe. Die Arbeit wurde in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Bonn, den 22. November 2006

Andrea Lahn

Zusammenfassung

Internet-Suchmaschinen bieten die Möglichkeit, über die Eingabe von Schlüsselwörtern Informationen im World Wide Web zu finden. Bei dieser Form der Suche werden nur Seiten angezeigt, die den eingegebenen Suchbegriff enthalten. Der Benutzer hat jedoch nicht nur genaue Vorstellungen über das gewünschte Thema einer Seite, sondern auch darüber, welche Form das Ergebnis haben soll. Mit der Form ist dabei der Inhalt eines Dokumentes gemeint. Eine Internet-Seite kann zu einem Thema beispielsweise einen langen technischen Artikel, kurze und präzise Antworten auf spezielle Fragen oder sehr viele Links zu anderen Seiten mit dem gleichen Thema enthalten. Anhand der Einträge, die auf der Ergebnisseite einer Suchanfrage stehen, ist jedoch schwer erkennbar, ob eine Seite für den Suchenden relevant ist oder nicht.

Das so genannte Genre gruppiert Dokumente aufgrund der Form sowie des Stils einer Seite und ist unabhängig vom Thema. Die Angabe des Genres kann dem Suchenden die Entscheidung, ob eine von der Suchmaschine gefundene Seite der gewünschten Form entspricht, erleichtern. Die Zielstellung dieser Diplomarbeit ist es, das Genre eines beliebigen Web-Dokumentes zu bestimmen. Die besondere Herausforderung liegt darin, Eigenschaften zu finden, welche die einzelnen Genres zuverlässig voneinander trennen. Da die Klassifikation in Echtzeit erfolgen soll, müssen diese Eigenschaften möglichst einfach zu bestimmen sein, um einen hohen rechnerischen und somit zeitlichen Aufwand zu vermeiden.

Für die Genre-Analyse wird eine große Anzahl von Beispieldokumenten gesammelt und per Hand den verschiedenen Genres zugeordnet. Aus dieser Trainingsmenge werden für jedes Dokument vorher definierte Eigenschaften, die Features, extrahiert. Anschließend wird mithilfe der Diskriminanzanalyse eine Evaluation durchgeführt, um die Güte unterschiedlicher Feature-Kombinationen zu bestimmen. Um die Zusammenstellung und Bestimmung dieser Kombinationen zu unterstützen, wird eine geeignete Software entwickelt. Mit dem gewonnenen Wissen können dann für die Klassifikation geeignete Eigenschaften selektiert werden. Mit diesen Features und der Diskriminanzanalyse ist das Vorhersagen eines Genres neuer Web-Dokumente möglich.

Um nun den Benutzer einer Suchmaschine, wie oben beschrieben, bei der Entschei-

dung, ob ein Eintrag in der Ergebnisliste für ihn relevant ist oder nicht, zu unterstützen, wird zu jedem dieser Einträge in Echtzeit das zugehörige Genre bestimmt und angezeigt. Damit jeder diese Genre-Analyse verwenden kann, wird für den Firefox-Browser eine Erweiterung implementiert. Diese erkennt automatisch das Starten einer Google-Suchanfrage. Wurde eine solche Anfrage gestartet, filtert die Erweiterung aus dem Ergebnis der Suche die gefundenen Seiten heraus und lässt sie klassifizieren. Das ermittelte Genre wird dann in der Ergebnisseite unmittelbar hinter jedem einzelnen Eintrag eingeblendet und kann direkt abgelesen werden.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1. Einleitung

1989 begann Tim Berners-Lee in einem Projekt am CERN, der Europäischen Organisation für Kernforschung in Genf, die Entwicklung des World Wide Web (kurz: Web), um einen Internet-basierten weltweiten Informationsaustausch zu ermöglichen [54]. Die Grundlage dafür bieten so genannte Hypertext-Dokumente, Textdokumente, die zusätzlich zum Inhalt Links zu anderen Dokumenten enthalten, und Server, welche auf Anfrage Hypertext-Dokumente liefern. Zunächst konnte nur reiner Text angezeigt werden und die Bedienung war nur über Kommandozeile möglich [53]. Die Entwicklung von einfach benutzbaren Browsern und die Möglichkeit, verschiedenste Dateiformate verarbeiten und darstellen zu können, machte das Web auch für Nichtwissenschaftler zugänglich.

Das Potenzial dieses Online-Netzwerkes wurde frühzeitig erkannt. Neben wissenschaftlichen Artikeln findet man unter anderem zahlreiche Firmenauftritte, Vereine, Verkaufs- oder Diskussionsangebote. Durch die immer noch enorm ansteigende Anzahl von Web-Servern und die somit wachsende Zahl von Web-Dokumenten [34] wird das World Wide Web immer unübersichtlicher. Bereits 1991 wurde der erste Vorläufer einer Suchmaschine, Gopher, entwickelt, um die gezielte Suche nach Informationen zu unterstützen, ohne den Namen oder den Ort eines Dokumentes wissen zu müssen. Heutige Suchmaschinen liefern innerhalb kürzester Zeit umfangreiche Suchergebnisse aus dem weltweiten Netz.

Das Web ermöglicht den Zugang zu verschiedensten Informationen und ist somit für viele Menschen zu einem unerlässlichen Werkzeug für die Informationsgewinnung geworden. Obwohl die Suchmaschinen dabei sehr hilfreich sind, liefern sie trotz Einschränkungsmöglichkeiten bei der Suchanfrage auch Seiten, die für den jeweiligen Nutzer momentan nicht von Interesse sind. Namenhafte Suchmaschinenbetreiber wie Google bieten deshalb beispielsweise mit den *Google Directories* [17] die Möglichkeit, in thematisch sortierten Verzeichnissen zu suchen. Die Web-Dokumente wurden hierbei ihrem Inhalt entsprechend zuvor unterschiedlichen Kategorien zugeordnet. Diese Kategorisierung geschieht jedoch manuell, was bei der hohen Dynamik und Wachstumsrate der verfügbaren Web-Inhalte einen erheblichen Aufwand bedeutet.

Web-Dokumente können nicht nur nach dem Thema, sondern auch nach dem Genre

gruppiert werden. Die Genres unterscheiden Dokumente durch deren Form und Typ. Die Form beschreibt die Art eines Inhaltes. Eine Seite kann zum Beispiel einen langen Bericht oder Diskussionsbeiträge enthalten. Auch die Verwendung von Wortarten und Satzzeichen wird betrachtet. Auf Seiten mit Linksammlungen finden sich beispielsweise kaum vollständige Sätze. Der Typ beschreibt die Präsentation des Inhaltes. Die Genres sind damit unabhängig vom Thema eines Dokumentes.

Gäbe es die Möglichkeit, Internet-Seiten automatisch nach Genres zu indizieren, könnte der Suchkomfort nochmals deutlich erhöht werden. Die Aufgabe würde der eines Spam-Filters für E-Mails ähneln: Ein Spam-Filter soll eintreffende Dokumente analysieren und entscheiden, ob es sich bei einer Nachricht um Spam handelt oder nicht. Der Empfänger muss nun nicht mehr selbst jede einzelne E-Mail öffnen und sortieren, was bei der steigenden Anzahl dieser unerwünschten Post einen wichtigen Zeitvorteil bringt. Ein ähnliches Verfahren könnte die Ergebnisse einer Suchmaschine in Echtzeit klassifizieren. Das ermittelte Genre kann dann zusätzlich zum bisherigen Eintrag angezeigt werden.

Der Benutzer einer Suchmaschine weiß genau, zu welchem Thema er Resultate möchte und welche Form diese haben sollen. Jedoch ist das Bewerten eines Ergebnisses einer Suchanfrage allein anhand des Eintrags in der Liste sehr schwer. Viele Benutzer gehen die Ergebnisse der Reihe nach durch, um zum Ziel zu gelangen.

Ein Programm, welches die einzelnen Seiten der Ergebnisliste automatisch im Hintergrund analysiert und das zugehörige Genre bestimmt, könnte den Wert einer Seite für einen Benutzer verdeutlichen. Die vorhergesagten Genres könnten zusätzlich hinter jedem Link angezeigt werden und damit die Entscheidung, ob das Ergebnis interessant ist oder nicht, unterstützen. Der Suchende kann das Aufrufen für ihn irrelevanter Seiten minimieren und spart Zeit und Nerven.

Gegenstand dieser Diplomarbeit ist es, die automatische Klassifikation von Web-Dokumenten zu verbessern und eine Genre-Analyse in Echtzeit zu ermöglichen. Dafür werden unterschiedliche Eigenschaften von Internet-Seiten untersucht und die, welche für die Klassifizierung geeignet sind, ausgewählt. Anhand dieser Eigenschaften werden die Web-Dokumente mittels der Diskriminanzanalyse klassifiziert. Es wird ein Programm entwickelt, welches ausgewählte Eigenschaften von Internet-Seiten extrahiert und in verschiedenen Formaten exportiert.

Um, wie oben beschrieben, den Benutzer bei der Internet-Suche zu unterstützen, wird eine Firefox-Erweiterung implementiert. Startet ein Benutzer eine Google-Suchanfrage, wird dies von der Erweiterung erkannt. Sie extrahiert die URLs der Einträge in der

Ergebnisseite und ruft ein Servlet auf, welches die zugehörigen Web-Dokumente klassifiziert. Die Firefox-Erweiterung verändert dann die Google-Ergebnisseite, indem sie hinter jedem von der Suchmaschine gefundenen Eintrag das ermittelte Genre anzeigt.

2. Der Begriff Genre

Der Begriff Genre kommt aus der französischen Sprache und bedeutet Gattung, Art oder Sorte. Bei Webster wird er geführt als Kategorie in der Kunst, der Musik oder der Literatur, um deren Werke aufgrund von Stil, Form oder Inhalt zusammenzufassen [16]. So gibt es in der Musik beispielsweise die Richtungen Jazz, Rock und Klassik und in der Literatur Romane, Novellen oder Prosa.

Bei der Klassifikation von Texten wird jedem Dokument eine Kategorie zugeordnet. Laut Santini lässt sie sich bezüglich des diskriminierenden Merkmals in folgende drei Arten unterteilen:

- Thema (text categorization, information filtering)
- Autor (authorship attribution)
- Genre (genre identification/detection/classification/categorization)

Alle drei Klassifikationsarten haben das gleiche Ziel, Texte zu gruppieren, aber sie unterscheiden sich anhand der Eigenschaften, die zur Erkennung der Kategorie verwendet werden [39].

In dem folgenden Abschnitt wird der Begriff Genre in Bezug auf die Klassifikation von Dokumenten näher erläutert. Anschließend wird die Genre-Analyse von Web-Dokumenten und deren verschiedenen Ansätze dargestellt.

2.1. Genres digitaler Dokumente

Es gibt viele Bezeichnungen für den „Typ eines Dokumentes“, der automatisch festgestellt werden soll, wie beispielsweise „Genre“, „Texttyp“, „Stil“, „stilistisches Genre“ und „funktionale Rolle“ [40].

Ebenso ungenau sind die Definitionen für den Begriff Genre. Dewdney et al. definieren das Genre eines Dokumentes als Kennzeichnung der Art, wie die Information dargeboten wird. So beinhaltet es das Format und den Sprachstil eines Textes [8]. Laut Finn und Kushmerick beschreibt das Genre den Typ eines Dokumentes und nicht das

Thema [12; 13]. Santini gruppiert Dokumente mit gleichen lexikalischen, syntaktischen und Layout-Features verbunden mit den gleichen kommunikativen Absichten in ein Genre [38].

Trotz der etwas unterschiedlichen Definitionen zeichnet sich ab, dass es sich beim Genre nicht um das Thema, sondern vielmehr um den Aufbau und den Typ eines Dokumentes handelt. Es ist somit orthogonal zur Thematik eines Textes. Das bedeutet, Texte mit verschiedenen Themen können das gleiche Genre haben und Texte mit dem gleichen Thema unterschiedlichen Genres angehören. Baayen et al. fanden heraus, dass sich die Texte eines Autors in unterschiedlichen Genres mehr unterscheiden als die Texte unterschiedlicher Autoren im gleichen Genre [2].

Die einzelnen Genres, die einer Klassifikation zugrunde liegen, können sehr unterschiedlich sein. So wurden Texte in allgemeine Kategorien wie:

- informativ vs. fiktiv (Karlgren und Cutting [23])
- erzählend oder nicht erzählend (Kessler et al. [25])
- positiv oder negativ (Finn und Kushmerik [12; 13])
- objektiv oder subjektiv (Finn und Kushmerik [12; 13])

geteilt. Oft sind sie aus der „Papierwelt“ entnommen [41], wie beispielsweise:

- Presse (Reportagen, Kommentare, Rezension), Vermischtes, Fakten, Fiktion (Karlgren und Cutting [23])
- Reportage, Kommentar, wissenschaftliche und technische Artikel, Artikel über Gerichte und Regierung, Fiktion, Nicht-Fiktion (Kessler et al. [25])
- Kommentar, Brief an den Editor, Reportage, Sportbericht (Stamatatos et al. [45])

Zusätzlich finden sich bei der Klassifikation von Web-Dokumenten auch Genres, die, wie beispielsweise die private Homepage, kein Äquivalent in der Papierwelt haben. Zum Beispiel wurden die folgenden Web-Genres verwendet:

- Inserat, schwarzes Brett, FAQ[1], Forum, Radionachrichten, Reuters Nachrichtennotiz, Fernsehnachrichten (Dewdney et al. [8])
- Artikel, Diskussion, Download, Hilfe, Linksammlungen, privates Portrait, nicht privates Portrait, Shop (Meyer zu Eißen und Stein [30])

[1] FAQ: Frequently Asked Questions, eine Rubrik in der häufig gestellte Fragen aufgelistet und beantwortet werden.

- Private Homepage, Firmen-Homepage, Homepage von Vereinen (Kennedy und Shepherd [24])

In dieser Arbeit bezieht sich das Genre auf die Form und den Typ eines HTML-Dokumentes. Die Form betrachtet den Inhalt des Dokumentes und der Typ die Art und Weise der Präsentation des Inhaltes. So kann ein Thema beispielsweise in einem wissenschaftlichen Artikel erläutert oder in einem Forum diskutiert werden. Auch gibt es Seiten mit sehr vielen Links, die damit Dokumente mit gleichen Themen verbinden.

2.2. Genres im Web

Wie bereits einleitend erläutert, ist das Web noch sehr jung und weist durch die Verwendung von Hypertext-Dokumenten eine neuartige Struktur auf. Die einzelnen Dokumente sind über Links miteinander verbunden und eine Internet-Seite kann die verschiedensten Inhalte darstellen. Daraus folgt, dass die Frage, was gehört zu einem Genre, überdacht werden muss. Es ergeben sich folgende drei Möglichkeiten:

- Ein HTML-Dokument kann mehrere Genres enthalten. Santini beispielsweise sagt, dass sich Genres nicht gegenseitig ausschließen und mehrere Genres in einem Dokument miteinander verbunden sein können [43].

- Ein HTML-Dokument hat genau ein Genre. Dieser Ansatz entspricht der klassischen Verfahrensweise und wurde am meisten angewandt (Dewdney et al. [8], Meyer zu Eißen und Stein [30], Kennedy und Shepherd [24]).

- Alle Dokumente, die durch Hyperlinks verbunden sind, gehören zu einem Genre. Laut Rehm sollte die Menge der Dokumente, die miteinander verbunden sind, zur Identifikation des Genres „Akademische Mitarbeiter-Homepage“ genutzt werden [36].

Aber nicht nur der Zusammenhang zwischen Genre und Dokument, sondern auch die Abgrenzung der Genre untereinander wird hinterfragt. Klassischerweise gehört ein Dokument nur einem Genre an und die Genres untereinander überschneiden sich nicht. Dimitrova et al. hingegen bewerten Dokumente mithilfe von drei Genre-Dimensionen. Sie beurteilen ein Web-Dokument nach dem Grad der Sachkenntnis, der für das Verständnis des Inhaltes erforderlich ist, dem Grad der Ausführlichkeit (kurze Zusammenfassung oder detaillierte Beschreibung) und dem Grad der Subjektivität [9].
Roussinov et al. stellten fest, dass unterschiedliche Nutzer unter ein und demselben Genre nicht zwangsläufig exakt das Gleiche verstehen und empfehlen deshalb eine unscharfe Trennung der Genre-Definitionen [37].
Rehm schlägt eine hierarchische Anordnung der Web-Genres vor. Er nimmt an, dass es

allgemeine Web-Genre-Typen gibt, welche die Grundlage für ein Web-Genre bilden [36].

Da in dieser Arbeit die automatische Genre-Klassifikation für die Suche im Internet eingesetzt werden soll, wird der Ansatz von Meyer zu Eißen und Stein weiterverfolgt. Das bedeutet, ein einzelnes HTML-Dokument gehört zu einem Genre und die einzelnen Genres sind klar voneinander abgetrennt.

Mit dem World Wide Web stehen sehr viele unterschiedliche Dokumente für die Suche von Informationen zur Verfügung, deren Zahl stetig weiter wächst. Meyer zu Eißen und Stein prüften anhand einer Benutzerstudie, ob die automatische Genre-Klassifikation im Rahmen einer Internet-Suche sinnvoll ist. Dabei stellte sich heraus, dass 64 % der Befragten sie als sehr und 29 % als manchmal nützlich einstufen. Da aufgrund der Informationsvielfalt die verschiedensten Genres vorstellbar sind, wurden von ihnen auch diverse Genres vorgeschlagen und die Probanden sollten entscheiden, wie hilfreich diese für sie wären. Bei der Evaluation der Umfrage haben sich folgende Genres als nützlich erwiesen:

- Hilfe: Questions and Answers (Q&A), FAQ, Nachschlagewerke
- Artikel: Dokumente mit längeren Textpassagen
- Diskussion: Foren, Discussion Boards, Mailing-Listen
- Shop: Seiten, die etwas zum Kauf anbieten
- Portrait (nicht privat): Web-Auftritte von Firmen, Universitäten, öffentliche Einrichtungen, Vereinen
- Portrait (privat): Private Seiten einzelner Personen
- Linksammlung: Dokumente, die primär Links auflisten
- Download: Seiten, welche beispielsweise Software, Freeware und Shareware zur Verfügung stellen

Bei der anschließenden Klassifikation erwies sich die Identifizierung der acht Genres als schwierig, da sich mit den ausgewählten Features beispielsweise die Genres *Hilfe*, *Portrait (nicht privat)* und *Linksammlung* nicht deutlich unterscheiden. Um den verschiedenen Benutzer-Typen gerecht zu werden, definieren Meyer zu Eißen und Stein unterschiedliche Profile. Sie fassten einzelne Genres zusammen und reduzierten somit die Anzahl der zu identifizierenden Kategorien, wodurch sich das Ergebnis der Klassifizierung verbesserte [30].

Da sich das Web über die Zeit hinweg stetig verändert und erneuert, kann die Internet-Suche nur dauerhaft unterstützt werden, wenn über längere Zeit hinweg Dokumente zuverlässig klassifiziert werden können. Boese und Howe haben die Dauerhaftigkeit der Klassifizierer von Web-Dokumenten untersucht [5]. Dabei stellten sie fest, dass die Klassifizierungsergebnisse über längere Zeit hinweg stabil bleiben. Ebenso fanden sie heraus, dass Internet-Seiten verschiedener Genres unterschiedlich häufig aktualisiert werden. So verändern sich Artikel kaum, wenn sie einmal ins Web gestellt sind, während auf den Diskussionsseiten regelmäßig neue Einträge hinzukommen. Sie schlagen vor, dieses Wissen für die Verbesserung der Effizienz von Webcrawler[2] der Suchmaschinen zu nutzen, da nicht mehr alle Seiten in kleinen Abständen besucht werden müssten.

Zusammenfassend muss noch hinzugefügt werden, dass bei der Genre-Analyse von Web-Dokumenten der Korpus meist aus HTML-Dokumenten besteht. Somit fallen im Web bereitgestellte Dateien wie beispielsweise PDFs heraus. Auch werden Internet-Seiten mit dynamischem Inhalt wie Flash oder jene, die komplett aus JavaScript heraus erzeugt werden, aufgrund von schwer auswertbarem Inhalt ausgeschlossen. Ebenso werden eingebundene Bilder oder Audio-/Videodateien nicht betrachtet.

[2] Webcrawler: Spiders oder auch Robots, sie durchsuchen das Web und analysieren Internet-Seiten.

3. Genre-Analyse

Bei der Genre-Analyse werden Dokumente aufgrund ihrer Eigenschaften in unterschiedliche Genres eingeordnet. Damit diese Analyse automatisiert werden kann, muss die Vorgehensweise, die ein Mensch beim Einsortieren anwendet, abstrahiert und in der Art umformuliert werden, dass sie von einem Computer durchgeführt werden kann.

Die Grundlage der Genre-Analyse ist eine Sammlung von Beispieldokumenten, der so genannte Korpus. Jedem Dokument aus dem Korpus wird per Hand ein Genre zugeordnet. Um automatisch die Ähnlichkeiten von Beispieldokumenten eines Genres und Unterschiede zwischen den Genres erkennen zu können, werden aus den Dokumenten Eigenschaften, die Features, extrahiert. Ein Klassifizierer kann dann mittels der Feature-Werte trainiert werden. Das bedeutet, über einen Lernalgorithmus wird versucht, anhand der Trainingsdaten eine Funktion zu finden bzw. anzunähern, welche jedem Beispiel die zugehörige Klasse zuweist. Nach Beendigung der Trainingsphase kann der Klassifizierer neue Dokumente unbekannten Genres zuordnen.

Im Folgenden werden die einzelnen Schritte, welche für die Klassifikation von Dokumenten erforderlich sind, näher erläutert.

3.1. Dokumentrepräsentation

Um die Komplexität der Dokumente des Korpus zu reduzieren und um sie einfacher verarbeiten zu können, werden die Beispiele in eine zahlenbasierte Repräsentation überführt. Die klassischen Retrieval-Modelle abstrahieren ein Dokument zu einer Datenstruktur auf Basis von Indextermen, eine Menge von repräsentativen Schlüsselwörtern. Dabei wird zu jedem Wort ein Zahlenwert berechnet [3].

Ein Beispiel für ein solches Modell ist das Boolesche Modell. Hier werden die Indexterme lediglich nach $\{0, 1\}$ abgebildet. Es wird also nur das Vorhandensein von Wörtern überprüft, ohne dass später Aussagen über deren Häufigkeit getroffen werden können. Für die Beschreibung des Inhaltes ist es jedoch ein Unterschied, ob ein Wort nur einmal vorkommt oder im Text eine zentrale Stellung einnimmt.

Beim Vektorraummodell werden die Indexterme gewichtet, damit die Bedeutung eines

Terms für ein Dokument berücksichtigt werden kann. Es wird eine endliche Menge an Termen gegeben und für jedes Dokument des Korpus sei zu jedem Indexterm ein Gewicht gegeben. Bei der Gewichtung hat sich der *tf-idf*-Ansatz bewährt [11]:

- Die Termhäufigkeit (*tf*) bezieht sich auf die Häufigkeit eines Terms in einem Dokument. Es wird davon ausgegangen, dass Terme, die häufig auftreten, bedeutender für den Inhalt sind als die selteneren.
- Auf einen Korpus bezogen, ist ein Term gut zur Beschreibung des Inhaltes geeignet, wenn er nur in wenigen Dokumenten vorkommt. Dieser Sachverhalt wird mit der inversen Dokumenthäufigkeit (*idf*) abgebildet.

