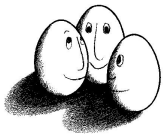
Diplomarbeit

Filterung von Ergebnislisten von Suchmaschinen

Norbert Basmaci



Diplomarbeit
am Fachbereich Informatik
der Universität Dortmund

Dortmund, 31. Oktober 2006

Betreuer:

Prof. Dr. Katharina Morik
Dipl.-Inform. Timm Euler

Danksagung

Ich möchte mich an dieser Stelle herzlich bei meinen Betreuern Katharina Morik und Timm Euler sowie den weiteren Mitarbeitern des Lehrstuhls für künstliche Intelligenz für ihre Kommentare, Ratschläge und Hilfestellungen bei der Anfertigung dieser Arbeit bedanken.

Ebenso danken möchte ich meiner Freundin und meinen Freunden für das Korrekturlesen dieser Arbeit.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1. Einleitung

Das Internet besteht aus einer nahezu unzählbar großen Anzahl an Webseiten. Je nach aktuellem Interesse des Nutzers beinhalten diese Webseiten mehr oder weniger hilfreiche Informationen. Das Auffinden von Informationen geschieht dabei häufig über Internet-Suchmaschinen, allen voran Google (*www.google.de*), dessen Index nach eigenen Angaben[1] mehr als acht Milliarden URLs umfasst.

Internet-Suchmaschinen beschränken sich in der Regel darauf, Webseiten sortiert aufzulisten, die den bzw. die gesuchten Begriffe (im Folgenden wird von mehreren Begriffen ausgegangen) enthalten oder mit ihnen in Zusammenhang gebracht werden können, beispielsweise durch die Beschriftung der Links von anderen Webseiten. Die Reihenfolge, in der die gefundenen Treffer angezeigt werden, wird dabei nach bestimmten Kriterien errechnet, z. B. nach Position und Häufigkeit der Suchbegriffe im Text oder nach der Anzahl der Links von anderen Webseiten.

Nachteil dieser sogenannten Volltextsuche ist, dass die gefundenen Webseiten die Suchbegriffe zwar enthalten, viele von ihnen dem Nutzer bei seiner aktuellen Anfrage aber nicht weiterhelfen. So sucht ein Nutzer mit einer Suche nach „Fernseher kaputt“ vielleicht nach Reparaturwerkstätten für TV- und HiFi-Geräte, bekommt aber in erster Linie Foren gelistet, die ihm Tipps bei bestimmten Problemen geben. In einem anderen Fall könnte er sich aber gerade für diese Webseiten interessieren, um die Reparatur mit Ratschlägen und Informationen aus einem Forum selbst in die Hand zu nehmen. Ein anderer, ähnlicher Fall wäre die Suche nach einem bestimmten Mobiltelefon. Sie liefert sowohl Webseiten, die über das Mobiltelefon informieren und Testergebnisse oder Erfahrungen anderer Personen beschreiben, als auch Webseiten von Onlineshops, die es lediglich zum Verkauf anbieten.

Mit Methoden der maschinellen Textklassifikation ist es sicherlich möglich, Webseiten nach vorher vom Benutzer definierten Kriterien zu klassifizieren und so Treffer auszublenden, die den Nutzer bei seiner aktuellen Anfrage nicht interessieren. Es stünde ihm dann eine kürzere Trefferliste zur Verfügung, in der sich die gewünschten Informationen schneller finden ließen.

Eine weitere Kategorie von Webseiten, durch deren Ausblendung man dem Nutzer einiges an Arbeit und Zeit beim Suchen ersparen könnte, sind Webseiten, die sich inhaltlich nur wenig bis kaum unterscheiden. So spiegeln viele Webseiten einfach nur das Open Directory Project (*www.dmoz.org*) und bereiten die Inhalte mit eigenem Layout und eventuell zusätzlichen Vorschaubildern auf. Weitere Gründe für Spiegelungen sind beispielsweise eine bessere Ressourcenverteilung oder ein-

[1] www.google.de/intl/de/why_use.html

fach nur der Wunsch eines Webseitenbetreibers, mehrmals in den Trefferlisten der Internet-Suchmaschinen zu stehen und so die Besucherzahl der eigenen Webseite(n) zu erhöhen. Dabei ist es vom Nutzer einer Internet-Suchmaschine nur selten gewünscht, unter den Treffern mehrmals eine inhaltlich gleiche Webseite in eventuell etwas veränderter Darstellung zu finden.

Bedingt durch den starken Einsatz von Internet-Suchmaschinen versuchen viele Webseitenbetreiber, ihre Webseiten möglichst hoch in den Trefferlisten zu platzieren und an möglichst viele Suchbegriffe anzupassen, um mehr Benutzer auf ihre Webseiten zu locken. Dabei werden die unterschiedlichsten Methoden verwendet, z. B. mehrfaches, sinnloses Aufzählen von Begriffen am Ende der Webseite, oft auch in der Farbe des Hintergrundes, um sie dann vor dem Besucher zu verstecken. Oder es finden sich Webseiten in der Ergebnisliste, die selbst wiederum nur Internet-Suchmaschinen sind. Kaum ein Benutzer wird einen Suchbegriff in eine Internet-Suchmaschine eingeben, um dann bei einer anderen Internet-Suchmaschine landen zu wollen. Derartige Webseiten beinhalten für den Nutzer in den meisten Fällen keine relevanten Inhalte und beanspruchen unnötig Zeit bei der Suche, da er eine Webseite zunächst betrachten muss, um festzustellen, dass sie völlig uninteressant ist.

Im Rahmen dieser Diplomarbeit werden Methoden untersucht, mit denen sich die Ergebnislisten von Internet-Suchmaschinen so reduzieren lassen, dass dem Nutzer in erster Linie Treffer präsentiert werden, die von ihm bei seiner aktuellen Anfrage auch gewünscht sind. Die Arbeit konzentriert sich dabei auf Google als derzeitigen Marktführer, die verwendeten Methoden lassen sich in gleicher Weise auch auf andere Internet-Suchmaschinen anwenden.

Google liefert in seiner Trefferliste zu jedem Eintrag die Internetadresse und den Titel der Webseite sowie einen kleinen Ausschnitt aus der Webseite, in dem die Suchbegriffe enthalten sind. Eine Filterung der Trefferliste allein anhand dieser Informationen ist in wesentlich kürzerer Zeit als eine Filterung anhand des gesamten Webseite möglich, da hierfür nicht die einzelnen Webseiten heruntergeladen werden müssen und wesentlich weniger Informationen zu verarbeiten sind. Die Frage, die sich hier stellt und in späteren Untersuchungen verfolgt wird, ist, ob und wie sich eine Filterung über den gesamten Inhalt der Webseite von einer Filterung ausschließlich über die von Google gelieferten Informationen unterscheidet. Ist ein Filtern allein anhand der Informationen aus der Trefferliste möglich, oder müssen die Webseiten heruntergeladen und verarbeitet werden?

1.1. Aufbau der Arbeit

In Kapitel 2 werden die eben genannten Probleme genauer erläutert und mit praktischen Beispielen unterlegt. Es werden Beispielklassen beschrieben, für die eine Filterung der Ergebnisliste untersucht und bewertet wird.

Da es sich bei einer Einteilung der Suchtreffer in interessante und uninteressante

Webseiten in erster Linie um Textklassifikation in zwei Klassen handelt, führt Kapitel 3 kurz in diese Thematik ein und geht etwas näher auf das später zur Anwendung kommende Verfahren der Support Vector Machines ein.

Kapitel 4 beschreibt zusätzliche Merkmale von Webseiten, die als Ergänzung zum reinen Textinhalt bei einer Klassifikation hilfreich sein können.

In Kapitel 5 werden Versuche zur Klassifikation von Webseiten durchgeführt. Zunächst wird die Entwicklung eines Programms mit grafischer Benutzeroberfläche beschrieben, mit dessen Hilfe Suchtreffer von Hand klassifiziert und gesammelt werden. Anschließend wird die zum Einsatz kommende Testumgebung beschrieben, der Ablauf der Versuche erläutert und die Ergebnisse ausgewertet.

Mit der Filterung inhaltlich ähnlicher Webseiten beschäftigt sich Kapitel 6. Dort wird ein Verfahren beschrieben, mit dem in Kapitel 7 Versuche zur Erkennung inhaltlich ähnlicher Webseiten durchgeführt und ausgewertet werden.

Kapitel 8 fasst die gewonnenen Erkenntnisse zusammen, nennt weitere mögliche Verbesserungsvorschläge und gibt Anregungen für eine praktische Umsetzung.

2. Problemklassen

Google liefert als Volltextsuchmaschine alle Webseiten, die die gesuchten Begriffe enthalten. Eine Einschränkung der Volltextsuche nach bestimmten Themengebieten ist, von einer Suche nach Bildern abgesehen, nicht möglich. Mit Google-News existiert seit einiger Zeit zwar auch eine Suche nach Nachrichten, die dabei durchsuchten Quellen sind allerdings vorgegeben, so dass sich mit dieser Funktion nicht das gesamte Internet durchsuchen lässt (für deutschsprachige Nachrichten benutzt Google 700 Nachrichtenquellen[1]).

Die zu einer Suche gefundenen Webseiten sortiert Google zur Auflistung nach dem eigens eingeführten „PageRank“, einem von den Google Gründern Sergey Brin und Larry Page entwickelten Rankingverfahren. Hierbei werden Webseiten nach der Anzahl der Links von anderen Webseiten, sogenannten Backlinks, bewertet, wobei ankommende Links von höher bewerteten Webseiten stärker gewichtet werden. Hintergrund des Verfahrens ist, dass Google das Setzen eines Links auf eine Webseite als ein Votum für diese Webseite interpretiert, so dass das Internet quasi über die Relevanz von Webseiten abstimmt[2].

Absicht dieser Diplomarbeit ist es nicht, die eigentliche Such- und Rankingfunktion von Google zu verbessern. Abgesehen davon, dass Google hier keinerlei Eingriffsmöglichkeit gewährt, ist das grundsätzliche Vorgehen, das gesamte Internet einzucrawlen und anschließend alle Webseiten sortiert auszuliefern, die die Suchbegriffe enthalten, sicher nicht unangebracht. Der Erfolg der auf dem PageRank basierenden Internet-Suchmaschine spricht außerdem für das Ranking über Backlinks.

Vielmehr sollen hier Möglichkeiten untersucht werden, die von Google sortiert ausgegebene Ergebnisliste nach den Bedürfnissen des Benutzers zu reduzieren, um unnötige Seitenaufrufe zu vermeiden. Im Folgenden werden drei Problemklassen diskutiert, die häufig zu unnötigen Seitenaufrufen führen. Es lassen sich auch weitere Bereiche finden, in denen eine maschinelle Vorauswahl sinnvoll wäre.

2.1. Benutzerpräferenzen

Eine Einschränkung auf eine bestimmte Kategorie oder ein Themengebiet würde die Trefferliste, je nach Kategorie und verwendeten Suchbegriffen, in vielen Fäl-

[1] www.google.de/intl/de_de/about_google_news.html
[2] www.google.de/intl/de_de/why_use.html

len deutlich reduzieren und so dem Benutzer das Auffinden von für ihn relevante Webseiten erheblich erleichtern.

Im Folgenden werden mögliche Kategorien vorgestellt, nach denen eine Trennung der Ergebnisliste denkbar ist. Es lassen sich aber beliebig weitere Kategorien festlegen.

2.1.1. Forenseite

Als (Internet-)Forum wird eine virtuelle Plattform im Internet bezeichnet, auf der sich Nutzer austauschen Können. Nutzer können dort eigene Probleme schildern und Fragen öffentlich stellen, zu denen andere Benutzer Hilfestellungen und eigene Erfahrungen liefern können. Der Austausch geschieht dabei über das Verfassen von Beiträgen, die meist zeitlich sortiert auf einer Webseite abgelegt und von anderen Nutzern eingesehen werden können. Eine solche Diskussion wird in Internetforen als *Thread* bezeichnet, ein einzelner Beitrag heißt *Post*.

Vor allem zu technischen, aber auch zu einer Vielzahl von anderen Themen, bieten sich im Internet etliche Foren mit einer Vielzahl von Nutzern an. Da der gesamte Verlauf der Diskussion öffentlich einsehbar ist, ist es bei Problemen oft hilfreich, nach Foren zu suchen, in denen das Problem bereits diskutiert wurde und Ratschläge vorliegen.


Telebau **Telnet** DSL LAN DSL-Modem Preisvergleich - Preise bei idealo.de
Telebau **Telnet** DSL LAN DSL-Modem: Preis ab 79,95 EUR (17.02.2006) - ... Preisvergleich für DSL-Modem AVM **FRITZ!Box** Fon 5050 · Preisvergleich für AVM **FRITZ!** ...
www.idealo.de/preisvergleich/ OffersOfProduct/131132.html - 51k - Im Cache - Ähnliche Seiten
[Weitere Ergebnisse von www.idealo.de]

Computerhilfen.de: Hilfe: Neue Funktion für AVM **Fritz Box**
Also wenn du per AVM **Fritz Box** im Netzwerk telefonierst, können selbst Tools ... von Windows oder Linux und starten Sie **telnet** mit der IP-Adresse der **Box**, ...
www.computerhilfen.de/hilfen-2-98453-0.html - 67k - Im Cache - Ähnliche Seiten

FRITZ BOX 2030 DSL-ROUTER - Internet - für 92.53 EUR
Der Router **FRITZ BOX** 2030 DSL-ROUTER für 92.53 EUR überzeugt mit ... PPPoE Remoteverwaltungsprotokoll: SNMP, **Telnet** Datenübertragungsrate: 100 Mbps ...
www.internet-traffic.name/ internet-shop/router/router-281.php - 14k -
Im Cache - Ähnliche Seiten

Sprachqualität bei **Fritz Box** Fon WLAN 7050..... - onlinekosten.de ...
geht irgendwie über **telnet**, glaub ich. ich hab keine **fritzbox** also kann ich dir dazu leide nix genaueres sagen, sorry. Aber ein thema dazu war schon mal, ...
www.onlinekosten.de/forum/showthread.php?t=72679 - 61k - Im Cache - Ähnliche Seiten

Abbildung 2.1.: Google-Ausschnitt zur Suche nach „Fritz Box TelNet“

Abbildung 2.1 zeigt die Suche nach „Fritz Box TelNet“, mit der ein Nutzer vielleicht Informationen darüber erhalten möchte, wie er über das Netzwerkprotokoll TelNet Kontakt zu seinem Fritz!Box genannten DSL-Router aufnehmen kann. Da die Verwendung dieses Protokolls vom Hersteller der Fritz!Box für den Nutzer nicht vorgesehen und im Handbuch demnach nicht dokumentiert ist, empfiehlt es sich, eines der zahlreichen Foren aufzusuchen, in denen die Freischaltung und

Verwendung dieser Funktion ausführlich beschrieben ist. Die Suche liefert sowohl Forenseiten, die Anleitungen zur Freischaltung und Verwendung von TelNet geben, als auch Webseiten mit Produktinformationen, die bei der Suche nach Foren stören und unnötig aufhalten. Durch eine Reduzierung der Trefferliste auf Forenseiten würde dem Nutzer eine kürzere Trefferliste zur Verfügung stehen, in der sich die gewünschten Informationen schneller finden lassen.

2.1.2. Nachrichtenseiten

Eine andere Kategorie für eine Einschränkung der Trefferliste bilden Nachrichtenseiten. Bei einer Suche nach einem Land könnte sich ein Nutzer für Information über das Land und die Bevölkerung, in einem anderen Fall vielleicht eher für Nachrichten interessieren, die dieses Land betreffen. Wie Abbildung 2.2 zeigt, sind bei einer Suche nach „Irak“ unter den ersten vier Treffern zwei Webseiten mit Nachrichten (Yahoo! Nachrichten und Telepolis) sowie zwei Webseiten mit Informationen über den Irak (Wikipedia und Auswärtiges Amt), so dass bei einer Einschränkung (keine oder ausschließlich Nachrichtenseite) von diesen vier Webseiten nur noch zwei ausgegeben werden müssten.

Abbildung 2.2.: Google-Ausschnitt zur Suche nach „Irak“

Ein weiteres Beispiel bildet die Suche nach „Vogelgrippe“, die in Abbildung 2.3 dargestellt ist. Sie liefert sowohl Webseiten, auf denen über die Vogelgrippe informiert wird (Gefahren für den Mensch, Infektionsrisiko für Auslandsreisende, Krankheitssymptome, usw.) als auch Nachrichtenseiten, die von neuen Funden toter Tiere berichten. Auch hier ließe sich die Trefferliste durch Ausblenden oder ausschließliches Anzeigen der Nachrichtenseiten stark reduzieren, so dass dem Nutzer unter den ersten Treffern wesentlich mehr potenziell interessante Webseiten geboten würden.

Abbildung 2.3.: Google-Ausschnitt zur Suche nach „Vogelgrippe“

2.1.3. Onlineshops

Eine weitere Kategorie, die sich zur Einschränkung bestimmter Anfragen eignet, sind kommerzielle Webseiten. Eine Suche nach einem bestimmten Gegenstand liefert Webseiten mit Informationen oder Berichten zu diesem Artikel und Webseiten von Onlineshops (im Folgenden als Shops bezeichnet), die ihn zum Verkauf anbieten.

Abbildung 2.4.: Google-Ausschnitt zur Suche nach „Nokia 6230i“

Die Suche nach einem bestimmten Handy liefert sowohl informative Webseiten,

die Details und Funktionen des Handys beschreiben oder Testberichte und Erfahrungen von Nutzern liefern, als auch Webseiten, die es lediglich zum Verkauf anbieten. Abbildung 2.4 zeigt die ersten vier Treffer zur Suche nach „Nokia 6230i“. Eine vorherige Festlegung, ob man nach Shops suchen oder keine Shops angezeigt bekommen möchte, würde auch hier zu einer kürzeren Trefferliste führen und dem Benutzer das Suchen nach für ihn relevanten Webseiten erleichtern.

2.1.4. Wissenschaftliche Webseiten

Bei einer Suche nach einem Fachbegriff finden sich in der Trefferliste Webseiten, die den Begriff fundiert erklären oder ihn in einem fachlichen Zusammenhang verwenden und Webseiten, die den Begriff zwar enthalten, jedoch nicht als fachlich oder wissenschaftlich zu bezeichnen sind.


Das Cocktailparty-Problem: Neue Verfahren zur ...
Die Gruppe beschäftigt sich schwerpunktmäßig mit computergestützten Datenanalysemethoden wie Independent Component Analysis (ICA), **Support Vektor** Maschinen ...
ddi.cs.uni-potsdam.de/HyFISCH/Spitzenforschung/Mueller.htm - 24k - Im Cache - Ähnliche Seiten

Diplomarbeit Martina Ott
Als Vergleichsverfahren wurden im experimentellen Teil der Minimum-Distanz-Klassifikator, ein mehrschichtiges Perzeptron (MLP) und eine **Support Vektor ...**
www.ifp.uni-stuttgart.de/lehre/diplomarbeiten/ott/ott.htm - 21k - Im Cache - Ähnliche Seiten

SeeYou Hilfe **Support**
Brauchen Sie Hilfe/**Support**? Bitte prüfen Sie zuerst die folgenden Dokumente, ob sich eine Antwort auf Ihre Frage findet oder hinterlassen eine Mitteilung im ...
www.ldittmer.de/seeyou/**support**-de.htm - 11k - Im Cache - Ähnliche Seiten

CAD
Raster - **Vektor** - Konvertierung Natürlich gibt es auch Pläne, die sich für eine ... Wussten Sie, dass der **Support** für unsere Seminarkunden bis zu 3 Monaten ...
www.polzin.de/data/cadd.htm - 10k - Im Cache - Ähnliche Seiten

Abbildung 2.5.: Google-Ausschnitt zur Suche nach „Support Vektor“

Abbildung 2.5 soll dies etwas genauer verdeutlichen. Der erste der dargestellten Treffer behandelt Verfahren zur Signalquellentrennung, Treffer zwei vergleicht Klassifikationsverfahren zur Detektion von Oberflächenfehlern. Bei beiden Treffern handelt es sich um fachliche Ausarbeitungen. Treffer drei führt zu einem Supportformular eines Softwareherstellers, unter Treffer vier stellt sich ein CAD-Zeichenbüro vor. Diese beiden Treffer bieten sicherlich keine fachlichen Informationen.

2.1.5. Umsetzung der Benutzerpräferenzen

Die Filterung der Trefferliste nach den soeben eingeführten Beispielklassen wird im Rahmen dieser Diplomarbeit mit maschineller Textklassifikation realisiert, die

um webseitenspezifische Merkmale ergänzt wird. Kapitel 3 beschreibt theoretisch, wie eine maschinelle Klassifikation der Webseiten möglich ist, das darauf folgende Kapitel beschreibt mögliche zusätzliche Merkmale. Später wird untersucht, wie gut sich für die einzelnen Kategorien eine maschinelle Trennung vornehmen lässt.

2.2. Ähnliche Webseiten

Ein anderer Problembereich bei der Suche nach Informationen im Internet ist, dass manche Webseitenbetreiber ihre Inhalte unter verschiedenen Internetadressen ins Netz stellen. Diese inhaltlich gleichen Webseiten tauchen in den Trefferlisten der Internet-Suchmaschinen dann mehrfach auf.

2.2.1. Beispiel XML-Kurse

Abbildung 2.6 zeigt die ersten vier Treffer für die Suchanfrage „Datenmodellierung Transformation“. Ein Besuch der Webseiten zeigt, dass sie sich inhaltlich nicht unterscheiden. Sie wurden lediglich in einem anderen Layout unter einer anderen Domain abgelegt. Unter den ersten 20 Treffern taucht die Webseite 13 mal auf! Die hohe Platzierung aller Webseiten wird unter anderem durch eine massive Verlinkung der vielen Domains untereinander erreicht, wie die Backlink-Abfrage[3] einiger der gelisteten Domains zeigt.


XML - Intensiv **Datenmodellierung** - Seminare * XML - Intensiv ...
XML - Intensiv **Datenmodellierung** Seminarangebote Weiterbildung Trainer ... im Dokument zur DB-**Datenmodellierung** - **Transformation** eines Dokuments mit Hilfe ...
www.marcus-wiederstein.de/deut4_1025288.php - 28k - Im Cache - Ähnliche Seiten

XML - Intensiv **Datenmodellierung** - Kurse - XML - Intensiv ...
XML - Intensiv **Datenmodellierung** Training Beratung Fortbildung Kurse ... im Dokument zur DB-**Datenmodellierung** - **Transformation** eines Dokuments mit Hilfe von ...
www.oop-consulting.com/deut4_1025288.php - 39k - Im Cache - Ähnliche Seiten

XML - Intensiv **Datenmodellierung** - Kurse | XML - Intensiv ...
XML - Intensiv **Datenmodellierung** Training Firmenseminare Weiterbildung Kurse ... im Dokument zur DB-**Datenmodellierung** - **Transformation** eines Dokuments mit ...
www.oop-programming.com/deut4_1025288.php - 39k - Im Cache - Ähnliche Seiten

XML - Intensiv **Datenmodellierung** - Fortbildung · XML - Intensiv ...
XML - Intensiv **Datenmodellierung** Firmenseminare Beratung Weiterbildung ... im Dokument zur DB-**Datenmodellierung** - **Transformation** eines Dokuments mit Hilfe ...
www.uml-consulting.com/deut4_1025288.php - 39k - Im Cache - Ähnliche Seiten

Abbildung 2.6.: Trefferliste für „Datenmodellierung Transformation“

[3] Bei Google über link:<url> möglich

2.2.2. Beispiel Open Directory Project

Ein weiteres Beispiel für viele ähnliche Webseiten liefert das unter *www.dmoz.org* erreichbare Open Directory Project. Es ist ein offenes Internet-Verzeichnis, das von jedem editiert und erweitert werden kann und von vielen Webseitenbetreibern in ihre Homepage eingebunden wird. Dabei sind lediglich optische Unterschiede, wie beispielsweise andere Farben, ein geändertes Layout oder zusätzliche Vorschaugrafiken der verlinkten Webseiten auszumachen. Der enthaltene Text und die Verlinkungen sind in der Regel identisch, von eventuellen Ergänzungen und eingeblendeter Werbung einmal abgesehen.