Die reellen Zahlenwerte der Indexterme eines Dokumentes werden zu einem Vektor, dem Dokumentvektor, zusammengefasst. Der Vektor repräsentiert ein Dokument im so genannten Vektorraum.

Da nicht alle Wörter gleich bedeutend für die Beschreibung des Inhaltes sind, wird beim Information Retrieval der Korpus vor der Bestimmung der einzelnen Intexterme aufbereitet, um die Anzahl der Indexterme zu verringern. Dabei werden Layout-Informationen, wie HTML-Tags, sowie häufige und gleichverteilte Wörter, die Stoppwörter (siehe Abschnitt 3.2.1), entfernt. Anschließend erfolgt zur Verallgemeinerung der Indexterme eine Stammformreduktion. Diese Reduktion wird auch als Stemming bezeichnet. So zählen beispielsweise die Wörter *compute*, *computation* und *computer* zu einem Indexterm *comput*.

Bei der Genre-Analyse sind jedoch, wie in Kapitel 2 beschrieben, die Informationen über das Layout eines Textes und die Stoppwörter von Bedeutung und dürfen somit nicht entfernt werden. Für die Genre-Analyse werden ebenso Informationen über den Satzbau und die verwendeten Wortarten eines Textes genutzt. Zur Bestimmung dieser Informationen werden ebenfalls ungekürzte Wörter und vollständige Sätze benötigt. Weiterhin werden Texteigenschaften, wie der Gebrauch von Satzzeichen, zur Klassifikation verwendet. Die Modelle berücksichtigen neben den Wortstämmen keine weiteren Textinformationen. Sie sind demnach für die Genre-Analyse nicht geeignet.

Dennoch werden die extrahierten Features bei der Genre-Analyse meist in Vektoren abgebildet. Pro Dokument wird ein Vektor erzeugt, in dem die Features in einer festen Reihenfolge aufgelistet sind (siehe Abbildung 3.1). Somit gleichen sich idealerweise die Dokumentvektoren eines Genres und es können, zum Beispiel über die Euklidische Distanz, Ähnlichkeitsberechnungen durchgeführt werden. Gleichzeitig macht die Vektorform ein Dokument für den Klassifizierer nutzbar.

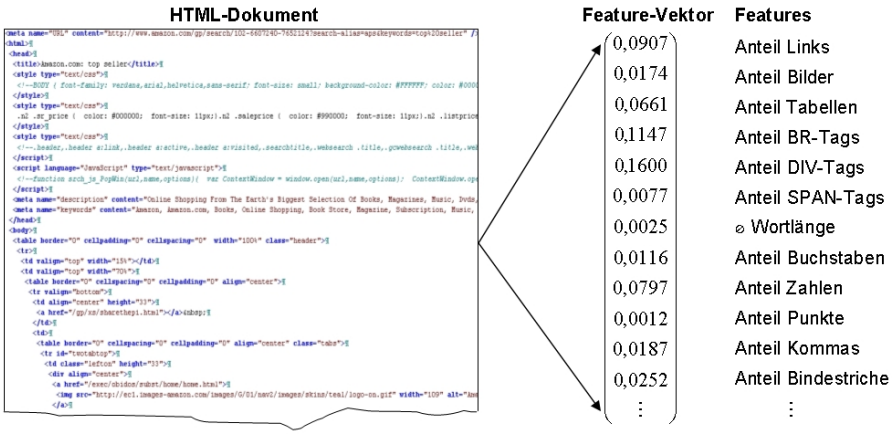

Abbildung 3.1.: Extrahieren von Features aus einem Web-Dokument

3.2. Features zur Genre-Klassifikation

Bei der Genre-Analyse werden Dokumente anhand von Features, Eigenschaften eines Dokumentes, klassifiziert. Eine Feature-Menge, die bei einer Klassifikation extrahiert wird, kann unterschiedliche Arten von Eigenschaften umfassen. Die Wahl der Eigenschaften hängt von den Dokumenten und den zu bestimmenden Klassen ab. Sollen beispielsweise Web-Dokumente in private oder nicht private Portraits aufgeteilt werden, gibt es zahlreiche Unterscheidungsmöglichkeiten. Es kann geprüft werden, ob eine Person oder eine Firma vorgestellt wird, ob ein Lebenslauf oder Geschäftsbedingungen vorhanden sind. Soll nun bestimmt werden, ob es sich bei einem Web-Dokument um einen Online-Shop handelt, sind wiederum andere Eigenschaften zur Erkennung hilfreich.

Damit diese Klassifikation automatisch ablaufen kann, müssen die Eigenschaften eines Dokumentes so definiert werden, dass ein Programm sie extrahieren kann. Der rechnerische Aufwand, der zur Bestimmung benötigt wird, teilt die Eigenschaften in Features mit niedrigem, mittlerem oder höherem Berechnungsaufwand [30]. Da das Ziel dieser Arbeit darin besteht, Features zu finden, die Web-Dokumente nicht nur möglichst gut, sondern vor allem in Echtzeit klassifizieren, werden bei der Evaluation nur Eigenschaften mit niedrigem und mittlerem Berechnungsaufwand untersucht (siehe Abschnitt 4.3).

Im Folgenden werden verschiedene Feature-Klassen, welche bei der Genre-Analyse benutzt werden, erläutert. Dabei werden zunächst die bereits etablierten klassischen Features und anschließend neue Features aufgeführt. Bei den Feature-Klassen, die später bei der Evaluation einbezogen werden, wird zusätzlich erläutert, wie die Feature-Werte bestimmt werden.

3.2.1. Klassische Features

Im Anschluss werden Features beschrieben, die sich als hilfreich erwiesen haben. Zu ihnen gehören die folgenden Klassen:

Textstatistik

Die Textstatistik beruht darauf, dass unter anderem die relative Anzahl der Satzzeichen bei den verschiedenen Genres unterschiedlich ist. Seiten mit Artikeln enthalten viele und vollständige Sätze und somit eine große Anzahl an Kommas und Punkten. In Foren und FAQ-Seiten werden Fragen gestellt, welche meist kurz beantwortet werden. Demnach ist bei diesen Seiten der Anteil der Fragezeichen relativ hoch. Auf Seiten mit Linksammlungen befinden sich kaum Satzzeichen, sondern vorrangig Wörter oder kurze Wortgruppen, die als Links auf die verbundenen Seiten zeigen.

Um diese Unterschiede in Features abzubilden, werden Elemente eines Textes, wie bespielsweise die Anzahl von Buchstaben, Zahlen und einzelner Satzzeichen, gezählt. Die Bedeutung der Wörter oder der inhaltliche Zusammenhang sind hierbei irrelevant, nur die Zeichen an sich werden betrachtet. Die Feature-Werte werden oft ins Verhältnis zur absoluten Wort- oder Zeichenanzahl gesetzt. Auch Berechnungen der durchschnittlichen Wort- oder Satzlänge sind möglich.

Die Features der Textstatistik werden auch „Character-Level Cues“ genannt [25] oder als „hand-crafted“ [12] bezeichnet. Stamatatos et al. [45] zeigten, dass unter Mitbetrachtung von Satzzeichen bei der Klassifikation die Güte deutlich verbessert werden kann.

Dass mit statistischen Features Aussagen über einen Text getroffen werden können, zeigte Flesch mit der Entwicklung einer Lesbarkeitsformel [15]. Er erkannte, dass für einen Menschen zum Verstehen eines Textes unter anderem die Wort- und die Satzlänge bedeutend sind. Je größer deren durchschnittliche Länge ist, desto komplizierter wird ein Text. Die von ihm aufgestellte Formel 3.1 beschreibt diesen Zusammenhang.

$$R = 206.835 - 1.015 \cdot \frac{W}{S} - 84.6 \cdot \frac{T}{W} \tag{3.1}$$

W beschreibt die Anzahl der Wörter, S die Anzahl der Sätze und T die Anzahl der Zeichen eines Textes. Die Konstanten wurden in einer Evaluation ermittelt. Der resultierende Index R reicht von 0 für sehr schwierig bis 100 für sehr einfach zu verstehende Texte.

Da die Textstatistik-Features mit geringem Berechnungsaufwand zu ermitteln sind,

werden sie in die Evaluation dieser Arbeit mit einbezogen. Bestimmt wird hierbei die relative Anzahl von Buchstaben, Großbuchstaben, Zahlen, Apostrophen, Doppelpunkten, Kommas, Punkten, Semikolons sowie Frage- und Ausrufezeichen. Dabei wird die absolute Anzahl eines Zeichens ins Verhältnis zur Anzahl aller Zeichen eines Textes gesetzt.

Die durchschnittliche Wortlänge eines Textes DW wird nach der folgenden Formel bestimmt.

$$\begin{aligned} DW &= \frac{\sum_{i=0}^{n} f_{wl}(w_i)}{n} \\ DWnormiert &= \begin{cases} \frac{DW}{20}, & DW < 20 \\ 1, & DW > 20 \end{cases} \end{aligned} \tag{3.2}$$

n ist die Anzahl der Wörter eines Textes und $f_{wl}(w_i)$ bestimmt die Länge des Wortes w_i. Damit alle Feature-Werte bei der Evaluation im Wertebereich von 0 und 1 liegen, wird die durchschnittliche Wortlänge normiert.

Präsentationsbezogende Eigenschaften

Diese Eigenschaften beschreiben das Layout eines Textes. In HTML-Dokumenten bietet sich die Möglichkeit, die Art und Weise der Textdarstellung über die so genannten HTML-Tags zu analysieren. Neben dem Zählen von Überschriften, Tabellen oder Bildern können auch die Attribute der Tags ausgewertet werden, um Untergruppen, wie beispielsweise Anker- oder Mail-Links zu definieren [30; 24].

Die Ermittlung der Häufigkeiten von HTML-Tags ist mit geringem Berechnungsaufwand verbunden, deshalb werden auch diese Features bei der Evaluation untersucht. Gezählt werden alle gebräuchlichen Tags, wie beispielsweise Anzahl von Bildern, Link-, DIV-, BR- und Tabellen-Tags[1]. Dabei wird für jedes HTML-Tag ein Feature-Wert berechnet. Nur bei dem Feature *Form tags* werden die Tags: <*font*>, <*b*>, <*i*>, <*u*> und <*strong*> zu einem Wert zusammengefasst.

Um die relative Häufigkeit eines Tags in einem Dokument zu bestimmen, wird deren absolute Anzahl ins Verhältnis zur Gesamtanzahl aller Tags gesetzt. Für alle Features, welche den Anteil spezieller Links beschreiben, wie zum Beispiel Ankor-Links, Mail-Links

[1] Eine vollständige Liste der untersuchten Features befindet sich im Anhang B.

oder JavaScript-Links, werden die relativen Häufigkeiten über das Verhältnis zur Anzahl aller Links bestimmt.

Part-of-Speech (POS)

Bei der Part-of-Speech-Analyse werden die im Text verwendeten Wörter in Bezug auf ihre Funktion im Satz oder ihre Wortart untersucht. Sie kann dazu benutzt werden, einen Überblick über den Sprachstil einer Seite, unabhängig vom Thema zu erhalten.

Bevor die Häufigkeit der einzelnen Wortarten ermittelt werden kann, wird der Text eines Dokumentes von einem so genannten POS-Tagger markiert. Dabei wird jedes Wort des Textes mit einer Markierung versehen, welche die Wortart enthält. Wenn die POS-Tags gesetzt sind, können gleiche Markierungen aufsummiert werden. So kann man einen Vektor erzeugen, dessen Komponenten die verwendeten Wortarten quantifizieren.

Für ein einzelnes Wort kann nicht immer eindeutig die Wortart vorhergesagt werden. Beispielsweise kann das englische Wort *run* als Verb und auch als Substantiv verwendet werden. Um mögliche Mehrdeutigkeiten aufzulösen, gibt es zwei Verfahren: Regelbasiert und stochastisch. Bei den regelbasierten Taggern werden aus einem Wörterbuch zunächst für jedes Wort alle POS-Tags herausgesucht, die das Wort losgelöst vom Kontext haben kann. Gibt es für ein Wort mehrere Möglichkeiten, wird über ein Regelsystem das zutreffende Tag ausgewählt. Bei den stochastischen Taggern wird die Auswahl des Tags über Wahrscheinlichkeitsberechnungen bestimmt [4].

Die jeweiligen POS-Tags können einzeln oder auch kombiniert in einen Feature-Wert eingehen. So definierte beispielsweise Santini so genannte POS-Trigramme, jeweils drei aufeinanderfolgende POS-Tags, welche sie für die Genre-Analyse [39; 41] verwendete.

Die Part-of-Speech-Features haben bei den Untersuchungen von Meyer zu Eißen und Stein [30] eine deutliche Verbesserung der Klassfizierungsrate bewirkt. Deshalb werden trotz hohem Berechnungsaufwand auch diese Features in der Evaluation untersucht. Für die Feature-Bestimmung wird lediglich der Text einer Web-Seite zugrunde gelegt. Das bedeutet, alle Zeichen, die sich innerhalb von HTML-Tags befinden, werden ignoriert. Die Wortarten werden mithilfe von QTag [29] ermittelt. Dies ist ein wahrscheinlichkeitsbasierter POS-Tagger von Oliver Mason, der laut seinen Studien sehr robust ist. Gezählt werden Wortarten, zum Beispiel Substantive, Verben und Adjektive (Vollständige Liste im Anhang B). Die relative Anzahl der Wörter einer Wortart WW wird über die folgende Formel bestimmt:

$$
\begin{aligned}
WW &= \frac{\sum_{i=0}^{n} f_{wa}(w_i)}{n} \\
f_{wa}(w_i) &= \begin{cases} 0, & w_i \notin M_{wa} \\ 1, & w_i \in M_{wa} \end{cases}
\end{aligned}
\tag{3.3}
$$

f_{wa} bestimmt, ob ein Wort w_i zu einer Wortart gehört. n ist die Anzahl aller Wörter eines Textes und M_{wa} die Menge der Wörter einer Wortart.

Syntactic-Group-Analysis

Die Syntactic-Group-Analysis bezieht sich auf die einzelnen Wörter und Attribute eines Satzes. Diese Features werden auch als „Linguistic Facets“ bezeichnet. Bei der Analyse wird unter anderem auch die Verwendung von Zeitformen, Relativsätzen und der Gebrauch von Aktiv- und Passivsätzen untersucht. Zur Bestimmung der Eigenschaften wird ein Parser verwendet, der den Text mit Anmerkungen versieht.

Dewdney verwendet beispielsweise als Feature den Wechsel der Zeitformen in einem Text [8] und Santini konstruiert aus den Linguistic Facets einfache Muster, deren Vorkommen im Text gezählt werden [42]. Laut Santini sind diese Features von hohem Informationsgehalt. Da die Berechnung dieser Features sehr aufwendig ist, werden sie nur selten benutzt und auch in der Evaluation, die im Rahmen dieser Arbeit durchgeführt wird, nicht untersucht.

Average-Word-Frequency-Class

Der Average-Word-Frequency-Class liegt zugrunde, dass einzelne Wörter in der Sprache unterschiedlich oft verwendet werden. Dabei sind es eher wenige Worte, die besonders häufig und viele Worte, die sehr selten auftreten. Diese Verteilung entspricht dem Zipf'schen Gesetz und kann durch die Worthäufigkeitsklassen wiedergegeben werden. Die Average-Word-Frequency-Class ist somit ein Indikator für den Wortgebrauch eines Textes.

Es sei C der Text-Korpus, $f(w)$ die Häufigkeit des Wortes $w \in C$ und w^* das häufigste Wort in C, so wird entsprechend [52] die Häufigkeitsklasse $c(w)$ eines Wortes $w \in C$ über die folgende Formel bestimmt:

$$
c(w) = \lfloor log_2(f(w^*)/f(w)) \rfloor \tag{3.4}
$$

In dieser Arbeit wird die Häufigkeitsklasse eines Wortes mithilfe der „Sydney Morning Herald Word Database“[2] von Simon Dennis [7] ermittelt. In der Datenbank befinden sich 97.031 englische Wörter, die unter anderem mit ihrer Häufigkeitsklasse versehen wurden. Es gibt 20 Häufigkeitsklassen, wobei das häufigste Wort „the“ die Klasse 0 hat und die seltensten Wörter zur Klasse 19 gehören.

Für die Evaluation wurden die durchschnittliche, minimale und maximale Worthäufigkeitsklasse eines Textes ermittelt. Damit die Feature-Werte wieder zwischen 0 und 1 liegen, werden die ermittelten Klassen ins Verhältnis zur Anzahl der Häufigkeitsklassen gesetzt.

Stoppwörter

Zu den Stoppwörtern zählen Wörter, die in Texten sehr häufig vorkommen, aber keine thematischen Inhalt haben, wie unter anderem Präpositionen oder Artikel. Da sie nicht zur inhaltlichen Beschreibung dienen, werden sie oft beispielsweise bei der Ähnlichkeitssuche eliminiert (siehe auch Abschnitt 3.1).

Stamatatos et al. [45] zeigen, dass diese Wörter nicht unwichtig sind und bei der Klassifikation von Genres sehr hilfreich sein können. Bei Nigam et al. [35] hatte zum Beispiel das Wort *my* einen der höchsten Werte beim Information Gain[3] und war besonders gut geeignet, private Homepages zu identifizieren.

Closed-Class Word Sets

Manche Wörter oder Zeichen kommen auf bestimmten Internet-Seiten häufiger vor, wie zum Beispiel Datumsangaben in Diskussionsforen oder Währungszeichen bei Shops. Um nicht die Häufigkeiten einzelner Wörter zu zählen, kann man sie in so genannten Closed-Class Word Sets zusammenfassen. Das sind Listen mit Wörtern und Symbolen, die frei definierbar sind. Bei der Feature-Bestimmung wird pro Dokument und Closed-Class Word Set ein Wert bestimmt, beispielsweise die relative Häufigkeit.

Man kann in einer Liste alle Wochentage aufnehmen und bei der Feature-Bestimmung wird jedes Wort eines Textes überprüft, ob es in dieser Liste steht. Ist dies der Fall, wurde ein Wochentag gefunden. Auf diese Weise kann man die Anzahl der Wochentage in einem Text ermitteln. Diese Anzahl kann auch ins Verhältnis zur Wortanzahl gesetzt werden. Ein hoher Anteil von Wochentagen kann beispielsweise auf ein Diskussionsforum

[2] Die „Sydney Morning Herald Word Database“ enthält alle Wörter, die 1994 in den Ausgaben des *Sydney Morning Herald* mindestens in zwei unterschiedlichen Artikeln vorkamen.

[3] Information Gain: Informationsgewinn eines Features bestimmt, wie viel Information beim Weglassen eines Features verloren geht oder gewonnen wird [59].

hindeuten, bei denen das Datum eines Eintrags immer angegeben wird.

Meyer zu Eißen und Stein haben unter anderem Wortlisten mit Vornamen, Nachnamen und Datumsangaben sowie Listen mit Genre-spezifischen Wörtern benutzt [30]. Diese Features werden bei Kessler et al. auch „Lexical Cues“ genannt [25].

Bei der Verwendung von Closed-Class Word Sets ist zu beachten, dass sie je nach Wahl der Wörter sprachabhängig sind und somit die spätere Verwendung des Klassifizierers einschränken.

Auch im Rahmen der Evaluation werden Features verschiedener Closed-Class Word Sets untersucht. Es wird die Hypothese aufgestellt, dass es für jedes Genre spezifisches Vokabular gibt, dessen Verwendung unabhängig vom Thema ist. Beispielsweise finden sich auf Seiten mit Diskussionen häufig die Wörter *forum*, *post* und *reply*, bei Download-Seiten *freeware*, *install* und *mirror*. Für jedes Genre wird eine Wortliste mit englischen Schlüsselwörtern zusammengestellt, die ihrerseits zur Identifikation dieses Genres dienen sollen. Es werden aber auch Wörterbücher untersucht, die unabhängig von den Genres zusammengestellt wurden. Sie enthalten beispielsweise Datumsangaben oder Vornamen.

Pro Wörterbuch erhält man einen Feature-Wert. Dieser gibt an, wie viele Wörter eines Textes in dem jeweiligen Wörterbuch stehen. Dieser Wert wird stets ins Verhältnis zur Wortanzahl des Textes gesetzt.

Um Rechtschreibschwächen oder Akronyme, wie sie in technischen Beschreibungen oft vorkommen, zu erkennen, werden nur die Wörter gezählt, die nicht in einem Wörterbuch stehen. Bei dem in dieser Evaluation verwendeten Wörterbuch handelt es sich um einen Auszug aus Webster's Wörterbuch. Es ist in der entwickelten Software zur Feature-Berechnung unter „Spelling Dictionaries“ zu finden.

3.2.2. Neue Features

Neben den bereits genannten klassischen Features wurden im Rahmen dieser Diplomarbeit neue Eigenschaften berechnet und getestet, die im Folgenden erläutert werden.

Closed-Class Word Sets für Titel, URL und Meta-Tags

Bisher wurde nur der Text eines Web-Dokumentes auf Wörter aus den Closed-Class Word Sets durchsucht. Internet-Seiten stellen aber noch weitere Informationen zur Verfügung. Wenn es sich zum Beispiel um einen Online-Verkauf handelt, steht im Titel der Seite häufig der Begriff *Shop*. Auch besitzt eine Internet-Seite unter Umständen

eine URL, die bei genauerer Betrachtung bereits einen Hinweis auf das Genre enthalten kann. Die URL *http://sneakykitchen.com/**forum**/recipes2.htm* beispielsweise enthält das Wort *Forum*, welches auf eine mögliche Diskussionsseite hindeutet. Ähnlich verhält es sich mit den Meta-Tags, die oftmals in HTML-Dokumenten vorhanden sind.

Bei der Evaluation wird neben dem Text der Titel, die URL und die Meta-Tags separat nach Wörtern aus den Closed-Class Word Sets durchsucht. Das heißt, pro Wörterbuch werden vier Features berechnet. Da es beispielsweise nicht sinnvoll ist, in einer URL nach Datumsangaben oder in einem Titel einer Seite nach HTML-Tags zu suchen, werden die zu verwendenden Wörterbücher für Text, Titel, URL und Meta-Tag getrennt voneinander angegeben.

Window Chunking

Mit den Closed-Class Word Sets kann das Vorkommen von Wörtern im Text erfasst werden. Auf diese Art wird ein Gesamtwert bestimmt. Aus diesem Wert ist nicht erkennbar, ob sich die Wörter eines Wörterbuches über den gesamten Text gleich verteilen oder ob sie sich in einem Bereich konzentrieren. Beispielsweise konzentrieren sich bei einem Internet-Shop die Wörter *offer*, *price* oder *buy* im Hauptinhalt der Seite, während sich an den Rändern Menüs oder Werbung befinden.

Um eine Konzentration von Wörtern zu erkennen, wird der Text mittels Window Chunking in Abschnitte, die so genannten Chunks, unterteilt. Die einzelnen Chunks werden dann nach Wörtern aus den Closed-Class Word Sets durchsucht. Anschließend müssen die Häufigkeiten der einzelnen Abschnitte zusammengefasst werden. In der durchgeführten Evaluation wird das Maximum der Werte als Feature-Wert verwendet. Somit erhält man mithilfe des Window Chunking einen Wert, der die maximale Konzentration von Wörtern in einem Textabschnitt wiedergibt.

Um die Features bestimmen zu können, muss man neben den zu verwendenden Wörterbüchern noch zwei weitere Parameter angeben:

- **Fenstergröße:** Anzahl der Wörter, die zu einem Chunk gehören
- **Sprungweite:** Anzahl der Wörter, um die das Fenster weiter geschoben wird

Die Abbildung 3.2 zeigt Chunk x und seinen Nachfolger $x+1$ mit einer Fenstergröße von 20 Wörtern und einer Sprungweite von 5 Wörtern. Mit dem entwickelten Software zur Feature-Berechnung (siehe Abschnitt 4.2) kann man auf diese Weise den gesamten HTML-Code und den Text eines Dokumentes separat durchsuchen.

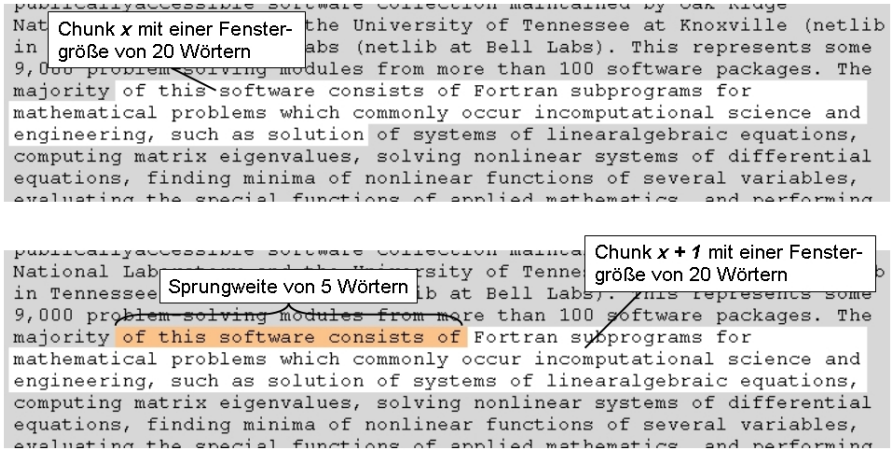

Abbildung 3.2.: Beipiel für einen Chunk x (oben) und $x+1$ (unten) mit einer Fenstergröße von 20 Wörtern und einer Sprungweite von 5 Wörtern

3.3. Klassifikation

Das Ziel der Klassifikation ist es, eine große Menge von Objekten oder Situationen aufgrund von Eigenschaften in unterschiedliche Gruppen einzuteilen. Bei automatisierten Klassifikationsaufgaben kann eine große Anzahl von Eigenschaften der Elemente berücksichtigt werden. Ein solches Verfahren findet beispielsweise bei der Entscheidung über die Vergabe von Krediten Anwendung. Ein Bankmitarbeiter muss lediglich die geforderten Daten eines Kunden angeben und erhält nach der Klassifikation Auskunft, ob ein Kredit für den Kunden bewilligt werden kann.

Bei der Genre-Analyse werden hauptsächlich zwei Klassifikationsverfahren verwendet. Zum einen werden Verfahren des Maschinellen Lernens benutzt und zum anderen findet die Diskriminanzanalyse Anwendung. Beide Methoden werden im Folgenden näher erläutert.

Das entscheidende bei den Ansätzen ist, dass stets alle Features eines Dokumentes berechnet und für die Klassifikation bereitgestellt werden. Die Berechnung der Eigenschaften wird nicht vorzeitig abgebrochen, auch wenn ein eindeutig diskriminierendes Feature vorhanden ist. Die Klassifikation findet erst nach der Feature-Bestimmung statt.

3.3.1. Maschinelles Lernen

Ein Computerprogramm lernt, wenn es anhand von Beispielen Gesetzmäßigkeiten erkennt und durch die gewonnene Erfahrung verallgemeinern kann, um unbekannte Daten beurteilen zu können [31].

Es sei C eine Menge von Beispielen und $M = \{c_1, ..., c_n\}$ eine Menge von Klassen. Für die Klassifizierung wird eine Funktion $f : C \rightarrow M$ benötigt. Ein Lernalgorithums versucht diese Funktion f zu bestimmen bzw. anzunähern. Nach der Art des Lernens können die Verfahren in folgende Klassen einteilt werden [49]:

- **Überwachtes Lernen**
  Beim überwachten Lernen, auch „Supervised Learning“ oder „Lernen aus Beispielen“ genannt, werden alle Beispiele der Trainingsmenge mit den Eigenschaften und der Klassenzugehörigkeit als zusätzliches Merkmal bereitgestellt. Dieses Wissen kann bei der Anpassung des Algorithmus genutzt werden, der so im Allgemeinen effektiver wird. Da das System die Klasse eines Beispiels kennt, wird bei einem Fehler der Algorithmus automatisch angepasst. Um das Verfahren anzuwenden, muss eine große Anzahl von klassifizierten Elementen für die Trainings- und die Testmenge zur Verfügung stehen. Beispiele für diese Lernart sind Support-Vektor-Maschinen und neuronale Netze.

- **Verstärkendes Lernen (Reinforcement Learning)**
  Bei dieser abgeschwächten Form des überwachten Lernens sind nur die Eigenschaften der Beispiele gegeben. Die zugehörigen Klassen werden nicht bereitgestellt. Durch Belohnung oder Bestrafung wird die Klassifikation eines Beispiels bewertet. Dies ist weniger aufwendig, aber es stehen auch weniger Informationen für die Parametrisierung des Algorithmus zur Verfügung. Angewendet wird dieses Verfahren bei Agentensystemen. Die Agenten sollen dabei selbstständig zu einem gegebenen Zustand die passende Aktion auswählen. Durch das Belohnen oder Betrafen lernen sie, wie sie sich zu verhalten haben.

- **Unüberwachtes Lernen**
  Hier stehen zum Lernen zwar Beispiele, aber keine bekannte Klassifikation bereit. Es ist nicht bekannt, welche Klassen es gibt. Anhand der Eigenschaften werden Gruppen aus ähnlichen Beispielen gebildet. Zu dieser Lernart gehört die Cluster-Analyse. Sie wird angewendet, um eine Vorstellung über eine Datenmenge zu bekommen und so eventuell eine neue Klassifikation herzuleiten [41].

Bei der Genre-Analyse wird meist das überwachte Lernen angewandt, da die anzuwendende Klassifikation vorgegeben wird und die Trainingsdaten bereits per Hand ihren Klassen zugeordnet wurden.

Im Folgenden werden häufig verwendete Verfahren kurz vorgestellt.

Cluster-Analyse

Die Cluster-Analyse basiert auf Distanzfunktionen, mit denen man die Ähnlichkeit zwischen zwei Elementen bestimmen kann. Das Ziel ist es, ähnliche Beispielelemente

in Gruppen beziehungsweise Clustern zusammenzufassen. Bei dem hierarchischen agglomerativen[4] Verfahren werden beispielsweise bei jedem Schritt die Elemente oder Cluster mit dem geringsten Abstand gesucht und zu einem neuen Cluster verbunden. Das Verfahren endet, wenn alle Elemente einem Cluster angehören.

Bei dem K-means-Verfahren, wie es beispielsweise Santini [41] benutzt, wird eine Anzahl von Cluster-Zentren (auch Zentroide) vorgegeben. Jedes Element wird dann dem Cluster zuordnet, bei dem der Abstand zum Cluster-Zentrum minimal ist. Der Schwerpunkt des jeweils gebildeten Clusters ist der neue Zentroid. Das Verfahren wird so lange wiederholt, bis es keine neuen Zuordnungen mehr gibt.

Entscheidungbäume

Bei einem Entscheidungsbaum wird die Klassifikation durch einen Baum abgebildet. In ihm repräsentieren die Wurzel und die inneren Knoten die Features. Für jeden Wert, den ein Feature annehmen kann, führt eine Kante vom Knoten weg. Die Blätter des Baumes beinhalten die Klasse. In der Lernphase wird ein solcher Entscheidungsbaum aufgebaut. Bei der Klassifizierung startet man an der Wurzel des Baumes. An jedem Knoten fragt man den Wert des entsprechenden Features ab und folgt der Kante mit dem Feature-Wert. Dieser Vorgang wird wiederholt bis ein Blatt erreicht wurde, welches die zugehörige Klasse darstellt.

Die Entscheidungsbäume unterscheiden sich in der Art, wie die einzelnen Features während der Trainingsphase im Baum einsortiert werden. Finn und Kushmerick [12] sowie Dewdney et al. [8] verwendeten zur Klassifikation von Textdokumenten den C4.5-Algorithmus, einen Entscheidungsbaum, welcher den Information Gain zur Sortierung der Features beziehungsweise der Knoten benutzt. Je höher der Informationsgehalt eines Features ist, desto näher zur Wurzel wird ein Feature angeordnet.

Neuronale Netze

Nach dem Vorbild der Natur wurden die neuronalen Netze entwickelt. Eine Nervenzelle erhält über Synapsen unterschiedliche Impulse, die sie addiert. Überschreitet das Ergebnis einen festgelegten Schwellwert, wird ein Signal gesendet. Diese gerichtete Informationsverarbeitung wird bei den neuronalen Netzen mit so genannten Perzeptrons abgebildet. Das bedeutet, ein Perzeptron erhält Eingangswerte, anhand derer es wie ein Neuron einen Ausgangswert ermittelt.

Ein Perzeptron verfügt über ein oder mehrere Eingänge sowie Ausgänge. Die Eingänge sind gewichtet, das heißt Eingangswerte werden vor dem Zusammenfassen mit den

[4] agglomerativ: zusammenfassend, verschmelzend

Gewichten verrechnet. Ein Perzeptron fasst die Werte der Eingänge zusammen und bestimmt über eine Schwellwertfunktion den Ausgangswert (siehe Abbildung 3.3 links). Bei einem neuronalen Netz wird eine beliebige Anzahl von Perzeptrons miteinander verschaltet (siehe Abbildung 3.3 rechts).

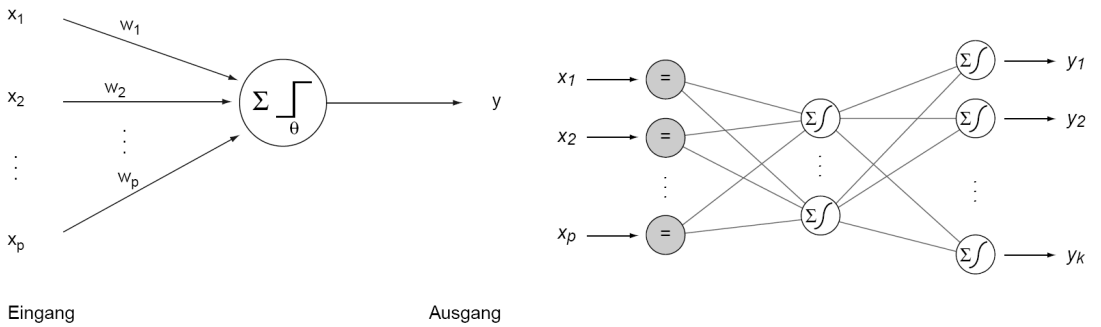

Abbildung 3.3.: Links: Einzelnes Perzeptron, rechts: Multilayer-Perzeptron, von Stein [47]

In der Lernphase wird jeweils ein Beispiel klassifiziert und danach überprüft, ob das berechnete Ergebnis richtig ist. Bei einer falschen Klassifizierung werden die am Fehler beteiligten Gewichte verändert. War das Ergebnis richtig, erfolgt keine Veränderung, da die aktuellen Gewichte für das Beispiel korrekt eingestellt sind.

Die Neuronalen Netze wurden unter anderem von Kessler et al. [25], Meyer zu Eißen und Stein [30] sowie Kennedy und Shepherd [24] zur Klassifikation verwendet.

Naive-Bayes

Bei einem Bayes-Klassifizierer wird die Klassenzugehörigkeit eines Beispiels über Wahrscheinlichkeiten bestimmt. Klassen und Feature-Werte werden als Ereignisse betrachtet. Während der Lernphase werden anhand der klassifizierten Beispiele die a-Priori-Wahrscheinlichkeiten[5] und die Wahrscheinlichkeiten für die Zusammenhänge zwischen den Daten und den Klassen bestimmt. Durch die Kombination der Wahrscheinlichkeiten lässt sich dann ein neues Element klassifizieren.

Näheres darüber, wie man mit einem Bayesschen Netzwerk lernen kann, beschreibt Heckerman in einem Tutorial [18]. Zur Genre-Klassifikation wurden Naive-Bayes-Netzwerke beispielsweise von Dewdney et al. [8], Lee und Myaeng [27] sowie Santini [39] benutzt. Es hat sich gezeigt, dass auch wenn die Features, wie beim Bayes-Klassifizierer

[5] a-Priori-Wahrscheinlichkeit: Ein Wahrscheinlichkeitswert, welcher mittels Vorwissen bestimmt wird.

gefordert, nicht stochastisch unabhängig[6] sind, sich doch sehr gute Klassifikationsergebnisse erreichen lassen [46].

Support-Vektor-Maschine (SVM)

Bei einer Support-Vektor-Maschine werden die Klassen der Trainingsdaten durch so genannte Hyperebenen getrennt. Die Lage der Hyperebene ist optimal, wenn der Abstand zu den Klassen maximal ist. Die Lage dieser Ebene wird durch die Support-Vektoren (tragende Vektoren) bestimmt. Das sind die Punkte, die der Hyperebene am nächsten liegen. Da die Trennung der Klassen in der entsprechenden Dimension oft kompliziert ist, werden die Daten in eine höhere Dimension, dem Feature-Space, abgebildet, wo die Lage der Hyperebene einfacher zu bestimmen ist. Die Abbildung 3.4 stellt die Transformation von einem 2- in einen 3-dimensionalen Raum dar.

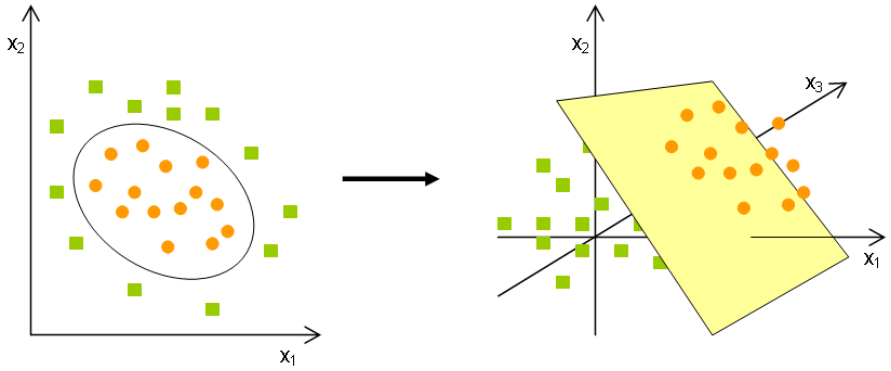

Abbildung 3.4.: Überführung der 2-dimensionalen Daten in eine 3-dimensionalen Raum, dem Feature-Space, nach Markowetz [28]

Für detaillierte Informationen sei auf das Tutorial von Smola und Schoelkopf [51] verwiesen. Neben Dewdney et al. [8] haben auch Santini [42] und Meyer zu Eißen und Stein [30] dieses Verfahren bei der Genre-Analyse angewandt. Es hat gute Ergebnisse bei der Klassifikation neuer Beispiele gezeigt.

3.3.2. Diskriminanzanalyse

Die Diskriminanzanalyse ist ein statistisches Verfahren, bei dem die Gruppenzugehörigkeit von Elementen analysiert wird. Anhand vorsortierter Beispiele werden Diskriminanzfunktionen berechnet, welche gemeinsam die unterschiedlichen Klassen trennen. Die Funktionen werden so gewichtet, dass sie die Unterschiede zwischen den Gruppen maximieren und somit die Überschneidungen minimieren. Die Diskriminanzanalyse gibt Aufschluss darüber, welche Eigenschaften sich am besten zur Vorhersage

[6] Stochastisch unabhängig bedeutet, dass sich Ereignisse bezüglich ihrer Wahrscheinlichkeiten nicht beeinflussen. Somit müssten die Werte der einzelnen Features unabhängig von einander sein.

der Gruppenzugehörigkeit eignen und ob mithilfe der Attribute signifikante Unterschiede zwischen den Gruppen zu erkennen sind [1].

Die zu bestimmende Klasse wird auch als Gruppenvariable und die Eigenschaften als Merkmalsvariablen bezeichnet. Die folgenden Bedingungen werden bei der Diskriminanzanalyse an den Datensatz gestellt:

- Nominal skalierte Gruppenvariablen
- Metrische Skalierung der Merkmalsvariablen
- Normalverteilung der Beispiele
- Mehr Eigenschaften als Gruppenvariablen

Eine Diskriminanzfunktion Y hat die folgende Form:

$$Y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_j x_j$$

Dabei bezeichnen $x_1 \ldots x_j$ die Werte der jeweiligen Merkmalsvariablen und $b_1 \ldots b_j$ die Diskriminanzkoeffizienten, mit denen die Variablen in der Funktion gewichtet werden. b_0 ist die Konstante von Y. Zur Klassifizierung von k Klassen werden $k - 1$ Diskriminanzfunktionen verwendet.

Bei der Diskriminanzanalyse werden zuerst anhand des Datensatzes die Koeffizienten der Funktionen geschätzt. Nach Huber [20] liefert die Diskriminanzfunktion für jedes Element i einer Klasse k mit dem Merkmalswert x_{jki} einen Diskriminanzwert Y_{ki}. Sei I_k die Anzahl der Elemente in k, so kann für jede Klasse der mittlere Diskriminanzwert, auch Zentroid genannt, wie folgt bestimmt werden:

$$\overline{Y}_k = \frac{1}{I_k} \sum_{i=1}^{I_k} Y_{ki}$$

Die absolute Differenz $|\overline{Y}_a - \overline{Y}_b|$ beschreibt somit den Unterschied zwischen den zwei Klassen A und B. Auf einer so genannten Diskriminanzachse y können die Zentroiden so abgetragen werden, dass der Abstand maximal wird (siehe Abbildung 3.5).

Y^* in Abbildung 3.5 ist nach Huber der kritische Diskriminanzwert oder das Trennkriterium. Er ermöglicht die Zuordnung neuer Elemente, da die folgende Regel angewendet

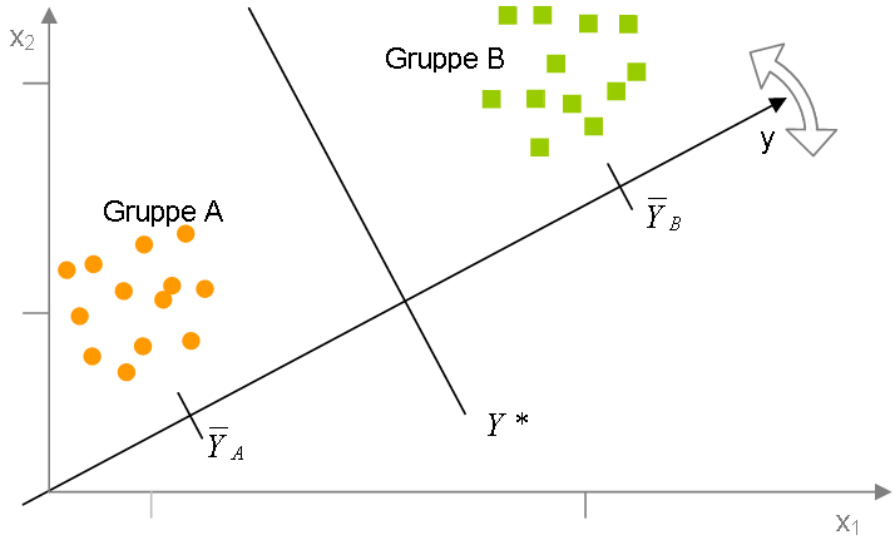

Abbildung 3.5.: Abbilden der Gruppenzentroiden auf die Diskriminanzachse y, nach Huber [20]

werden kann:

$$Y_i < Y^* \rightarrow i \in Gruppe\ A$$
$$Y_i > Y^* \rightarrow i \in Gruppe\ B$$

Bei der Diskriminanzanalyse wird der Abstand eines Elementes i zu allen Klassenzentroiden berechnet und verglichen. i ist ein Element der Klasse, bei der der Abstand minimal ist.

Bereits 1994 verwendeten Karlgren und Cutting [23] dieses Verfahren und zeigten, dass es für die Genre-Analyse durchaus geeignet ist. Ebenso wurde es von Stamatatos et al. [45] benutzt.

Dieses statistische Verfahren wurde auch bei der Evaluation dieser Arbeit angewandt.

4. Evaluation

In diesem Kapitel werden die einzelnen Phasen der im Rahmen dieser Diplomarbeit durchgeführten Genre-Analyse erläutert. Zunächst wird die Zusammenstellung des verwendeten Korpus erklärt und dann die entwickelte Software zum Extrahieren von Features vorgestellt. Des Weiteren werden die untersuchten Eigenschaften analysiert und die für die Genre-Analyse selektieren Features benannt. Zum Abschluss wird die Güte der Klassifikation einzelner Feature-Mengen untersucht.

4.1. Korpus-Konstruktion

Bevor mit dem Aufbau des Korpus begonnen werden kann, müssen die Genres festgelegt werden. Hier fand die Einteilung von Meyer zu Eißen und Stein [30] Anwendung, die über eine Benutzerstudie gewonnen wurde (siehe Abschnitt 2.2). Ebenso wurde deren Korpus in Form einer Bookmark-Datei mit bereits klassifizierten englischen Internet-Seiten übernommen. Da für das Trainieren eines Klassifizierers eine große Anzahl von Dokumenten repräsentativer ist, wurde der Korpus erweitert. Gleichzeitig wurden die gespeicherten Links der Bookmark-Datei auf ihre Zuordnung und Gültigkeit hin überprüft. Schließlich standen für die Evaluation 1704 Web-Dokumente zur Verfügung. Die Verteilung der Dokumente auf die einzelnen Genres zeigt Tabelle 4.1.

Genre	Dokumentanzahl
Artikel	181
Diskussion	242
Download	200
Hilfe	198
Linkliste	233
Portrait (nicht privat)	213
Portrait (privat)	191
Shop	246
Summe	1704

Tabelle 4.1.: Zusammenstellung der Web-Dokumente des Korpus

Beim Sammeln der Dokumente wurde darauf geachtet, dass die Beispiele eines Genres möglichst eine breite Menge der unterschiedlichen Ausprägungen widerspiegeln,

um Overfitting zu verhindern. Das Overfitting beschreibt das Problem, dass die Trainingsmenge nicht alle Facetten der Gesamtmenge enthält und dann eventuell der Klassifizierer bei neuen Datensätzen schlechtere Ergebnisse liefert.