Personliche Home Pages :: J
... Jehli, Peter - **Hochzeit** und erstellte Internetseiten.http://www.jehli.ch/. Jenzen, Thomas - **Biertest**, Reinheitsgebot, **Trinkspiele** für die Freizeit oder auf ...
www.personlichehomepages.info/directory/ 1675033f91d988d34dfaa24e2ecfcd41/20_j.htm - 19 k - Zusätzliches Ergebnis - Im Cache - Ähnliche Seiten

Open Directory - World: Deutsch: Gesellschaft: Menschen ...
Jenzen, Thomas - **Biertest**, Reinheitsgebot, **Trinkspiele** für die Freizeit oder auf Partys ... Johannes, Bianca und Markus - Vorstellung, Infos zur **Hochzeit**, ...
dmoz.org/World/Deutsch/Gesellschaft/ Menschen/Persönliche_Homepages/J/ - 29k - Im Cache - Ähnliche Seiten

Krify Web Directory - > World> Deutsch> Gesellschaft> Menschen> ...
... Jehli, Peter - - **Hochzeit** und erstellte Internetseiten. ... Jenzen, Thomas - - **Biertest**, Reinheitsgebot, **Trinkspiele** für die Freizeit oder auf ...
www.directory.krify.com/index.php?browse=/ World/Deutsch/Gesellschaft/Menschen/Persönliche_Homepages/J/ - 66k - Zusätzliches Ergebnis - Im Cache - Ähnliche Seiten

Svensk TuristGuide
... Jehli, Peter **Hochzeit** und erstellte Internetseiten. ... Jenzen, Thomas **Biertest**, Reinheitsgebot, **Trinkspiele** für die Freizeit oder auf Partys oder ...
www.smorgasbord.se/cgi-bin/odp/index.cgi?/ World/Deutsch/Gesellschaft/Menschen/Persönliche_Homepages/J/ - 25k - Zusätzliches Ergebnis - Im Cache - Ähnliche Seiten

Abbildung 2.7.: Trefferliste für „Hochzeit Trinkspiele Biertest“

Abbildung 2.7 zeigt vier Treffer zur Suche nach „Hochzeit Trinkspiele Biertest“. Die Treffer enthalten alle Spiegelungen des Open Directory Project. Die Anfrage ist in diesem Fall zwar konstruiert, verdeutlicht aber das Problem. Bei einer Suche nach „Trinkspiele Biertest“ befinden sich immer noch vier Spiegelungen des Open Directory Project unter den ersten zehn Treffern. Für den Google-Nutzer ist es in der Regel ausreichend, nur eine dieser Webseiten in der Ergebnisliste präsentiert zu bekommen.

2.2.3. Umsetzung der Filterung ähnlicher Webseiten

Für den Nutzer einer Internet-Suchmaschine ist es in der Regel ausreichend, von gleichen oder inhaltlich ähnlichen Webseiten nur den ersten Treffer mit einem Hinweis auf weitere Treffer präsentiert zu bekommen.

Kapitel 6 beschreibt ein Ähnlichkeitsmaß, das hier zur Anwendung kommen kann, um wiederholtes Vorkommen gleicher oder inhaltlich ähnlicher Webseiten zu erkennen. In Kapitel 7 folgen Untersuchungen darüber, wie gut sich damit inhaltlich ähnliche Webseiten erkennen lassen.

2.3. Unerwünschte Webseiten

Den dritten Problembereich stellen Webseiten dar, die vom Nutzer als Suchtreffer generell nicht erwünscht sind.

In den Trefferlisten einer Internet-Suchmaschine finden sich immer wieder Webseiten, die dem Nutzer keinen Inhalt bieten. Viele von ihnen sind lediglich dazu da, den PageRank anderer Webseiten durch das Setzen von Links zu erhöhen oder Geld mit eingeblendeter Werbung zu verdienen. Andere Treffer sind selbst Internet-Suchmaschinen, die vom Nutzer bei der Suche nach relevanten Webseiten nicht gewünscht sind.

2.3.1. Internet-Suchmaschinen

Bei einer Suche nach „Fernseher reparieren“ liefert einer der Treffer die Katalogsuchmaschine *suchnase.de* und dort als Einstiegsseite die Rubrik *Hobby / Technik / Audio*. Abbildung 2.8 zeigt einen Ausschnitt der Webseite. Ein weiterer Treffer führt zur Internet-Suchmaschine *Alexana.de*, die in Abbildung 2.9 dargestellt ist.

Abbildung 2.8.: Ausschnitt der Webseite *www.suchnase.de*

Internet-Suchmaschinen als Treffer einer Suche sind für den Nutzer uninteressant, wenn er nicht gerade auf der Suche nach einer Spezialsuchmaschine ist. Sie kosten ihn Zeit, da aus dem von Google präsentierten Ausschnitt nicht immer direkt hervorgeht, dass es sich dabei um eine Internet-Suchmaschine handelt. Dies wird beispielsweise dadurch erreicht, dass Google zum Indizieren eine andere Webseite geliefert wird als die, die der Nutzer beim Besuch der Webseite angezeigt bekommt.

Abbildung 2.9.: Ausschnitt der Webseite *www.alexana.de*

Die Erkennung und Filterung von Internet-Suchmaschinen könnte prinzipiell auch als Klassifikationsaufgabe aufgefasst werden. Sie wird hier aber zum Filtern unerwünschter Webseiten, die generell ausgeblendet werden sollten, gezählt, da Internet-Suchmaschinen in den seltensten Fällen vom Benutzer als Treffer gewünscht sind.

2.3.2. Webseiten mit Begriffs-Auflistungen

Einige Webseiten versuchen, durch Aufzählen vieler Begriffe auf möglichst viele Suchanfragen zu passen. Abbildung 2.10 zeigt einen Google-Ausschnitt zur Suchanfrage „Hund Haustier“. Die dazugehörige Webseite ist in Abbildung 2.11 dargestellt. Ziel der Webseitenbetreiber ist es in diesem Fall, dass der Besucher auf eine der von Google bereitgestellten, thematisch zum aufgelisteten Text passenden Anzeigen[4] in der oberen Hälfte der Webseite klickt. Die Webseite an sich bietet dem Nutzer keinen Inhalt.

Abbildung 2.10.: Google-Ausschnitt zur Suche nach „Hund Haustier“

Ein weiteres Beispiel liefert die in Abbildung 2.12 dargestellte Webseite. Auch hier sind im oberen Teil massenhaft Begriffe aufgezählt, so dass die Webseite für viele Anfragen zum Treffer wird. Der untere Teile besteht aus einer Linkauflistung, wahrscheinlich um die Anzahl der Backlinks anderer (Partner-)Webseiten zu erhöhen, da die meisten der gelisteten URLs wieder zu Webseiten mit Linkauflistungen oder kommerziellem Inhalt führen.

[4] https://www.google.com/adsense

Abbildung 2.11.: Ausschnitt der Webseite *www.halle-infos.de*

Derartige Webseiten stellen für den Benutzer einer Internet-Suchmaschine eine Belästigung dar. Mit einer zuverlässigen Erkennung solcher Webseiten könnte man die Trefferliste reduzieren und dem Nutzer die Suche im Internet erleichtern. Dabei muss aber gesichert werden, dass es sich wirklich um unerwünschte Webseiten handelt, damit keine relevanten Treffer entfernt werden.

Mögliche Indizien sind beispielsweise, dass Begriffe aufgelistet werden, ohne dass sich irgendwelche Satzzeichen zwischen ihnen befinden. Es handelt sich also nicht um einen natürlichsprachlichen Text. Ein weiterer Hinweis sind die vielen aufeinanderfolgenden Links, zwischen denen sich kein Text befindet. Seriöse Linksammlungen kommentieren ihre Links in der Regel.

Diese Arbeit macht sich derartige Eigenschaften zu Nutze, um die Erkennung solcher Webseiten durch zusätzliche Attribute zu verbessern. Eine Auflistung weiterer Merkmale ist in Kapitel 4 zu finden.

Abbildung 2.12.: Ausschnitt der Webseite *www.apanot.de*

3. Textklassifikation

3.1. Maschinelles Lernen

Maschinelles Lernen bezeichnet (stark vereinfacht dargestellt) den Erwerb von Wissen aus Erfahrungen. Ein System lernt dabei aus Beispielen, um das gewonnene Wissen anschließend auf neue, unbekannte Eingaben übertragen zu können. Eine ausführliche Erörterung des Begriffes „Lernen“ und eine Einführung in maschinelles Lernen findet sich beispielsweise in [14].

Die am häufigsten verwendete Form des maschinellen Lernens ist das *überwachte Lernen* (engl. supervised learning), bei dem aus vorgegebenen Beispielen gelernt wird. Ein Beispiel ist dabei ein Paar aus Eingabe und zugehöriger Ausgabe. Weitere Formen sind das *unüberwachte Lernen* (engl. unsupervised learning), bei dem keine Ausgaben vorgegeben werden, sowie das *verstärkende Lernen* (engl. reinforcement learning), bei dem der Computer durch Belohnung und Bestrafung lernt, sein Verhalten zu optimieren.

3.1.1. Aufgabe der Textklassifikation

Die Textklassifikation als Teilgebiet des maschinellen Lernens befasst sich mit der Einordnung neuer Texte in vom Benutzer vorgegebene Klassen. Die Festlegung dieser Klassen erfolgt anhand von Beispieltexten, zu denen jeweils eine Klasse angegeben ist. Es handelt sich demnach um das soeben beschriebene überwachte Lernen. Die Klasse bzw. das sogenannte Klassenlabel wird bei einer Klassifikation in zwei Klassen meist mit den Werten -1 oder 1 belegt.

Anhand der vorgegebenen Beispiele, den sogenannten Trainingsdaten, soll das Lernverfahren ein Modell erzeugen, dass die Zuordnung der Texte zu den Klassen erklärt und sich auf neue, noch nicht klassifizierte Texte übertragen lässt. Eine Beschreibung gängiger Lernverfahren zur Textklassifikation folgt in Kapitel 3.3.

Oft werden die Begriffe Textklassifikation und Textkategorisierung synonym verwendet. Tatsächlich handelt es sich bei der Kategorisierung von Texten aber um eine unüberwachte Lernaufgabe, bei der eine Menge von Texten in vorher nicht definierte Kategorien eingeteilt werden soll. Die Eingabe ist also eine Menge von Texten, die Kategorien werden durch das Lernverfahren generiert.

3.1.2. Repräsentation von Texten

Um Texte inhaltlich mit maschinellen Lernverfahren verarbeiten zu können, müssen sie zunächst in eine für das Lernverfahren geeignete Form gebracht werden. Dazu werden sie auf eine Menge von Attributen reduziert, für deren Erstellung es verschiedene Verfahren gibt, die im Folgenden kurz erläutert werden.

Zeichenkettenbasierte Verfahren betrachten einen Text als eine Aneinanderreihung von Zeichen und benötigen keinerlei Wissen über die verwendete Sprache. Das bekannteste zeichenkettenbasierte Verfahren benutzt n-Gramme, also Ketten von n aufeinanderfolgenden Zeichen. Hierbei wird ein Fenster mit fester Länge über den Text geschoben, schrittweise immer ein Zeichen weiter. Der Inhalt des Fensters bildet in jedem Schritt einen Indexterm. Die 4-Gramme der Überschrift dieses Kapitels lauten demnach *Repr*, *eprä*, *präs*, usw. Die Repräsentation des gesamten Textes erfolgt dann meist über einen Vektor, der für jedes n-Gramm die Anzahl seines Vorkommens im Text angibt. Die Indexterme bilden für ein Lernverfahren die Attribute eines Textes und werden im Folgenden auch so bezeichnet.

N-Gramme sind unabhängig von der verwendeten Sprache und relativ robust gegen Schreibfehler. Ein Nachteil ist, dass die üblicherweise kleinen Werte für die Länge der n-Gramme zu wesentlich mehr n-Grammen führt, als Wörter im Text enthalten sind. Die entsprechend gebildeten Vektoren werden dadurch sehr lang.

Wortbasierte Textrepräsentationen verwenden dagegen ganze Wörter als Attribute und benötigen Wissen darüber, woran ganze Wörter zu erkennen sind. Eine Trennung nur über Leerzeichen reicht hierbei nicht aus, was bereits an Satzzeichen klar wird. Folgt ein Punkt oder ein Komma direkt auf ein Wort, so wird bei einer rein leerzeichenbasierten Erkennung von Wörtern das Satzzeichen als Bestandteil des Wortes betrachtet und so zu einem anderen Attribut führen, als wenn dem Wort kein Satzzeichen folgt. Weitere, zu berücksichtigende Problemfälle sind beispielsweise Zahlen oder Bindestriche. Möchte man Zahlen ignorieren oder für jede Zahl ebenfalls ein Attribut bilden? Was geschieht mit Kombinationen aus Zahlen und Buchstaben, wie z. B. in *Airbus A380*? Betrachtet man mit einem Bindestrich verbundene Wörter als ein oder zwei Wörter?

Die wortbasierte Textrepräsentation führt in der Regel zu weniger Attributen als die Repräsentation über n-Gramme. Eine weitere Reduktion der Anzahl an Attributen kann durch das Ausfiltern von Stoppwörtern erreicht werden. Hierbei handelt es sich um Wörter, die in natürlichsprachlichen Texten häufig vorkommen, im Allgemeinen aber keine Aussagekraft für die Erfassung eines Textinhalts haben. Sie übernehmen meist nur grammatikalische oder syntaktische Funktionen. Zur Entfernung von Stoppwörtern werden häufig Stoppwortlisten verwendet, andere Ansätze bedienen sich eines Lexikons und entfernen alle Wörter bestimmter Wortarten, beispielsweise Artikel, Konjunktionen oder Präpositionen.

Verschiedene Formen eines Wortes führen bei der wortbasierten Textrepräsentation zu verschiedenen Attributen. Über eine Stammformreduktion, sogenanntes *Stemming*, kann hier eine Generalisierung der Attribute erreicht werden. Dabei

wird ein Wort auf seine Grundform reduziert, so dass verschiedene Flexionsformen desselben Wortes auf dasselbe Attribut abgebildet werden können. Die bekanntesten Stemmer werden in [11] und [15] beschrieben. Es ist leicht ersichtlich, dass eine Stammformreduktion entsprechendes Wissen über die verwendete Sprache benötigt. Während die Reduktion einer äußeren Flexion unter Benutzung von Suffixen (z. B. *schneller / schnell*) noch relativ einfach ist, wird für die Reduktion von Wörtern mit inneren Flexionen (z. B. *lief / laufen*) erhebliches sprachliches Wissen benötigt.

Auch bei der wortbasierten Repräsentation von Texten bietet sich eine Darstellung der Attribute eines Dokuments als Vektor an. Der Vektor, für Wortbasierte Repräsentationen im Folgenden als Wortvektor bezeichnet, enthält dabei so viele Stellen, wie es verschiedene Wörter in der gesamten Textsammlung gibt, damit die Wortvektoren aller Dokumente miteinander vergleichbar sind ([19]). Der Wert der Attribute, die in dem jeweiligen Text nicht als Wort enthalten sind, beträgt dann 0. Es ist leicht ersichtlich, dass sich bereits für eine kleine Textsammlung sehr lange Wortvektoren ergeben, in denen viele Stellen mit 0 belegt sind. Die Reihenfolge, in der die Wörter im Text vorkommen, geht bei diesem sogenannten Bag of Words-Modell allerdings verloren.

Als Ausprägung für die Attribute kann die Anzahl der Fundstellen im Text verwendet werden. Eine bessere Annäherung an die Charakteristik eines Wortes für einen Text bietet das TF-IDF-Maß (term frequency - inverse document frequency, [18]). Es beruht auf der Annahme, dass für einen Text charakteristische Wörter in diesem Text häufig, in anderen Texten der Textsammlung aber seltener vorkommen. Die Gewichtung eines Wortes richtet sich dabei also nicht ausschließlich nach dem betrachteten Text, sondern nach der gesamten Textsammlung. Der TF-IDF-Wert des Wortes w in Dokument d ist definiert als

$$TFIDF_{w,d} = TF_{w,d} \cdot \log \frac{N}{DF_w} \quad .$$

$TF_{w,d}$ bezeichnet dabei die Häufigkeit des Wortes w im Dokument d, N die Anzahl an Dokumenten in der Textsammlung und DF_w die Anzahl der Dokumente, in denen das Wort w vorkommt.

3.1.3. Bewertungsmaße

Die Bewertung eines Lernverfahrens für eine Lernaufgabe erfolgt meist im Rahmen einer Kreuzvalidierung. Dabei werden die zur Verfügung stehenden Beispiele in x Teilmengen geteilt und x Durchgänge gestartet, in denen jeweils die x-te Menge zum Test des auf den restlichen $x-1$ Teilmengen gelernten Modells dient. Die in einem Durchgang zum Lernen benutzten Beispiele werden als Trainingsmenge, die zum Testen benutzte Menge als Testmenge bezeichnet. Das Testen besteht darin, die Texte der Testmenge mit dem auf der Trainingsmenge gelernten Modell zu

klassifizieren und das Ergebnis für jeden Text mit der vorgegebenen Klasse des Textes zu vergleichen.

Zur Bewertung der Güte können beispielsweise die Maße *Accuracy*, *Precision* oder *Recall* verwendet werden, die sich im Bezug auf die Klassifikation von Texten wie folgt beschreiben lassen:

- Die Accuracy entspricht dem prozentualen Anteil der nach dem Training richtig klassifizierten Texte.
- Die Precision entspricht dem prozentualen Anteil der tatsächlich relevanten Texte aus der Menge der vom System als relevant eingestuften Texten.
- Der Recall entspricht dem Anteil der vom System als relevant eingestuften Texte aus der Gesamtmenge der relevanten Texte.

Eine formale Beschreibung dieser Bewertungsmaße ist beispielsweise in [6] und in [5] zu finden. Relevante Texte sind in diesem Zusammenhang die Texte, die der gesuchten Klasse angehören. Die Werte der Bewertungsmaße werden im Rahmen der Kreuzvalidierung als Mittelwert der x Durchgänge ermittelt.

3.2. Internet-Suchmaschinen und Maschinelles Lernen

Die Verbindung von Internet-Suchmaschinen mit maschinellem Lernen beschränkt sich meist darauf, die Reihenfolge zu verändern, in der die Suchtreffer angezeigt werden.

In [10] wird beschrieben, wie sich das Ranking durch eine implizite Bewertung der Treffer durch den Nutzer optimieren lässt, ohne den Nutzer nach einer Bewertung der einzelnen Webseiten zu fragen. Treffer, die der Nutzer in der Trefferliste überspringt, scheinen für ihn bei seiner Anfrage nicht interessant zu sein. Ruft er bei seiner Suche lediglich die Treffer zwei und drei auf, so hat er sich anhand der Angaben in der Trefferliste gegen Treffer eins entschieden. Anhand dieser impliziten Bewertung kann gelernt werde, das Ranking besser an einen einzelnen oder eine Gruppe von Nutzern anzupassen. Mit *STRIVER*[1] existiert eine Umsetzung dieser Idee, mit der sich die Webseiten der Cornell Universität durchsuchen lassen.

Ein jüngerer Ansatz schlägt das sogenannte Eye Tracking zur impliziten Bewertung von Suchtreffern vor ([7]). Dazu werden Treffer durch Beobachtung der Pupille des Nutzers bewertet. Relevante Kriterien sind beispielsweise, wie lange und in welcher Reihenfolge der Nutzer die Treffer betrachtet werden.

Diese Ansätze unterscheiden sich von der in dieser Diplomarbeit verfolgten Idee dadurch, dass sie Webseiten in Ahängigkeit zu dem verwendeten Suchbegriff gewichten. Die Trefferliste wird dann durch Umsortierung der Treffer optimiert. Der

[1] www.cs.cornell.edu/~tj/striver/

hier mit der Umsetzung von Benutzerpräferenzen verfolge Ansatz besteht dagegen darin, die Trefferliste nach vom Nutzer vorgegebenen Klassen zu filtern. Treffer, die nicht der gewählten Klasse angehören, sollen entfernt werden. Ein Treffer kann dabei, je nach gewählter Klasse, für den gleichen Suchbegriff in einem Fall vom Nutzer erwünscht, in einem anderen Fall unerwünscht sein. Es reicht also nicht aus, die Relevanz der Treffer allein anhand der verwendeten Suchbegriffe zu bestimmen.

3.3. Algorithmen zur Textklassifikation

Im Folgenden werden einige Algorithmen zur Textklassifikation kurz vorgestellt, die sich auch zur Klassifikation von Webseiten eignen. Anschließend erfolgt die Auswahl eines Algorithmus für die später durchgeführten Versuche.

3.3.1. k-Nearest-Neighbour

Ein einfach aufgebauter Algorithmus, der bei der Klassifikation von Texten dennoch gute Ergebnisse liefert, ist der kNN-Algorithmus ([13]). Er betrachtet die Wortvektoren als Punkte im n-dimensionalen Raum, wobei n die Dimension der Wortvektoren, also die Anzahl unterschiedlicher Wörter in der gesamten Textsammlung ist. Bei einem neu zu klassifizierenden Dokument wird sein Wortvektor mit den Wortvektoren der gegebenen, bereits klassifizierten Dokumente verglichen und die k Dokumente mit dem geringsten Abstand im n-dimensionalen Raum bestimmt. Als Distanzfunktion eignet sich beispielsweise die euklidische Distanz. Für zwei Punkte x und y mit $x = (x_1, ..., x_n)$ und $y = (y_1, ..., y_n)$ ist sie definiert als

$$d(x, y) = \sqrt{\sum_{i=0}^{n} (x_i - y_i)^2} \quad .$$

Die Klasse des neu zu klassifizierenden Dokuments entspricht der Klasse der Mehrzahl seiner k nächsten Nachbarn. Durch Wahl eines ungeraden k kann dabei eine eindeutige Mehrheit sichergestellt werden. Im Beispiel in Abbildung 3.1 ist das neu zu klassifizierende, durch Punkt X dargestellte Dokument für $k = 5$ in die Klasse der mit einem schwarzen Punkt dargestellten Dokumente einzuordnen, der vier seiner fünf nächsten Nachbarn angehören.