Da Internet-Seiten einem ständigen Wandel unterliegen und die Verfügbarkeit eines Dokumentes vom Zustand eines Servers abhängt, wurden die Dateien für die Evaluation heruntergeladen. Dies geschah unter Verwendung der im folgenden Abschnitt beschriebenen Software *FeCo*. Beim Download wurde die URL jeder Seite zusätzlich in die Datei geschrieben, um sie später in die Auswertung einschließen zu können. Auch der Inhalt von eventuell vorhandenen Frames wurde in die heruntergeladene Datei geschrieben, so dass der gesamte sichtbare Inhalt analysiert werden kann. Die lokale Speicherung der Dokumente friert den aktuellen Zustand der Web-Dokumente ein und der Korpus kann für spätere Vergleichsbetrachtungen herangezogen werden. Weitere Details zum Datei-Download werden in Abschnitt 4.2.1 erläutert.

4.2. Software zum Extrahieren von Features

FeCo (**Fe**ature **Co**mputer) ist eine Bezeichnung für das im Rahmen dieser Diplomarbeit entstandene Programm zur Sicherung des Korpus sowie zur Gewinnung und zum Export von Features der gespeicherten Web-Dokumente. Dieses Programm ist mithilfe von Eclipse 3.1 [10] vollständig in Java Version 5.0 [50] implementiert worden und damit plattformunabhängig[1]. Es wurde erfolgreich unter Windows und Linux getestet.

In den folgenden Abschnitten werden die Programmabläufe, die Benutzeroberfläche und die umfangreichen Einstellungsmöglichkeiten erläutert.

4.2.1. Ablauf des Programms

Das Programm unterstützt die drei wesentlichen Arbeitsschritte zum Extrahieren von Features aus HTML-Dokumenten. Diese sind:

1. Herunterladen der HTML-Dateien
2. Berechnen von Features lokal gespeicherter HTML-Dateien
3. Exportieren aller Feature-Werte in eine Datei

Abbildung 4.1 veranschaulicht die Reihenfolge der Teilaufgaben. Sie zeigt auch, dass es drei mögliche Einstiegspunkte gibt. Es muss nicht zwingend an Punkt *a* begonnen werden. Ist bereits eine Sammlung von Dokumenten vorhanden und in der festgelegten

[1] Java-Applikationen werden in einer Virtual Machine (VM) ausgeführt und sollten auf allen Systemen funktionieren, auf denen auch eine VM installiert ist.

Struktur gespeichert, kann sofort bei *b* gestartet werden. Ebenso ist es möglich, gleich mit *c*, dem Export der Feature-Werte, anzufangen, wenn schon Dateien mit berechneten Features verfügbar sind.

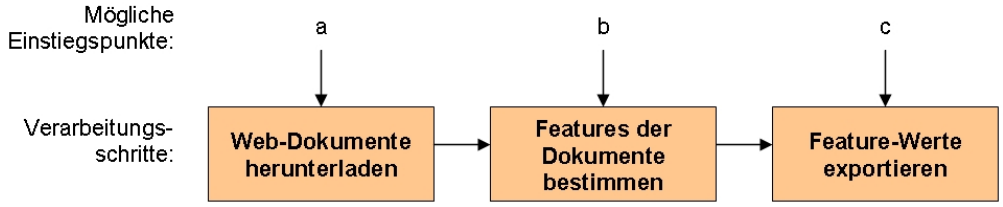

Abbildung 4.1.: Die einzelnen Verarbeitungsschritte und mögliche Einstiegspunkte des Programms FeCo

Der Zielordner für den Datei-Download beziehungsweise der Ordner für die Speicherung der ermittelten Features muss vor dem Start angegeben werden. In diesen Verzeichnissen wird dann jeweils pro Genre ein Unterordner mit dem Genre-Namen angelegt, in dem wiederum die Beispieldateien der einzelnen Kategorien abgelegt werden. Die Abbildung 4.2 zeigt eine von FeCo angelegte Ordnerstruktur.

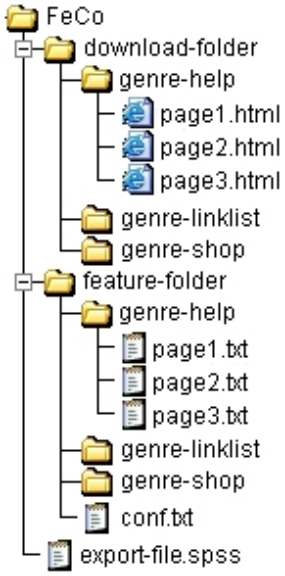

Abbildung 4.2.: Von FeCo angelegte Ordnerstruktur

Im Folgenden werden die drei Arbeitsschritte näher erläutert.

Download des Korpus

FeCo ist in der Lage, eine aus dem Firefox-Browser exportierte, im HTML-Format gespeicherte Bookmark-Datei zu parsen. Die Ordner, in denen die Verweise im Browser gruppiert sind, werden als Genres und die verlinkten Seiten selbst als Beispiele eines Genres interpretiert.

Vor dem Programmstart besteht die Möglichkeit, aus allen in der Bookmark-Datei erkannten Genres eine Auswahl zu treffen (siehe Abschnitt 4.2.2). Heruntergeladen werden nur HTML-Dokumente, eingebundene Bilder oder andere verknüpfte Dateien werden nicht betrachtet. Gespeichert wird ein Dokument unter einem Namen, welcher aus der URL generiert wurde[2]. Das ermöglicht die Zuordnung von verlinkter Seite und gespeichertem Dokument für einzelne gezielte Untersuchungen, falls beispielsweise eine Datei leer ist[3]. Gleichzeitig wird dem Dokument die URL als Meta-Tag hinzugefügt.

Die verlinkten Seiten der Bookmark-Datei werden aufgerufen und heruntergeladen. Zum Untersuchen der HTML-Datei wird der *HTMLParser* von Sourceforge benutzt [19]. Nicht selten befindet sich in der heruntergeladenen Datei jedoch nicht der eigentliche Inhalt der Seite, sondern nur eine Weiterleitung auf eine andere Seite. Eine erfolgreiche Klassifikation des Dokumentes wäre dann unmöglich. Das Programm ist daher in der Lage, einfache Weiterleitungen mittels HTML-Tags oder JavaScript zu erkennen. Es läd die „neue“ Seite herunter und schreibt den Inhalt zusätzlich in das bereits lokal gespeicherte HTML-Dokument. Ebenso wird mit den Inhalten von Frames verfahren. Da die verlinkte Seite wiederum Weiterleitungen oder Frames enthalten kann, wird in dieser nochmals nach Verweisen gesucht und wie oben beschrieben operiert. Bei einer Tiefe von 2 wird abgebrochen, um die Datenmenge zu begrenzen.

Berechnung der Features

Bei der Feature-Berechnung werden die lokal gespeicherten HTML-Dokumente der ausgewählten Genres geöffnet und untersucht. Dabei wird für jede HTML-Datei eine Textdatei mit gleichem Namen angelegt, in der die berechneten Feature-Werte hineingeschrieben werden. Leere HTML-Dokumente werden ignoriert.

Neben den eigentlichen Textdokumenten mit den entsprechenden Feature-Werten wird in dem Ordner, wie in Abbildung 4.2 zu sehen ist, eine Konfigurationsdatei *conf.txt* angelegt. Diese enthält die Namen der Eigenschaften in genau der Reihenfolge, wie

[2] Bei der Namensgenerierung wird die URL auf 30 Zeichen gekürzt, alle Punkte werden mit einem Minus und alle anderen für einen Dateinamen unzulässigen Zeichen durch einen Unterstrich ersetzt.

[3] Leere HTML-Dokumente entstehen, wenn der Server den Zugriff auf die Datei verweigert oder beim Herunterladen ein Fehler auftritt.

sie in der Feature-Datei aufgeführt werden. So dokumentiert das Programm, welche Features ermittelt wurden und zum Export zur Verfügung stehen.

Bevor neue Eigenschaften berechnet werden, wird stets die aktuelle Feature-Auswahl mit der eventuell im Zielordner vorhandenen Konfigurationsdatei verglichen. Um Inkonsistenz zu vermeiden, werden bei Nicht-Übereinstimmung oder Fehlen dieser Datei alte Feature-Dateien im Zielverzeichnis gelöscht.

Wie genau das Programm die Feature-Werte der einzelnen Klassen berechnet, ist im Abschnitt 3.2 beschrieben. Eine Liste aller mit FeCo berechenbaren Features befindet sich im Anhang B.

Feature-Export

Beim Exportieren werden alle benötigten Feature-Dateien eingelesen und die Werte in dem gewählten Exportformat abgelegt. Der Speicherort und das Format der Exportdatei werden vor dem Start angegeben.

Für den Export stehen die folgenden drei Formate zur Verfügung:

- **SPSS**-Datei, die mit dem Statistik- und Analyseprogramm SPSS® [44] eingelesen werden kann.
- **SVM**-Datei für SVM^{light} [21], die Support-Vektor-Maschine und SVM^{multiclass} [22], die Support-Vektor-Maschine für Multi-Klassen von Thorsten Joachims.
- **Weka**-Datei für die Data Mining Software Weka [55] von der Universität von Waikato.

In jedem Programmlauf wird nur jeweils eine Exportdatei erzeugt. Sind aber die Feature-Werte einmal bestimmt und gespeichert, kann unter Angabe des Feature-Ordners direkt eine Datei in einem anderen Exportformat erstellt werden.

4.2.2. Benutzeroberfläche

Nach dem Start des Programms ist das Hauptfenster zunächst leer (siehe Abbildung 4.3 links). Über den Menüpunkt *File* kann ein neues oder ein bestehendes Projekt geöffnet werden. Dann erscheinen drei Registerkarten, im Folgenden auch Tabs genannt, in denen die Konfiguration vorgenommen wird, bevor das Programm gestartet werden kann (siehe Abbildung 4.3 rechts).

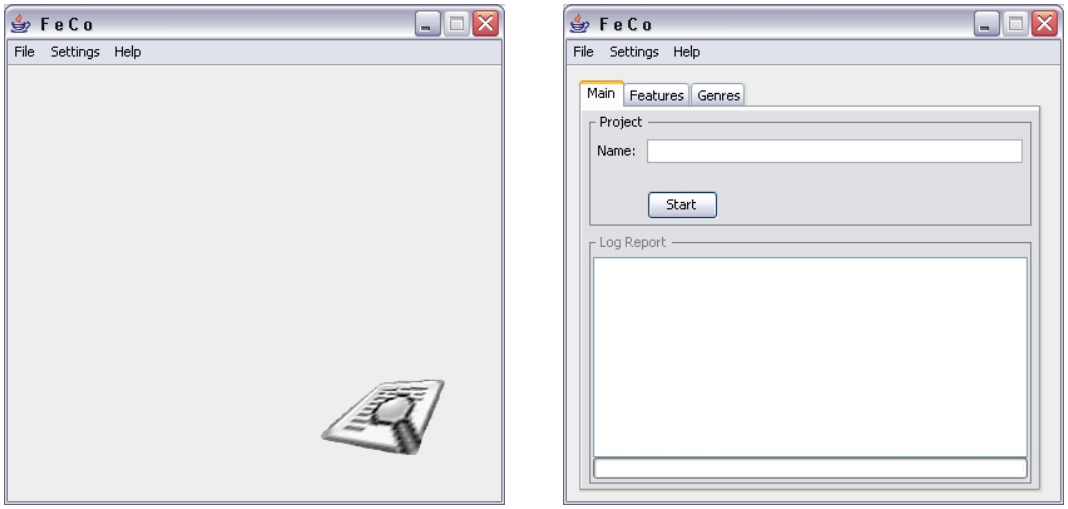

Abbildung 4.3.: FeCo nach dem Programmstart (links) und nach dem Öffnen eines neuen Projektes (rechts)

FeCo-Projekt

Das Programm bietet die Möglichkeit, die getroffenen Einstellungen in einer Projektdatei zu speichern. Sie enthält alle Parameter, um die Feature-Bestimmung von HTML-Dokumenten reproduzieren zu können. Darin werden unter anderem die ausgewählten Ordner und Features festgehalten. Der Speicherort der Projektdatei ist frei wählbar.

Menü

Neben den programmüblichen Menüpunkten *File* und *Help* enthält die FeCo-Software unter dem Eintrag *Settings* folgende zwei Punkte:

1. **Source:** Bestimmen des Einstiegpunktes durch Wahl von *Corpus Source* und Angabe der benötigten Dateien (siehe Abbildung 4.4 links)
2. **Export:** Festlegen der Exportdatei (siehe Abbildung 4.4 rechts)

Tabs

In dem Hauptfenster des Programms erscheinen drei Registerkarten: Main, Features und Genres (siehe Abbildung 4.5). Diese werden im Folgenden näher erläutert.

a) **Main-Tab**

Auf dem Main-Tab kann man den Namen des Projektes festlegen und über den Start-Button das Programm starten sowie stoppen (Abbildung 4.5 links). Im unteren Teil befindet sich eine Log-Konsole, die den Verlauf und den Zustand des

Abbildung 4.4.: Dialog für das Festlegen des Source (links) und für das Bestimmen der Exportdatei (rechts)

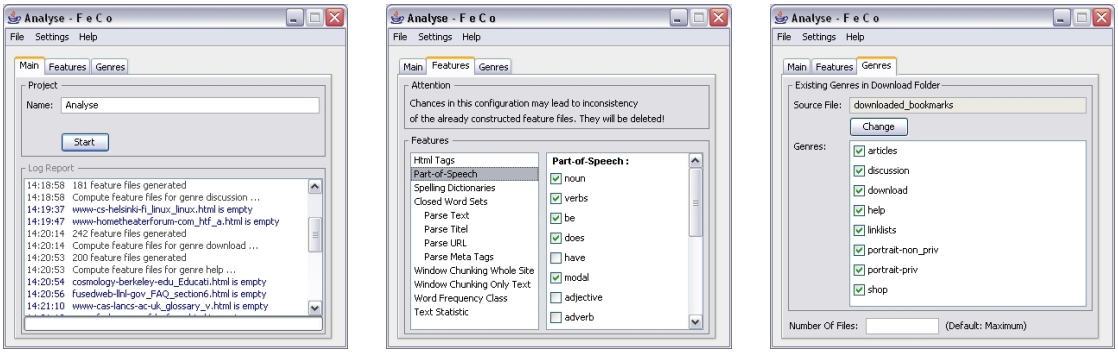

Abbildung 4.5.: Die drei Tabs des Programms: Main, Features und Genres (von links)

Programms dokumentiert. Die Einträge haben je nach Bedeutung drei verschiedene Farben.

- Informationen: Meldungen über den Programmstatus
- Warnungen: Hinweise auf ungültige URLs, leere oder fehlende Dateien
- Errors: Ereignisse, die keinen regulären Programmablauf ermöglichen

b) **Features-Tab**

Das Programm wurde entwickelt um auf einfache Weise unterschiedliche Feature-Mengen definieren und extrahieren zu können. Für jedes selektierte Feature wird ein Feature-Wert berechnet. Die Auswahl der gewünschten Features wird auf den Features-Tab getroffen (Abbildung 4.5 mitte). Die verfügbaren Features sind zur besseren Übersichtlichkeit in ihre Klassen aufgeteilt, welche auf der linken Seite angezeigt werden. Bei Auswahl einer Klasse werden die zugehörigen Features im rechten Fenster angezeigt.

Eine Selektion der gewünschten Features ist immer möglich, egal mit welchem der in Abschnitt 4.2 beschriebenen Arbeitsschritte man beginnt. Eine Beschränkung gibt es nur, wenn lediglich Feature-Werte exportiert werden sollen. In diesem Fall sind selbstverständlich alle Eigenschaften, die nicht berechnet wurden und somit nicht zur Verfügung stehen, inaktiv und demnach nicht auswählbar.

c) **Genres-Tab**

Wie bereits beschrieben kann das Programm an drei Einstiegspunkten gestartet werden (siehe Abbildung 4.1). Das Genres-Tab (Abbildung 4.5 rechts) zeigt im oberen Bereich, den ausgewählten Startpunkt unter Genre-Source (siehe Abschnitt 4.2.2) an. Er wird über den Dialog *Source* verändert. Zu diesem Dialog kommt man über den Menüpunkt *Settings* oder durch Klicken des Buttons *change*.

In der Mitte werden die in der Quelle erkannten Genres aufgelistet. Hier besteht die Möglichkeit, nach Belieben Genres zu selektieren.

Im unteren Teil kann man die Anzahl der Beispieldateien pro Genre begrenzen. Dabei stellt der Wert eine obere Grenze dar. Befinden sich mehr als die gewünschte Menge in der Bookmark-Datei oder dem Verzeichnis, werden die restlichen Einträge vernachlässigt. Sind jedoch weniger vorhanden, wird lediglich zur Information für den Nutzer eine Warnung in der Log-Konsole ausgegeben. Das Programm selbst

läuft ganz normal weiter. Wird keine Anzahl festgelegt, werden alle verfügbaren Beispiele verwendet.

4.3. Optimierung der Feature-Menge

In Abschnitt 3.2 wurden verschiedene Feature-Klassen, die bei der Genre-Analyse angewendet werden können, erläutert. Dennoch gibt es entscheidende Gründe, nicht alle Features zur Klassifikation zu verwenden und eine Selektion der Eigenschaften vorzunehmen. Es gibt Eigenschaften, die über alle Genres hinweg gleichverteilt sind und somit nicht zur Identifikation beitragen können, wie beispielsweise die Anzahl der Titel-Tags. Hinzu kommt, dass die Reduktion der Feature-Anzahl eine Verkürzung der Berechnungszeit bringt und gleichzeitig den Aufwand für die Klassifizierung senkt.

Im Folgenden werden die mit FeCo extrahierbaren Features analysiert und für die Klassifikation der gewählten Genres geeignete selektiert. Zum Abschluss werden die ausgewählten Eigenschaften benannt.

4.3.1. Feature-Analyse

Um einen Überblick über die einzelnen Feature-Klassen zu erhalten, werden sie im Folgenden näher untersucht. Zunächst werden sie anhand ihres Berechnungsaufwandes analysiert und anschließend hinsichtlich ihrer Klassifizierungsrate betrachtet.

Berechnungsaufwand

Die Genre-Analyse soll nicht nur zuverlässig klassifizieren, sondern auch in Echtzeit durchgeführt werden. Dafür ist es notwendig die einzelnen Feature-Klassen in Bezug auf den zeitlichen Aufwand, der bei der Berechnung erforderlich ist, zu untersuchen. Das Ergebnis wird in der Abbildung 4.6 dargestellt.

Es wurde die Zeit gemessen, die FeCo zum Extrahieren[4] einzelner Feature-Mengen aus den 1704 Beispieldokumenten (insgesamt 58,3 MB) benötigte. Um den Durchsatz zu bestimmen, wurde anschließend die Datenmenge durch die benötigte Zeit dividiert. Das Säulendiagramm in der Abbildung 4.6 vergleicht die resultierdenen Werte. Dabei zeigen die dunklen Säulen den Wert für die Berechnung aller Features einer Klasse und die hellen die Werte für ausgewählte Features.

Die für die Evaluation ausgewählten Feature-Klassen Spelling, Textstatistik und Word-Frequency-Class sind mit einem sehr geringen Rechenaufwand verbunden und

[4] Rechnerparameter: AMD Athlon™64 X2 Dual Core Processor 4200+; 2,21 GHz; 2,00 GB RAM

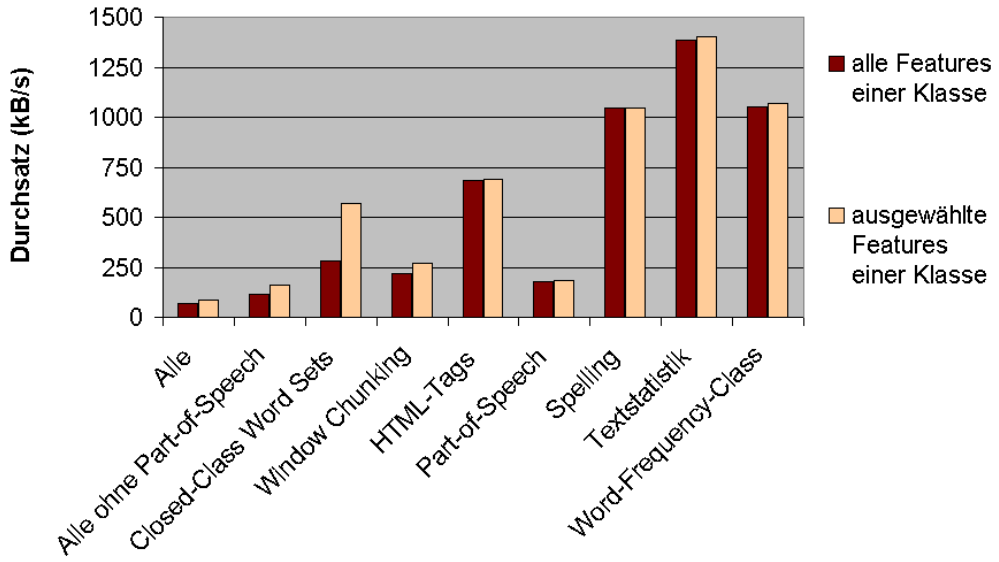

Abbildung 4.6.: Vergleich des Berechnungsaufwandes einzelner Feature-Mengen. Die genauen Werte stehen im Anhang in Tabelle A.1.

schaffen einen Durchsatz von mehr als 1000 Kilobytes pro Sekunde (kB/s). Auch die Features der HTML-Tags erzielen mit etwa 680 kB/s einen hohen Durchsatz. Zeitaufwendiger ist die Berechnung der Features mittels Closed-Class Word Sets und Window Chunking. Sie erreichen etwa 280 bzw. 216 kB/s. Am aufwendigsten ist die Bestimmung der Part-of-Speech-Features. Hier wurde nur ein Durchsatz von 180 kB/s erreicht.

Die Berechnung aller Features verringert den Durchsatz auf 70 bzw. 85 kB/s. Nimmt man die POS-Features aus der Feature-Menge heraus, kann der Durchsatz fast verdoppelt werden. Im Folgenden wird noch geprüft, ob sich der Mehraufwand bei der Berechnung der POS-Features durch einen deutlichen Anstieg der Klassifizierungsrate lohnt.

Die Abbildung 4.6 verdeutlicht auch, dass eine Selektion der Features innerhalb einer Klasse nicht immer einen Vorteil beim Berechnungsaufwand bringt. Bei den HTML-Tags beispielsweise müssen stets alle Tags erkannt werden, um die ausgewählten zu zählen. Somit bleibt der Aufwand für die Berechnung nahezu konstant. Bei den Features, die mithilfe von Wörterbüchern bestimmt werden, wie die Closed-Class Word Sets, und für jedes Wort eines Textes geprüft wird, ob es in einer Liste vorkommt oder nicht, ist ein Zeitgewinn bei Verringerung der Wörterbuchanzahl erkennbar.

Klassifizierungsrate

Im Folgenden wird untersucht, wie gut sich die einzelnen Feature-Klassen zu Identifikation der Genres eignen. Für jede Feature-Klasse werden alle zugehörigen Features extrahiert, welche anschließend mithilfe der Diskriminanzanalyse hinsichtlich ihrer Klassifikationsgüte untersucht werden.

Einen Vergleich der resultierenden Klassifizierungsraten zeigt das Säulendiagramm in Abbildung 4.7. Die Klassifizierungsrate bestimmt, wie viele Dokumente eines Genres durch die Diskriminanzanalyse korrekt zugeordnet wurden. Die gestrichelte Linie entspricht der Klassifikationsrate bei zufälliger Zuordnung der Genre-Klasse.