Der Vorteil des kNN-Algorithmus ist sein einfacher Aufbau, der ihn leicht implementierbar macht und dennoch zu guten Ergebnissen in der Textklassifikation führt ([22] und [8]). Ein großer Nachteil ist seine Laufzeit, da ein neu zu klassifizierendes Dokument mit allen anderen Dokumenten der Textsammlung verglichen werden muss, um seine nächsten Nachbarn zu bestimmen. Andere Algorithmen führen einen Teil ihrer Berechnungen während der Trainingsphase schon im Vorfeld aus und reduzieren dadurch die Laufzeit einer späteren Klassifikation.

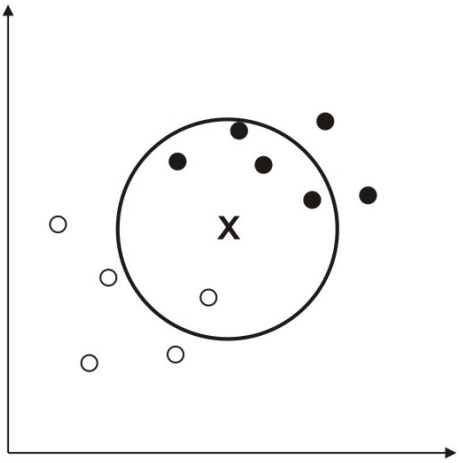


Abbildung 3.1.: Beispiel für kNN mit $k = 5$

3.3.2. Naive Bayes

Das Naive Bayes Verfahren ([13]) ist ein mathematisches Verfahren für die maschinelle Zuordnung von Texten zu Klassen. Es beruht auf der vom Bayes-Theorem ([1]) abgeleiteten Formel für bedingte Wahrscheinlichkeiten. Das Verfahren wird als *naiv* bezeichnet, da die ihm zugrunde liegende Annahme, dass jedes Attribut (jedes Wort) nur vom Klassenlabel und nicht von anderen Attributen abhängt, in der Realität selten zutrifft. So folgt auf „Mit freundlichen“ mit hoher Wahrscheinlichkeit „Grüßen“.

Dennoch liefern Naive Bayes-Klassifizierer in der Praxis häufig gute Ergebnisse und sind daher bei der Klassifikation von Texten, vor allem bei der Erkennung von Spam, weit verbreitet.

Das Verfahren berechnet eine relative Häufigkeit der Wörter bezüglich der zugehörigen Klasse des Dokuments. Diese Häufigkeiten bilden das sogenannte Vorwissen, auch a priori-Wahrscheinlichkeiten genannt. Zur Klassifikation neuer Dokumente wird anhand der berechneten Häufigkeiten der vorkommenden Wörter mit Hilfe des Bayes-Theorems die Klasse bestimmt, der das Dokument am wahrscheinlichsten angehört.

Mit b als Behauptung und M als beobachtetem Merkmal gilt nach dem Bayes-Theorem

$$P(b|M) = \frac{P(M|b) \cdot P(b)}{P(M)}$$

wobei hier $P(b|M)$ die Wahrscheinlichkeit ist, dass Behauptung b gilt, wenn Merkmal M beobachtet wird. Eine Behauptung wäre hier, dass ein Dokument einer bestimmten Klasse angehört, ein Merkmal, dass ein bestimmtes Wort vorkommt oder nicht. $P(M|b)$ ist die Wahrscheinlichkeit, dass Merkmal M auftritt, wenn Behauptung b zutrifft. $P(b)$ ist die Wahrscheinlichkeit für die Behauptung b, die sich

aus dem Verhältnis der klassifizierten Dokumente ergibt. $P(M)$ ist die Wahrscheinlichkeit, dass die Beobachtung M auftritt. Sie ist bei jedem Problem konstant und kann daher auch weggelassen werden.

Ein kleines Beispiel soll das Verfahren veranschaulichen. Gesucht wird die Wahrscheinlichkeit, dass eine Webseite unerwünscht ist. Die Behauptungen sind:

- b_1: Die Webseite ist erwünscht
- b_2: Die Webseite ist unerwünscht

Die beobachteten Daten sind:

- M_1: Die Webseite enthält den Begriff „Shop“
- M_2: Die Webseite enthält den Begriff „Shop“ nicht

Mit den folgenden a priori-Wahrscheinlichkeiten

$$\begin{array}{rr} P(b_1)=0,80 & P(b_2)=0,20 \\ P(M_1)=0,33 & P(M_2)=0,67 \\ P(M_1|b_1)=0,10 & P(M_2|b_1)=0,90 \\ P(M_1|b_2)=0,75 & P(M_2|b_2)=0,25 \end{array}$$

ist die Wahrscheinlichkeit, dass eine Webseite, die den Begriff „Shop“ enthält, erwünscht ist

$$\begin{aligned} P(b_1|M_1) &= \frac{P(M_1|b_1) \cdot P(b_1)}{P(M_1)} \\ &= \frac{0,10 \cdot 0,80}{0,33} \\ &= 0,\overline{24} \quad . \end{aligned}$$

Die Wahrscheinlichkeit, dass die entsprechende Webseite unerwünscht ist beträgt

$$\begin{aligned} P(b_2|M_1) &= \frac{P(M_1|b_2) \cdot P(b_2)}{P(M_1)} \\ &= \frac{0,75 \cdot 0,20}{0,33} \\ &= 0,\overline{45} \end{aligned}$$

und ist somit höher. Es ist also wahrscheinlicher, dass die Webseite zur Klasse der unerwünschten Webseiten gehört.

3.3.3. Neuronale Netze

Die Funktionsweise künstlicher neuronaler Netze orientiert sich an der von Neuronen (Nervenzellen), wie sie im Gehirn vorkommen. Dort sind Nervenzellen so miteinander verbunden, dass jede Nervenzelle mehrere eingehende Verbindung besitzt, über die sie Reize von anderen Zellen empfängt. Eine Nervenzelle verarbeitet die sie erreichenden Reize und gibt selbst wieder einen Reiz an andere Zellen weiter.

Diese Funktionsweise wird von künstlichen neuronalen Netzen nachgeahmt. In einem gerichteten Graph stellen gewichtete Kanten die Verbindung zwischen Neuronen dar. Eine Auswertungsfunktion in jedem Neuron berechnet aus den Eingangswerten der eingehenden Verbindungen und den zugehörigen Gewichten der jeweiligen Verbindung einen Aktivierungszustand. Unter Verwendung einer Schwellwertfunktion wird anhand des Aktivierungszustands bestimmt, was über die ausgehenden Verbindungen weitergegeben wird. Dabei wird an alle nachgelagerten Neuronen der gleiche Wert weitergegeben.

Nach [20] ist ein künstliches Neuron formal ein Tupel $(\vec{x}, \vec{w}, f_a, f_o, o)$ bestehend aus einem Eingabevektor $\vec{x} = (x_1, ..., x_n)$, einem Gewichtsvektor $\vec{w} = (w_1, ..., w_n)$, einer Aktivierungsfunktion f_a mit $f_a : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$ und einer Ausgabefunktion f_o, für die $f_o : \mathbb{R} \mapsto \mathbb{R}$ gilt. Durch $f_o(f_a(\vec{x}, \vec{w})) = o$ wird dabei der Ausgabewert des Neurons erzeugt und über die ausgehende Verbindung an die nachfolgenden Neuronen weitergeleitet.

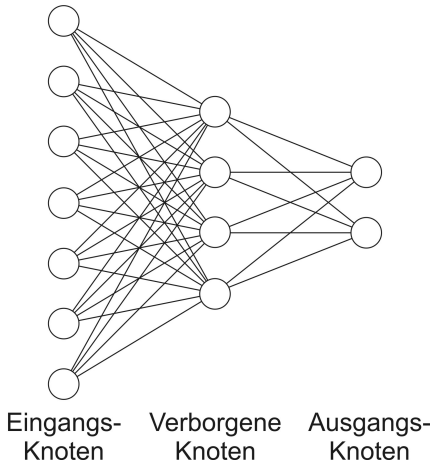


Abbildung 3.2.: Aufbau eines neuronalen Netzes

Ein künstliches neuronales Netz besteht in der Regel aus einer Eingangsschicht, einer oder mehreren Zwischenschichten (in denen die so genannten verborgenen Knoten liegen) und einer Ausgangsschicht. Abbildung 3.2 zeigt ein einfaches neuronales Netz mit zwei Ausgangsknoten und einer verborgenen Schicht. Ein neuronales Netz zur Klassifikation von Texten muss für jedes vorkommende Wort einen

Eingangsknoten besitzen, der entsprechende Wortvektor bildet den Eingabevektor $\vec{c}$ für das Neuronale Netz.

Das Training eines künstlichen neuronalen Netzes besteht darin, die Gewichte der Verbindungen anzupassen. Hierfür gibt es verschiedene Verfahren, beispielsweise das Backpropagation-Verfahren ([16]).

3.3.4. Support Vector Machines

Ein relativ junger Ansatz sind die Support Vector Machines (SVM) nach [4], die erstmals in [9] zur Klassifikation von Texten verwendet wurden.

Wie beim kNN-Algorithmus (vgl. 3.3.1) werden auch hier alle Wortvektoren als Punkte im n-dimensionalen Raum betrachtet. Die SVM errechnet nun eine Hyperebene (im zweidimensionalen Raum wäre es eine Gerade, im dreidimensionalen Raum eine Fläche), die die durch Wortvektoren gegebenen Punkte unter Berücksichtigung der jeweiligen Klasse mit größtmöglichem Abstand in zwei Hälften trennt. Dazu werden Punkte gesucht, die an der Grenze der beiden zu trennenden Klassen liegen. Sie werden als Stützvektoren bezeichnet (engl. Support Vectors). Die gesuchte Hyperebene verläuft zwischen diesen Stützvektoren. In der Regel gibt es unendlich viele solcher Hyperebenen. Die SVM errechnet diejenige, die den größten Abstand (im Bezug auf SVMs spricht man von Rand) zu beiden Klassen aufweist, um eine möglichst deutliche Trennung zu erreichen.

Zur Klassifikation eines neuen Dokuments wird sein Wortvektor mit der durch die SVM errechneten Hyperebene verglichen und bestimmt, auf welcher Seite der Hyperebene der entsprechende Punkt liegt.

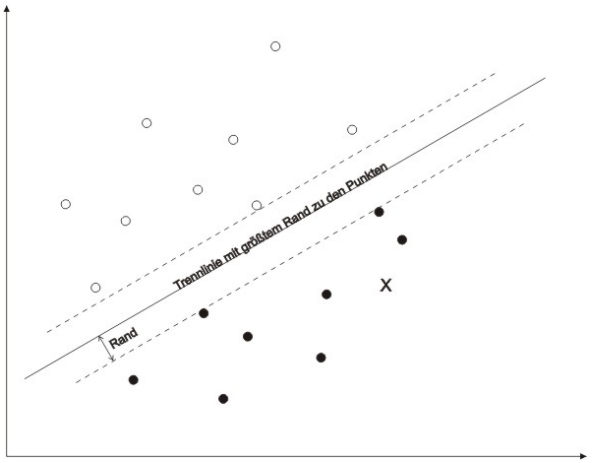


Abbildung 3.3.: Hyperebene bei einer SVM

Abbildung 3.3 zeig dies beispielhaft im zweidimensionalen Raum. Die durchgezogene Gerade ist in diesem Fall die Hyperebene mit dem größten Rand zu

beiden Klassen. Das Dokument des zu Punkt X gehörenden Wortvektors ist hier in die Klasse der durch einem schwarzen Punkt dargestellten Dokumente einzuordnen. An der Abbildung lässt sich leicht veranschaulichen, dass es in der Regel unendlich viele Hyperebenen gibt, die die Klassen eindeutig trennen: durch Parallelverschiebung der durchgezogenen Trennlinie innerhalb der beiden gestrichelten Linien erhält man weitere Hyperebenen, die die beiden Klassen eindeutig trennen.

Formal besteht die Aufgabe der SVM darin, eine Hyperebene H der Form

$$\vec{w} \cdot \vec{x} + b = 0$$

mit Normalenvektor $\vec{w}$ und Verschiebung vom Ursprung b zu finden, die die gegebenen Punkte unter Berücksichtigung der jeweiligen Klasse in zwei Hälften trennt und den größtmöglichen Rand zu den Punkten aufweist. Als Optimierungsaufgabe lässt sich dies wie folgt schreiben:
Minimiere $||w||^2$, so dass für alle i

$$\begin{aligned} f(x_i) = \vec{w} \cdot \vec{x_i} + b \geq 1 \qquad & \text{für } y_i = 1 \qquad \text{und} \\ f(x_i) = \vec{w} \cdot \vec{x_i} + b \leq -1 \qquad & \text{für } y_i = -1 \end{aligned}$$

gilt, wobei $y_i \in \{-1; 1\}$ die Klasse des Dokuments i angibt.

Ist, wie bislang angenommen, eine eindeutige Trennung der beiden Klassen möglich, so spricht man von einem linear separierbaren Merkmalsraum. Ist keine eindeutige Trennung der Klassen möglich, verwendet man so genannte weiche Ränder, erlaubt also in einer geringen Entfernung der Hyperebene noch Instanzen der anderen Klasse. Ein anderer Ansatz versucht, durch eine Transformation des Merkmalsraumes eine Trennung zu erhalten. Dies ist beispielsweise in [21] und [3] erklärt, wird hier aber nicht weiter betrachtet, da nach [9] eine Transformation bei Texten meist nicht benötigt wird.

Zur Realisierung der weichen Ränder werden sogenannte Schlupfvariablen ξ_i eingeführt, mit denen die SVM für falsch klassifizierte Trainingsbeispiele bestraft wird. Die Optimierungsaufgabe der SVM besteht dann in der Minimierung des Ausdrucks

$$||w||^2 + C \sum_{i=1}^{n} (\xi_i)$$

für einen Kostenparameter $C \in \Re_{>0}$.

Mit höherem Wert für C erlaubt man der SVM dadurch weniger falsch klassifizierte Trainingsbeispiele, da die Kosten für eine Fehlklassifikation steigen. Die SVM muss ihr gelerntes Modell dann genauer an die Trainingsdaten anpassen, was oft zu sogenanntem Overfitting führt: das gelernte Modell wird genau auf die Trainingsbeispiele zugeschnittenen, generalisiert aber schlecht.

Abbildung 3.4 dient der Veranschaulichung dieses Problems in einem linear separierbaren Merkmalsraum und verdeutlicht zugleich, warum auch hier weiche

Ränder oft zu einer besseren Generalisierung führen. Der mit X markierte Punkt wurde, beispielsweise durch Fehlmessung, in die Klasse der schwarzen Punkte einsortiert. Setzt man den Kostenparameter C zu hoch, wird die SVM im Training die Hyperebene 2 zur Trennung der beiden Klassen auswählen, um den Punkt X nicht falsch einzusortieren. Erlaubt man eine gewisse Fehlklassifikation, wäre die SVM in der Lage Trennlinie 1 zu wählen, die zwar Punkt X falsch einsortiert, davon abgesehen aber einen wesentlich höheren Rand zu den anderen Punkten aufweist.

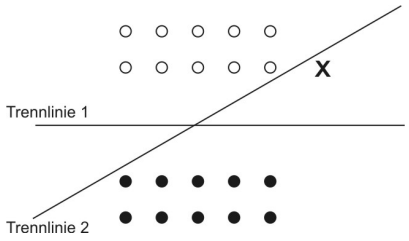


Abbildung 3.4.: Beispiel für weiche Ränder

3.4. Eignung zur Klassifikation von Suchtreffern

Alle der soeben vorgestellten Verfahren zur Textklassifikation eignen sich prinzipiell auch zur Klassifikation von Webseiten. Neben der Genauigkeit der Verfahren ist hier aber auch eine Auswahl nach Geschwindigkeitsaspekten zu treffen. Eine Trefferliste sollte in möglichst kurzer Zeit gefiltert werden. Das langwierige Erstellen einer Liste, die das Auffinden von relevanten Webseiten beschleunigen soll, erscheint unangemessen.

Diesbezüglich schneidet der kNN-Algorithmus besonders schlecht ab, da er die gesamte Berechnung erst während der Klassifikation durchführt. Die anderen hier vorgestellten Algorithmen verlagern den größten Teil ihrer Berechnungen in das Training und können dadurch deutlich schneller klassifizieren.

Eine gegenüberstellende Bewertung der einzelnen Algorithmen ist in [2] zu finden. Die Algorithmen werden dort auf ihre Eignung zur Erkennung von Spam, für die zusätzliche Attribute aus den Header-Daten der E-Mails generiert werden, verglichen. Demnach stellen die Support Vector Machines einen guten Kompromiss zwischen der Geschwindigkeit im Training und der Geschwindigkeit in der Klassifikation dar und können sehr gut inkrementell lernen, neue Trainingsbeispiele können also hinzugefügt werden, ohne dass die gesamte Berechnung neu durchgeführt werden muss. In [9] wird außerdem nachgewiesen, dass SVMs bei der Klassifikation von Texten sehr gute Ergebnisse liefern.

Ein weiterer Vorteil der Support Vector Machines für die in Kapitel 5 folgenden Versuche ist, dass sich die von ihnen errechneten Attributgewichte auf die einzel-

nen Attribute beziehen und leicht ausgeben lassen. So können die in Kapitel 4 zusätzlich erzeugten Attribute einzeln bezüglich ihrer Relevanz bewertet werden.

Für die folgenden Versuche werden die SVMs als Verfahren zur Textklassifikation ausgewählt. Wie bereits erwähnt, können hier prinzipiell auch die anderen beschriebenen Verfahren verwendet werden.

Bei einer Klassifikation von Suchtreffern über die von Google gelieferten Ausschnitte stehen nur sehr kurze Textteile zur Verfügung. Es stellt sich die Frage, wie gut die jeweiligen Verfahren mit diesen kurzen Texten zurecht kommen. Sollten die SVMs in den späteren Versuchen auf den Google-Schnipseln zu keinen guten Ergebnissen kommen, sind weitere Versuche mit anderen Lernverfahren denkbar.

4. Zusätzliche Merkmale

Die Textklassifikation, wie sie in Kapitel 3 beschrieben wird, arbeitet ausschließlich mit dem Textinhalt, ohne jedoch auf weitere Eigenschaften der Dokumente einzugehen. In [17] wurde bereits nachgewiesen, dass sich das Klassifikationsergebnis der Filterung von Spam durch die Hinzunahme von anwendungsspezifischen Merkmalen weiter verbessern lässt. Zusätzliche Merkmale sind dort unter anderem das Vorkommen von Sonderzeichen (vor allem Dollar- und Ausrufungszeichen) im Betreff der E-Mail, Uhrzeit des Mailversands oder bestimmte Phrasen, wie z. B. „Free money“. Es ist nahe liegend, auch für Webseiten weitere Merkmale heranzuziehen, um die Klassifikation zu verbessern.

Ein besonderer Fokus liegt hier auf der URL (Uniform Resource Locator), da diese als Teil der Google-Trefferliste schon vor dem Besuch der jeweiligen Webseite bekannt ist. Weitere, von Google direkt bereitgestellte Informationen sind der Titel der Webseite sowie ein kurzer Ausschnitt um den bzw. die Suchbegriffe.

Merkmale der vollständigen Webseiten, die vor allem bei der Klassifikation unerwünschter Webseiten von Bedeutung sein könnten, sind beispielsweise die Verteilung von Satzzeichen, Stoppwörtern und Links auf der Webseite.

Vermutlich lässt sich allein anhand dieser Merkmale keine zuverlässige Klassifikation durchführen, sie könnten aber als Zusatz zum Textinhalt der Webseiten zu einer Verbesserung der Klassifikation führen. Im Folgenden werden die Merkmale genauer erläutert. In den in Kapitel 5 durchgeführten Versuchen wird unterschieden, ob und welche zusätzlichen Merkmale jeweils zur Klassifikation verwendet werden. So lässt sich untersuchen, bei welchen Kategorien mit welchen Merkmalen eine Verbesserung erzielt werden kann.

4.1. Internetadresse

Als URL wird ein „einheitlicher Ortsangeber für Ressourcen“ bezeichnet, im deutschen Sprachgebrauch oft auch einfach Internetadresse genannt.

Der Aufbau einer URL folgt einem fest definierten Schema. Zuerst wird das Übertragungsprotokoll angegeben. Zur Übertragung von Webseiten ist dies in der Regel HTTP oder das verschlüsselte Pendant HTTPS, bei Dateiübertragungen kommt oft FTP zum Einsatz. Dem Protokoll folgen ein Doppelpunkt und zwei Schrägstriche. Anschließend folgt der Hostname, angegeben als (Sub-)Domain oder IP-Adresse, eventuell gefolgt von weiteren Verzeichnisangaben und dem Dateinamen, der wiederum durch einen Schrägstrich von dem Domainnamen bzw. den

Verzeichnisangaben getrennt wird. Dieser eigentlichen Adresse zu einer Datei oder einer Webseite können weitere Parameter folgen, die durch Fragezeichen von der Adresse getrennt und serverseitig Variablen zugeordnet werden.

Ein Beispiel für eine URL mit Subdomain und einem Parametern ist *http://subdomain.domain.de/Verzeichnis1/datei.html?parameter=12345*

HTTP wird von aktuellen Browsern als Standard verwendet, wenn kein Protokoll angegeben wird. Wird kein Dateiname angegeben, liefern Server bei Verwendung von HTTP als Übertragungsprotokoll in der Regel die Datei *index.html* aus. Daher liefert die Eingabe von *www.google.de* in einen Browser dieselbe Webseite wie die Eingabe von *http://www.google.de/index.html*.

4.1.1. URL-Spezifische Merkmale

Aus den in Kapitel 5.1 gesammelten Datensätzen wurde zu jeder Klasse eine Liste der entsprechenden URLs der positiven und negativen Klasse erzeugt und auf relevante Unterschiede untersucht, die bei der Klassifikation von Webseiten zu Verbesserungen gegenüber dem rein wortbasierten Ansatz beitragen können. Die Unterschiede werden im Folgenden beschrieben.

Forenseiten

- In den meisten Foren tritt der Begriff *Forum* oder *Board* auch irgendwo in der URL auf. Bei reinen Forenseiten oft als Teil der Domain (z. B. *www.medizin-forum.de*), bei Webseiten die zusätzlich zum eigentlichen Inhalt ein Forum betreiben, entweder als Bezeichnung der Subdomain (z. B. *forum.freenet.de*) oder im Pfad zum Forum (z. B. *www.computerbase.de/forum/index.html*).

- Foren verfügen nicht selten über eine große Anzahl an Webseiten, die üblicherweise automatisch durchnummeriert werden. Zur Auswahl einer bestimmten Webseite ist ihre Nummer meist in der URL enthalten, entweder als Name der HTML-Datei (z. B. *board.protecus.de/t17626.htm*) oder als zusätzlicher Parameter (z. B. *www.mcseboard.de/showthread.php?p=384778*).

- Webseiten, die nicht als durchnummerierte HTML-Dateien gespeichert werden, sondern denen die Seitennummer als Parameter übergeben wird, besitzen in der Regel die Dateiendung .php.

- Die meisten URLs enthalten die Begriffe *Beitrag*, *Thread* oder *Topic*, entweder als Teil einer Parameterbezeichnung (z. B. *threadid=13427*) oder als Teil des Dateinamen (*z. B. topic13952-0.html*).

Nachrichtenseiten

- Wie schon bei Foren dargestellt, arbeiten auch die meisten Nachrichtenseiten aufgrund der großen Anzahl an Webseiten mit Nummerierungen. Hierbei finden sich in den URLs auch durch Kommata getrennte Ziffernblöcke (z. B. *0,1518,333173,00.html*), vermutlich um Kategorien oder Themen zu selektieren.

- In vielen Fällen wird die Kategorisierung der Nachrichten serverseitig auch durch Verzeichnisse abgebildet, was zu Ausschnitten wie */politik/ausland/* oder */wirtschaft/unternehmen/* führt.

- Nachrichtenseiten enthalten häufig die Begriffe *News* oder *Zeitung* in der URL.

Onlineshops

- Auch hier enthalten die URLs oft Ziffern, mit denen die Shopseiten nummeriert werden.

- Viele Shopseiten enthalten den Begriff *Shop* irgendwo in der URL (im Domainnamen, der Subdomain oder als Verzeichnisangabe).

- Häufig taucht der Begriff *Produkt* im Dateinamen oder der Verzeichnishierarchie auf.

- Webseiten, die konkrete Artikelbezeichnungen als Dateinamen benutzen, enthalten oft „-“ oder „_“ anstelle der in URLs nicht zulässigen Leerzeichen, was zu Dateinamen wie *spiel_ monopoly_ fur.html* führt.

Wissenschaftliche Webseiten

- Viele der wissenschaftlichen Webseiten enthalten den Begriff *Uni* und die Bezeichnung von Fachbereichen (z. B. *Physik* oder *Informatik*) in der URL.

- Die Top-Level-Domain (TLD) *.edu* kommt in den gesammelten Datensätzen nur bei den wissenschaftlichen Webseiten vor.

- Bei wissenschaftlichen Webseiten handelt sich in der Regel um HTML-Dateien, PHP kommt seltener zum Einsatz.

Unerwünschte Webseiten

- Die Begriffe *Suche*, *Suchmaschine* und *Katalog* sind häufig in der Internetadresse unerwünschter Webseiten vertreten.

Bei den gesammelten Daten zu unerwünschten Webseiten konnten kaum Unterschiede in den URLs der positiven und der negativen Klasse ausgemacht werden. Für Foren, Nachrichtenseiten, Onlineshops und wissenschaftliche Webseiten scheint es in der URL dagegen deutliche Hinweise zu geben.

4.1.2. Umsetzung in Attribute

N-Gramme

Da viele der Begriffe, die in den URLs einer Klasse häufig enthalten sind, auch als Teil eines zusammengesetzten Wortes vorkommen können (z. B. *Thread* in *Showthread* und *Threadid*), wird die URL in n-Gramme zerlegt (vgl. Kapitel 3.1.2). So lassen sich auch Zusammenhänge zwischen Begriffen wie *Showthread* und *Viewthread* erkennen, die beide die n-Gramme aus *Thread* enthalten.

Wählt man die Länge der n-Gramme zu klein, so ergeben sich unter Umständen viele gleiche n-Gramme in unterschiedlichen Wörtern. Ein einfaches Beispiel liefern die Wörter *Hund* und *Mund*. Die 2-Gramme (auch Bigramme) dieser Wörter unterscheiden sich lediglich in einem der jeweils drei 2-Gramme. In den 4-Grammen, hier jeweils das gesamte Wort, findet sich keine Übereinstimmung.

Da die Länge der gebildeten n-Gramme einen Einfluss auf die Klassifikationsleistung hat, wird in den später durchgeführten Versuchen zuerst für jede Klasse eine optimale Länge bestimmt.

Beim Hinzufügen der n-Gramme als zusätzliche Attribute ist zuerst die Menge aller n-Gramme aus allen beteiligten URLs zu bilden, da die Dimension aller Dokumentenvektoren gleich sein und in jeder Dimension das gleiche Attribut stehen muss, um vergleichbar zu sein. Die in der URL zu einer Webseite nicht enthaltenen n-Gramme bekommen, wie schon bei der Bildung der Wortvektoren, den Wert 0 zugewiesen.

Umsetzung der weiteren Merkmale

Alle weiteren genannten Merkmale werden entweder als numerische (z. B. Anzahl der Ziffern, Anzahl der Fragezeichen) oder als binäre Attribute (z. B. Dateiendung ist *.php*, Top-Level-Domain ist *.de*) für jede Webseite hinzugefügt. Eine vollständige Auflistung aller Attribute für die weiteren Merkmale aus den URLs findet sich in Tabelle 4.1.

Binäre Attribute	Numerische Attribute
Dateiendung ist .asp	Anzahl der Bindestriche
Dateiendung ist .htm(l)	Anzahl der Unterstriche
Dateiendung ist .jsp	Anzahl der Binde- und Unterstriche
Dateiendung ist .php	Anzahl der Fragezeichen
Sonstige Dateiendung	Anzahl der Gleichzeichen
Dateiname ist index.htm(l)	Anzahl der Kommata
Enthält Ziffern	Anzahl der Punkte
Enthält mindestens fünf Ziffern	Anzahl der Ausrufungszeichen
Protokoll ist HTTPS	Anzahl der Schrägstriche
TLD ist .biz	Anzahl der Ziffern
TLD ist .com	Länge der URL
TLD ist .de	
TLD ist .edu	
TLD ist .info	
TLD ist .net	
TLD ist .org	
Sonstige TLD	

Tabelle 4.1.: Liste der URL Attribute

4.2. Synonyme und semantisch ähnliche Wörter

Der reine Bag of Words-Ansatz, wie er in Kapitel 3 beschrieben wurde, bildet für jedes Wort ein eigenes Attribut, dessen Wert mit TF, TF-IDF (vgl. S. 17) oder ähnlichen Maßen ermittelt wird. Dabei werden Synonyme, also unterschiedliche Wörter mit gleicher oder ähnlicher Bedeutung, als unterschiedliche Attribute angesehen. Zur Textklassifikation scheint es aber sinnvoll, synonym benutzte Wörter zu einem Attribut zusammenzuziehen, also beispielsweise die Wörter *ungefähr*, *circa* und *etwa* auf dasselbe Attribut abzubilden.

Gleiches gilt je nach Klassifikationsaufgabe auch für semantisch ähnliche Wörter. Bei der Klassifikation in Nachrichten- und sonstige Webseiten erscheint es hilfreich, Namen politischer Personen auf dasselbe Attribut abzubilden, um trotz unterschiedlicher Wörter einen Zusammenhang zwischen verschiedenen Texten herstellen zu können. So kann ein Zusammenhang zwischen Sätzen wie „Merkel trifft George W. Bush“ und „Stoiber besucht Blair“ hergestellt werden, obwohl sie nicht ein Wort gemeinsam haben. Als semantisch ähnlich werden hier beispielsweise die Wörter *Merkel*, *Stoiber*, *Blair* und *Bush* betrachtet, bei denen es sich um Politiker, aber offensichtlich nicht um Synonyme im eigentlichen Sinne handelt.

Vermutlich lässt sich mit diesen Attributen für die Klasse der unerwünschten Webseiten keine wesentliche Verbesserung erzielen, da sie sich, von Internet-Suchmaschinen und Webverzeichnissen einmal abgesehen, inhaltlich an keinem

Themengebiet orientieren und gleiche Wörter oder Synonyme daher nicht unbedingt ebenfalls auf eine unerwünschte Webseite schließen lassen. Das würde dann aber auch bedeuten, dass eine Textklassifikation über die Wortvektoren im Vergleich mit den anderen Kategorien schlechter abschneidet.

4.2.1. Umsetzung in Attribute

Die Universität Leipzig hält unter dem Projekt *Wortschatz* (*wortschatz.uni-leipzig.de*) unter anderem eine Synonymdatenbank bereit, die über einen Webservice abgefragt werden kann. Dieser lässt sich aus einer Java-Anwendung heraus nutzen und bietet sich daher an, um bei der Erstellung der für die Klassifikation benötigten Wortvektoren Synonyme und semantisch ähnliche Wörter zu finden und zu einem Attribut zusammenzufassen. Der Webservice gibt unter der Suche nach Synonymen auch die hier als semantisch ähnlich bezeichneten Wörter aus.

Zur Bildung der Wortgruppen wird zuerst die Menge aller Wörter in der gesamten Textsammlung bestimmt. Anschließend wird für jedes Wort eine Abfrage in der Synonymdatenbank durchgeführt. Die Rückgabe enthält eine Menge von Wörtern, die synonym oder semantisch ähnlich zur Anfrage sind. Diese Wörter werden zu einer Gruppe zusammengefasst und auf ein neues Attribut abgebildet.

4.3. Stoppwörter und Satzzeichen

Die zehn häufigsten Wörter der deutschen Sprache sind nach Angaben der Universität Leipzig[1] in den zur Berechnung verwendeten Quellen die Wörter *der*, *die*, *und*, *in*, *den*, *von*, *zu*, *das*, *mit* und *sich*. Es ist leicht ersichtlich, dass diese Wörter zur Klassifikation von Texten nicht hilfreich sind. Solche, als Stoppwörter bezeichneten Wörter werden daher zur Textklassifikation in der Regel ausgefiltert (vgl. Kapitel 3.1.2), um die Anzahl der Attribute zu reduzieren und so die Klassifikation zu beschleunigen.

Das Vorkommen von Stoppwörtern lässt sich hier aber eventuell unterstützend nutzen, um Webseiten nach den vorher genannten Kategorien zu klassifizieren. So lässt sich unterscheiden, ob in einem Dokument ein „*normales*“ Verhältnis von Stoppwörtern zur Textlänge besteht oder ob verhältnismäßig wenige Stoppwörter vorkommen.

Analog zu Stoppwörtern kommen auch Satzzeichen in natürlichsprachlichen Texten in einem bestimmten Verhältnis zur Textlänge vor. Diese lassen sich ebenfalls zur Untersuchung von textuellen Eigenschaften nutzen.

Was als normale Anzahl an Stoppwörtern und Satzzeichen der einzelnen Klassen anzusehen ist, wird nicht untersucht oder festgelegt. Dies ist im Rahmen der folgenden Untersuchungen aber auch nicht nötig, da für eine Klassifikation lediglich

[1] wortschatz.uni-leipzig.de/html/wliste.html

entscheidend ist, ob sich für zwei gegebene Klassen Unterschiede ergeben.

Stoppwort- und Satzzeichenattribute sollen in erster Linie zu einer besseren Erkennung von Webseiten führen, die durch häufiges Aufzählen oft gesuchter Begriffe versuchen, bei möglichst vielen Suchanfragen als Ergebnis gelistet zu werden, wie in Kapitel 2.3.2 erläutert wurde. Es ist aber durchaus denkbar, dass sie auch zu einer besseren Klassifikation der anderen Kategorien aus Kapitel 2.1 führen. So werden in Onlineshops vermutlich wenige Stoppwörter benutzt, da dort meist viele Informationen stichpunktartig aufgelistet werden. Foren- und wissenschaftliche Webseiten enthalten dagegen häufig viel ausformulierten Text und demnach auch einen gewissen Anteil an Stoppwörtern und Satzzeichen.

4.3.1. Umsetzung in Attribute

Sowohl für Stoppwörter, als auch für Satzzeichen, wird ihre absolute Anzahl sowie ihre Anzahl im Verhältnis zur Länge des gegebenen Textes bestimmt. Für Satzzeichen geschieht dies zudem nochmal einzeln für Punkt, Komma, Semikolon, Frage- und Ausrufungszeichen. Damit kann unterschieden werden, ob manche Kategorien bestimmte Sonderzeichen häufiger verwenden als andere. Es wird beispielsweise erwartet, dass sich in Forenseiten mehr Fragezeichen als in Shops oder in Nachrichtenseiten finden.

Der Anfang einer HTML-Datei besteht in der Regel fast ausschließlich aus HTML-Code, unter anderem für Layout, Farben oder Werbung. Der eigentliche, im Browser angezeigte Text der Webseite steht weiter in der Mitte der Datei. Daher wird jede Datei in fünf gleich große Abschnitte unterteilt und in jedem Abschnitt noch einmal die Anzahl der Stoppwörter und Satzzeichen bestimmt (wieder absolut und im Verhältnis zur Länge des Abschnitts).

Um auch die jeweilige Verteilung der Stoppwörter über den Text abbilden zu können, wird die Anzahl der Stoppwörter bestimmt, zwischen denen maximal 3, 4-7, 8-11, 12-15, 16-19 oder mindestens 20 Nicht-Stoppwörter liegen, jeweils absolut und im Verhältnis zur Länge des Dokuments. Hiermit soll festgestellt werden, ob innerhalb des gesamten Textes zwar eine normale Anzahl an Stoppwörtern vorhanden ist, diese sich dort aber an einer Stelle häufen. Es kann z. B. sein, dass ein Teilbereich oder Absatz keine Stoppwörter enthält, ein anderer aber übermäßig viele, so dass sich die gesamte Anzahl wieder ausgleicht.

Analog dazu wird die Verteilung auch für Satzzeichen auf Attribute abgebildet, wobei hier nicht zwischen den einzelnen Satzzeichen unterschieden wird. Als Attribute dienen hier die Kategorien maximal 4, 5-9, 10-14, 15-19, 20-24 und mindestens 25 Wörter zwischen zwei Satzzeichen, jeweils absolut und im Verhältnis zur Textlänge der Datei.

4.4. HTML-Attribute

Neben den bisher genannten textuellen Eigenschaften gibt es weitere Eigenschaften, die speziell für Webseiten gelten. Dazu zählt beispielsweise die Verteilung von Links und Bildern über die Webseite, die Verwendung von iFrames oder JavaScript und der Einsatz von Passwortfeldern.

4.4.1. Links

Als Link wird ein Verweis auf eine Datei im Internet, meist eine Webseite, bezeichnet. Ein Link ist also nichts anderes als eine URL (vgl. Kapitel 4.1), die durch einen bestimmten HTML-Code als Verweis markiert ist. Im Browser werden Links in der Regel farblich hervorgehoben. Durch einen Klick auf einen Link wird die verlinkte Datei in den Browser geladen beziehungsweise heruntergeladen, falls es sich dabei nicht um eine Webseite handelt. Als interne Links werden Verweise auf Webseiten oder Dateien derselben Domain, als externe Links Verweise auf Ressourcen einer anderen Domain bezeichnet.

Sicherlich enthalten bestimmte Kategorien von Webseiten durchschnittlich mehr Links als andere. Auch das Verhältnis der externen und internen Links zueinander ist interessant. Vermutlich enthalten Shops vor allem interne Links, wogegen in Foren häufiger auf externe Webseiten, beispielsweise auf Herstellerseiten oder andere Webseiten mit weiteren Informationen, verlinkt wird.

Umsetzung in Attribute

Zur Umsetzung der Merkmale in Attribute wird die Anzahl der Links insgesamt sowie die Anzahl der internen und externen Links bestimmt, jeweils absolut und im Verhältnis zur Länge der Webseite. Ein weiteres, binäres Attribut gibt an, ob mehr interne oder mehr externe Links vorhanden sind.

Anschließend wird die Datei in fünf gleich große Abschnitte unterteilt und für jeden Abschnitt die Anzahl der internen und der externen Links bestimmt, jeweils absolut und im Verhältnis zur Länge des Abschnittes. Damit soll unterschieden werden, ob sich die Links lediglich am Anfang der Datei häufen, was z. B. auf eine Art Wegweiser oder Inhaltsverzeichnis deutet, oder ob auch Links in der Mitte und am Ende der Webseite stehen.

4.4.2. Bilder

Analog zu Links kann auch die Anzahl von Bildern auf der Webseite charakteristisch für eine Klasse sein. So setzen viele Foren Avatare (kleine, für den jeweiligen Nutzer stellvertretend stehende Grafiken) ein, mit dem alle Beiträge eines Nutzers markiert werden. Eine solche Forenseite enthält dann unter oder neben jedem Bei-

trag ein kleines Bild. Auch für die anderen Kategorien ist denkbar, dass es je nach Kategorie eine typische Anzahl und Anordnung der Bilder über die Webseite gibt.

Umsetzung in Attribute

Auch für Bilder wird ihre Anzahl absolut und im Verhältnis zur Länge der Webseite bestimmt und auf ein Attribut abgebildet. Anschließend wird die Datei wieder in fünf gleich große Abschnitte unterteilt und für jeden Abschnitt die Anzahl der Bilder bestimmt, jeweils absolut und im Verhältnis zur Anzahl der Bilder insgesamt.

4.4.3. Sonstige HTML-Merkmale

Das Vorhandensein eines Feldes zur Passworteingabe, in dem jedes Zeichen als ein Stern angezeigt wird, deutet in der Regel auf einen Login-Bereich hin, der wohl eher in Foren und Onlineshops als auf Nachrichtenseiten zu finden ist. Das Vorkommen eines solchen Feldes wird auf ein binäres Attribut abgebildet.

Auf Forenseiten werden vermutlich mehr E-Mail-Adressen angegeben als auf Nachrichtenseiten oder in Onlineshops. Die Anzahl der E-Mail-Adressen bildet daher ein weiteres, numerisches Attribut.

Mit JavaScript werden kleine Programme realisiert, die im Browser ausgeführt werden. Sie werden beispielsweise verwendet, um Benutzereingaben auf ihre Plausibilität zu prüfen, bevor die Daten an den Server geschickt werden. Dadurch wird der Server von unnötiger Arbeit entlastet. JavaScript wird auch dazu verwendet, das Erscheinungsbild einer Webseite zu verändern, beispielsweise um die Farbe und das Aussehen eines Links zu verändern, sobald der Mauszeiger über ihn fährt. Wissenschaftliche Webseiten, die sich häufig in einem schlichten Layout präsentieren, verwenden vermutlich weniger Scripte als Forenseiten oder Onlineshops, die meist ein recht aufwändiges Layout verwenden und mit vielen Zusatzfunktionen ausgestattet sind. Die Anzahl der JavaScripte bildet also ein weiteres Attribut, dessen Ausprägung charakteristisch für eine bestimmte Klasse von Webseiten sein kann.

Eingebette Frames, sogenannte iFrames, sind Bereiche innerhalb einer HTML-Datei, in die fremde Quellen, vor allem andere HTML-Dateien, eingebettet werden können. Auch hier ist denkbar, dass bestimmte Kategorien von Webseiten mehr iFrames einsetzen als andere. Die Anzahl der iFrames bildet daher ein weiteres, numerisches Attribut.

5. Versuche zur Klassifikation nach Benutzerpräferenzen

5.1. Sammeln von Beispieldaten

Um die Klassifikation von Webseiten und vor allem die Klassifikation mit Hilfe der zusätzlichen Attribute bewerten zu können, ist es zunächst notwendig, Beispiele zu sammeln und von Hand zu klassifizieren. Dabei ist der Weg über das Webinterface einer Internet-Suchmaschine sehr aufwendig, da zu jedem Suchtreffer die verlinkte Webseite von Hand gespeichert werden müsste. Möchte man eine Klassifikation von Suchtreffern nur über die von Google gelieferten Informationen vornehmen und bewerten, ohne die jeweiligen Webseiten zu besuchen, so müsste man die einzelnen Einträge der Auflistung jeweils von Hand in eine Textdatei kopieren, was sicherlich einen erheblichen Aufwand darstellt.

Google verfügt über einen Webservice, durch den sich das Sammeln von Beispielen vereinfachen lässt.

5.1.1. Google-API

Die sogenannte Google-API ermöglicht es, Suchanfragen aus einem Java-Programm heraus zu stellen. Um sie nutzen zu können ist es zunächst notwendig, sich unter *www.google.com/apis/index.html* zu registrieren. Man erhält einen eigenen Schlüssel, der zu maximal 1000 Suchanfragen pro Tag berechtigt. Das ebenfalls dort erhältliche „Google Web APIs Developer's Kit“ enthält eine .jar-Bibliothek, die alle nötigen Java-Klassen enthält, um den Webservice nutzen zu können.

Das Stellen einer Suchanfrage ist denkbar einfach: man erstellt ein Objekt der Klasse `GoogleSearch`, setzt den eigenen Schlüssel, den Suchbegriff und optional den Startindex mit Hilfe der Methoden `setKey()`, `setQueryString()` und `setStartResult()` und ruft auf dem erzeugten Objekt die Methode `doSearch()` auf. Rückgabewert der Methode ist ein Objekt der Klasse `GoogleSearchResult`, das bis zu zehn Elemente vom Typ `GoogleSearchResultElement` enthält. Jedes `GoogleSearchResultElement` repräsentiert einen Suchtreffer.

Um die nächsten zehn Treffer zu erhalten, muss lediglich der Startindex auf 10 gesetzt (die Nummerierung der Treffer beginnt bei 0) und die Methode `doSearch()` erneut aufgerufen werden.

Zu jedem Treffer lassen sich unter anderem folgende, hier interessante Informationen abrufen:

- URL der Webseite über die Methoden `getURL()`
- Titel der Webseite über die Methode `getTitle()`
- Ausschnitt, den Google auch bei einer Suche über die eigene Webseite anzeigt, über die Methode `getSnippet()`

Für weitere Informationen zur Google-API sei hier auf die von Google zur Verfügung gestellte Dokumentation verwiesen, die unter der oben genannten URL abgerufen werden kann.

5.1.2. Benutzeroberfläche für die Google-API

Um Google komfortabel zu nutzen und Webseiten entsprechend in zwei Klassen einteilen zu können, wurde eine grafische Oberfläche entwickelt (siehe Abbildung 5.1), die die Google-API nutzt. Nach Eingabe der Suchanfrage und eventuell des Startindexes kann die Suche über den Go-Button gestartet werden. Die URL und der von Google gelieferte Ausschnitt der Webseite erscheinen in jeweils einem Textfeld, der Titel der Webseite als Tooltip auf den beiden Textfeldern. Über den Button rechts neben den Textfeldern wird die jeweilige Webseite im Browser angezeigt. Links neben den Textfeldern kann die Klasse (1, 2 oder Treffer auslassen) ausgewählt werden, der der Treffer zugeordnet werden soll.

Die manuell klassifizierten Treffer werden nach Betätigung des Speichern-Buttons wie folgt auf die Festplatte geschrieben:

- Die jeweils von Google verlinkte Webseite wird heruntergeladen und in einen Ordner in dem jeweiligen Unterordner (1 oder 2) gespeichert, sofern eine der beiden Klassen gewählt wurde.
- In einem anderen Ordner wird parallel zur Webseite für jeden Treffer eine Textdatei in den jeweiligen Unterordner (1 oder 2) geschrieben, die die genaue URL der Webseite, den Titel sowie den von Google gelieferten Ausschnitt enthält.

5.1.3. Erstellung von Datensätzen

Beim Erstellen manuell klassifizierter Datensätze ist darauf zu achten, dass sowohl in der positiven als auch in der negativen Klasse zu jedem Suchbegriff gleich viele Treffer gesammelt werden. Es macht wenig Sinn, nur Suchbegriffe wie „Shop“ oder „Warenkorb“ für die Suche nach Onlineshops zu verwenden und die negative Klasse mit Treffern zu Suchbegriffen wie „Textklassifikation“ oder „Data Mining“ zu füllen.