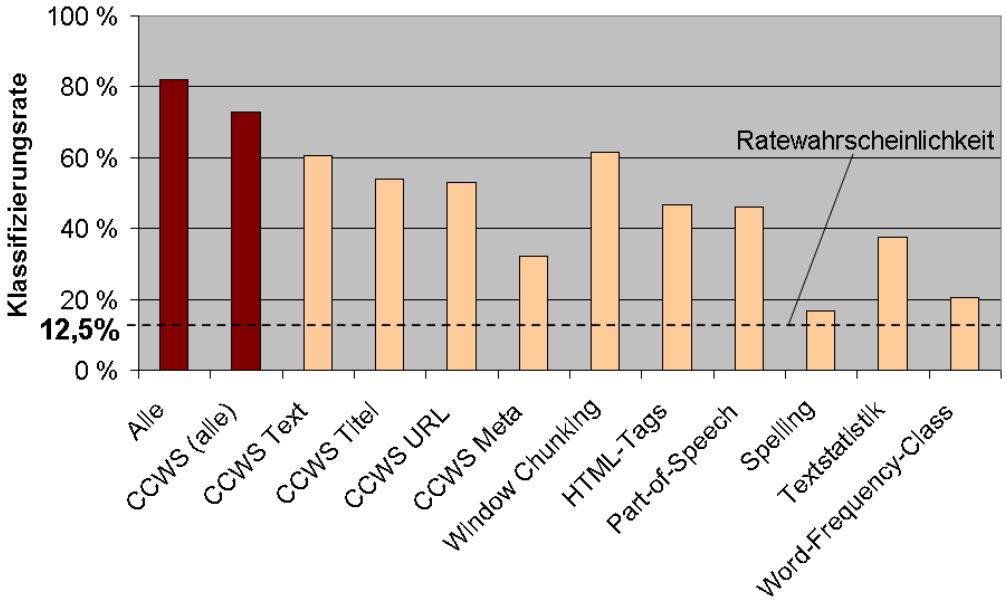

Abbildung 4.7.: Vergleich der Klassifizierungsraten der einzelnen Feature-Klassen (siehe auch Tabelle A.2)

Untersucht wurden alle Features einer Klasse, die sich mittels FeCo bestimmen lassen. Vergleicht man die einzelnen Feature-Klassen untereinander (helle Säulen), zeigt sich, dass die Closed-Class Word Sets (CCWS) im Durchschnitt am besten klassifizieren, wohingegen die Werte der Klassen Word-Frequency-Class und Spelling deutlich schlechter sind. Der vergleichsweise hohe Wert für das Window Chunking lässt sich mit der Verwandtschaft zu den Closed-Class Word Sets begründen (siehe Abschnitt 3.2.2). Verwendet man alle Closed-Class Word Sets zusammen, so liegt die Klassifizierungsrate nur 9,1 % unter dem Ergebnis, welches mit allen Feature-Klassen erzielt wurde (dunkle Säulen).

Da sich die Web-Dokumente durch die Features der Closed-Class Word Sets am besten identifizieren lassen, wurden diese genauer untersucht. Abbildung 4.8 zeigt, wie gut die einzelnen Genres mittels der Closed-Class Word Sets klassifiziert werden können.

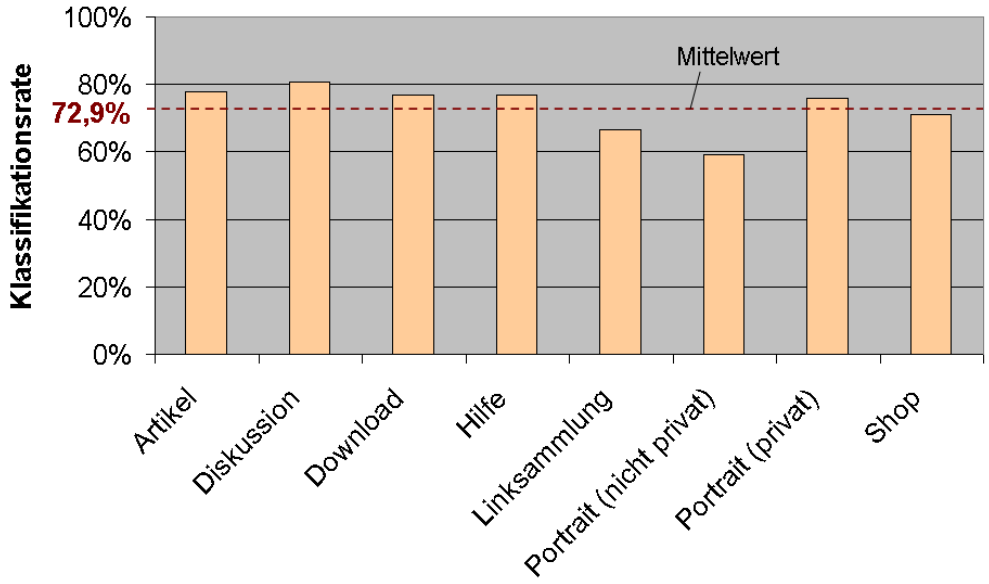

Abbildung 4.8.: Vergleich der Klassifizierungsraten der Closed-Class Word Sets (siehe auch Tabelle A.3)

Es wird deutlich, dass sie sich für die Identifikation der Genres Artikel, Diskussion, Download, Hilfe und Portrait (privat) sehr gut eignen. Dagegen lassen sich die nicht privaten Portraits deutlich schwerer erkennen und der Wert liegt 13,7 % unter dem Durchschnitt.

In Abschnitt 3.2.1 wurde die Hypothese aufgestellt, dass innerhalb eines Genres spezielles vom Thema unabhängiges Vokabular verwendet wird. Aus diesem Grund wurden Genre-spezifische Closed-Class Word Sets zusammengestellt. Das heißt, für jedes Genre wurde eine Liste mit typischen Wörtern zusammengetragen. Die folgende Untersuchung bestätigt diese Hypothese.

Zunächst wurden für jedes Dokument die Feature-Werte der Genre-spezifischen Closed-Class Word Sets berechnet. Pro Closed-Class Word Set wurden anschließend die Werte innerhalb eines Genres gemittelt. Dieser Durchschnitt gibt an, wie viele Wörter eines Closed-Class Word Sets in den Dokumenten eines Genres durchschnittlich enthalten waren. Um zu ermitteln, in welchem Genre die meisten Wörter eines Wörterbuches vorkamen, wurden pro Closed-Class Word Set die Mittelwerte ins Verhältnis gesetzt.

Das Säulendiagramm in Abbildung 4.9 stellt die durchschnittliche Verteilung der Feature-Mittelwerte von den Genre-spezifischen Closed-Class Word Sets auf die einzelnen Genres dar. Die Mittelwerte aller Genres ergeben zusammen pro Closed-Class Word Set 100 %. Jede Säule zeigt das Verhältnis der Feature-Werte eines Genre-spezifischen Closed-Class Word Sets. Ein farbiger Abschnitt in einem Balken kennzeichnet, wie viel Prozent der Gesamtsumme auf das entsprechende Genre entfallen. Das Diagramm lässt erkennen, dass die Wörter in den Closed-Class Word Sets, die jeweils für ein Genre spezifiziert sind, auch geschickt gewählt wurden. Analysiert man beispielsweise die Verteilung der Werte für das Genre-spezifischen Closed-Class Word Set *Shop*, so wird deutlich, dass 49,7 % der Wörter, die in den HTML-Dokumenten enthalten waren und in diesem Wörterbuch stehen, auch aus Web-Dokumenten des Genres *Shop* entstammen (letzte Säule). Wie die anderen Säulen zeigen, ist die Verteilung nicht immer so eindeutig, dennoch entfallen auf das entsprechende Genre, für das ein Closed-Class Word Set zusammengestellt wurde, die meisten Treffer. Das bestätigt die Verwendung von Genre-spezifischem Vokabular.

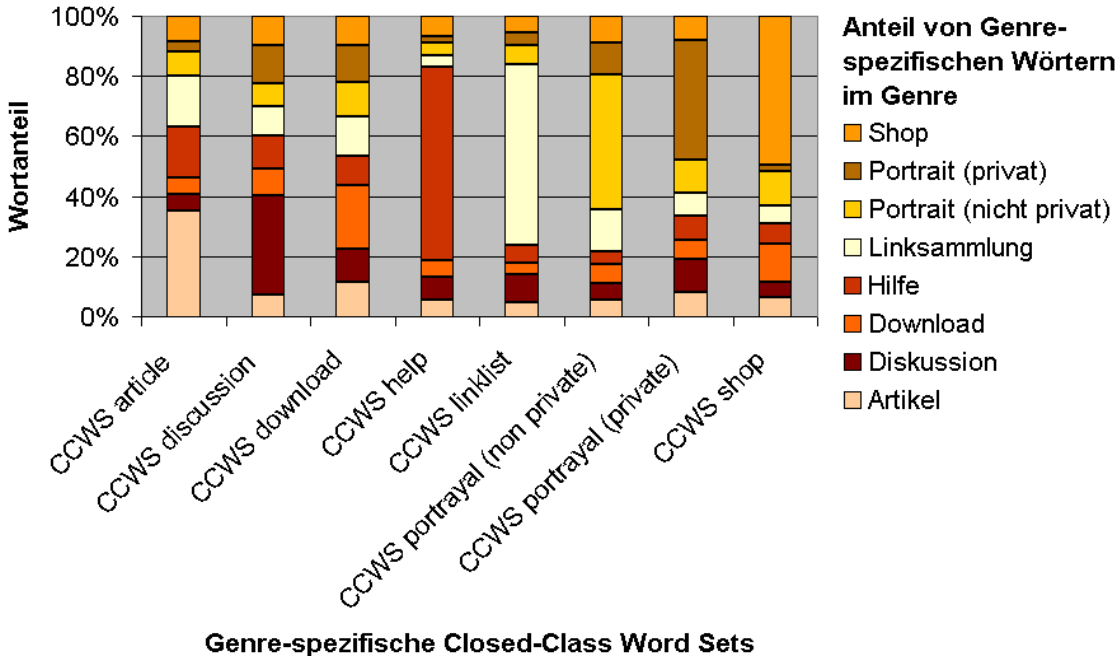

Abbildung 4.9.: Durchschnittliche Verteilung der Features von den Genre-spezifischen Closed-Class Word Sets in Bezug auf die Genres (siehe Tabelle A.4)

Abschließend wurde untersucht, inwieweit sich die Klassifizierungsrate der klassischen Features (wie in Abschnitt 3.2 beschrieben) durch die zusätzliche Analyse von Titel, URL und Meta-Tag sowie mit den durch Window Chunking ermittelten Features verbessern

lässt. Dafür wurde für die einzelnen Feature-Mengen eine Diskriminanzanalyse durchgeführt. Die Ergebnisse stehen in Tabelle 4.2. Es zeigt sich, dass die anfängliche Klassifizierungsrate von 72,6 % auf 81,3 % gesteigert werden konnte. Die in Abschnitt 3.2.2 beschriebenen neuen Features haben sich somit für die Klassifikation als sehr hilfreich erwiesen.

	Klassische Features (KF)	KF + Window Chunking	KF + Features aus Titel, URL, Meta-Tag	alle zusammen
Klassifizierungsrate	72,6%	75,2%	80,1%	81,3%

Tabelle 4.2.: Auswirkung der Hinzunahme von Analyse von Titel, URL und Meta-Tag sowie mit den durch Window Chunking ermittelten Features zu den klassischen Features auf das Klassifikationergebnis

4.3.2. Feature-Selektion

Wie bereits in Abschnitt 3.1 erwähnt, kommt es bei der Auswahl der Features darauf an, die Eigenschaften mit einem hohen Informationsgehalt herauszufiltern und diejenigen zu entfernen, welche keinen oder nur sehr geringen Einfluss auf die Klassifikation haben. Die Selektion hängt von den folgenden drei Faktoren ab [27]:

- Wie verteilen sich die Features zwischen den Genres?
- Wie gleichmäßig ist ein Feature innerhalb eines Genres verteilt?
- Wie stark kann ein Feature die Genres voneinander unterscheiden?

Auch ist die absolute Häufigkeit eines Features zu beachten. Eigenschaften, die innerhalb einer Klasse hohe Werte aufweisen und sonst kaum auftreten, sind sehr gut geeignet, diese eine Klasse abzugrenzen. Seltene oder sehr häufige Features, die in allen Klassen gleich verteilt sind, haben keine Auswirkung auf die Klassifikation.

Zur Selektion der relevanten Features stehen verschiedene Methoden zur Auswahl, wie Information Gain oder Mutual Information [59]. Hier wurde die Software SPSS zur Bewertung der unterschiedlichen Eigenschaften benutzt. Dabei wurde mit allen Feature-Werten, die mit FeCo ermittelt werden können, eine Diskriminanzanalyse (siehe Abschnitt 3.3.2) durchgeführt, welche neben den Klassifizierungsergebnissen auch Tabellen ausgibt, die einen Überblick über die einzelnen Feature-Werte verschaffen. Beispielsweise geben die kanonischen Diskriminanzfunktionskoeffizienten die Faktoren eines Feature-Wertes an, mit dem er in die einzelnen Diskriminanzfunktionen eingeht,

die der Klassifizierung zu Grunde liegen. Die standardisierten Koeffizienten erhält man über eine Transformation der Werte, so dass der Mittelwert 0 und die Standardabweichung 1 beträgt. Diese Werte sagen aber noch nichts über die Einflussgröße aus. Dazu dient die Struktur-Matrix. In dieser wird die Größe der absoluten Korrelation zwischen eines Features und der jeweiligen Diskriminanzfunktion aufgeführt und somit der Zusammenhang zwischen diesen verdeutlicht.

Da ein niedriger absoluter Wert auf einen geringen Einfluss einer Variablen hindeutet, wurde jeweils das Feature mit den geringsten Werten in der Struktur-Matrix aus der Feature-Menge entfernt. Auch Eigenschaften, die in mehreren Funktionen nahezu gleiche Werte hatten, wurden herausgenommen. Nach jeder Auslese wurde zur Kontrolle eine Diskriminanzanalyse durchgeführt. Diese Selektion wurde manuell durchgeführt, um eventuell gestrichene Features mit sehr geringen Werten, aber nicht unbedeutendem Einfluss, wieder aufnehmen zu können.

Die Güte einer Feature-Menge erkennt man in der SPSS-Ausgabe an den Klassifizierungsergebnissen. Sie geben an, wie gut die berechneten Diskriminanzfunktionen die Dokumente des Korpus klassifizieren. Die Tabelle 4.3 zeigt die Klassifizierungsergebnisse nach Abschluss der Selektion. In einer Zeile stehen die Dokumente eines Genres, die ihm manuell zugeordnet wurden. Beispielsweise stehen in der ersten Zeile alle 181 Dokumente des Genres Artikel. Sie wurden entsprechend ihrer Klassifizierung durch die Diskriminanzfunktionen in verschiedene Spalten aufgeteilt. Die Funktionen haben unter anderem 152 Beispiele dem Genre Artikel zugeordnet und eins dem Genre Diskussion. In der Diagonalen, von links oben nach rechts unten, stehen die Dokumente die korrekt klassifiziert wurden.

Die Tabelle 4.4 gibt die Ergebnistabelle in Prozent an. In den Zeilen stehen wiederum die manuell zugeordneten Dokumente eines Genres, die entsprechend dem vorhergesagten Genre in die einzelnen Spalten aufgeteilt wurden. Die 152 Beispiele aus dem Genre Artikel, die korrekt einsortiert wurden, entsprechen 84,0 % der Dokumente des Genres. Die durchschnittliche Klassifizierungsrate dieser Analyse beträgt 81,3 %. Das bedeutet, 81,3 % der Dokumente wurden korrekt klassifiziert.

Da für die Genre-Analyse interessant ist, wie gut die Features jedes einzelne Genre identifizieren können, sind hier nur die kreuzvalidierten Ergebnisse aufgeführt. Das heißt, bei der Kreuzvalidierung wurde jeweils eine Klasse durch die Diskriminanzfunktionen klassifiziert, die unter Ausschluss dieser Klasse ermittelt wurden.

Die Tabelle 4.5 zeigt die Klassifizierungsergebnisse vor der Feature-Selektion (Zeile 1), nach Abschluss der Selektion (Zeile 2) und für die selektierten Features ohne die Part-of-Speech-Features (Zeile 3). Es ist zu sehen, dass mit allen 132 Features eine durchschnittliche Rate von 82,0 % erreicht werden konnte. Bei der Selektion wurde fast

	Artikel	Diskussion	Download	Hilfe	Linksammlung	Portrait (nicht privat)	Portrait (privat)	Shop
Artikel	**152**	1	2	6	1	15	2	2
Diskussion	8	**208**	4	4	2	9	3	4
Download	1	2	**164**	4	0	18	3	8
Hilfe	19	4	1	**155**	0	11	4	4
Linksammlung	3	2	4	0	**184**	28	5	7
Portrait (nicht privat)	13	2	13	1	6	**163**	6	9
Portrait (privat)	13	0	2	0	2	8	**165**	1
Shop	2	3	5	6	2	33	1	**194**

Tabelle 4.3.: Klassifizierungsergebnisse der Diskriminanzanalyse mithilfe der selektierten Features (absolute Verteilung)

	Artikel	Diskussion	Download	Hilfe	Linksammlung	Portrait (nicht privat)	Portrait (privat)	Shop
Artikel	**84,0%**	0,6%	1,1%	3,3%	0,6%	8,3%	1,1%	1,1%
Diskussion	3,3%	**86,0%**	1,7%	1,7%	0,8%	3,7%	1,2%	1,7%
Download	0,5%	1,0%	**82,0%**	2,0%	0,0%	9,0%	1,5%	4,0%
Hilfe	9,6%	2,0%	0,5%	**78,3%**	0,0%	5,6%	2,0%	2,0%
Linksammlung	1,3%	0,9%	1,7%	0,0%	**79,0%**	12,0%	2,1%	3,0%
Portrait (nicht privat)	6,1%	0,9%	6,1%	0,5%	2,8%	**76,5%**	2,8%	4,2%
Portrait (privat)	6,8%	0,0%	1,0%	0,0%	1,0%	4,2%	**86,4%**	0,5%
Shop	0,8%	1,2%	2,0%	2,4%	0,8%	13,4%	0,4%	**78,9%**

Tabelle 4.4.: Klassifizierungsergebnisse der Diskriminanzanalyse mithilfe der selektierten Features (prozentuale Verteilung)

die Hälfte der Features entfernt. Dennoch liegt die durchschnittliche Klassifizierungsrate nach der Selektion mit den verbliebenen 76 Features inklusive Part-of-Speech-Features bei 81,3 % und ohne bei 80,0 %. Diese Abnahme ist gering, betrachtet man die enorme Reduktion der Feature-Anzahl. Zusätzlich muss man in diese Betrachtung die Analyse des Berechnungsaufwandes in Tabelle A.1 einbeziehen. Die Geschwindigkeit bei der Feature-Berechnung beträgt für alle Features etwa 70 kB/s und für die selektierten etwa 85 kB/s. Lässt man die Bestimmung der POS-Features weg, so lässt sich der Durchsatz mit 158 kB/s verdoppeln.

	Artikel	Diskussion	Download	Hilfe	Linksammlung	Portrait (nicht privat)	Portrait (privat)	Shop	Gesamt
alle (132)	82,3%	87,6%	81,5%	78,3%	79,0%	77,5%	88,5%	81,3%	82,0%
sel. (76)	84,0%	86,0%	82,0%	78,3%	79,0%	76,5%	86,4%	78,9%	81,3%
sel. ohne POS (66)	81,8%	86,0%	81,0%	77,3%	76,8%	74,2%	85,3%	78,5%	80,0%

Tabelle 4.5.: Vergleich der Klassifizierungsergebnisse vor der Feature-Selektion, nach Abschluss der Selektion und für die selektierten Features ohne die Part-of-Speech-Features (von oben nach unten)

4.3.3. Die selektierten Features

Nachdem im vorherigen Abschnitt die Kriterien für die Selektion beschrieben wurden, sollen nun die ausgewählten Features in Tabelle 4.6 benannt werden. Die Features werden in der Reihenfolge aufgelistet, wie sie in FeCo sortiert sind.

HTML Tags	All links	FTP links	Form tags
	Links in a list	Images	Paragraph tags
	Domain links	Bullet tags	DIV tags
	Ankor links	Table tags	SPAN tags
	Internet links	Table rows	Formular tags
	Mail links	Table columns	BR tags
Spelling Dictionaries	Webster's		
Closed-Class Word Sets			
Text	Discussion	Help	Non private
	Download	HTML table	Private
	Family names	Linklist	Shop
	First names	Months days	
Title	Discussion	Help	Private
	Download	Linklist	Shop

	Family names	Non private	
URL	Discussion	Help	Private
	Download	Linklist	Shop
	Family names	Non private	
Meta Tags	Article	Help	
	Download	Non private	
Window Chunking Text	Article	HTML table	Private
	Discussion	Linklist	Shop
	Download	Non private	
Window Chunking Whole Site	Links		
Word Frequency Class	Average word frequency class		
Text Statistics	Average word length	Capitals	Commas
	Letter	Colons	Dots

Tabelle 4.6.: Die selektierten Features

In der ersten Spalte der Tabelle 4.6 werden die Feature-Klassen genannt und in den drei darauf folgenden die dazugehörigen Features aufgelistet. Bei den *Spelling Dictionaries* und den *Closed-Class Word Sets* sind die Namen der selektierten Wörterbücher eingetragen. Die Wörterbücher *Article*, *Discussion*, *Download*, *Help*, *Linklist*, *Non private*, *Private* und *Shop* enthalten Wörter, die für das jeweilige Genre typisch sind. In *HTML table* stehen spezielle HTML-Entities und in *Months days* sind unter anderem Monatsnamen sowie Wochentage enthalten. Für das Window Chunking wurde eine Fenstergröße von 20 für den Text und 15 Wörtern für die gesamte Seite sowie eine Sprungweite von 3 Wörtern gewählt.

Da die Firefox-Erweiterung das Genre in Echtzeit bestimmen soll und der Unterschied bei der Klassifizierungsrate gering sowie der Mehraufwand bei der Feature-Berechnung sehr hoch ist, wurden die folgenden Part-of-Speech-Features aus dem Feature-Set für die Erweiterung herausgenommen.

Part-of-Speech	Noun	Modal	Copula
	Verbs	Article	Alphanumeric
	Be	Relative pronoun	
	Does	Preposition	

Tabelle 4.7.: Die entfernten POS-Features

4.4. Experimente

Im Folgenden werden drei Experimente erläutert, welche die Auswirkungen unterschiedlicher Feature-Mengen und verschiedener Genres auf die Klassifizierungsrate untersuchen.

Bei den Tests wurden stets alle Dokumente des Korpus, die FeCo-Software zur Bestimmung der Feature-Werte und die Diskriminanzanalyse von SPSS zur Klassifizierung verwendet.

4.4.1. Analyse von Texten vs. Web-Dokumenten

In Kapitel 2 wurde bereits erläutert, dass es einen Unterschied zwischen der Analyse von purem Text und der von Web-Dokumenten gibt. Wie in Abschnitt 3.2.2 beschrieben wurde, können aus letzteren Meta-Daten, die über den Inhalt des Textes hinaus gehen, gewonnen werden. Nun wird überprüft, ob das Auswerten dieser zusätzlichen Informationen wirklich einen Vorteil bringt und ob der steigende Berechnungsaufwand vertretbar ist.