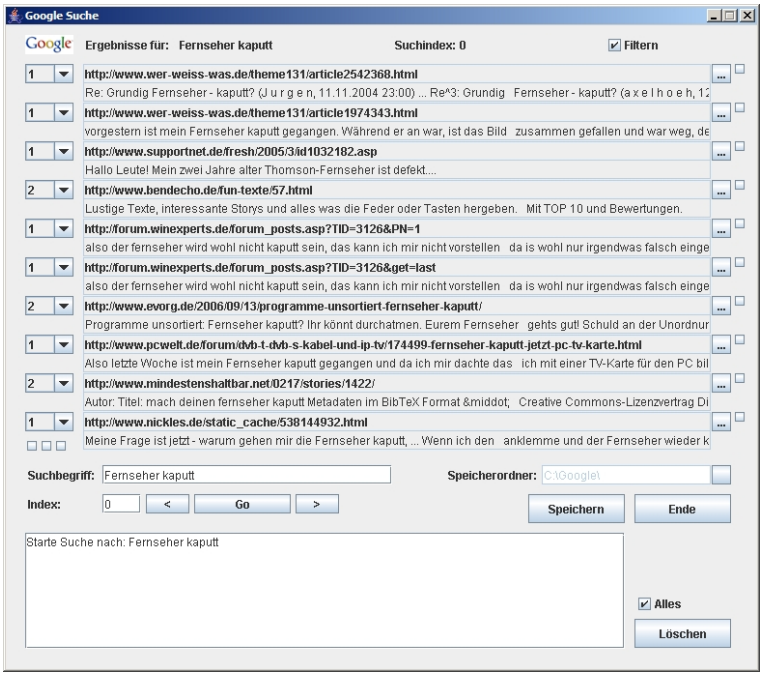


Abbildung 5.1.: Benutzeroberfläche zur manuellen Klassifikation von Suchtreffern

Da der Suchbegriff in der Regel immer in dem von Google bereitgestellten Textausschnitt enthalten ist, wäre eine vernünftige Bewertung der Klassifikationsleistung nicht möglich. Eine SVM würde allein anhand der dann nur in der entsprechenden Klasse vorkommenden Suchbegriffe zu einer guten Klassifikation kommen.

Mit Hilfe der in Abbildung 5.1 dargestellten Benutzeroberfläche wurden zum Thema Foren, Nachrichten, Shops und wissenschaftliche Webseiten jeweils 200 positive und 200 negative Beispiele manuell klassifiziert. Pro verwendetem Suchbegriff wurden dabei mindestens fünf, maximal zehn positive und genau so viele negative Treffer gespeichert.

Zu den dabei gefundenen 117 Webseiten, die der Klasse der unerwünschten Webseiten zuzuordnen sind, wurden ebenfalls 117 Webseiten mit dem jeweils gleichen Suchbegriff gesucht, die nicht der Klasse unerwünschter Webseiten angehören. Der so entstandene Datensatz hat also 117 positive und genauso viele negative Beispiele zu unerwünschten Webseiten. Da eine gezielte Suche nach unerwünschten Webseiten sehr langwierig ist, fällt dieser Datensatz im Vergleich zu den anderen vier kleiner aus.

5.2. Versuchsumgebung und -ablauf

5.2.1. Yale

Yale ([12]) ist eine am Lehrstuhl für Künstliche Intelligenz der Universität Dortmund erstellte (Test-)Umgebung für maschinelles Lernen und Data Mining, die unter *yale.sf.net* zum freien Download bereit steht. Sie wird hier mit dem dort ebenfalls zur Verfügung stehenden WordVectorTool PlugIn, das der Erstellung der für die SVM benötigten Wortvektoren dient, für die Versuche zur Textklassifikation verwendet. Das WordVectorTool stellt beim Einlesen von Dokumenten (Dateien) sicher, dass alle erzeugten Wortvektoren die gleiche Dimension haben. Als SVM wird die in Yale zur Verfügung stehende LibSVM verwendet, da sie in ersten Testdurchläufen im Vergleich mit der ebenfalls zur Verfügung stehenden JMySVM und dem MyKLRLearner die besten Ergebnisse lieferte.

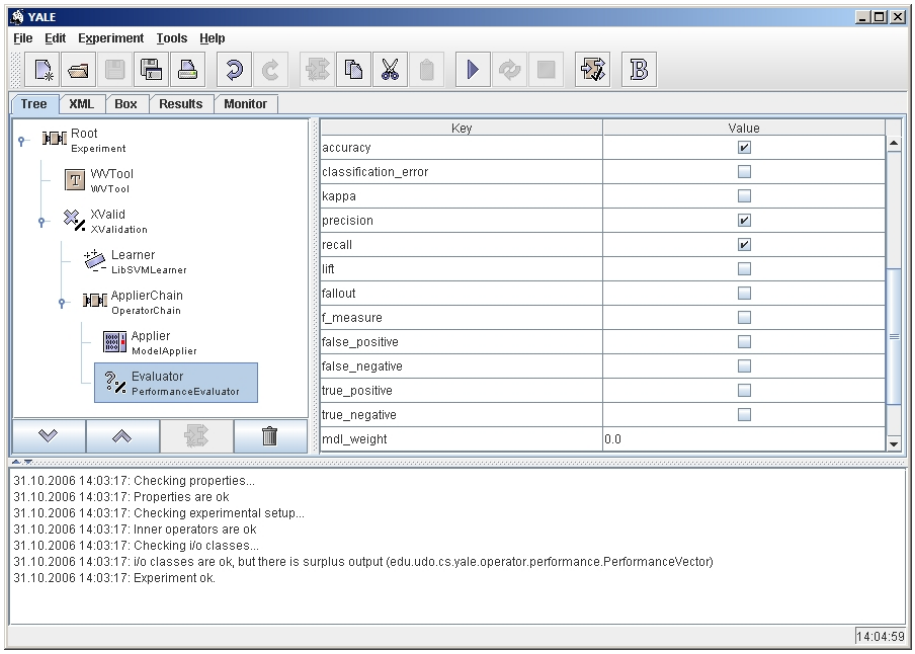


Abbildung 5.2.: Kreuzvalidierung in Yale

5.2.2. Versuchsablauf

Zuerst wird zu einem Thema mit dem WordVectorTool die positive und die negative Klasse eingelesen. Die erstellten Wortvektoren werden normalisiert, d. h. alle Attribute werden auf den Zahlenbereich von 0 bis 1 skaliert. Die normalisierten Wortvektoren werden exportiert, um sie für eine spätere Erweiterung um zusätzliche Attribute nicht neu erstellen zu müssen.

Die normalisierten Wortvektoren werden zur Kreuzvalidierung mit einer SVM benutzt (siehe Kapitel 3.1.3). Für die Versuche wird eine Kreuzvalidierung mit zehn Durchgängen verwendet.

Um einen Ausgangswert zu erhalten, an dem die zusätzlichen Attribute aus Kapitel 4 gemessen werden können, wird zunächst eine Kreuzvalidierung nur mit den Wortvektoren der Google-Schnipsel und eine Kreuzvalidierung nur mit den HTML-Dateien durchgeführt. Anschließend wird mit den Google-Schnipseln und den HTML-Dateien zu jeder Gruppe zusätzlicher Merkmale je eine Kreuzvalidierung mit und ohne die Wortvektoren durchgeführt. Dabei gelten in allen später dargestellten Tabellen und Beschreibungen folgende Bezeichnungen:

- **Synonymgruppen** - Die mit dem Wortschatz der Universität Leipzig zu Gruppen zusammengefassten Synonyme und semantisch ähnlichen Wörter.
- **URL n-Gramme** - Die aus den URLs gebildeten n-Gramme.
- **URL Attribute** - Die restlichen, aus der URL gebildeten Attribute aus Kapitel 4.1.2 (ohne n-Gramme).
- **URL gesamt** - Zusammenfassung der Gruppen URL n-Gramme und URL Attribute, so dass hier alle aus der URL generierten Attribute enthalten sind.
- **HTML-Attribute** - Anzahl und Verteilung der Links und Bilder über die Webseite sowie die weiteren Attribute aus Kapitel 4.4.
- **Satzzeichen** - Anzahl und Verteilung von Satzzeichen über die Webseite (siehe Kapitel 4.3.1).
- **Stoppwörter** - Anzahl und Verteilung von Stoppwörter über die Webseite (siehe Kapitel 4.3.1).
- **Alle Attribute** - Zusammenfassung aller zusätzlich generierten Attribute.

5.3. Versuchsergebnisse

In einem Vorlauf wurde zu jedem Datensatz die Länge der URL n-Gramme bestimmt, die zur besten Klassifikationsleistung führt. Für Forenseiten und Shops liegt sie bei 4, auf den restlichen Klassen konnten die besten Ergebnisse bei einer Länge von 3 erreicht werden. Diese Werte werden in den Durchläufen zur Erzeugung der URL n-Gramme verwendet.

Im Folgenden werden die Ergebnisse der Versuchsdurchläufe beschrieben, wobei hier zum Vergleich, ob eine Verbesserung oder Verschlechterung eingetreten ist, lediglich die Accuracy (siehe Kapitel 3.1.3) betrachtet wird. Bis auf wenige Ausnahmen sind bei einer höheren Accuracy in den durchgeführten Versuchen auch die Werte für Precision und Recall höher.

Ein Durchlauf von mehreren Kreuzvalidierungen auf denselben Daten lieferte Schwankungen von bis zu einem Prozent zwischen der höchsten und der niedrigsten Accuracy. Die Schwankungen erklären sich dadurch, dass bei der Kreuzvalidierung die Beispiele zufällig in Trainings- und Testdaten eingeteilt werden. Je nach Einteilung erhält man unterschiedliche Werte, da auf anderen Trainingsdaten ein etwas anderes Modell gelernt wird, das auf den dann zur Verfügung stehenden Testdaten zu einer anderen Accuracy führt. Die Kreuzvalidierung bildet den Schnitt aus mehreren Durchläufen (hier zehn), aber auch dieser schwankt durch die unterschiedlichen Werte in den einzelnen Durchläufen. Aus diesem Grund werden in den folgenden Versuchen Veränderungen um weniger als einem Prozent als normale Schwankungen angesehen und nicht weiter betrachtet.

5.3.1. Forenseiten

Bei einer Kreuzvalidierung nur mit den Wortvektoren der Google-Schnipsel (siehe Tabelle 5.1) erreicht die SVM eine Accuracy von 90,17%. Sie wird hier allein durch die Hinzunahme der Synonymgruppen auf 92,25% gesteigert. Durch eine Ergänzung der Wortvektoren nur um die Attribute der 4-Gramme der URL kann eine Accuracy von 92,50% erreicht werden. Bemerkenswert ist auch, dass die Klassifikation von Forenseiten nur über die 4-Gramme der URL, also ohne die Wörter aus den Google-Schnipseln zu betrachten, bereits zu einer Accuracy von 88,92% führt.

Die URL Attribute (wie auch Satzzeichen- und Stoppwortattribute) verschlechtern die Accuracy leider etwas, wobei auffällt, dass die Kombination von Wortvektoren, URL n-Grammen und URL Attributen mit einer Accuracy von 93,75% zu einem besseren Ergebnis führt als ohne die URL Attribute. Als Ergänzung zu den n-Grammen führen die URL Attribute also zu einer weiteren Verbesserung.

Bei Hinzunahme aller zusätzlichen Attribute zu den Wortvektoren kommt die SVM auf eine Accuracy von 93,17%. Sie ist damit um 3% besser als bei der Klassifikation nur über die Wortvektoren, aber wiederum schlechter als nur mit Wortvektoren und allen aus der URL generierten Attributen. Die Satzzeichen- und Stoppwortattribute scheinen hier das Ergebnis wieder etwas zu verschlechtern. Lässt man diese Attribute weg, führt also nur eine Kreuzvalidierung mit den Wortvektoren, den Synonymgruppen und allen aus der URL generierten Attributen durch (nicht in der Tabelle dargestellt), erreicht die SVM eine Accuracy von 94,25%.

Die Erkennung von Foren anhand der Google-Schnipsel, also ohne die jeweiligen Webseiten herunterzuladen, eignet sich mit einer Accuracy von weit über 90% bereits für eine praktische Anwendung!

Die auf den HTML-Dateien (siehe Tabelle 5.2) erreichte Accuracy liegt mit 93,83% über der Accuracy der Textklassifikation auf den Google-Schnipseln, aber unter der auf den Google-Schnipseln mit zusätzlichen Attributen erreichbaren Accuracy (94,25% mit Synonymgruppen und URL gesamt). Doch auch hier kann mit zusätzlichen Attributen eine Verbesserung der Klassifikation erreicht werden.

Attribute	Precision	Recall	Accuracy
nur Wortvektoren	87,78	93,33	90,17
Synonymgruppen ohne Wortvektoren	78,08	82,50	79,67
Synonymgruppen mit Wortvektoren	90,05	95,00	92,25
URL Attribute ohne Wortvektoren	76,38	76,00	76,25
URL Attribute mit Wortvektoren	88,24	90,00	89,00
URL n-Gramme ohne Wortvektoren	87,84	90,33	88,92
URL n-Gramme mit Wortvektoren	90,09	95,50	92,50
URL gesamt ohne Wortvektoren	87,22	89,83	88,33
URL gesamt mit Wortvektoren	92,00	95,83	93,75
Satzzeichen ohne Wortvektoren	69,94	76,00	71,67
Satzzeichen mit Wortvektoren	86,57	91,33	88,58
Stoppwörter ohne Wortvektoren	54,38	49,50	54,00
Stoppwörter mit Wortvektoren	86,30	91,33	88,42
Alle Attribute ohne WVTool	86,44	91,33	88,50
Alle Attribute mit WVTool	90,98	95,83	93,17

Tabelle 5.1.: Forenseiten (Google-Schnipsel)

Die Synonymgruppen verbessern die Accuracy einer Klassifikation der HTML-Dateien auf 94,92%, die URL n-Gramme sogar auf 96,17%. HTML- und Stoppwortattribute verbessern die Klassifikation nur unwesentlich, Satzzeichenattribute führen zu einer minimalen Verschlechterung der Klassifikation. Warum eine Verschlechterung durch zusätzliche Attribute möglich ist, wird in Kapitel 5.4.3 diskutiert.

Zusammen mit allen zusätzlichen Attributen kann die Accuracy der Klassifikation der HTML-Dateien von 93,83% auf 97,00% gesteigert werden. Die Fehlklassifikationsrate wurde also von 6,17% auf 3,00% um mehr als die Hälfte reduziert!

5.3.2. Nachrichtenseite

Die Klassifikation von Nachrichtenseiten nur über die Wortvektoren der Google-Schnipsel (siehe Tabelle 5.3) fällt mit einer Accuracy von 84,42% schlechter als bei den Foren aus. Dieser Wert kann hier durch die Synonymgruppen nicht verbessert werden. Aus der URL lassen sich aber wieder Attribute generieren, die zu einer besseren Klassifikation führen. Sowohl die verwendeten URL Attribute als auch die URL n-Gramme führen zu einer Accuracy von jeweils 86,00%.

Die Satzzeichenattribute verbessern die Klassifikation nur minimal, Stoppwortattribute führen zu einem deutlichen Abfall der Accuracy um über 2,5% auf 81,75%. Auch hier sei auf Kapitel 5.4.3 verwiesen, in dem diskutiert wird, warum eine Verschlechterung durch zusätzliche Attribute überhaupt möglich ist.

Attribute	Precision	Recall	Accuracy
nur Wortvektoren	91,48	96,67	93,83
Synonymgruppen ohne Wortvektoren	86,00	87,00	86,42
Synonymgruppen mit Wortvektoren	93,68	96,33	94,92
URL Attribute ohne Wortvektoren	76,51	76,00	76,33
URL Attribute mit Wortvektoren	92,25	95,17	93,58
URL n-Gramme ohne Wortvektoren	87,52	88,83	88,08
URL n-Gramme mit Wortvektoren	94,82	97,67	96,17
URL gesamt ohne Wortvektoren	87,24	90,00	88,42
URL gesamt mit Wortvektoren	94,34	97,33	95,75
HTML-Attribute ohne Wortvektoren	78,58	79,50	78,92
HTML-Attribute mit Wortvektoren	92,19	96,50	94,17
Satzzeichen ohne Wortvektoren	74,78	84,00	77,83
Satzzeichen mit Wortvektoren	91,71	95,83	93,58
Stoppwörter ohne Wortvektoren	70,86	74,17	71,83
Stoppwörter mit Wortvektoren	92,13	97,50	94,58
Alle Attribute ohne WVTool	94,55	95,50	95,00
Alle Attribute mit WVTool	95,93	98,17	97,00

Tabelle 5.2.: Forenseiten (HTML-Dateien)

Zusammen mit den Wortvektoren und allen zusätzlichen Attributen kann eine Accuracy von 86,17% erreicht werden, die durch das Weglassen der Stoppwortattribute nicht weiter gesteigert werden kann.

Auf den HTML-Dateien kann, wie Tabelle 5.4 zeigt, der im Vergleich mit den Google-Schnipseln etwas höhere Ausgangswert von 88,50% durch die URL Attribute und URL n-Gramme nur noch jeweils um knapp einen Prozentpunkt auf 89,42% gesteigert werden. Die anderen Attribute führen hier zu keiner nennenswerten Verbesserung. Die Stoppwortattribute, die auf den Google-Schnipseln die Accuracy noch deutlich verschlechtert haben, führen auf den HTML-Dateien zu einer geringen Verbesserung innerhalb des als Schwankung festgelegten Bereiches, verschlechtern die Klassifikation hier also nicht mehr.

Unter Verwendung aller zusätzlichen Attribute kann die Accuracy von 88,50% auf 89,92% gesteigert werden.

5.3.3. Onlineshops

Die Werte des Durchlaufs mit den Google-Schnipseln der Onlineshops ist in Tabelle 5.5 dargestellt. Die Accuracy sinkt durch Hinzunahme der Synonymgruppen von 82,33% auf 81,17%. URL- und Stoppwortattribute führen hier ebenfalls zu einer Verschlechterung.

Attribute	Precision	Recall	Accuracy
nur Wortvektoren	83,79	85,33	84,42
Synonymgruppen ohne Wortvektoren	63,03	66,50	63,75
Synonymgruppen mit Wortvektoren	84,20	83,50	83,92
URL Attribute ohne Wortvektoren	80,88	82,50	81,50
URL Attribute mit Wortvektoren	86,36	85,50	86,00
URL n-Gramme ohne Wortvektoren	80,47	79,67	80,17
URL n-Gramme mit Wortvektoren	85,88	86,17	86,00
URL gesamt ohne Wortvektoren	83,79	82,67	83,33
URL gesamt mit Wortvektoren	86,62	84,17	85,58
Satzzeichen ohne Wortvektoren	68,00	68,00	68,00
Satzzeichen mit Wortvektoren	85,05	85,33	85,17
Stoppwörter ohne Wortvektoren	63,67	60,17	62,92
Stoppwörter mit Wortvektoren	83,13	79,67	81,75
Alle Attribute ohne WVTool	83,75	84,17	83,92
Alle Attribute mit WVTool	86,53	85,67	86,17

Tabelle 5.3.: Nachrichtenseiten (Google-Schnipsel)

Attribute	Precision	Recall	Accuracy
nur Wortvektoren	85,32	93,00	88,50
Synonymgruppen ohne Wortvektoren	84,06	86,17	84,92
Synonymgruppen mit Wortvektoren	86,41	92,17	88,83
URL Attribute ohne Wortvektoren	80,88	82,50	81,50
URL Attribute mit Wortvektoren	86,44	93,50	89,42
URL n-Gramme ohne Wortvektoren	80,20	80,33	80,25
URL n-Gramme mit Wortvektoren	85,78	94,50	89,42
URL gesamt ohne Wortvektoren	82,39	82,67	82,50
URL gesamt mit Wortvektoren	86,05	92,50	88,75
HTML-Attribute ohne Wortvektoren	79,39	86,00	81,83
HTML-Attribute mit Wortvektoren	85,72	92,00	88,33
Satzzeichen ohne Wortvektoren	66,74	55,17	63,83
Satzzeichen mit Wortvektoren	85,83	92,83	88,75
Stoppwörter ohne Wortvektoren	77,82	77,17	77,58
Stoppwörter mit Wortvektoren	85,73	93,17	88,83
Alle Attribute ohne WVTool	86,61	87,33	86,92
Alle Attribute mit WVTool	87,72	92,83	89,92

Tabelle 5.4.: Nachrichtenseiten (HTML-Dateien)

Attribute	Precision	Recall	Accuracy
nur Wortvektoren	79,75	86,67	82,33
Synonymgruppen ohne Wortvektoren	71,35	64,33	69,25
Synonymgruppen mit Wortvektoren	79,68	83,67	81,17
URL Attribute ohne Wortvektoren	55,16	62,83	55,92
URL Attribute mit Wortvektoren	78,41	85,33	80,92
URL n-Gramme ohne Wortvektoren	75,25	76,50	75,67
URL n-Gramme mit Wortvektoren	82,25	86,50	83,92
URL gesamt ohne Wortvektoren	73,40	72,67	73,17
URL gesamt mit Wortvektoren	80,03	86,83	82,58
Satzzeichen ohne Wortvektoren	59,09	67,17	60,33
Satzzeichen mit Wortvektoren	80,85	85,17	82,50
Stoppwörter ohne Wortvektoren	67,77	64,83	67,00
Stoppwörter mit Wortvektoren	79,02	82,83	80,42
Alle Attribute ohne WVTool	76,52	78,17	77,08
Alle Attribute mit WVTool	83,31	86,50	84,58

Tabelle 5.5.: Onlineshops (Google-Schnipsel)

Die URL n-Gramme führen zu einer leichten Verbesserung der Accuracy auf 83,92%. Die Verbesserung durch Hinzunahme der Satzzeichenattribute zu den Wortvektoren ist nur sehr gering. Bei einer Kreuzvalidierung mit den Wortvektoren, den URL n-Grammen und den Satzzeichenattributen (nicht in der Tabelle dargestellt) erreicht die SVM eine Accuracy von 86,45%, so dass die Satzzeichenattribute im Zusammenspiel mit den URL n-Grammen zu einer weiteren Verbesserung führen.

Auf den HTML-Dateien der Onlineshops kann, wie Tabelle 5.6 zeigt, leider keine nennenswerte Verbesserung erzielt werden. Lediglich die URL n-Gramme, die bereits bei den Google-Schnipseln zu einer Verbesserung geführt haben, steigern die Accuracy gegenüber des Durchlaufs nur mit den Wortvektoren von 91,33% auf 92,00%, was aber innerhalb des hier als einfache Schwankung festgelegten Bereiches liegt.

5.3.4. Wissenschaftliche Webseiten

Beim Erkennen wissenschaftlicher Webseiten nach der Definition aus Kapitel 2.1 kann sowohl für die Google-Schnipsel (Tabelle 5.7) als auch für die vollständigen HTML-Dateien (Tabelle 5.8) eine Verbesserung erzielt werden.

Die Verbesserung der Klassifikation über die Google-Schnipsel fällt hier mit einer Steigerung von 74,83% auf 82,17% besonders deutlich aus. Mit Ausnahme der Stoppwortattribute, die das Ergebnis gegenüber der Klassifikation nur über die

Attribute	Precision	Recall	Accuracy
nur Wortvektoren	91,06	91,67	91,33
Synonymgruppen ohne Wortvektoren	80,36	82,50	81,17
Synonymgruppen mit Wortvektoren	91,79	91,33	91,58
URL Attribute ohne Wortvektoren	56,09	62,17	56,75
URL Attribute mit Wortvektoren	90,45	91,50	90,92
URL n-Gramme ohne Wortvektoren	74,96	76,33	75,42
URL n-Gramme mit Wortvektoren	92,00	92,00	92,00
URL gesamt ohne Wortvektoren	73,76	72,17	73,25
URL gesamt mit Wortvektoren	90,88	91,33	91,08
HTML-Attribute ohne Wortvektoren	70,74	68,50	70,08
HTML-Attribute mit Wortvektoren	91,78	91,17	91,50
Satzzeichen ohne Wortvektoren	64,71	77,00	67,50
Satzzeichen mit Wortvektoren	90,89	91,50	91,17
Stoppwörter ohne Wortvektoren	77,71	63,33	72,58
Stoppwörter mit Wortvektoren	90,95	90,50	90,75
Alle Attribute ohne WVTool	84,31	84,17	84,25
Alle Attribute mit WVTool	91,25	92,17	91,67

Tabelle 5.6.: Onlineshops (HTML-Dateien)

Wortvektoren etwas verschlechtern, führen alle anderen zusätzlichen Attribute zu einer Verbesserung. Hier ist eine Klassifikation nur über die URL n-Gramme, also ohne die aus dem Ausschnitt der Webseite gebildeten Wortvektoren, mit einer Accuracy von 77,08% bereits besser als die Klassifikation über die Wortvektoren.

Bei den HTML-Dateien fällt die Verbesserung nicht mehr ganz so deutlich aus, aber auch hier kann allein durch die Einbeziehung der URL (URL Attribute und URL n-Gramme) in die Klassifikation die Accuracy von 87,00% auf 89,25% gesteigert werden. Synonymgruppen führen ebenfalls zu einer Verbesserung um 1,25% gegenüber der Klassifikation nur über die Wortvektoren. Die restlichen Attribute haben kaum Einfluss auf die Klassifikationsleistung. Mit allen zusätzlichen Attributen kann eine Accuracy von 89,67% erreicht werden.

5.3.5. Unerwünschte Webseiten

Tabelle 5.9 stellt die Ergebnisse der Kreuzvalidierung der Google-Schnipsel von unerwünschten Webseiten dar. Hier führen die vor allem für diese Klasse eingeführten Satzzeichen- und Stoppwortattribute zu einer Steigerung der Accuracy von 69,37% auf jeweils über 72%. Die Einbeziehung der aus der URL generierten Attribute führt ebenfalls zu einer Verbesserung, Synonymgruppen verbessern das Ergebnis nicht. Mit allen zusätzliche Attributen kann die Accuracy von 69,37%

Attribute	Precision	Recall	Accuracy
nur Wortvektoren	73,50	77,67	74,83
Synonymgruppen ohne Wortvektoren	59,60	61,00	59,83
Synonymgruppen mit Wortvektoren	78,46	78,33	78,42
URL Attribute ohne Wortvektoren	71,08	67,17	69,92
URL Attribute mit Wortvektoren	79,29	78,50	79,00
URL n-Gramme ohne Wortvektoren	77,31	76,67	77,08
URL n-Gramme mit Wortvektoren	81,49	78,50	80,33
URL gesamt ohne Wortvektoren	76,96	76,83	76,92
URL gesamt mit Wortvektoren	80,82	79,33	80,25
Satzzeichen ohne Wortvektoren	72,43	58,67	68,17
Satzzeichen mit Wortvektoren	77,37	77,50	77,42
Stoppwörter ohne Wortvektoren	60,44	59,33	60,25
Stoppwörter mit Wortvektoren	72,32	75,33	73,25
Alle Attribute ohne WVTool	78,31	78,83	78,50
Alle Attribute mit WVTool	83,06	80,83	82,17

Tabelle 5.7.: Wissenschaftliche Webseiten (Google-Schnipsel)

Attribute	Precision	Recall	Accuracy
nur Wortvektoren	84,58	90,50	87,00
Synonymgruppen ohne Wortvektoren	76,96	86,83	80,42
Synonymgruppen mit Wortvektoren	86,49	90,67	88,25
URL Attribute ohne Wortvektoren	72,04	67,00	70,50
URL Attribute mit Wortvektoren	86,10	89,83	87,67
URL n-Gramme ohne Wortvektoren	76,66	77,17	76,83
URL n-Gramme mit Wortvektoren	86,96	90,00	88,25
URL gesamt ohne Wortvektoren	78,70	77,00	78,08
URL gesamt mit Wortvektoren	88,29	90,50	89,25
HTML-Attribute ohne Wortvektoren	67,31	64,50	66,58
HTML-Attribute mit Wortvektoren	84,69	88,50	86,25
Satzzeichen ohne Wortvektoren	78,39	71,33	75,83
Satzzeichen mit Wortvektoren	85,33	90,17	87,33
Stoppwörter ohne Wortvektoren	78,12	88,67	81,92
Stoppwörter mit Wortvektoren	85,69	90,83	87,83
Alle Attribute ohne WVTool	86,86	88,17	87,42
Alle Attribute mit WVTool	88,02	91,83	89,67

Tabelle 5.8.: Wissenschaftliche Webseiten (HTML-Dateien)

Attribute	Precision	Recall	Accuracy
nur Wortvektoren	69,66	68,66	69,37
Synonymgruppen ohne Wortvektoren	65,90	57,26	63,82
Synonymgruppen mit Wortvektoren	68,22	71,51	69,09
URL Attribute ohne Wortvektoren	58,40	58,40	58,41
URL Attribute mit Wortvektoren	69,30	72,65	70,23
URL n-Gramme ohne Wortvektoren	63,83	59,83	62,96
URL n-Gramme mit Wortvektoren	69,33	74,07	70,66
URL gesamt ohne Wortvektoren	66,11	67,81	66,52
URL gesamt mit Wortvektoren	70,09	76,07	71,79
Satzzeichen ohne Wortvektoren	58,63	58,97	58,69
Satzzeichen mit Wortvektoren	71,79	73,22	72,22
Stoppwörter ohne Wortvektoren	66,10	64,96	65,81
Stoppwörter mit Wortvektoren	71,95	73,79	72,51
Alle Attribute ohne WVTool	72,55	76,07	73,65
Alle Attribute mit WVTool	74,66	78,06	75,78

Tabelle 5.9.: Unerwünschte Webseiten (Google-Schnipsel)

auf 75,78% gesteigert werden.

In der in Tabelle 5.10 dargestellten Klassifikation unerwünschter Webseiten anhand der HTML-Dateien kann mit den bislang kaum nützlichen HTML-Attributen (u. a. Link- und Bildverteilung) eine Verbesserung der Accuracy von 84,33% auf 87,61% erreicht werden. Satzzeichenattribute führen hier ebenfalls zu einer Steigerung der Accuracy auf 85,47%.

Die Klassifikation mit allen zusätzlichen Attributen schneidet hier schlechter ab als die Klassifikation nur über die Wortvektoren, was vor allem an den aus der URL generierten Attributen liegt, die bereits als Zusatz zu den Wortvektoren die Accuracy von 84,33% auf 82,76% verschlechtern. Führt man eine Klassifikation nur über die Attribute durch, die hier zu einer Verbesserung führen (HTML- und Satzzeichenattribute), erreicht die SVM eine Accuracy von 88,03% (nicht in der Tabelle dargestellt).

5.4. Auswertung der Ergebnisse

Im Folgenden werden die soeben präsentierten Ergebnisse genauer betrachtet. Interessant ist sowohl ein Vergleich der Klassifikation der Google-Schnipsel mit der Klassifikation anhand der vollständigen HTML-Dateien, als auch eine Betrachtung dessen, welche Attribute für die einzelnen Klassen zu Verbesserungen führen. Im Anschluss wird erläutert, warum die Klassifikation an einigen Stellen durch die Hinzunahme zusätzlicher Attribute verschlechtert wurde.

Attribute	Precision	Recall	Accuracy
nur Wortvektoren	85,56	82,62	84,33
Synonymgruppen ohne Wortvektoren	72,39	67,24	70,80
Synonymgruppen mit Wortvektoren	86,77	80,34	84,05
URL Attribute ohne Wortvektoren	59,56	60,40	59,69
URL Attribute mit Wortvektoren	86,38	79,49	83,47
URL n-Gramme ohne Wortvektoren	66,13	60,11	64,67
URL n-Gramme mit Wortvektoren	84,78	80,91	83,19
URL gesamt ohne Wortvektoren	68,86	65,53	67,95
URL gesamt mit Wortvektoren	84,02	80,91	82,76
HTML-Attribute ohne Wortvektoren	73,54	75,21	74,07
HTML-Attribute mit Wortvektoren	89,29	85,47	87,61
Satzzeichen ohne Wortvektoren	75,34	80,06	76,92
Satzzeichen mit Wortvektoren	84,88	86,32	85,47
Stoppwörter ohne Wortvektoren	76,65	62,68	71,79
Stoppwörter mit Wortvektoren	84,20	83,48	83,90
Alle Attribute ohne WVTool	81,68	77,49	80,05
Alle Attribute mit WVTool	85,85	81,20	83,90

Tabelle 5.10.: Unerwünschte Webseiten (HTML-Dateien)

5.4.1. Vergleich Google-Schnipsel / HTML-Dateien

Tabelle 5.11 stellt die Accuracy der Textklassifikation ohne zusätzliche Attribute noch einmal für die Google-Schnipsel und die HTML-Dateien gegenüber. Die Werte der Klassifikation mit allen zusätzlichen Attributen sind in Tabelle 5.12 dargestellt. Die in Klammern dargestellten Werte können erreicht werden, wenn Attributgruppen weggelassen werden, die sich dort negativ auf die Klassifikation auswirken.

Klasse	Google-Schnipsel	HTML-Dateien
Foren	90,17	93,83
Nachrichten	84,42	88,50
Shops	82,33	91,33
Wissenschaftliche Webseiten	74,83	87,00
Unerwünschte Webseiten	69,37	84,33

Tabelle 5.11.: Ergebnisse ohne zusätzliche Attribute

Es wird deutlich, dass für Foren- und Nachrichtenseiten der Unterschied zwischen einer Klassifikation der Google-Schnipsel und der HTML-Dateien nicht besonders groß ist, so dass es sich nur bedingt lohnt, die Zeit für eine Klassifikation

Klasse	Google-Schnipsel	HTML-Dateien
Foren	93,17 (94,25)	97,00
Nachrichten	86,17	89,92
Shops	84,58	91,67
Wissenschaftliche Webseiten	82,17 (86,45)	89,67
Unerwünschte Webseiten	75,78	83,90 (88,03)

Tabelle 5.12.: Ergebnisse mit zusätzlichen Attributen

der HTML-Dateien, die dafür erst aus dem Internet geladen werden müssen, abzuwarten. Vor allem Forenseiten lassen sich mit einer Accuracy von 93,17% bei Verwendung aller zusätzlichen Attribute (bzw. 94,25% bei einer Ergänzung der Wortvektoren lediglich um Synonymgruppen und alle aus der URL generierten Attribute) für den normalen Nutzer bereits sehr gut über die Google-Schnipsel klassifizieren.

Für Onlineshops und wissenschaftliche Webseiten ist der Unterschied zwischen den Google-Schnipseln und den HTML-Dateien um einiges größer, kann durch zusätzliche Attribute aber etwas verringert werden. Mit einer Differenz der Accuracy von jeweils etwas über 6% empfiehlt es sich dennoch, die Klassifikation von Shops und wissenschaftlichen Seiten über die HTML-Dateien vorzunehmen.

Die Erkennung unerwünschter Webseiten über die Google-Schnipsel kann durch zusätzliche Attribute zwar verbessert werden, arbeitet aber deutlich unzuverlässiger als eine Klassifikation HTML-Dateien und empfiehlt sich nicht für einen Praxiseinsatz.

5.4.2. Attributgewichte

Wie in Kapitel 3.3.4 beschrieben wurde, ist die SVM in der Lage, zu jedem Attribut ein entsprechendes Gewicht auszugeben. Das Vorzeichen des Gewichts gibt an, ob das Attribut eher Indiz für die eine oder die andere Klasse ist. Je höher der Betrag des Gewichts eines Attributs ist, desto höher ist der Einfluss des einzelnen Attributs auf die Klassifikation.

Für jeden Datensatz werden jeweils nach der Kreuzvalidierung die Gewichte aller Attribute exportiert und absteigend nach ihrem Betrag sortiert. Betrachtet werden nun die 100 Attribute mit dem höchsten Gewichtsbetrag. Diese Anzahl ist willkürlich gewählt, reicht aber hier zur Untersuchung, welche Attribute die SVM als besonders aussagekräftig für die Klassifikation bewertet.

Attributgewichte auf den Google-Schnipseln

Tabelle 5.13 stellt für jede Klasse die Verteilung der 100 Attribute mit dem höchsten Gewichtsbetrag für die Google-Schnipsel dar. Jeweils etwas mehr als die Hälfte

der 100 am höchsten bewerteten Attribute sind Wortattribute aus den Wortvektoren, der Rest setzt sich aus zusätzlichen Attributen zusammen. Hier nehmen vor allem die URL n-Gramme einen großen Teil ein, wobei zu beachten ist, dass ein Begriff in der Regel aus mehreren n-Grammen besteht, die dann oft alle hoch bewertet werden.

Google-Schnipsel	Foren	Nachr.	Shops	Wissensch.	Unerw.
Wortattribute (aus WVTool)	53	58	58	53	66
Synonymgruppen	9	5	10	4	7
URL Attribute	6	6	1	7	1
URL n-Gramme	29	25	25	29	20
Satzzeichen	3	5	1	3	1
Stoppwörter	0	1	5	4	5

Tabelle 5.13.: Verteilung der Attribute für Google-Schnipsel

Forenseiten Bei der Klassifikation von Forenseiten werden folgende URL Attribute von der SVM mit einem hohen Gewicht bewertet: PHP als Dateiformat, Anzahl der Fragezeichen, Anzahl der Punkte und Anzahl der Gleichheitszeichen als Indiz für eine Forenseite sowie die Anzahl der Schrägstriche und eine sonstige Dateiendung (vgl. 4.1.2) als Indiz gegen eine Forenseite.

Die 4-Gramme aus der URL der Foren, die von der SVM mit einem hohen Gewicht bewerteten werden, sind die 4-Gramme aus *Forum*, *Board*, *Viewtopic*, *Showtopic* und *Showthread*. Betrachtet man die URLs der beiden Klassen, ist dies naheliegend: 151 der 200 Webseiten enthalten mindestens einen der (Teil-)Begriffe *Forum*, *Thread*, *Topic*, *Show* und *View*. Dem gegenüber stehen nur sechs Vorkommen in den 200 URLs der anderen Klasse.

Bei den Forenseiten gehören zu den hoch bewerteten Satzzeichenattribute die Anzahl der Fragezeichen absolut sowie im Verhältnis zur Länge (was bei den häufig gleich langen Google-Schnipseln vermutlich keinen großen Unterschied macht) und die Anzahl der Sonderzeichen im ersten Fünftel der erstellten Dateien. Da bei allen Dateien die URL der verlinkten Webseite in den Anfang der Datei geschrieben wurde, beziehen sich die Werte nicht unbedingt nur auf die Google-Schnipsel sondern auch auf die URL. Vor allem das Attribut „Anzahl der Sonderzeichen im ersten Fünftel“ wird sich nur auf die URL beziehen, deren Anzahl an Fragezeichen schon bei den URL Attributen ein hohes Gewicht bekommen hat.

Keines der verwendeten Stoppwortattribute hat es hier unter die 100 Attribute mit dem höchsten Gewicht geschafft.

Nachrichtenseiten URL Attribute, die bei der Klassifikation von Nachrichtenseiten hervorzuheben sind, sind das Vorkommen von Ziffern und vielen Schräg-

strichen sowie eine etwas längere URL. Gegen eine Nachrichtenseite spricht die Verwendung von *index.html* als Dateiname der Webseite.

Die für die SVM am interessantesten 3-Gramme sind das 3-Gramm *zei* (aus *Zeitung* und *Schlagzeile*) sowie die jeweils zwei 3-Gramme aus *News* und *Info*, die für eine Nachrichtenseite sprechen. Von anderen Wörtern sind nur Teile der 3-Gramme unter den 100 am höchsten bewerteten Attributen. So gewichtet die SVM das 3-Gramm *wel* aus *Welt* hoch, das 3-Gramm *elt* hat ein eher geringes Gewicht. Eine genauere Betrachtung zeigt, dass es in der Klasse der Nicht-Nachrichtenseiten mehrfach als Teil von *Eltern* vorkommt, was die Abwertung durch die SVM erklärt. Gleiches gilt beispielsweise auch für die 3-Gramme *mel* und *ldu* aus *Meldung*. Sie bekommen von der SVM ein hohes Gewicht, während die restlichen 3-Gramme aus *Meldung* nicht besonders hoch gewichtet werden. Das 3-Gramm *eld* kommt in der Klasse der Nicht-Nachrichtenseiten mehrfach als Teil von *Geld*, die 3-Gramme *dun* und *ung* unter anderem als Teil von *Bildung* und *Abfindung* vor. N-Gramme, die hier gegen eine Nachrichtenseite sprechen und von der SVM ein hohes Gewicht erhalten haben, sind z. B. *sei* und *ite* aus *Seite*, während das auch in *Zeit* vorkommende 3-Gramm *eit* in beiden Klassen ähnlich häufig vertreten ist und daher geringer gewichtet wird.

Für die Satzzeichen besteht, wie in den folgenden Klassen auch, das gleiche Problem wie bei den Foren. Bezüglich der Satzzeichenattribute auf den Dateien mit den Google-Schnipseln ist bei lediglich hervorzuheben, dass die Anzahl der Ausrufungszeichen in den Ausschnitten der Nachrichtenseiten von der SVM ein hohes Gewicht erhält, das nicht auf die URL zurückzuführen ist.

Von den Stoppwortattributen hat es ein Attribut (Stoppwörter im ersten Fünftel der Datei) unter die 100 Attribute mit dem höchsten Gewicht geschafft, aber auch hier ist die Aussage dieses Attributs zweifelhaft, da der Teil vollständig von der URL eingenommen wird.

Onlineshops Unter den URL Attributen der Shopseiten hat es nur ein Attribut unter die 100 Attribute mit dem höchsten Gewicht geschafft. Die Anzahl der Punkte in der URL wird von der SVM als ein Indiz gegen eine Shopseite gewertet.

Die interessanten 4-Gramme aus den URLs sind hier das 4-Gramm *shop* (das mit einem Verhältnis von 76 Vorkommen in den URLs der Shopseiten zu einem Vorkommen in den Nicht-Shopseiten als deutlicher Hinweis auf einen Shop von der SVM nach dem Wort *Shop* aus den Wortvektoren das zweit höchste Gewicht aller Attribute bekommt) sowie die 4-Gramme aus *Preisvergleich*, wobei die zwei 4-Gramme aus *Preis* ein höheres Gewicht als die restlichen 4-Gramme erhalten. Dies liegt daran, dass *Preis* zusätzlich als Teil der auch in den URLs vorkommenden Begriffe *Preisroboter* oder *Preissuchmaschine* vorkommt. Hoch bewertete Attribute, die gegen eine Shopseite sprechen, sind die zwei 4-Gramme aus *Forum*.

Hoch gewichtete Stoppwortattribute sind unter anderem die Anzahl der Stoppwörter insgesamt und im Verhältnis zur Länge der Datei, was bei den in der Regel

fast gleich großen Dateien vermutlich keinen Unterschied macht. Viele Stoppwörter sind ein Indiz gegen eine Shopseite.

Wissenschaftliche Webseiten Zur Erkennung wissenschaftlicher Webseiten gibt es mehrere interessante URL Attribute. Die Anzahl der Punkte und der Schrägstriche werden von der SVM als ein gutes Indiz für eine wissenschaftliche Webseite gesehen und erhalten ein relativ hohes Gewicht. Viele Gleichheits- und Fragezeichen in der URL gelten dagegen ebenso als Hinweis gegen eine wissenschaftliche Webseite, wie die Top-Level-Domain *.com* oder die Verwendung von PHP.

Das 3-Gramm *uni* wird von der SVM mit dem höchsten Gewicht der für die Erkennung wissenschaftlicher Webseiten verwendeten Attribute (einschließlich der Wortvektoren) versehen. Das 3-Gramm *php* gilt für die SVM ebenso wie die 3-Gramme *suc*, *uch*, *chm* und *hma* aus *Suche* und *Suchmaschine* als deutliches Indiz gegen eine wissenschaftliche Webseite. Die restlichen 23 der 29 URL n-Gramme aus den 100 Attributen mit dem höchsten Gewicht nehmen hauptsächlich den hinteren Teil der 100 Attribute ein. Hierbei handelt es sich in erster Linie um Teile aus Städtenamen, die Teile der Internetadresse einer Universität sind (z. B. *urg* aus *Hamburg*, *Augsburg* oder *Magdeburg*) sowie um Teile aus Fachrichtungen (z. B. *Physik* oder *Informatik*).

Wie bereits bei den Shopseiten gewichtet die SVM unter anderem die Anzahl der Stoppwörter insgesamt und die Anzahl der Stoppwörter im Verhältnis zur Länge der Datei hoch, wobei sie hier Indiz für eine wissenschaftliche Webseite sind.

Unerwünschte Webseiten Bei den unerwünschten Webseiten gewichtet die SVM wieder lediglich ein URL-Attribut so hoch, dass es zu den 100 Attributen mit den höchsten Gewichten gezählt wird. Demnach gelten viele Ziffern in der URL als Indiz gegen eine unerwünschte Webseite.

Die 3-Gramme, die von der SVM hoch gewichtet werden, kommen in der Regel entweder nur in der positiven oder nur in der negativen Klasse vor, jeweils aber nicht besonders häufig, so dass sie nur bei wenigen Webseiten als zusätzliches Attribut zu einer Verbesserung führen. Das höchste Gewicht unter den 3-Grammen erhält *for* aus *Forum* als Indiz gegen eine unerwünschte Webseite. Die 3-Gramme *pag* und *age* aus *Page* als Indiz gegen eine unerwünschte Webseite erhalten ebenfalls ein hohes Gewicht, wobei *age* als Teil von *Gebrauchtwagen* auch in den URLs der unerwünschten Webseiten vorkommt und ein geringeres Gewicht erhält. Weitere 3-Gramme sind beispielsweise *omi* und *mio* als Teil von *www.promio.net*. Bei nur zwei Fundstellen in den nicht unerwünschten Webseiten gegenüber keiner Fundstelle in den unerwünschten Webseiten führt dieses Attribut allerdings bei den meisten Webseiten zu keiner Verbesserung der Klassifikation.