In diesem Experiment werden die Klassifikationsergebnisse von zwei unterschiedlichen Feature-Mengen gegenübergestellt. Die eine für die Web-Dokumenten-Analyse enthält alle in Abschnitt 4.3.3 vorgestellten Feature-Klassen und bei dem anderen wurden nur die ausgewählt, welche sich ausschließlich auf den Text eines Dokumentes beziehen. Die Tabelle 4.8 zeigt die Zusammenstellung der beiden Feature-Mengen. Die Analysen wurden jeweils mit und ohne die Part-of-Speech-Features durchgeführt.

Text-Analyse	Web-Dokument-Analyse
Closed-Class Word Sets im Text	Feature-Menge der Text-Analyse plus:
Textstatistik	HTML-Tags
Worthäufigkeitsklasse	Closed-Class Word Sets im Titel, der
Spelling	URL und den Meta-Tags
Window Chunking Text	
(+ Part-of-Spech)	(+ Part-of-Speech)

Tabelle 4.8.: Zusammensetzung der Feature-Mengen für die Text- und die Web-Dokumenten-Analyse

In der Abbildung 4.10 werden die aus dem Experiment resultierenden Klassifizierungsraten dargestellt. Es zeigt sich, dass das Ergebnis mit den klassischen Text-Analyse-Features mit 70,9 % deutlich hinter der zweiten Feature-Menge mit 80,0 % zurückbleibt. Auch unter Verwendung der Part-of-Speech-Features gelingt es nicht, die Differenz von 9,1 % zu verringern, da sich die Feature-Werte selbst in beiden Fällen nicht unterscheiden. Das bedeutet, die Ausnutzung der Meta-Daten bringt einen enormen Vorteil und verbessert die Klassifizierungsrate deutlich.

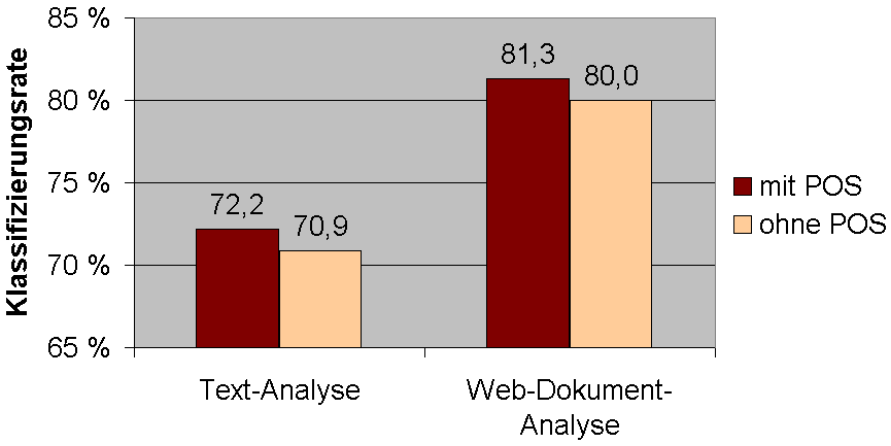

Abbildung 4.10.: Vergleich der Klasifizierungsraten für die Text- und die Web-Dokumenten-Analyse

4.4.2. Single-Genre

Bei diesem Experiment wurde die Klassifizierungsrate einzelner Genres gegenübergestellt. Jedes einzelne der acht Genres wurde allein gegen die anderen getestet. Pro Genre wurden wiederum zwei Tests durchgeführt, wobei die vorgenannte Feature-Menge zur Web-Dokumenten-Analyse (siehe Tabelle 4.8) mit und ohne die Part-of-Speech-Features zur Klassifizierung verwendet wurde. Der Klassifizierer musste bei dieser Untersuchung jeweils nur entscheiden, ob ein Dokument zum jeweiligen Genre gehört oder nicht.

	Shop	Portrait (privat)	Portrait (nicht privat)	Down-load	Diskus-sion	Artikel	Link-samm-lung	Hilfe
ohne POS	94,7%	95,8%	85,6%	95,1%	96,0%	91,8%	96,0%	95,5%
mit POS	94,7%	96,4%	86,0%	95,2%	96,2%	92,0%	96,6%	96,0%

Tabelle 4.9.: Klassifizierungsergebnisse von Single-Genre-Klassifizierern

Es hat sich herausgestellt, dass die Entscheidung zwischen zwei Klassen deutlich einfacher ist, als die Zuordnung in acht. Die Tabelle 4.9 zeigt in Prozent, wie viele der Dokumente im jeweiligen Test im Durchschnitt korrekt zugeordnet wurden. In der ersten Zeile stehen die entsprechenden Werte ohne und in der zweiten die mit Berücksichtigung der POS-Features. Die Klassifizierungsraten reichen von 85,6 % bis 96,0 % bzw. 86,0 % bis 96,6 % und liegen damit über dem Ergebnis der Klassifikation in acht Genres, bei dem im Durchschnitt lediglich 80,0 % bzw. 81,3 % erreicht wurden. Auffällig ist, dass das Genre *Portrait (nicht privat)* nicht nur am schwersten zu erkennen ist, sondern auch mit einer Differenz von etwa 6 % deutlich abgeschlagen hinter dem vorletzten Wert liegt. Das zeigt deutlich, dass dieses Genre durch die selektierten Features nicht extakt identifiziert

werden kann.

4.4.3. Profile

Meyer zu Eißen und Stein [30] weisen darauf hin, dass die Menge der interessanten Genres für verschiedene Nutzergruppen unterschiedlich ist. Sie stellen dabei die folgenden drei Benutzerprofile vor, welche die individuellen Interessen berücksichtigen sollen:

- *„Edu“*: Zu der Gruppe gehören Personen aus dem Bereich Bildung, der Fokus liegt auf den Genres Artikel, Linksammlung und Hilfe.
- *„Geek“*: Die Computer-Freaks sind sehr an Downloads, Diskussionen, Artikeln, Linksammlungen und Hilfe interessiert.
- *„Private“*: Die privaten Nutzer surfen bevorzugt, um zu shoppen und sich private Internet-Seiten anzusehen.

In dieser Untersuchung wird das Verhalten der Klassifizierungsraten unter den verschiedenen Profilen überprüft. In Tabelle 4.10 ist zu sehen, welche Genres benutzt beziehungsweise zu einem zusammengefasst wurden, da sie im jeweiligen Profil nicht von Interesse sind. Gleichzeitig sind die entsprechenden Klassifizierungsergebnisse der Diskriminanzanalyse eingetragen. In der oberen Reihe stehen die Werte ohne und in der unteren Reihe die mit den Part-of-Speech-Features.

<table>
<tr><th></th><th>Shop</th><th>Portrait (privat)</th><th>Portrait (nicht privat)</th><th>Download</th><th>Diskussion</th><th>Artikel</th><th>Linksammlung</th><th>Hilfe</th><th>Gesamt</th></tr>
<tr><td>Profil „Edu“</td><td colspan="4">91,9%
91,8%</td><td>82,9%
84,5%</td><td>78,1%
82,8%</td><td>77,3%
78,8%</td><td>87,4%
88,3%</td></tr>
<tr><td>Profil „Geek“</td><td colspan="3">85,1%
85,2%</td><td>82,5%
86,0%</td><td>85,5%
85,1%</td><td>83,4%
84,5%</td><td>77,7%
80,7%</td><td>78,3%
78,3%</td><td>82,9%
83,8%</td></tr>
<tr><td>Profil „Private“</td><td>82,5%
82,9%</td><td>88,0%
91,1%</td><td colspan="5">92,9%
93,4%</td><td>90,8%
91,7%</td></tr>
</table>

Tabelle 4.10.: Klassifizierungsraten der Profile

Bei Betrachtung dieser Tabelle ist erkennbar, dass sich die Aussage von Karlgren und Cutting [23] bestätigt. Die Klassifizierungsrate nimmt mit steigender Anzahl von Genres ab. Das Profil „Privat“ hat drei Klassen und liegt mit den Durchschnittswerten von 90,8 % und 91,7 % über den Ergebnissen der anderen. Auch „Edu“ schneidet mit seinen vier Klassen und einem Durchschnitt von 87,4 % beziehungsweise 88,3 % besser ab als „Geek“, welches mit den Ergebnissen durchschnittlich bei 82,9 % beziehungsweise 83,8 % liegt.

Zur Veranschaulichung dieser Ergebnisse wurden die Streuungsdiagramme der drei Profile und der Genre-Analyse mit den 8 Genres generiert. Sie werden in der Abbildung 4.11 dargestellt. An den Achsen sind jeweils die Funktionswerte der beiden ersten Diskriminanzfunktionen abgetragen. Pro Web-Dokument wird ein Viereck eingezeichnet, dessen Lage von den jeweiligen Funktionswerten dieser zwei Diskriminanzfunktionen abhängt. Die einzelnen Genres haben verschiedene Farben, so kann die Gruppenzugehörigkeit einfach abgelesen werden.

Pro Diskriminanzanalyse werden $n-1$ Funktionen berechnet, wobei n die Anzahl der zu klassifizierenden Genres darstellt. Demnach stehen für die Profile *Edu* und *Private* nur drei beziehungsweise zwei Funktionen zur Verfügung. Da in diesen Diagrammen die Funktionswerte der ersten beiden Diskriminanzfunktionen eingetragen werden und die Klassifizierungsrate bei den zwei Profilen mit durchschnittlich 91 % und 88 % sehr hoch war, kann man in den zwei unteren Diagrammen sehen, dass sich für jedes einzelne Genre kleine Cluster gebildet haben, die sich untereinander kaum überlagern. Das zeigt, dass sich die Genres mit den ersten zwei Funktionen sehr gut klassifizieren lassen.

In den oberen Diagrammen sind die Werte der Genre-Analyse mit allen acht Genres und des Profils *Geek* eingetragen. Da hier für die Klassifikation acht beziehungsweise sechs Klassen verwendet und demnach auch sieben sowie fünf Diskriminanzfunktionen berechnet werden, ist die Trennung zwischen den Genres nicht mehr deutlich erkennbar. Die Cluster der Genres *Hilfe*, *Artikel*, *Linksammlung* und *Diskussion* sind in beiden Abbildungen sichtbar. Die anderen Genres sind noch sehr verstreut. Deren Abgrenzung erfolgt mit den restlichen Diskriminanzfunktionen.

Aus den Klassifikationsergebnissen der Profile lässt sich schlussfolgern, dass eine Personalisierung über eine Festlegung eigener Gewichtungen für die Genres und eine daraus resultierende Zusammenlegung von Genres zu einer Verbesserung der Ergebnisse der Genre-Analyse führt. Die Zufriedenheit des Benutzers könnte auf diese Weise gesteigert werden.

4.5. Zusammenfassung

Für die Durchführung einer Genre-Analyse sind ein repräsentativer Korpus und die Auswahl einer geeigneten Feature-Menge wichtige Grundvoraussetzungen. So wurde die Linksammlung von Meyer zu Eißen und Stein deutlich erweitert. Es wurde darauf geachtet, dass die ausgewählten Dokumente möglichst alle Facetten der jeweiligen Genres widerspiegeln, damit der Klassifizierer später unbekannte Dokumente zuverlässig einordnen kann.

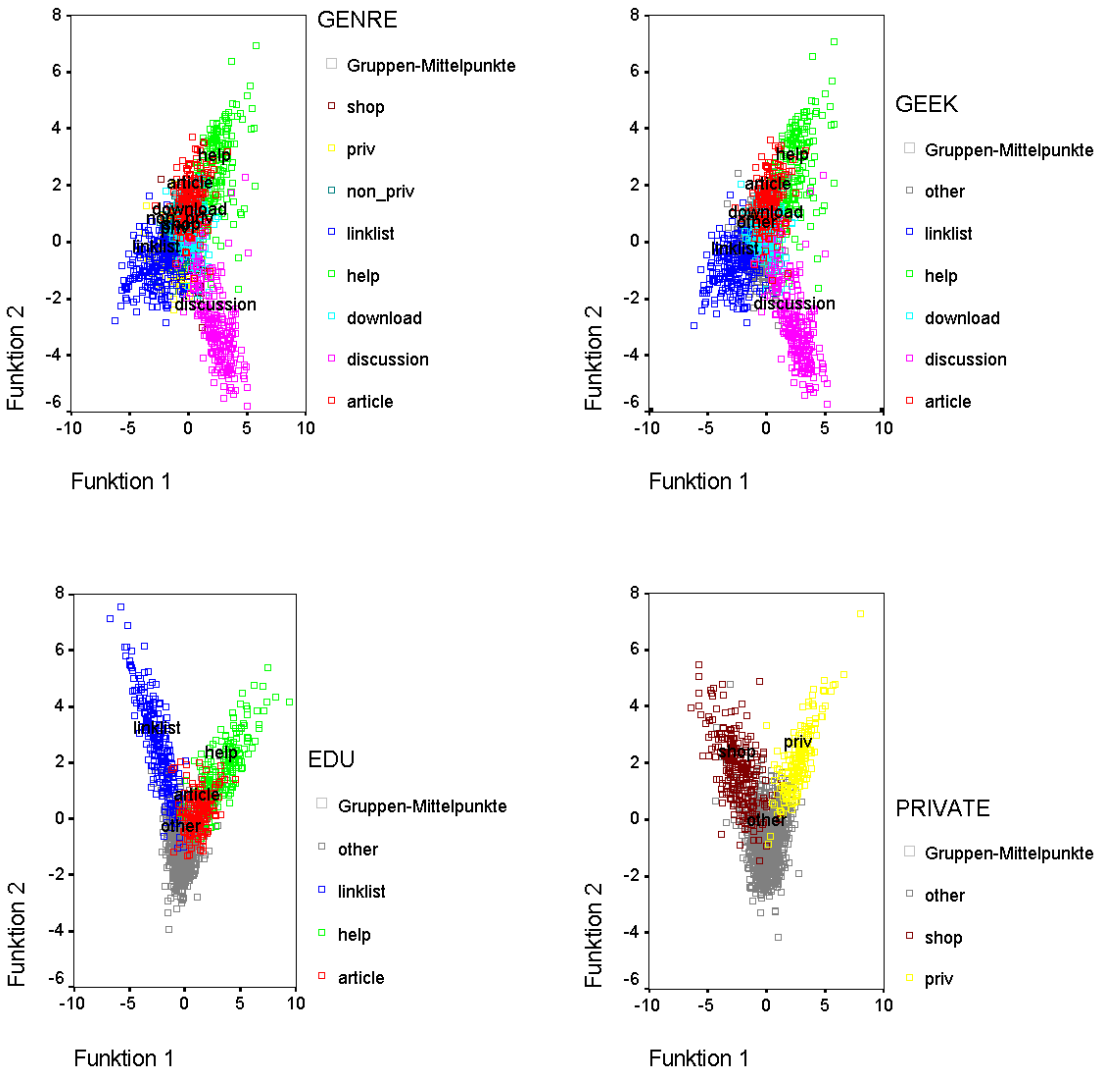

Abbildung 4.11.: Streuungsdiagramm unterschiedlicher Profile
oben: mit allen acht Genres (links), Profil Geek (rechts)
unten: Profil Edu (links), Profil Private (rechts)

Aus der Vielzahl aller Features müssen diejenigen selektiert werden, die zur Identifizierung der ausgewählten Genres am besten geeignet sind. Da später Web-Dokumente nicht nur zuverlässig, sondern auch in Echtzeit klassifiziert werden sollen, wurden vorwiegend einfach bestimmbare Eigenschaften ausgewählt und evaluiert. Um die verschiedensten Feature-Mengen ohne großen Aufwand berechnen und zusammenstellen zu können, wurde die Software FeCo entwickelt, welche eine Auswahl per Mausklick ermöglicht.

Durch Auswertung der Struktur-Matrix der Diskriminanzanalyse wurden schließlich aus den insgesamt 132 Features, welche das Programm derzeit anbietet, 66 für die anschließende Klassifizierung für die Firefox-Erweiterung selektiert. Zwar bestätigte sich der intuitive Ansatz, je mehr Features, desto besser das Ergebnis, aber der zeitliche Mehraufwand war hier entscheidend. Denn die Rechenzeit zur Bestimmung der Features konnte durch die Selektion um mehr als die Hälfte gesenkt werden.

Die Analyse der ausgewählten Feature-Menge hat gezeigt, dass die Klassifizierungsrate auch ohne die Part-of-Speech-Features mit durchschnittlich 80 % sehr hoch ist. Die Verwendung von so genannten Profilen kann diese Ergebnisse nochmals verbessern.

Bei den Untersuchungen zeigte sich, dass sich die Linksammlungen und private Portraits am besten identifizieren lassen. Das Genre *Portrait (nicht privat)* stellte sich als am schwersten zu erkennende Klasse heraus.

5. WEGA: Eine Firefox-Erweiterung für Real-Time Genre-Analyse

Mit Suchmaschinen im Internet nach Informationen zu suchen, ist eine alltägliche Handlung geworden. Man gibt Suchbegriffe ein und erhält Listen mit Ergebnissen. Anhand der Einträge ist es sehr schwer zu entscheiden, ob ein Ergebnis für die jeweilige Anfrage relevant ist oder nicht. Um den Suchenden bei der Entscheidung zu unterstützen, soll zusätzlich zu jedem Eintrag das zugehörige Genre eingeblendet werden.

Um die Genre-Analyse bei einer Internet-Suchanfrage zur Verfügung zu stellen, wurde eine Erweiterung für den Firefox-Browser entwickelt. Diese Erweiterung des Browsers erkennt durch Parsen der aktuellen URL die Durchführung einer Google-Suche, filtert die Links der Ergebnisseiten heraus und sendet die dazugehörigen URLs an ein Servlet. Dieses klassifiziert mithilfe des FeCo-Packages die Internet-Seiten und sendet die ermittelten Genres an die Erweiterung zurück. Schließlich werden die Genres hinter den jeweiligen Links direkt in der Google-Ergebnisseite eingeblendet. Die Abbildung 5.1 veranschaulicht die Kommunikation zwischen dem Client und dem Server.

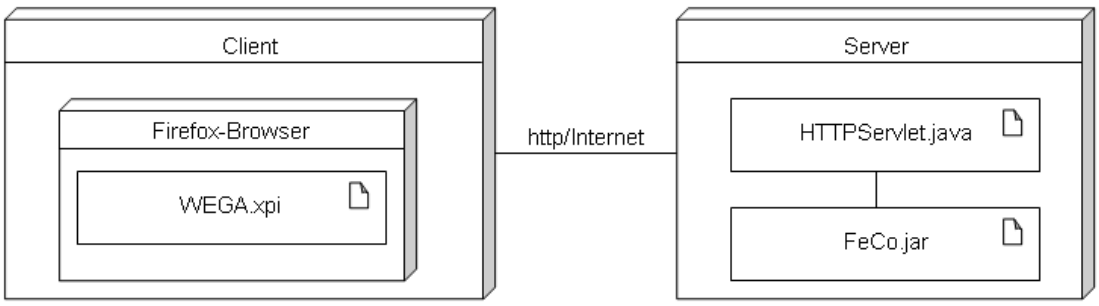

Abbildung 5.1.: Übersicht der Komponenten für die Real-Time Genre-Analyse

In den folgenden Abschnitten werden die Firefox-Erweiterung, das klassifizierende FeCo-Package und das Servlet näher erläutert.

5.1. Firefox-Erweiterung WEGA

Für die Bereitstellung der Genre-Analyse wurde die Erweiterung WEGA (Akronym für Web Genre Analysis) für den Firefox-Browser (ab Version 1.5) entwickelt. Im Firefox ist direkt ein Interface für Plugins und Erweiterungen vorgesehen. Als Plugin wird hierbei ein kleines Programm bezeichnet, welches mit dem Browser interagiert, um das Anzeigen beziehungsweise Abspielen von beliebigen MIME-Typen (Multipurpose Internet Mail Extensions) zu ermöglichen. Beispielsweise ermöglicht ein Plugin das Abspielen von Flash- oder Audio-Dateien im Browser selbst unter Zuhilfenahme von auf dem Rechner installierten Programmen. Im Gegensatz zu Plugins verändern oder ergänzen Erweiterungen die Funktionalität und das Layout eines Firefox-Browsers ohne Hilfe von externen Programmen. Bei der Real-Time Genre-Analyse werden unter Verwendung von JavaScript Google-Ergebnisseiten verändert, deshalb ist es eine Firefox-Erweiterung.

In den Folgenden Abschnitten werden die Funktionsweise und Struktur der Firefox-Erweiterung erläutert.

5.1.1. Funktionsweise

Immer, wenn im Browser eine neue Seite geladen wird, überprüft die Erweiterung, ob es sich dabei um eine Google-Suchanfrage handelt. Ist das der Fall, wird die Ergebnisseite geparst und die von Google gelieferten Links herausgesucht. Aus diesen Links werden die URLs extrahiert und zum Klassifizieren an den Server geschickt. Das ermittelte Genre wird dann direkt hinter dem Link in der Ergebnisseite eingeblendet. Das Activity-Diagramm in Abbildung 5.2 veranschaulicht die Abläufe.

Zur Überprüfung, ob es sich bei der aktuellen Seite um eine Google-Suchanfrage handelt, wird die URL anhand eines Text-Pattern (siehe Listing 5.1) überprüft. Wurde eine Suchanfrage erkannt, werden die Links der Ergebnisse anhand ihrer Tag-Klasse herausgesucht und die enthaltenen URLs extrahiert. Die gefundenen Links werden mit einem kleinen schwarzen Punkt versehen. Der Punkt ist ein Platzhalter für den Eintrag des ermittelten Genres. Um die „Fundstellen“ später den Serverantworten zuordnen zu können, werden die Platzhalter in Form von SPAN-Tags mit einer eindeutigen ID eingefügt.

```
P = "http://www\.google\. .* search\? .*";
```

Listing 5.1: Pattern zur Erkennung einer Google-Suchanfrage

Die URLs und die zugehörigen IDs werden mittels Ajax (Asynchronous JavaScript And XML) an den Server geschickt. Damit die Genre-Analyse parallel ablaufen kann,

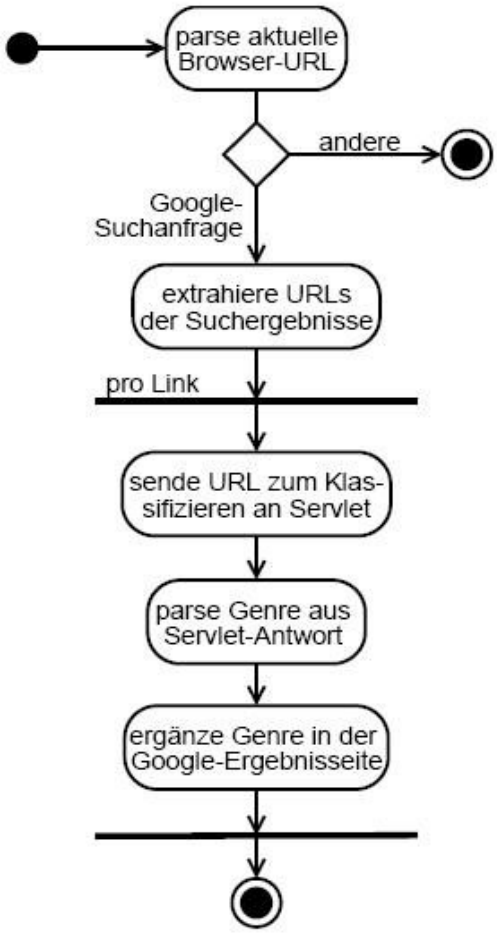

Abbildung 5.2.: Activity-Diagramm für die Firefox-Erweiterung WEGA

wird pro URL genau eine Anfrage an den Server gesendet. Wie das Wort „asynchron“ in Ajax bereits verrät, wird nicht auf die jeweilige Antwort des Servers mit dem Genre einer URL gewartet, bevor die nächste Anfrage gesendet wird. Vielmehr werden erst alle Server-Aufrufe gestartet und dann auf das Eintreffen der Ergebnisse gewartet.