Viele Stoppwörter gelten als Indiz gegen eine unerwünschte Webseite.

Attributgewichte auf den HTML-Dateien

HTML-Dateien	Foren	Nachr.	Shops	Wissensch.	Unerw.
Wortattribute (aus WVTool)	62	62	75	53	57
Synonymgruppen	6	6	8	5	12
URL Attribute	4	3	0	4	0
URL n-Gramme	15	23	12	29	16
HTML	8	4	3	1	7
Satzzeichen	3	1	2	4	7
Stoppwörter	2	1	0	4	1

Tabelle 5.14.: Verteilung der Attribute für HTML-Dateien

Die entsprechende Verteilung der 100 höchsten Attributgewichte für die HTML-Dateien ist in Tabelle 5.14 dargestellt. Die hoch bewerteten URL Attribute und URL n-Gramme sind dieselben wie für die Google-Schnipsel, weshalb sie an dieser Stelle nicht nocheinmal untersucht werden.

Forenseiten Die von der SVM hoch bewerteten HTML-Attribute sind die Anzahl der Links sowie die Anzahl der internen Links (also Links auf andere Webseiten desselben Forums) insgesamt und im ersten und vierten Fünftel. Hoch gewichtet wurde auch die Anzahl der Bilder auf der Webseite insgesamt sowie die Anzahl der Bilder in den letzten beiden Fünfteln. Ob ein Feld zur Passworteingabe auf der Webseite enthalten ist, dient der SVM ebenfalls als relevantes Attribut für eine Forenseite.

Relevante Satzzeichenattribute sind dem Gewicht nach die Anzahl der Fragezeichen absolut und im Verhältnis zur Textlänge. Mit im Schnitt 14,3 Fundstellen in den Forenseiten und 3,1 Fundstellen in den nicht Forenseiten ist der Einfluss der in den Anfang der HTML-Datei geschriebenen URL hier vernachlässigbar, da in den URLs der Forenseiten im Schnitt lediglich 0.45 Fragezeichen enthalten sind.

Nachrichtenseiten Die Anzahl der Bilder insgesamt sowie in den mittleren drei Fünfteln dienen der SVM dem Gewicht nach als gutes Indiz für eine Nachrichtenseite.

Gegen eine Nachrichtenseite spricht das Vorhandensein vieler Satzzeichen im oberen Teil der Webseite.

Onlineshops Viele Satzzeichen im Verhältnis zur Länge der Webseite sowie viele kurz aufeinander folgende Satzzeichen (wie sie vor allem in Aufzählungen vorkommen) sind dem Gewicht nach ein gutes Attribut für eine Shopseite. Ein enthaltenes Passwortfeld dient der SVM ebenfalls als Indiz für eine Shopseite. Für einen Shop

sprechen viele interne Links, im Gegensatz dazu sind viele externe Links ein Indiz gegen einen Shop.

Wissenschaftliche Webseiten Das einzige hoch bewertete HTML-Attribut ist hier die Anzahl der Links im Verhältnis zur Textlänge. Dabei sprechen viele Links gegen eine wissenschaftliche Webseite.

Ebenfalls gegen eine wissenschaftliche Webseite sprechen viele kurz aufeinander folgende Satzzeichen, wogegen viele etwas weiter auseinander liegende Satzzeichen ein für wissenschaftliche Webseiten sprechendes Attribut sind.

Viele kurz aufeinander folgende Stoppwörter dienen als Hinweis auf eine wissenschaftliche Webseite, wobei viele Stoppwörter im ersten Fünftel der Webseite wiederum gegen eine wissenschaftliche Webseite sprechen.

Unerwünschte Webseiten Die SVM bewertet ein vorhandenes Passwortfeld, viele Bilder im Verhältnis zur Länge der Webseite sowie viele interne Links als Attribute gegen eine unerwünschte Webseite, viele externe Links sprechen wiederum für eine unerwünschte Webseite.

Viele kurz aufeinander folgende Satzzeichen, vor allem viele Punkte, werden hier mit einem hohen Gewicht als Attribut für eine unerwünschte Webseite belegt.

5.4.3. Untersuchung der Verschlechterung durch zusätzliche Attribute

Im Folgenden wird diskutiert, warum die zusätzlichen Attribute an einigen Stellen zu einer Verschlechterung der Klassifikation führen und warum eine Verschlechterung durch zusätzliche Attribute überhaupt möglich ist.

Overfitting

Eine mögliche Erklärung für die stellenweise Verschlechterung durch zusätzliche Attribute könnte darin liegen, dass die SVM ihr gelerntes Modell zu genau an die Trainingsdaten anpasst. Auf den gelernten Daten wird dann sehr gut klassifiziert, die Klassifikation neuer Daten gelingt hingegen nur schlecht. Dieses Problem wird im maschinellen Lernen mit Überanpassung (engl. overfitting) bezeichnet. Man kann es in gewisser Weise auch mit auswendig lernen vergleichen, ohne das gelernte Wissen dann auf neue Aufgaben übertragen zu können.

Ein im Allgemeinen guter Hinweis auf Overfitting ist ein geringer Trainingsfehler (Fehler bei der Klassifikation der Daten, auf denen gelernt wurde) aber eine schlechte Accuracy in der Kreuzvalidierung. Zur Bestimmung des Trainingsfehlers wird ein Modell auf allen Daten eines Datensatzes gelernt, das dann wieder zur Klassifikation derselben Daten verwendet wird. Klassifiziert die SVM hier nur

wenige oder keine Beispiele falsch, so hat sie ein Modell gelernt, welches die Trainingsdaten gut erklärt. Bei der Kreuzvalidierung wird dagegen nur auf einem Teil der Daten gelernt und das errechnete Modell dann auf den Rest der Daten angewendet wird. Macht die SVM dort dann viele Fehler, ist das gelernte Modell zu genau auf die Daten zugeschnitten auf denen gelernt wurde, lässt sich jedoch nicht auf die restlichen Daten übertragen.

Da die Verschlechterung durch zusätzliche Attribute am deutlichsten bei den Google-Schnipseln der Nachrichtenseiten durch Hinzunahme der Stoppwortattribute ist (vgl. Tabelle 5.3), wird dieser Fall exemplarisch untersucht.

Ein Training nur auf den Wortvektoren der Google-Schnipsel führt mit dem auch bei der Kreuzvalidierung verwendeten Kostenparameter $C = 1000$ dazu, dass die SVM mit dem auf diesen Daten gelernten Modell alle 400 Webseiten richtig klassifiziert. Ein Training mit anschließender Anwendung für die um die Stoppwortattribute erweiterten Wortvektoren führt ebenfalls dazu, dass die SVM alle 400 Beispiele richtig klassifiziert. Hier lässt sich also keine Aussage darüber treffen, ob die SVM im Fall der zusätzlichen Attribute aufgrund von Overfitting ein schlechteres Ergebnis liefert. Dasselbe Verhalten zeigt sich hier für $C = 100$. Senkt man den C-Wert weiter auf 10 herab, klassifiziert die SVM ohne die Stoppwortattribute fünf der 400 Dokumente falsch, mit den Stoppwortattributen werden 13 Webseiten falsch klassifiziert. Auch dies liefert keinen Hinweis auf Overfitting, da die Kreuzvalidierung mit $C = 10$ mit den Stoppwortattributen ebenfalls ein schlechteres Ergebnis als ohne liefert, hier aber keinen geringeren Trainingsfehler macht.

Dieses Verhalten zeigt sich auch in den anderen Fällen, in denen zusätzliche Attribute zu einer Verschlechterung der Klassifikation führen. Die Verschlechterung lässt sich hier also nicht auf Overfitting zurückführen.

Veränderung des Merkmalsraumes

Das Hinzufügen weiterer Attribute führt zu längeren Merkmalsvektoren (die Wortvektoren werden um zusätzliche Attribute ergänzt) und dadurch auch zu einem Merkmalsraum mit höherer Dimension. Folglich wird hier auch eine andere Hyperebene gefunden, die sich für die veränderte Aufgabe schlechter eignen kann, als die gefundene Hyperebene der ursprünglichen Klassifikationsaufgabe. Dadurch kann es in diesem Fall zu den geringen Verschlechterungen der Accuracy an einigen Stellen kommen.

5.5. Klassifikation der Google-Schnipsel durch ein mit HTML-Dateien gelerntes Modell

Die Klassifikation der HTML-Dateien liefert in den durchgeführten Versuchen bessere Ergebnisse als eine Klassifikation der Google-Schnipsel. Es ist durchaus denkbar, dass sich das anhand der HTML-Dateien gelernte Modell auch zur Klassifikation der Google-Schnipsel eignet und zu besseren Ergebnissen führt, als ein auf den Google-Schnipseln trainiertes Modell.

Um dies zu überprüfen, wird für jede Klasse eine SVM mit den HTML-Dateien trainiert und zur Klassifizierung der Google-Schnipsel verwendet. Damit sich das gelernte Modell der HTML-Dateien auf die Google-Schnipsel anwenden lässt, müssen in beiden Fällen die gleichen Wortvektoren verwendet werden. Sie werden daher beim Training mit den HTML-Dateien für jede Klasse exportiert und zur Erzeugung der Wortvektoren für die Klassifizierung der Google-Schnipsel verwendet. Das in Yale zur Erzeugung der Wortvektoren benutzte WordVectorTool PlugIn (vgl. Kapitel 5.2.1) bietet hierfür die benötigten Schnittstellen.

Tabelle 5.15 stellt für jede Klasse die Accuracy eines mit den HTML-Dateien trainierten Modells der Accuracy eines mit Google-Schnipseln trainierten Modells gegenüber. In beiden Fällen wird das Modell zur Klassifikation der Google-Schnipsel verwendet.

Klasse	Training mit HTML-Dateien	Training mit Google-Schnipseln
Foren	64,25	90,17
Nachrichten	65,25	84,42
Shops	86,00	82,33
Wissenschaftliche	76,75	74,83
Unerwünschte	80,77	69,37

Tabelle 5.15.: Training mit HTML-Dateien, Klassifikation der Google-Schnipsel

Für Foren- und Nachrichtenseiten ergibt sich eine höhere Accuracy, wenn zur Klassifikation der Google-Schnipsel das mit diesen Schnipseln trainierte Modell verwendet wird. Bei den Shops, wissenschaftlichen und unerwünschten Webseiten lassen sich die Google-Schnipsel mit dem anhand der HTML-Dateien trainierten Modell besser klassifizieren als mit dem Modell, das auf den Google-Schnipseln trainiert wurde.

Begründen lassen sich die Ergebnisse der Foren- und Nachrichtenseiten dadurch, dass viele Wörter, die in den HTML-Dateien vorhanden sind und die Lage der gelernten Hyperebene maßgeblich bestimmen, in den Google-Schnipseln nur selten oder gar nicht vorkommen. Ein gutes Beispiel hierfür liefert das Wort *registrieren*,

das in 127 der 200 HTML-Dateien von Forenseiten und in nur sieben der 200 Webseiten, die keine Foren sind, vorkommt. Die SVM gewichtet das Wort mit einem entsprechend hohen Gewicht für eine Forenseite. Das Wort kommt aber in keinem der 200 Google-Schnipsel zu Forenseiten vor. Dadurch lässt sich das gelernte Modell nur sehr schlecht zur Klassifikation der Google-Schnipsel verwenden.

Die Verbesserung der Klassifikation der Google-Schnipsel durch das mit HTML-Dateien trainierte Modell für Shops, wissenschaftliche und unerwünschte Webseiten lässt sich dadurch erklären, dass die SVM die Relevanz einzelner Wörter mit Hilfe der umfangreicheren HTML-Dateien genauer bestimmen kann, als mit den kurzen Google-Schnipseln. So kommt der Begriff *Mwst* in den Google-Schnipseln der Shopseiten in lediglich zwei der 200 Treffer vor, die SVM gewichtet den Begriff daher sehr niedrig. In den HTML-Dateien kommt der Begriff in 67 Shopseiten und in nur drei der Webseiten vor, die keine Shops sind. Das Attribut erhält hier ein entsprechend hohes Gewicht.

5.6. Fazit zur Klassifikation nach Benutzerpräferenzen

Die Versuche zur Klassifikation nach Benutzerpräferenzen haben gezeigt, dass sich auf den Google-Schnipseln und HTML-Dateien durch die in Kapitel 4 beschriebenen zusätzlichen Attribute für fast alle untersuchten Klassen eine Verbesserung der Klassifikation erreichen lässt. Vor allem bei den kurzen Google-Schnipseln können erhebliche Verbesserungen erzielt werden, was sich am deutlichsten bei den wissenschaftlichen und den unerwünschten Webseiten zeigt.

Nicht alle Attribute lieferten die gewünschten Erfolge. Die URL Attribute führten bei der Klassifikation von Nachrichten-, wissenschaftlichen und unerwünschten Webseiten zu teilweise deutlichen Verbesserungen, vor allem bei einer Klassifikation der Google-Schnipsel. Zur Filterung von Foren und Onlineshops waren sie dagegen wenig hilfreich. Die URL n-Gramme führten, mit Ausnahme der HTML-Dateien der unerwünschten Webseiten, bei allen Klassen zu einer besseren Klassifikation.

Durch die Verwendung von Synonymgruppen konnte lediglich die Klassifikation von Foren und wissenschaftlichen Webseiten verbessert werden, für die anderen Klassen waren sie wenig erfolgreich.

Die vor allem zur Filterung unerwünschter Webseiten eingeführten Stoppwort- und Satzzeichenattribute trugen zu einer besseren Erkennung solcher Webseiten anhand der Google-Schnipsel bei, für die HTML-Dateien konnte lediglich mit den Satzzeichenattributen eine Verbesserung erzielt werden. Entgegen der ursprünglichen Vermutung waren diese Attribute zur Erkennung von Shops wenig hilfreich, durch die Satzzeichenattribute konnte aber eine bessere Klassifikation von wissenschaftlichen Webseiten erreicht werden.

In den durchgeführten Versuchen stellte sich heraus, dass die Klassifikation von Webseite allein über die von Google gelieferten Informationen mit den hier zusätzlich erzeugten Attributen bei Foren- und Nachrichtenseiten nur geringfügig schlechter ist, als eine Klassifikation über die vollständigen HTML-Dateien, die allein aufgrund der Übertragungszeit für die Dateien wesentlich zeitintensiver ist.

6. Ähnlichkeit von Texten

Wie in Abschnitt 2.2 bereits vorgestellt wurde, ist ein weiteres, bei der Suche über eine Internet-Suchmaschine auftretendes Problem, dass sich in einer Trefferliste Webseiten befinden, die sich inhaltlich kaum oder überhaupt nicht unterscheiden.

Im Folgenden wird eine Methode erläutert, mit der sich die Ähnlichkeit von Texten berechnen lässt. Außerdem wird erläutert, auf welchen Daten eine Erkennung ähnlicher Suchtreffer sinnvoll erscheint. In Kapitel 7 wird untersucht, ob sich mit diesem Verfahren tatsächlich inhaltlich ähnliche Webseiten erkennen und ausfiltern lassen.

6.1. Ähnlichkeitsmaß: Cosinus des Winkels zwischen den Wortvektoren

Ein weit verbreitetes und einfaches Maß zur Bestimmung der Ähnlichkeit von Dokumenten ist das Cosinusmaß. Dabei wird der Cosinus des Winkels α (siehe Abbildung 6.1) zwischen den Wortvektoren zweier Dokumente bestimmt.

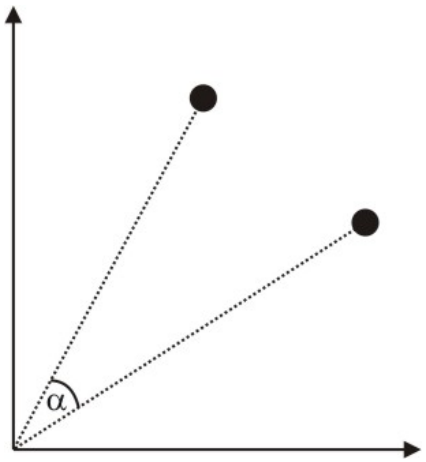


Abbildung 6.1.: Winkel zwischen zwei Wortvektoren

Bei einem Cosinus von genau 1 ist der Winkel zwischen den Wortvektoren 0, die Dokumente gelten als gleich. Ein kleiner Wert für den Cosinus entspricht einem

großen Winkel zwischen den beiden Wortvektoren, die Dokumente haben dann nur eine geringe Ähnlichkeit.

Der Cosinus c des Winkels zwischen den Wortvektoren $\vec{d_i}$ und $\vec{d_j}$ zweier Dokumente i und j errechnet sich über

$$c = \frac{\vec{d_i} \cdot \vec{d_j}}{||d_i|| \cdot ||d_j||}$$

Wie schon bei der Textklassifikation wird auch bei der Ähnlichkeitsberechnung durch die Textrepräsentation über Wortvektoren die Reihenfolge der Wörter im jeweiligen Dokument außer Acht gelassen.

Ebenfalls nicht berücksichtigt wird der Fall, dass der Winkel zwischen den Wortvektoren der Dokumente i und j auch dann genau 0 und der Cosinus somit 1 ist, wenn jedes Wort in Dokument i mit jeweils gleichem Faktor mehrfach so oft vorkommt wie in Dokument j. Die Wortvektoren verlaufen dann parallel. Dieser Fall tritt in der Praxis kaum auf und wird daher nicht weiter berücksichtigt.

6.2. Ähnlichkeit von Suchtreffern

Zur Berechnung der Ähnlichkeit von Suchtreffern stellt sich die Frage, auf welchen Daten eine Berechnung sinnvoll ist. Reichen die Informationen der Google-Schnipsel aus oder führt erst ein Vergleich der jeweils verlinkten Webseite zu einer zuverlässigen Erkennung ähnlicher Suchtreffer.

6.2.1. Google-Schnipsel

Ausgangssituation für die Filterung gleicher oder inhaltlich ähnlicher Suchtreffer anhand der Google-Schnipsel ist die Suche nach bestimmten Begriffen. Dabei liefert Google neben der URL zu jeder Webseite den Titel und einen Textausschnitt, der die Suchbegriffe enthält. Es ist anzunehmen, dass Google für zwei inhaltlich nahezu gleiche Webseiten, die beide in der Trefferliste zur selben Suchanfrage enthalten sind, auch den entsprechend gleichen Ausschnitt liefert.

Von zwei gleichen Ausschnitten lässt sich aber nicht automatisch auf den gleichen Inhalt der entsprechenden Webseiten schließen. Ein gutes Beispiel sind hier zwei Webseiten, die beide denselben Text zitieren und ihn dann unterschiedlich kommentieren. Kommen die Suchbegriffe nur in dem zitierten Text vor, so erhält man für beide Webseiten sehr wahrscheinlich den gleichen Ausschnitt in der Google-Auflistung. Der jeweilige Kommentar zu dem zitierten Text und damit der Rest der entsprechenden Webseite kann allerdings gänzlich verschieden sein.

Betrachtet man die URLs von zwei Webseiten zu verschiedenen Beiträgen in einem Forum oder einer Nachrichtenseite, so unterscheiden sie sich oft nur in wenigen Ziffern, die die Nummer des Beitrags angeben. Eine ähnliche URL lässt also

nicht auf zwei inhaltlich ähnliche Webseiten schließen. Für die Versuche in Kapitel 7 wird für eine Ähnlichkeitsberechnung von zwei Suchtreffern anhand der von Google gelieferten Informationen die URL daher außer Acht gelassen und lediglich der Titel der Webseite und der gelieferte Ausschnitt betrachtet.

6.2.2. HTML-Dateien

Bei den Spiegelungen des Open Directory Project aus Abbildung 2.7 auf Seite 10 wird von den Webseitenbetreibern meist ein unterschiedliches Layout zur Darstellung der Webseite verwendet. Häufig werden die Webseiten noch um Werbeeinblendungen oder eigenen Text ergänzt. Die HTML-Dateien der jeweiligen Webseiten enthalten dann zur Darstellung des unterschiedlichen Layouts auch unterschiedlichen HTML-Code. Es reicht also nicht aus, als Kriterium für zwei als inhaltlich gleich geltende Webseiten nahezu ähnliche HTML-Dateien, also einen Cosinus von fast 1 zu fordern.

Abbildung 6.2.: Spiegel Beitrag 1

Andererseits gibt es Webseiten, die ein festes Layout haben und jeweils nur einen anderen Artikel zeigen. Die beiden Beiträge aus Abbildung 6.2 und 6.3 behandeln unterschiedliche Themen, der Kopf der Webseite, die gesamte rechte Hälfte sowie die nicht abgebildeten Hinweise und Links am Ende der Webseite sind aber identisch.

Die HTML-Datei zu Abbildung 6.2 besteht aus 6545 Wörtern, der im Browser dargestellte Text enthält lediglich 1155 Wörter. Die restlichen 5390 Wörter stammen aus HTML-Code und werden zu einem großen Teil auch in der HTML-Datei der Webseite aus Abbildung 6.3 enthalten sein. Es ergibt sich daher automatisch eine starke Ähnlichkeit der HTML-Dateien, die zu einem großen Teil aus gleichem HTML-Code bestehen.

Abbildung 6.3.: Spiegel Beitrag 2

Es stellt sich die Frage, ob es ausreicht, nur die kompletten HTML-Dateien zu vergleichen. Wahrscheinlich ist es notwendig, zuvor mit einem geeigneten Parser den Textinhalt der Webseiten zu exrahieren, um so einen Vergleich nur über den Text durchführen zu können, welcher bei einer Betrachtung der Webseite auch im Browser angezeigt wird.

Zur Extraktion der Textinhalte der HTML-Dateien wird für die folgenden Versuche der unter *htmlparser.sourceforge.net* zum Download angebotene HTML Parser in der derzeit aktuellen Version 1.6 verwendet. Er ist in der Lage, HTML-Dateien vollständig von HTML-Code zu befreien und den reinen Textinhalt der entsprechenden Webseite auszugeben.

7. Versuche zur Erkennung ähnlicher Webseiten

Zunächst wird untersucht, auf welchen Daten eine Filterung ähnlicher Suchtreffer überhaupt Sinn macht. Lassen sich ähnliche Webseiten allein anhand der Informationen der Google-Schnipsel erkennen, oder müssen die jeweiligen Webseiten heruntergeladen und verglichen werden? Reicht es im Fall der heruntergeladenen Webseiten aus, einfach die gesamten HTML-Dateien zu vergleichen, oder muss zu jeder Webseite der im Browser angezeigte Text extrahiert werden, um nicht auch den in der Datei enthaltenen HTML-Code in den Vergleich mit einzubeziehen?

Anschließend stellt sich die Frage nach einem geeigneten Schwellwert, ab dem zwei Webseiten als gleich gelten und eine davon ausgefiltert werden kann. Dafür ist zu ermitteln, was für Cosinuswerte sich für inhaltlich ähnliche Webseiten ergeben und wie groß der Abstand zu Cosinuswerten inhaltlich verschiedener Webseiten ist.