Der Server antwortet mit einem XML-Dokument, in dem die ID und das ermittelte Genre stehen. Über die ID wird das zugehörige SPAN-Tag gefunden und anschließend die Klassifikation als Text eingetragen.

Wird als Genre eine *0* gesandt, so ist bei der Klassifizierung der Seite ein Fehler aufgetreten. In dem Fall ist in der XML-Datei eine Fehlermeldung enthalten. Das Format des SPAN-Tags verändert sich und statt dem Genre wird die Fehlermeldung angezeigt. Nähere Informationen über mögliche Fehler werden in Abschnitt 5.3.2

gegeben.

Die Abbildung 5.3 zeigt einen Ausschnitt des Ergebnisses einer Google-Suche mit eingebetteten Genre-Informationen.

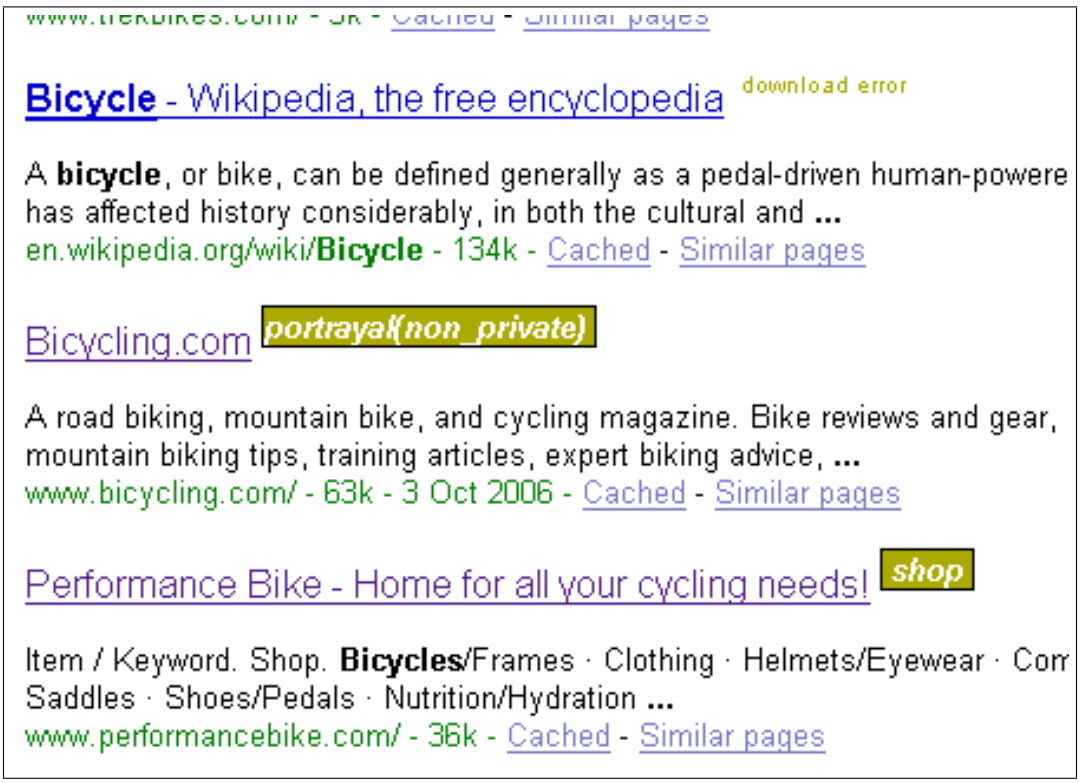

Abbildung 5.3.: Google-Suchanfrage mit eingebetteten Genre-Informationen

5.1.2. Struktur

Die Firefox-Oberfläche ist in XUL (Extensible User-Interface Language), einem XML-Dialekt, implementiert [58]. XUL-Dateien bestimmen, aus welchen Teilen die Benutzeroberfläche besteht und wie diese zueinander angeordnet sind. Sie definieren Elemente wie Fenster, Buttons sowie das Menü und die Statuszeile. Zur Veränderung des Layout werden CSS-Dateien (Cascading Style Sheets) benutzt. Unter Verwendung von JavaScript wurde das Event-Management, wie Menü- und Button-Selektionen sowie Maus- und Tastatureingaben umgesetzt. Mit den so genannten „Overlays“ wird ein leistungsfähiger Mechanismus bereitgestellt, der dazu dient, zusätzliche Elemente in die grafische Benutzeroberfläche des Browsers und Funktionalitäten einzubinden, ohne den Programmcode des Firefox-Browsers zu verändern.

Unter Verwendung der Overlays und JavaScript wurde die Firefox-Erweiterung WEGA entwickelt. Sie setzt sich aus mehreren Ordnern und Dateien zusammen, die in einem Extension-Paket, einer XPI-Datei (Cross-Platform Install), zur Verfügung gestellt werden. Dieses Format ist eine Entwicklung der Mozilla Foundation und stellt einen ZIP-Container dar, der ein Installationsskript und weitere Dateien enthält. In der Abbildung 5.4 ist die Zusammensetzung der Erweiterung dargestellt.

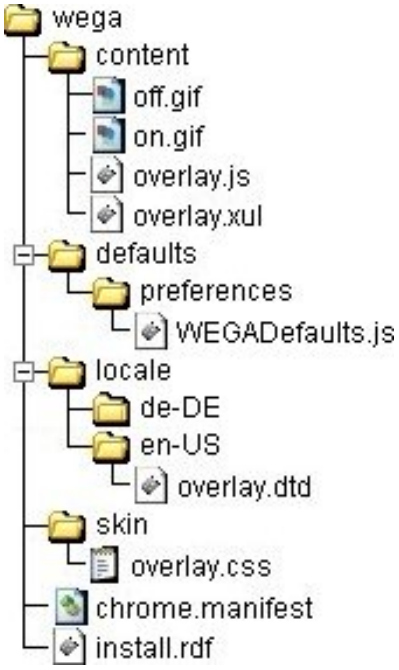

Abbildung 5.4.: Ordnerstruktur der Firefox-Erweiterung WEGA

Die Ordnerstruktur ist bei Programmierung einer Firefox-Erweiterung vorgegeben [33]. Abbildung 5.4 zeigt diese für die WEGA angepasste Struktur. Auf oberster Ebene des Containers muss die Installationsdatei (*install.rdf*) liegen, in der die Parameter der Erweiterung stehen, wie beispielsweise der Name, der Autor und eine Beschreibung der Anwendung. Ebenso werden in der Datei eine eindeutige ID und die Browser-Versionen bestimmt, für welche diese Erweiterung funktioniert. Eine Datei mit dem Namen *chrome.manifest* muss sich ebenfalls im Hauptverzeichnis befinden. Sie dient zur Definition der Pfade zu den Dateien einer Erweiterung. In den folgenden Unterordnern liegen die eigentlichen Anwendungsdateien:

- **content**
  Dieser Ordner ist das Herzstück jeder Erweiterung. Hier werden die XUL-Dateien abgelegt, welche die Benutzeroberfläche verändern und die JavaScript-Dateien, welche Methoden enthalten, die bei den festgelegten Events aufgerufen werden sollen.

Ebenso können hier Bilder abgelegt werden, die im Browser anzuzeigen sind. Beispielsweise befindet sich bei WEGA in diesem Ordner die JavaScript-Datei mit Methoden, die unter anderem eine Google-Suchanfrage erkennen und den Server zum Klassifizieren aufrufen.

- **defaults**
  In diesem Ordner kann man Präferenzen definieren. Hier stehen in einer DTD-Datei (Document Type Definition) Standardeinstellungen für das Plugin, wie im Fall dieser Firefox-Erweiterung die URL-Adresse des Servlets. Beim Start des Browsers werden die Dateien dieses Ordners stets geöffnet und die gespeicherten Einstellungen berücksichtigt. Die hier festgelegten Präferenzen können mit den JavaScript-Methoden aus dem *content*-Ordner verändert werden. Diese benutzerspezifischen Einstellungen bleiben über die Laufzeit des Browsers hinaus gespeichert [32].

- **locale**
  Der Firefox-Browser ist in einer Vielzahl von Sprachen verfügbar. Beim Start des Browsers wird anhand der verwendeten Browser-Sprache für jede installierte Erweiterung überprüft, ob Text (zum Beispiel die Titel der Menüeinträge) in der jeweiligen Sprache zur Verfügung steht. Die Beschriftungen können im *locale*-Ordner in dem Unterverzeichnis der entsprechenden Sprache innerhalb einer DTD-Datei definiert werden.

- **skin**
  Hier besteht die Möglichkeit, in einer CSS-Datei die Elemente der grafischen Benutzeroberfläche der Erweiterung optisch anzupassen.

5.1.3. GUI-Elemente

Nachdem die Erweiterung installiert wurde, ist sie standardmäßig aktiviert. Verändern lässt sich dieser Status im Menü über den Eintrag *Extras*, über das Icon in der Statusleiste oder über die Tastenkombination Strg + Umschalt + E. Der aktuelle Status kann an zwei verschiedenen Stellen im Browser abgelesen werden, in der Statuszeile und am Menüeintrag (siehe Abbildung 5.5). Dieser Status wird beim Beenden des Browsers gespeichert und beim erneuten Start auf den zuletzt gespeicherten Wert gesetzt.

5.2. Klassifizierendes FeCo-Package

Das im Abschnitt 4.2 vorgestellte Programm ist in der Lage, Web-Dokumente herunterzuladen und Features zu extrahieren sowie zu exportieren. Damit es auch klassifizieren kann, musste es wie folgt erweitert werden.

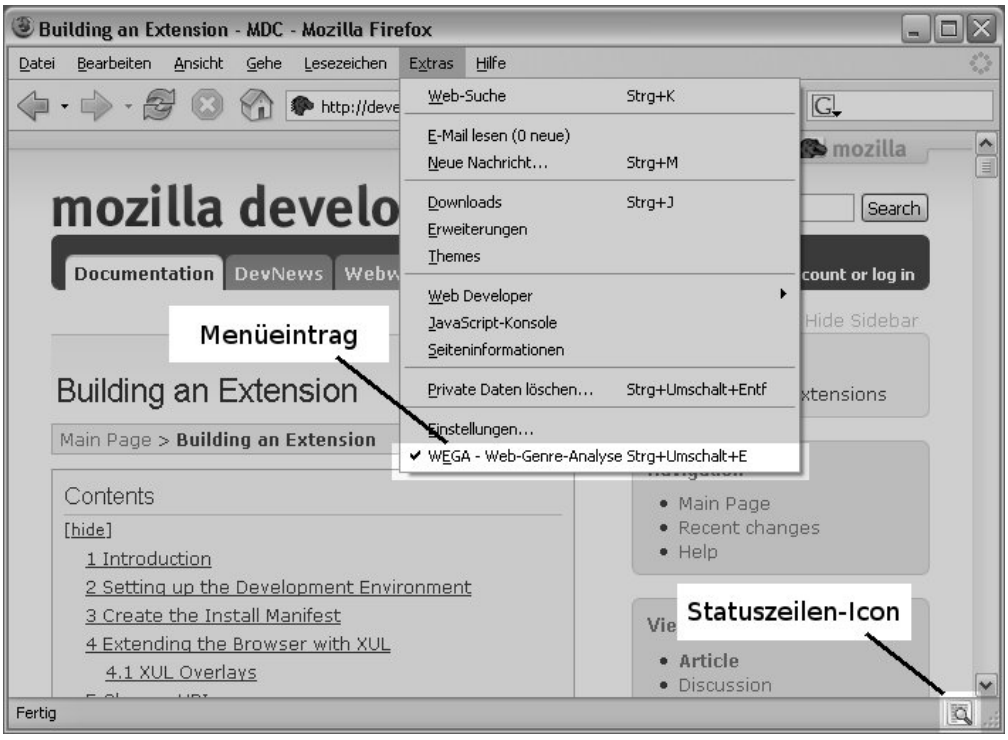

Abbildung 5.5.: Kennzeichnung des Status der Firefox-Erweiterung WEGA

Die Genre-Analyse wird auf Basis der Diskriminanzanalyse (siehe auch Abschnitt 3.3.2) durchgeführt. Mithilfe der Beispieldokumente des Korpus und der selektierten Feature-Menge wurden mittels SPSS die Diskriminanzfunktionen zur Trennung der Genres bestimmt. Die dabei ermittelten Koeffizienten der jeweiligen Diskriminanzfunktionen für die Gewichtung der einzelnen Features wurden in der Tabelle mit den kanonischen Diskriminanzfunktionskoeffizienten aufgelistet. Diese Werte wurden extrahiert und in einer Datei auf dem Server abgelegt. Ebenso wurden bei der Diskriminanzanalyse die mittleren Diskriminanzwerte von jedem Genre bestimmt. Die Ergebnisse stehen in der Tabelle „Funktionen bei den Gruppenzentroiden“, welche auch auf dem Server gespeichert wurde. Die Tabellen 5.1 und 5.2 zeigen vereinfacht zwei Beispieltabellen, die auf dem Server abgelegt sein könnten.

Feature	f_1	f_2
x	0,5	0,3
y	−0,4	−0,1
z	0,2	0,4
(Konstant)	−0,55	0,45

Tabelle 5.1.: Kanonische Diskriminanzfunktionskoeffizienten

Genre	f_1	f_2
A	−2,4	1,7
B	2,7	2,1
C	0,1	1,6

Tabelle 5.2.: Funktionen bei den Gruppen-Zentroiden

Bei der Klassifikation eines neuen Dokumentes werden die Werte der jeweiligen Diskriminanzfunktionen berechnet. Für die Beispieltabelle 5.1 ergeben sich folgende zwei Funktionen:

$$\begin{aligned} f_1 &= 0,5 * x - 0,4 * y + 0,2 * z - 0,55 \\ f_2 &= 0,3 * x - 0,1 * y + 0,4 * z + 0,45 \end{aligned} \tag{5.1}$$

Vom FeCo-Package werden die entsprechenden Koeffizienten mit den Feature-Werten multipliziert und aufsummiert. Die ermittelten Diskriminanzfunktionswerte beschreiben die Position eines Dokumentes im mehrdimensionalen Raum. Für ein Beispieldokument D mit den Feature-Werten (3; 1,5; 0,75) ergeben sich folgende Diskriminanzfunktionswerte.

$$\begin{aligned} f_1 &= 0,5 * 3 - 0,4 * 1,5 + 0,2 * 0,75 - 0,55 &= 0,5 \\ f_2 &= 0,3 * 3 - 0,1 * 1,5 + 0,4 * 0,75 + 0,45 &= 1,5 \end{aligned} \tag{5.2}$$

Zur Bestimmung des Genres werden nun die Euklidischen Distanzen zwischen der Position des zu klassifizierenden Dokumentes und den jeweiligen Zentroiden eines Genres

berechnet und miteinander verglichen. Das Genre des Zentroiden, zu dem der Abstand minimal ist, entspricht dem Genre eines Dokumentes. Für das Beispiel ergeben sich die folgende Distanzen d:

$$\begin{aligned} d(A,D) &= \left|\begin{pmatrix}-2,4\\1,7\end{pmatrix} - \begin{pmatrix}0,5\\1,5\end{pmatrix}\right| &= 2,9 \\ d(B,D) &= \left|\begin{pmatrix}2,7\\2,1\end{pmatrix} - \begin{pmatrix}0,5\\1,5\end{pmatrix}\right| &= 2,3 \\ d(C,D) &= \left|\begin{pmatrix}-0,1\\1,6\end{pmatrix} - \begin{pmatrix}0,5\\1,5\end{pmatrix}\right| &= 0,6 \end{aligned} \tag{5.3}$$

Das Dokument D gehört demnach zu Genre C.

5.3. Servlet

Damit das FeCo-Package Anfragen aus dem Internet bearbeiten kann, wurde ein Servlet implementiert. Dabei handelt es sich um ein Java-Programm, welches in einer auf einem Web-Server integrierten Java Virtual Machine läuft. Der Vorteil eines Servlets besteht darin, dass es über eine URL aufgerufen werden kann. Das ermöglicht es, beispielsweise mit JavaScript-Code eine Anfrage zu formulieren.

Das Servlet nimmt die Anfragen der Firefox-Erweiterung entgegen und beantwortet sie unter Zuhilfenahme des erweiterten FeCo-Packages. Als Protokoll wurde HTTP gewählt, da es sich bei den Anfragen mittels Ajax um einen so genannten *XMLHTTPRequest* handelt.

5.3.1. Initialisierung

Beim Start des Servlets werden die Parameter der Genre-Analyse aus einer Datei eingelesen. In dieser Konfigurationsdatei stehen beispielsweise die Sprache, welche mit dieser Einstellung klassifiziert werden kann, und die zu bestimmenden Features.

Bei der Initialisierung werden bereits zeitaufwendige Prozesse durchgeführt, um später eine Anfrage schneller verarbeiten zu können. So werden bereits einige der benötigten Instanzen des FeCo-Packages erzeugt und damit unter anderem die ausgewählten Wörterbücher eingelesen.

5.3.2. Verarbeitung einer Anfrage

In der Abbildung 5.6 wird die Verarbeitung eines HTTP-Requests mithilfe eines Sequenz-Diagramms veranschaulicht. Die einzelnen Arbeitsschritte werden im Anschluss näher erläutert.

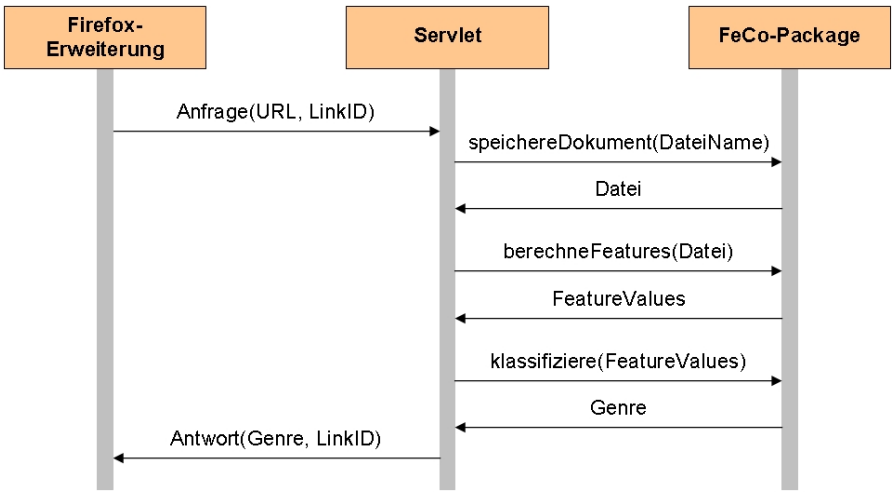

Abbildung 5.6.: Sequenz-Diagramm: Klassifizierung eines Web-Dokumentes

Erhält das Servlet eine Anfrage, werden zunächst die Parameter URL und Link-ID heraus gefiltert. Das Listing 5.2 zeigt ein Beispiel für einen HTTP-Request. Die zu klassifizierende Seite ist *http://www.ibm.com* und die Link-ID ist *fc1*.

```
http://.../servlets/genre?url=http://www.ibm.com&linkID=fc1
```

Listing 5.2: Beispiel für einen HTTP-Request

Da die Link-ID nur von der Firefox-Erweiterung für die Zuordnung des ermittelten Genres zum korrespondierenden Link in der Google-Ergebnisliste benötigt wird, ist dieser Parameter optional.

Das Servlet generiert aus der URL einen eindeutigen Namen, unter dem die Seite temporär auf dem Server gespeichert wird. Diese Zwischenspeicherung des Web-Dokumentes dient zur Beschleunigung der Feature-Gewinnung, denn je nach Auswahl der Features muss der Text des Dokumentes mehrmals geparst werden.

Wurde die Datei erfolgreich heruntergeladen, wird die Sprache des Dokumentes ermittelt. Das ist wichtig, weil bei der Klassifizierung auch Features benutzt werden, die sprachabhängig sind. Mit der in der Evaluation ermittelten Feature-Menge (siehe

Abschnitt 4.3.3) kann das Genre nur für englische Seiten bestimmt werden.

Die Spracherkennung erfolgt mittels so genannter Stoppwörter. Diese Wörter haben kaum einen Einfluss auf den Inhalt eines Textes, kommen aber sehr häufig in einem Text vor und sind deshalb für die Sprachbestimmung geeignet. Zu den Stoppwörtern zählen beispielsweise Präpositionen und Artikel. Zur Bestimmung der Sprache werden maximal die ersten 500 Wörter des Textes einer Seite benutzt. Besteht der Text jedoch aus weniger als 50 Wörtern, wird abgebrochen, da dann die Spracherkennung unzuverlässig wird und auch eine zuverlässige Klassifizierung nicht mehr möglich wäre.

Um die Sprache zu bestimmen, werden die Stoppwörter der jeweiligen Sprache, die im Text vorkommen, gezählt. Die Stoppwortliste, zu der es die meisten Übereinstimmungen gab, ist die Sprache, die im Web-Dokument verwendet wurde. Werden aus keiner Sprache mehr als fünf Stoppwörter gefunden, so kann die Sprache nicht festgelegt werden und die Seite ist wiederum nicht klassifizierbar. Wurde eine englischsprachige Seite erkannt, werden die Features bestimmt.

Anhand der Feature-Werte eines Web-Dokumentes wird mithilfe des erweiterten FeCo-Packages das zugehörige Genre ermittelt (siehe Abschnitt 5.2).

Die Antwort des Servlets (mit Genre und Link-ID) wird als XML-Datei an den Browser gesendet. Das Listing 5.3 zeigt als Beispiel das Ergebnis auf den Request aus Listing 5.2. Das ermittelte Genre ist dabei *portrayal(non-priv)*, eine nicht private Portrait-Seite.

```
<genre>
    <linkID>fc1</linkID>
    <name>portrayal(non_private)</name>
    <error> </error>
</genre>
```

Listing 5.3: Beispiel für eine XML-Antwortdatei

Es besteht auch die Möglichkeit, über die XML-Datei mögliche Fehler, die bei der Genre-Analyse aufgetreten sind, zu übermitteln. Dann bekommt das ermittelte Genre den Wert *0*, der Fehler steht im Tag *<error>* und die Firefox-Erweiterung zeigt, wie in Abbildung 5.3 zu sehen, die Art des Fehlers an. Es gibt drei mögliche Fehlerarten:

- **download error**
  Zu dieser Art gehören alle Fehler, die während des Downloads auftreten können, wenn beispielsweise die referenzierte Seite nicht verfügbar ist oder vom Server der

Zugriff verweigert wird. Auch wenn das Herunterladen des Web-Dokumentes länger dauert als fünf Sekunden[1], wird der Download-Fehler angezeigt.

- **unsupported language**
  Diese Meldung bedeutet, dass eine Genre-Analyse für die Sprache der jeweiligen Internet-Seite nicht verfügbar ist. Sie wird derzeit immer ausgegeben, wenn es sich um eine Seite mit nicht englischem Inhalt handelt.

- **not classifiable**
  Nicht klassifizierbar sind Web-Dokumente mit weniger als 50 Wörtern im Text. Mit diesem Fehler werden auch Seiten indiziert, die einen sehr hohen dynamischen Anteil haben und der Text beispielsweise fast nur aus JavaScript heraus generiert wird.

Zum Abschluss der Genre-Analyse wird vom Servlet das temporär gespeicherte Web-Dokument auf dem Server gelöscht.

5.4. Zusammenfassung

Da bei einer Google-Suchanfrage die einzelnen Einträge des Ergebnisses klassifiziert und das ermittelte Genre angezeigt werden soll, wurde für den Firefox-Browser die Erweiterung WEGA implementiert. Sie erkennt automatisch eine Google-Suche, filtert die Links der Ergebnisseite heraus und sendet diese zum Analysieren des Genres mittels des Ajax-Konzeptes an einen Server. Durch die Markierung der „Fundstellen“ im DOM-Baum der Google-Ergebnisseite kann das in der Antwort übermittelte Genre hinter dem entsprechenden Eintrag angezeigt werden.

Damit das Genre eines Web-Dokumentes ermittelt werden kann, wurde das FeCo-Package erweitert und die Klassifikation mittels der Diskriminanzanalyse implementiert. Ebenso wurde ein Servlet programmiert, welches über eine URL aufgerufen wird und das Genre mittels des FeCo-Packages bestimmt. Das Servlet speichert temporär das Dokument, berechnet die benötigten Features und ermittelt schließlich mithilfe der Diskriminanzanalyse das zugehörige Genre. Das Ergebnis der Klassifikation wird in einem XML-Dokument verpackt und als Antwort an die Firefox-Erweiterung zurückgeschickt. Durch die in der URL angegebene Link-ID kann eine Serveranfrage eindeutig einer entsprechenden Antwort zugeordnet werden.

Da bei der verwendeten Genre-Analyse auch Features benutzt werden, die sprachabhängig sind, können derzeit nur englische Internet-Seiten klassifiziert werden.

[1] Die maximale Download-Dauer ist ein Parameter des Servlets und kann verändert werden.

Die entwickelte Firefox-Erweiterung stellt eine praktische Anwendung der Genre-Analyse dar, die nun einer Vielzahl von Nutzern zur Verfügung steht. Das Servlet wird auf einem Server zur Verfügung gestellt und kann so über eine URL angesprochen werden. Somit ist die Klassifikation von Web-Dokumenten generell für jede Seite einsetzbar und könnte für Anfragen weiterer Programme verwendet werden.

6. Zusammenfassung und Ausblick

Die Informationsflut im World Wide Web nimmt stetig zu. Trotz ausgefeilter Suchmaschinen-Technologie wird es immer schwieriger, an die gewünschten Informationen zu kommen. Das Resultat einer Suchmaschine ist eine Liste mit den gefundenen Seiten. Für den Suchenden ist es sehr schwer, anhand des Eintrags zu entscheiden, ob ein Ergebnis für ihn relevant ist oder nicht. Eine Möglichkeit, diese Entscheidung zu unterstützen, stellt die automatische Genre-Analyse von Web-Dokumenten dar.

Das Genre eines Web-Dokumentes beschreibt die Form und den Typ eines Dokumentes. Die Form beschreibt den Inhalt der Seite und der Typ die Präsentation des Inhaltes. So beinhalten Seiten zum gleichen Thema beispielsweise einen langen Text, andere eine Linksammlung oder ein Diskussionsforum. Das heißt, eine Genre-Klassifikation erfolgt unabhängig vom Thema.

Für die Operationalisierung der Genre-Analyse werden Beispieldokumente zusammengetragen und manuell jeweils ihrem Genre zugeordnet. Aus den Dokumenten werden Features extrahiert, anhand derer eine Menge von Diskriminanzfunktionen lernt das Genre eines Web-Dokumentes erkennen. Mithilfe dieser Funktionen kann ein Klassifizierer später neue unbekannte Beispiele eingruppieren.

Das Ziel dieser Arbeit bestand darin, die automatische Klassifikation von Web-Dokumenten zu verbessern und sie in Echtzeit zu ermöglichen. So waren Features zu bestimmen, welche die einzelnen Genres zuverlässig repräsentieren, um so beliebige Web-Dokumente sicher klassifizieren zu lassen. So wurde zunächst der Korpus zusammengestellt und anschließend die Beispiele der jeweiligen Genres analysiert. Dabei stellte sich unter anderem heraus, dass der Titel, die URL und die Meta-Tags einer Internet-Seite bereits Hinweise auf das zugehörige Genre liefern können. Eine Software namens FeCo wurde entwickelt, die nicht nur die klassischen Features bestimmt, sondern die zusätzlichen Informationen aus Titel, URL und Meta-Tag nutzt und daraus Features berechnet.

Für die Klassifikation der Web-Dokumente wurde die Diskriminanzanalyse verwendet. Bei der anschließenden Evaluation erwiesen sich die neuen Features als sehr hilfreich und steigerten die durchschnittliche Klassifizierungsrate um 8,7 %. Um die Genre-Analyse

in Echtzeit zu ermöglichen, wurden nur die Features ausgewählt, die sich besonders gut zur Klassifikation der verwendeten Genres eignen und mit sehr geringem Berechnungsaufwand zu bestimmen sind. Deshalb wurden die zeitkritischen Part-of-Speech-Features aus der Feature-Menge entfernt. Damit wurde zwar die resultierende durchschnittliche Klassifizierungsrate von anfänglich 82,0 % auf 80,0 % gesenkt, doch der zeitliche Aufwand für die Berechnung der Eigenschaften wurde um mehr als die Hälfte verringert.

Damit dem Internet-Nutzer die Genre-Analyse auch für die Suche im Netz zur Verfügung steht, wurde die Firefox-Erweiterung WEGA implementiert. Die Erweiterung erkennt automatisch eine Google-Suchanfrage, lässt die einzelnen Einträge der Ergebnisseite klassifizieren und versieht sie anschließend mit dem entsprechenden Genre. Hierzu sendet WEGA die jeweilige URL eines Eintrags an ein Servlet, welches ebenso im Rahmen dieser Arbeit implementiert wurde. Dieses Servlet klassifiziert mittels dem erweiterten FeCo-Package das Web-Dokument und sendet die Klasse in einer XML-Datei an den Browser zurück. Die Firefox-Erweiterung filtert das Genre aus der Datei und zeigt es an entsprechender Stelle an.

Aufgrund der Verwendung von sprachspezifischen Eigenschaften kann die Genre-Analyse nur für englische Seiten angeboten werden. Aufbauend auf diese Arbeit bietet sich nun die Untersuchung an, inwieweit sich die selektierten Features nach entsprechender Sprachanpassung für die Klassifikation anderssprachiger Seiten eignen. Da die Sprache eines Dokumentes automatisch vom Servlet erkannt wird, könnte der neue Klassifizierer einfach eingebunden werden. Das hätte den Vorteil, dass, wären in einer Ergebnisliste einer Google-Suche Seiten unterschiedlicher Sprachen aufgelistet, alle Einträge klassifiziert werden könnten, ohne dass der Benutzer Einstellungen ändern müsste.

Eine Benutzerstudie könnte Aufschluss darüber geben, wie hilfreich diese Firefox-Erweiterung ist und wie nützlich die ausgewählten Genres wirklich sind. Da die Klassifikationsergebnisse bei kleinerer Genre-Anzahl besser sind, könnte auch eine Personalisierung in der Art angeboten werden, dass ein Benutzer Genres, die für ihn weniger wichtig sind, zu einem zusammenfassen kann.

Desweiteren könnte eine Internet-Suche so verändert werden, dass zusammen mit dem Suchbegriff direkt die gewünschten Genres mit angegeben werden könnten, welche somit die Reihenfolge der Ergebnisse beeinflussen. So würden für den Suchenden die interessantesten Ergebnisse zuerst angezeigt und er käme schneller an die gewünschte Information.

A. Evaluationsergebnisse

Feature Set	Durchsatz in Kilobyte pro Sekunde	
	Alle Features einer Klasse	Ausgewählte Features einer Klasse
Alle	69,08	85,18
Alle ohne Part-of-Speech	112,48	158,66
Closed-Class Word Sets	283,97	568,40
Window Chunking	216,82	268,83
Html-Tags	683,53	687,30
Part-of-Speech	180,33	181,20
Spelling	1043,33	1043,33
Textstatistik	1386,74	1403,04
Wort-Frequency-Class	1052,34	1068,92

Tabelle A.1.: Durchsatz (kB/s) bei der Berechnung einzelner Feature-Mengen (Werte zur Abbildung 4.6)

Feature-Klasse	Klassifizierungsrate
Alle	82,0%
Closed-Class Word Sets (alle)	72,9%
Closed-Class Word Sets Text	60,6%
Closed-Class Word Sets Titel	54,0%
Closed-Class Word Sets URL	53,0%
Closed-Class Word Sets Meta	32,1%
Window Chunking	61,6%
HTML-Tags	46,7%
Part-of-Speech	46,2%
Spelling	16,7%
Textstatistik	37,5%
Word-Frequency-Class	20,5%

Tabelle A.2.: Vergleich der Klassifizierungsraten der einzelnen Feature-Klassen (Werte zur Abbildung 4.7)

Genre	Klassifizierungsrate
Artikel	77,9%
Diskussion	80,6%
Download	77,0%
Hilfe	76,8%
Linksammlung	66,5%
Portrait (nicht privat)	59,2%
Portrait (privat)	75,9%
Shop	71,1%
Gesamt	72,9%

Tabelle A.3.: Vergleich der Klassifizierungsraten der Closed-Class Word Sets (Werte zur Abbildung 4.8)

	article	discussion	download	help	linklist	portrayal (non private)	portrayal (private)	shop
Artikel	35,3%	7,1%	11,3%	5,7%	4,7%	5,5%	8,2%	6,2%
Diskussion	5,2%	33,1%	11,1%	7,3%	9,1%	5,7%	11,0%	5,2%
Download	5,7%	9,1%	21,4%	5,7%	4,0%	6,3%	6,4%	12,7%
Help	16,9%	10,8%	9,5%	64,2%	6,0%	3,9%	7,7%	6,6%
Linksammlung	17,2%	9,7%	13,4%	3,9%	60,3%	14,3%	7,8%	6,2%
Portrait (nicht privat)	7,8%	7,7%	11,3%	4,2%	6,1%	44,7%	11,0%	11,2%
Portrait (privat)	3,6%	12,9%	12,2%	2,0%	4,4%	10,5%	39,6%	2,2%
Shop	8,4%	9,7%	9,9%	6,9%	5,3%	9,0%	8,2%	49,7%
Summe	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%

Tabelle A.4.: Durchschnittliche Werteverteilung der Features der Genre-spezifischen Closed-Class Word Sets (Spalten) in Bezug auf die Genres (Zeilen), Tabelle zu Abbildung 4.9

B. Liste der in FeCo verfügbaren Features

HTML Tags	All links	Definition list bullet	DIV tags
	Links in a list	Bullet tags	SPAN tags
	Domain links	Bullet list tags	Formular tags
	Ankor links	Table tags	INPUT tags
	Internet links	Table header	BR tags
	Mail links	Table rows	SCRIPT tags
	FTP links	Table columns	Remark tags
	Javascript links	META tags	OPTION tags
	Question links	Headings	Selection tags
	Other links	Text nodes	LABEL tags
	Base HREF tags	STYLE tags	FRAMESET tags
	Images	Form tags	FRAME tags
	Definition list	Paragraph tags	OBJECT tags
Part-of-Speech	Noun	Adjective	Copula
	Verbs	Adverb	To
	Be	Article	Alphanumeric
	Does	Pronoun	Foreign word
	Have	Relative pronoun	Symbol
	Modal	Preposition	Interjection
Spelling Dictionaries	Webster's		
Closed-Class Word Sets			
Text	Article	First names	Months days
	Discussion	Help	Non private
	Download	HTML table	Private
	Family names	Linklist	Shop
Title	Article	First names	Months days
	Discussion	Help	Non private
	Download	HTML table	Private
	Family names	Linklist	Shop
URL	Article	First names	Months days
	Discussion	Help	Non private
	Download	HTML table	Private
	Family names	Linklist	Shop
Meta-Tags	Article	First names	Months days
	Discussion	Help	Non private
	Download	HTML table	Private
	Family names	Linklist	Shop
Window Chunking Text	Article	First names	Months days
	Discussion	Help	Non private

	Download	HTML table	Private
	Family names	Linklist	Shop
Window Chunking Whole Site	Links		
Word Frequency Class	Average word frequency class	Minimal word frequency class	Maximal word frequency class
Text Statistics	Average word length	Apostrophes	Exclamation marks
	Letter	Colons	Question marks
	Capitals	Commas	Semicolon
	Digits	Dots	

Tabelle B.1.: Die mit FeCo berechenbaren Features

Literaturverzeichnis

[1] J.-S. Ahn. Diskriminanzanalyse. *http://wulv.uni-greifswald.de/2005_ah_maf/userdata/Diskriminanzanalyse%20081205.pdf*, 2005. Last Access: 09/10/2006.

[2] H. Baayen, H. Halteren, and F. Tweedie. Outside the cave of shadows: Using syntactic annotation th enhance autorship attribution. *Literary and Linguistic Computing*, 11, 1996.

[3] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999. ISBN: 0-201-39829-X.

[4] Markus Bestehorn. Part-of-Speech Tagging. In *Text Mining: Wissensgewinnung aus natürlichsprachigen Dokumenten*, Interner Bericht 2006-5, pages 59 – 82. Universität Karlsruhe, Fakultät für Informatik, Institut für Programmstrukturen und Datenorganisation (IPD), 2006. ISSN: 1432-7864.

[5] Elizabeth Sugar Boese and Adele E. Howe. Effects of web document evolution on genre classification. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 632 – 639, 2005.

[6] Ivan Bretan, Johan Dewe, Anders Hallberg, Niklas Wolkert, and Jussi Karlgren. Webspecific genre visualization. 1998. presented at WebNet '98, Orlando.

[7] Simon Dennis. The sydney morning herald word database. *http://www2.psy.uq.edu.au/CogPsych/Noetica/OpenForumIssue4/SMH.html*, 1995. Last Access: 22/09/2006.

[8] Nigel Dewdney, Carol VanEss-Dykema, and Richard MacMillan. The form is the substance: classification of genres in text. In *Proceedings of the workshop on Human Language Technology and Knowledge Management*, volume 2001, pages 1 – 8, 2001.

[9] Maya Dimitrova, Aidan Finn, Nicholas Kushmerick, and Barry Smyth. Web genre visualization. In *Conference on Human Factors in Computing Systems*, 2002.

[10] Eclipse Foundation: Eclipse 3.1. *http://www.eclipse.org/*. Last Access: 30/08/2006.

[11] Reginald Ferber. *Information Retrieval: Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web*. Dpunkt Verlag, 2003. ISBN: 3-89864-213-5.

[12] Aidan Finn and Nicholas Kushmerick. Learning to classify documents according to genre. In *IJCAI-03 Workshop on Computational Approaches to Style Analysis and Synthesis (Acapulco)*, 2003.

[13] Aidan Finn and Nicholas Kushmerick. Learning to classify documents according to genre. In *Journal of the American Society for Information Science and Technology (JASIST), Special Issue on Computational Analysis of Style*, volume 7, page 99.3, 2006.

[14] Aidan Finn, Nicholas Kushmerick, and Barry Smyth. Fact or fiction: Content classification for digital libraries. In *Joint DELOS-NSF Workshop on Personalisation and Recommender Systems in Digital Libraries*, 2001.

[15] R. Flesch. A new readability yardstick. *Journal of Applied Psychology*, 32:221–233, 1948.

[16] Merriam-Webster Online Dictionary: Genre. *http://www.m-w.com/cgi-bin/dictionary?book=Dictionary\&va=genre*. Last Access: 03/09/2006.

[17] Google Directory. *http://directory.google.com/*. Last Access: 10/09/2006.

[18] David Heckerman. A tutorial on learning with bayesian networks. 1995. Revised June 96.

[19] HTMLParser von Derrick Oswald. *http://htmlparser.sourceforge.net/*. Last Access: 30/08/2006.

[20] K. Huber. Diskriminanzanalyse. *http://www.wiwi.uni-passau.de/lehrstuehle/schweitzer/doc/Diskriminanzanalyse.ppt*.

[21] Thorsten Joachims. SVM^{light}. *http://svmlight.joachims.org/*. Last Access: 21/09/2006.

[22] Thorsten Joachims. Svm^{multiclass}. *http://svmlight.joachims.org/svm_multiclass.html*. Last Access: 21/09/2006.

[23] Jussi Karlgren and Douglass Cutting. Recognizing text genres with simple metrics using discriminant analysis. In *Proceedings of the 15th conference on Computational linguistics*, volume 2, pages 1071 – 1075, 1994.

[24] Alistair Kennedy and Michael Shepherd. Automatic identification of home pages on the web. In *Proceedings of the 38th Hawaii International Conference on System Sciences*, page 99.3, 2005.

[25] Brett Kessler, Geoffrey Numberg, and Hinrich Schütze. Automatic detection of text genre. In *Proceedings of the 35th annual meeting on Association for Computational Linguistics*, pages 32 – 38, 1997.

[26] Pär Lannerö. Dejavu: The web as we remember it. *http://www.dejavu.org/*. Last Access: 30/08/2006.

[27] 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.

[28] Florian Markowetz. Klassifikation mit Support Vector Machines. *http://lectures.molgen.mpg.de/statistik03/docs/Kapitel_16.pdf*, 2003. Last Access: 11/10/2006.

[29] Oliver Mason. Qtag. *http://www.english.bham.ac.uk/staff/omason/software/qtag.html*. Last Access: 21/09/2006.

[30] Sven Meyer zu Eißen and Benno Stein. Genre classification of web pages: User study and feasibility analysis. In *Proceedings of 27th German Conference on Artificial Intelligence (KI 2004)*, page 256 – 269. Springer LNAI 3228 (2004), 2004.

[31] Tom Mitchell. *Machine Learning*. McGraw Hill, 1997. ISBN: 0-07-115467-1.

[32] Mozilla Developer Center. Adding preferences to an extension: Establish the defaults. *http://developer.mozilla.org/en/docs/Adding_preferences_to_an_extension*. Last Access: 02/10/2006.

[33] Mozilla Developer Center. Building an extension. *http://developer.mozilla.org/en/docs/Building_an_Extension*. Last Access: 02/10/2006.

[34] Netcraft. June 2006 web server survey. *http://news.netcraft.com/*. Last Access: 30/08/2006.

[35] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2-3):103–134, 2000.

[36] Georg Rehm. Towards automatic web genre identification – a corpus-based approach in the domain of academia by example of the academic's personal homepage. In *Proceedings of the 35th Hawai'i International Conference on System Sciences*, 2002.

[37] D. Roussinov, K. Crowston, M. Nilan, B. Kwasnik, J. Cai, and X. Liu. Genre based navigation on the web. In *Proceedings of the 34th Annual Hawaii International*

Conference on System Sciences, page 4013, Washington, DC, USA, 2001. IEEE Computer Society.

[38] Marina Santini. Identification of genres on the web: a multi-faceted approach. In *Proceedings of the 26th European Conference on IR Research*, volume 2, 2004.

[39] Marina Santini. A shallow approach to syntactic feature extraction for genre classification. In *Proceedings of the 7th Annual Colloquium for the UK Special Interest Group for Computational Linguistics*, 2004.

[40] Marina Santini. ßtate-of-the-art on automatic genre identification". Technical report, 2004.

[41] Marina Santini. Genres in formation? an exploratory study of web pages using cluster analysis. In *Proceedings of the 8th Annual Colloquium for the UK Special Interest Group for Computational Linguistics*, 2005.

[42] Marina Santini. Linguistic facets for genre and text type identification: A description of linguistically-motivated features. Technical report itri-05-02, ITRI, University of Brighton (UK), 2005.

[43] Marina Santini. Interpreting Genre Evolution on the Web: Preliminary Results. EACL 2006 Workshop: NEW TEXT - Wikis and blogs and other dynamic text sources, 2006.

[44] SPSS. Statistik- und Analyseprogramm. *http://www.spss.com/de/spss/*. Last Access: 03/09/2006.

[45] E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Text genre detection using common word frequencies. In *Proceedings of the 18th conference on Computational linguistics - Volume 2*, pages 808 – 814, 2000.

[46] Benno Stein. Web-Technologie II (WS2005/06): Bayes-Klassifikation, Bauhaus-Universität Weimar. *http://www.uni-weimar.de/medien/webis/teaching/lecturenotes/webtec-advanced/part-machine-learning/unit-bayesian-learning.ps.pdf*, 2005. Last Access: 11/10/2006.

[47] Benno Stein. Web-Technologie II (WS2005/06): Neuronale Netze , Bauhaus-Universität Weimar. *http://www.uni-weimar.de/medien/webis/teaching/lecturenotes/webtec-advanced/part-machine-learning/unit-neural-networks.ps.pdf*, 2005. Last Access: 11/10/2006.

[48] Benno Stein. Web-Technologie II (WS2005/06): Retrieval-Modelle , Bauhaus-Universität Weimar. *http://www.uni-weimar.de/medien/webis/*

teaching/lecturenotes/webtec-advanced/part-information-retrieval/unit-rm-term-based.ps.pdf, 2005. Last Access: 11/10/2006.

[49] Dr. Johannes Steinmüller. Maschinelles Lernen (WS 2006/2007). *http://www-user.tu-chemnitz.de/%7Estj/lehre/masch.pdf*. Last Access: 09/10/2006.

[50] Inc. Sun Microsystems. Java 2 Platform Standard Edition (J2SE) 5.0. *http://java.sun.com/j2se/1.5.0/*. Last Access: 30/08/2006.

[51] A. Smola und B. Schoelkopf. A tutorial on support vector regression. *NeuroCOLT2 Technical Report NC2-TR-1998-030*, 1998.

[52] Universität Leipzig. Wortschatz. *http://wortschatz.uni-leipzig.de/html/faq/hkl.html*. Last Access: 06/10/2006.

[53] W3C. Line Based Browser. *http://www.w3.org/History/19921103-hypertext/hypertext/WWW/LineMode/Browser.html*. Last Access: 30/08/2006.

[54] W3C. Tim Berners-Lee. *http://www.w3.org/People/Berners-Lee/*. Last Access: 30/08/2006.

[55] Weka. Data Mining Software in Java. *http://www.cs.waikato.ac.nz/~ml/weka/*. Last Access: 21/09/2006.

[56] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, 2000. ISBN: 0-12-088407-0.

[57] Maria Wolters and Mathias Kirsten. Exploring the use of linguistic features in domain and genre classification. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, pages 142 – 149, 1999.

[58] XML User Interface Language. *http://www.mozilla.org/projects/xul/*. Last Access: 02/10/2006.

[59] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, 1997.