7.1. Vergleich zwischen Google-Schnipsel, HTML-Datei und extrahiertem Texten

Um zu überprüfen, auf welchen Daten eine Ähnlichkeitsberechnung überhaupt Sinn macht, werden zu inhaltlich ähnlichen und inhaltlich verschiedenen Webseiten jeweils die Cosinuswerte der Google-Schnipsel, der kompletten HTML-Dateien und der aus den HTML-Dateien extrahierten Texte verglichen.

Als ähnliche Treffer werden hier jeweils fünf inhaltlich gleiche Webseiten der in Abbildung 2.6 auf Seite 9 dargestellten Treffer zur Suche nach „Datenmodellierung Transformation“ (im Folgenden als XML-Kurse bezeichnet) und fünf inhaltlich gleiche Spiegelungen des Open Directory Project der in Abbildung 2.7 auf Seite 10 dargestellten Suche nach „Hochzeit Trinkspiele Biertest“ verwendet.

Als inhaltlich verschiedene Webseiten werden fünf Meldungen des Heise Newstickers (*www.heise.de/newsticker*) und fünf Wikipedia Artikel (*de.wikipedia.org*) untersucht. Die Webseiten präsentieren sich jeweils im gleichen Layout, die dargestellten Nachrichten und Artikel sind aber inhaltlich verschieden.

Für die vier eben genannten Beispielklassen wird zu jeder Klasse der Mittelwert aller paarweise errechneten Cosinuswerte der fünf Treffer ermittelt, je einmal für die Google-Schnipsel, die komplette HTML-Datei und für den durch den HTML

Parser gefilterten Text der Webseite. Tabelle 7.1 stellt die jeweiligen Mittelwerte gegenüber.

Webseiten	Google-Schnipsel	HTML-Datei	Gefilterter Text
XML-Kurse	0,9684	0,8441	0,9483
Open Directory Project	0,8978	0,6132	0,9929
Heise Newsticker	0,4355	0,9843	0,7721
Wikipedia	0,5274	0,9159	0,6052

Tabelle 7.1.: Vergleich der Cosinuswerte

Die Cosinuswerte der Google-Schnipsel sind für die inhaltlich gleichen Treffer der XML-Kurse (0,9684) und die Spiegelungen des Open Directory Project (0,8978) im Schnitt deutlich höher als die Cosinuswerte der inhaltlich unterschiedlichen Treffer zu Meldungen des Heise Newstickers (0,4355) und der Wikipedia Artikel (0,5274).

Die inhaltlich verschiedenen, aber im Layout gleichen Meldungen des Heise Newstickers und die verschiedenen Wikipedia Artikel scheinen bei einer Berechnung des Cosinus auf den kompletten HTML-Dateien mit einem Schnitt von 0,9843 (Heise Newsticker) und 0,9159 (Wikipedia) sehr ähnlich zu sein. Dagegen sind die inhaltlich nahezu identischen, aber im Layout verschiedenen Webseiten der XML-Kurse und der Spiegelungen des Open Directory Project mit Cosinuswerten von im Schnitt 0,8441 bzw. 0,6132 deutlich unterschiedlicher.

Ein ähnliches Bild wie bei den Google-Schnipseln zeigt sich bei einem Vergleich der aus den HTML-Dateien gefilterten Texte. Für die inhaltlich gleichen XML-Kurse beträgt der Cosinus im Schnitt 0,9483, für die Webseiten des Open Directory Project liegt er bei 0,9929. Die Cosinuswerte der inhaltlich unterschiedlichen Meldungen des Heise Newstickers und der verschiedenen Wikipedia Artikel sind mit einem Schnitt von 0,7721 (Heise Newsticker) bzw. 0,6052 (Wikipedia) deutlich geringer.

Das Erkennen ähnlicher Webseiten ist über einen Vergleich der ungefilterten HTML-Dateien also nicht möglich. Inhaltlich unterschiedliche Webseiten, die sich im selben Layout präsentieren, haben aufgrund des HTML-Codes wesentlich mehr Ähnlichkeit, als zwei inhaltlich nahezu gleiche Webseiten, die sich lediglich im Layout unterscheiden. Auf den Google-Schnipseln und den mit dem HTML Parser gefilterten HTML-Dateien gehen die Cosinuswerte für inhaltlich ähnliche und inhaltlich verschiedene Treffer hier in die richtige Richtung auseinander, für inhaltlich ähnliche Webseiten ergeben sich also hohe und für inhaltlich verschiedene Webseiten niedrige Cosinuswerte.

7.2. Versuche mit Google-Schnipseln und gefilterten HTML-Dateien

Um im Folgenden eine Filterung ähnlicher Webseiten über die Google-Schnipsel und die gefilterten HTML-Dateien weiter auf ihre Machbarkeit zu prüfen und nach einem geeigneten Schwellwert zu suchen, werden die in Kapitel 5 zur Klassifikation nach Benutzerpräferenzen gesammelten Daten auf ähnliche Treffer untersucht.

Da ein inhaltlicher Vergleich der vielen Webseiten manuell kaum möglich ist, wird für alle paarweisen Kombinationen unter den Treffern einer Klasse der Cosinus errechnet. Anschließend werden für Paare mit einem hohen Cosinuswert (mindestens 0,9) die entsprechenden Webseiten verglichen und überprüft, ob sie inhaltlich tatsächlich ähnlich sind. Auf diese Weise soll ermittelt werden, was für Werte sich für inhaltlich gleiche Treffer ergeben und wie groß der Unterschied zu den Cosinuswerten von inhaltlich verschiedenen Treffern ist.

Liefert eine Berechnung des Cosinus für zwei Treffer mit ähnlichem Seiteninhalt nur einen kleinen Wert, so bleiben auf diese Weise unter Umständen inhaltlich ähnliche Webseiten unerkannt. Zur Suche nach einem geeigneten Schwellwert, bei dem verschiedene Webseiten nicht fälschlich als ähnlich ausgefiltert werden, reicht dieses Vorgehen jedoch aus.

7.2.1. Vergleich der Google-Schnipsel

Bei einem Vergleich der Google-Schnipsel ergeben sich für viele Trefferpaare hohe Cosinuswerte, obwohl die entsprechenden Webseiten sich inhaltlich deutlich unterscheiden. Dies betrifft in der Regel verschiedene Webseiten desselben Forums, derselben Nachrichtenseite oder desselben Shops.

Für zwei Treffer der Suche nach „Auto Lachgas“ beträgt der Cosinus der Google-Schnipsel 0,9163, unter den Treffern zur Suche nach „Schreibaby“ finden sich zwei Treffer mit einem Cosinus von 0,9520 und für zwei Treffer zur Suche nach „Fritz Box“ liegt der Cosinus bei 0,9247. Bei allen Paaren handelt es sich jeweils um Webseiten desselben Forums, die inhaltlich unterschiedliche Beiträge darstellen.

Unter den Treffern zur Suche nach „Windows Fehlermeldung“ finden sich sogar zwei Forenseiten für die der Cosinus genau 1 beträgt, die inhaltlich aber zu einem sehr großen Teil verschieden sind. Der erste Beitrag der beiden Webseiten stammt vom selben Nutzer und ist auf beiden Webseiten identisch, die restlichen Beiträge anderer Nutzer auf den beiden Webseiten sind aber verschieden. Der Cosinus von genau 1 erklärt sich hier dadurch, dass Google bei der Suche nach „Windows Fehlermeldung“ zu beiden Treffern denselben Ausschnitt liefert, der aus dem ersten, auf beiden Webseiten identischen Beitrag stammt.

Es gibt aber auch Paare von Treffern mit einem hohen Cosinuswert, die sich inhaltlich kaum unterscheiden. Unter den Treffern zur Suche nach „Super Nanny“ finden sich zwei Treffer, für die der Cosinus der Google-Schnipsel 0,9538 beträgt.

Die beiden Webseiten stammen aus demselben Forum, bei einem der beiden Treffer handelt es sich um die Archiv- bzw. Druckversion des anderen Treffers. Die Webseiten unterscheiden sich also lediglich im Layout, die dargestellten Forenbeiträge sind identisch. Das Gleiche gilt für zwei Treffer der Suche nach „Thunderbird PGP“ mit einem Cosinus von 0,9744 und für zwei Treffer der Suche nach „Tretlager reparieren“ mit einem Cosinus von 0,9245. Auch hier ist der Inhalt jeweils gleich, einer der Treffer präsentiert sich lediglich in einem einfacheren Layout.

Die Cosinuswerte der inhaltlich gleichen Webseiten heben sich nicht signifikant von den zuvor aufgelisteten Beispielen unterschiedlicher Webseiten ab, selbst wenn man den Cosinus von 1 für die beiden Treffer zur Suche nach „Windows Fehlermeldung“ einmal ignoriert.

Bei den Treffern der anderen Klassen zeigt sich das gleiche Bild. Unter den Shoptreffern zur Suche nach „DVD Player DivX“ finden sich zwei Treffer mit einem Cosinus von 0,9459, die inhaltlich verschieden sind. Eine der beiden Webseiten bietet einen bestimmten DVD-Player zum Verkauf an, auf der anderen Webseite werden für verschiedene DVD-Player die jeweils günstigsten Onlineshops aufgelistet.

Ein Beispiel für inhaltlich gleiche Shopseiten mit einem hohen Cosinus bieten zwei Treffer zur Suche nach „Videorekorder“. Der Cosinus der Google-Schnipsel zu den inhaltlich bis auf den Titel nahezu gleichen Shops beträgt 0,9356. Er liegt damit aber etwas unter dem Cosinuswert der soeben erwähnten, inhaltlich verschiedenen Treffer zur Suche nach „DVD Player DivX“.

Diese Beispiele zeigen, dass es Paare von inhaltlich unterschiedlichen Treffern gibt, für die der Cosinus über dem Wert mancher Paare von inhaltlich gleichen Webseiten liegt. Auch unter den anderen Klassen gibt es Kombinationen von inhaltlich unterschiedlichen Webseiten, für die der Cosinus der Google-Schnipsel höher ist als der Cosinus von manchen inhaltlich gleichen Webseiten.

7.2.2. Vergleich der gefilterten HTML-Dateien

Für die jeweils zwei unterschiedlichen Treffer aus demselben Forum zur Suche nach „Auto Lachgas“, „Schreibaby“ und „Fritz Box“, für die der Cosinus der Google-Schnipsel teilweise deutlich über 0,9 lag, beträgt der Cosinus der gefilterten HTML-Dateien lediglich 0,7331, 0,6806 bzw. 0,7116. Die gefilterten HTML-Dateien lassen sich hier also klar als unterschiedlich identifizieren.

Der Cosinus der beiden Treffer zur Suche nach „Windows Fehlermeldung“, bei denen der erste Beitrag gleich und die restlichen Antworten unterschiedlich waren, liegt für die gefilterten HTML-Dateien mit 0,9627 immer noch zu hoch, um eindeutig auf zwei Webseiten mit unterschiedlichem Inhalt schließen zu können.

Auch dies ist wieder nicht das einzige Beispiel für zwei Webseiten mit einem hohen Cosinuswert, die sich im Inhalt unterscheiden. Unter den Treffern zur Suche nach „Windows Fehlermeldung“ existieren zwei weitere Webseiten, die sich inhaltlich zwar unterscheiden, deren Cosinuswert mit 0,9830 aber sehr hoch ist.

Die drei Paare von inhaltlich gleichen Forenseiten, von denen jeweils eine als Archivversion ein wesentlich schlichteres Layout besaß, lassen sich mit Cosinuswerten von 0,6620 (Treffer zur Suche nach „Super Nanny“), 0,8394 („Thunderbird PGP“) und 0,6733 („Tretlager reparieren“) nicht als inhaltlich gleich identifizieren, obwohl die jeweils dargestellten Beiträge sich nicht unterscheiden. Die Cosinuswerte der soeben erwähnten, inhaltlich unterschiedlichen Webseiten liegen deutlich höher.

Unter den gefilterten HTML-Dateien gibt es deutlich weniger Beispiele von inhaltlich verschiedenen Webseiten mit einem hohen Cosinus als unter den Google-Schnipseln. Das zuletzt genannte Beispiel der Forenseiten, die einmal in normalem Layout und einmal als optisch einfache Archivversion in der Trefferliste auftauchen, zeigt deutlich, dass der Einfluss der Rahmen- und Linktexte selbst auf den gefilterten HTML-Dateien noch zu hoch ist, um inhaltlich ähnliche Webseiten eindeutig erkennen zu können.

7.3. Versuche mit einem anderen Ähnlichkeitsmaß

Der Einsatz des Cosinusmaßes führt leider nicht zu einer sicheren Erkennung ähnlicher Webseiten bzw. der entsprechenden Suchtreffer. Vor allem inhaltlich verschiedene Webseiten, die nach einer Berechnung der Ähnlichkeit über den Cosinus sehr ähnlich scheinen, machen das Verfahren zur Erkennung ähnlicher Suchtreffer untauglich.

Zur Überprüfung, ob mit einem anderen Verfahren bessere Ergebnisse erzielt werden oder ob dies lediglich daran liegt, dass die Daten nicht die erforderlichen Ähnlichkeiten und Unterschiede aufweisen, werden die zuvor durchgeführten Versuche mit einem anderen Ähnlichkeitsmaß wiederholt. Als zweites Maß dient die Distanz zwischen den Wortvektoren (bzw. zwischen den entsprechenden Punkten im Raum). Zur Bestimmung der Distanz von zwei Wortvektoren wird die euklidische Distanz verwendet, der auf Seite 19 erläutert ist. Je geringer die Distanz zwischen zwei Wortvektoren, desto ähnlicher sind sich die jeweiligen Dokumente.

Tabelle 7.2 stellt die Mittelwerte der Klassen gegenüber, die bereits unter Verwendung des Cosinusmaßes zum Vergleich von inhaltlich ähnlichen Webseiten mit unterschiedlichem Layout und inhaltlich verschiedenen Webseiten mit gleichem Layout verwendet wurden. Es überrascht nicht, dass sich die Werte der ungefilterten HTML-Dateien aufgrund des enthaltenen HTML-Codes wieder nur zur Erkennung von Webseiten mit gleichem Layout eignen und eine Filterung inhaltlich ähnlicher Seiten auch hier lediglich über die Google-Schnipsel und die gefilterten HTML-Dateien sinnvoll ist.

Leider zeigt sich auch bei einer Ähnlichkeitsberechnung der zur Klassifikation nach Benutzerpräferenzen gesammelten Daten über die euklidische Distanz dasselbe Bild wie bei einer Ähnlichkeitsberechnung über das Cosinusmaß. Es finden sich viele Beispiele von inhaltlich verschiedenen Google-Schnipseln und gefilterten HTML-Dateien, zu denen die euklidische Distanz zwischen den Wortvektoren

Webseiten	Google-Schnipsel	HTML-Datei	Gefilterter Text
XML-Kurse	0,2810	1,0470	0,4544
Open Directory Project	0,5235	1,6577	0,1352
Heise Newsticker	1,7994	0,3485	1,7209
Wikipedia	1,3102	0,8012	1,8097

Tabelle 7.2.: Vergleich der Distanz zwischen den Wortvektoren

geringer ist als zwischen den Wortvektoren einiger Treffer, die sich inhaltlich unterscheiden.

7.4. Fazit zu den Ergebnissen

In den durchgeführten Versuchen wird klar, dass das Erkennen inhaltlich ähnlicher Webseiten weder über die Google-Schnipsel noch über die gefilterten HTML-Dateien möglich ist. Ein Vergleich der ungefilterten HTML-Dateien scheidet zur Bestimmung ähnlicher Webseiten aus, da die Ähnlichkeit dieser Dateien aufgrund des enthaltenen HTML-Codes für zwei inhaltlich verschiedene Webseiten mit gleichem Layout deutlich höher ist, als für zwei inhaltlich gleiche Webseiten, die sich in einem unterschiedlichen Layout präsentieren.

Unter den Google-Schnipseln finden sich etliche Treffer, für die der Cosinus trotz unterschiedlichem Inhalt der Webseite höher (bzw. die euklidische Distanz geringer) ist, als für inhaltlich gleiche Treffer. Es lässt sich also kein sinnvoller Schwellwert definieren, ab dem zwei Treffer als gleich gelten und ausgefiltert werden können, ohne dass Webseiten mit unterschiedlichem Inhalt fälschlich als stark ähnlich ausgefiltert werden würden.

Selbst auf den mit dem HTML Parser gefilterten HTML-Dateien gab es Paare von Webseiten mit einem Cosinus von fast 1 (bzw. einer euklidischen Distanz von fast 0), die sich inhaltlich dennoch unterschieden. Auch hier liegen die Cosinuswerte einiger Paare inhaltlich gleicher Webseiten deutlich niedriger (bzw. die euklidische Distanz höher).

8. Zusammenfassung und Ausblick

8.1. Zusammenfassung

Im Rahmen dieser Diplomarbeit wurden Methoden untersucht, mit denen sich die Trefferlisten von Internet-Suchmaschinen reduzieren lassen, damit dem Nutzer weniger Treffer präsentiert werden, die ihn (bei seiner aktuellen Anfrage) nicht interessieren und bei der Suche im Internet unnötig aufhalten.

Behandelt wurden dabei drei Aspekte: das Klassifizieren von Webseiten nach vorgegebenen Klassen, das Erkennen von Webseiten, die sich inhaltlich kaum oder gar nicht unterscheiden und das Ausfiltern generell unerwünschter Webseiten. Mit generell unerwünschten Webseiten waren dabei vor allem weitere Internet-Suchmaschinen und Webseiten gemeint, die einfach nur häufig gesuchte Begriffe aufzählen, um so für möglichst viele Suchbegriffe als Treffer mit ausgegeben zu werden.

Die Klassifikation von Webseiten nach vorgegebenen Kategorien konnte mit zusätzlichen text- und webseitenspezifischen Merkmalen gegenüber der reinen Textklassifikation für fast alle behandelten Klassen verbessert werden. Es wurde ebenfalls gezeigt, dass für manche Klassen eine Klassifikation über die in der Trefferliste der Internet-Suchmaschine enthaltenen Informationen, also ohne die entsprechende Webseite zu betrachten, bereits zu einer praktisch nutzbaren Klassifikation führt.

Die Erkennung generell unerwünschter Webseiten wurde ebenfalls mit Methoden der Textklassifikation behandelt und konnte auch durch zusätzliche Attribute verbessert werden.

Die Versuche zur Erkennung inhaltlich ähnlicher oder gleicher Webseiten haben gezeigt, dass eine Erkennung ohne weiteres weder über die in der Trefferliste der Internet-Suchmaschine enthaltenen Informationen noch über die entsprechenden Webseiten möglich ist. Zwei ähnliche Ausschnitte in der Trefferliste können aus einem kleinen Abschnitt der Webseiten stammen, die sonst aber völlig unterschiedlich sein können. Die Treffer schienen dann sehr ähnlich oder sogar exakt gleich zu sein, obwohl sich die entsprechenden Webseiten deutlich unterschieden. Desweiteren ergaben sich für die Treffer zu inhaltlich unterschiedlichen Webseiten oft ähnlich hohe oder teilweise höhere Ähnlichkeitswerte, als für die Treffer zu inhaltlich gleichen Webseiten, so dass eine Abgrenzung nicht möglich war.

Auf den vollständigen HTML-Dateien der kompletten Webseiten war eine Ähn-

lichkeitsberechnung nicht sinnvoll durchführbar, da der HTML-Code den Textinhalt der Webseiten an Umfang oft um ein Vielfaches überstieg. Inhaltlich verschiedene Webseiten, die exakt dasselbe Layout verwendeten, erschienen dadurch wesentlich ähnlicher als inhaltlich nahezu identische Webseiten, die sich im Layout unterschieden.

Ein Vergleich der extrahierten Inhalte der jeweiligen Webseiten führte ebenfalls zu keiner zuverlässigen Erkennung ähnlicher Webseiten. Auch hier gab es unter den untersuchten Daten zu viele Beispiele unterschiedlicher Webseiten, die einen höheren Ähnlichkeitswert als manche inhaltlich gleiche Webseiten hatten.

8.2. Umsetzung in eine Anwendung

In einer Anwendung scheint es zur Umsetzung der Benutzerpräferenzen sinnvoll, zunächst eine Klassifikation nur über die Google-Schnipsel vorzunehmen. Die Google-Schnipsel stehen schnell zur Verfügung, die Erzeugung der zusätzlichen Attribute geht sehr zügig, da es sich hierbei lediglich um sehr kurze Textteile und Internetadressen handelt.

Um Fehler beim Ausfiltern zu vermeiden, bietet es sich an, einen Schwellwert für die Konfidenz (im Falle einer SVM die Distanz zur trennenden Hyperebene) zu definieren, ab dem Treffer ausgefiltert werden.

Parallel zur Klassifikation über die Google-Schnipsel sollten im Hintergrund die entsprechenden HTML-Dateien nachgeladen und analysiert werden, was in Abhängigkeit der Internetanbindung und der Länge der Webseite etwas Zeit in Anspruch nehmen kann. Anhand dieser genaueren Klassifikation über die vollständige Webseite können dann eventuell weitere Treffer ausgefiltert oder fälschlich ausgefilterte Treffer wieder eingeblendet werden. Besucht der Nutzer die Webseite des ersten Treffers der anhand der Google-Schnipsel gefilterten Liste und stellt fest, dass ihn die entsprechende Webseite nicht interessiert, steht eventuell schon die etwas genauere, über die HTML-Dateien gefilterte Liste bereit.

Sofern es für den Nutzer bei der Suche keine Einschränkung darstellt, wenn zwei inhaltlich verschiedene Webseiten fälschlich als ähnlich klassifiziert werden, kann hier eine Filterung durchgeführt werden. Dabei sollten ähnliche Webseiten aber nicht einfach verworfen werden. Es erscheint sinnvoll, den ersten Eintrag, zu dem ähnliche Webseiten gefunden wurden, mit einer entsprechenden Funktion zu versehen, die es dem Nutzer ermöglicht, sich die inhaltlich ähnlichen Treffer anzeigen zu lassen. Üblich ist hierfür ein mit einem Plus-Zeichen versehener Button, der den Treffer expandiert und die weiteren Treffer dabei etwas eingerückt untereinander anzeigt. So hat der Nutzer auf Wunsch die Auswahl aus weiteren inhaltlich gleichen Alternativen, beispielsweise für den Fall, dass die Webseite zum ersten Treffer nicht erreichbar ist.

Google liefert pro Anfrage ein Paket mit zehn Treffern, die nächsten zehn Treffer erhält man dann auf Anforderung. In einer Endanwendung sollten nach den ersten

zehn schon weitere Treffer geladen und klassifiziert werden. Zum einen bleiben nach dem Filtern der ersten zehn Treffer unter Umständen nur noch wenige Treffer übrig, zum anderen können auf Anforderung des Nutzers dann ohne Wartezeit weitere, bereits klassifizierte Treffer präsentieren werden.

8.3. Weitere Untersuchungen und Verbesserungsmöglichkeiten

Abschließend folgen Anregungen für zusätzliche Versuche zur Klassifikation nach Benutzerpräferenzen und weitere Ideen, mit denen sich die Klassifikation eventuell weiter verbessern lässt.

8.3.1. Weitere Untersuchungen

Für die durchgeführten Versuche mussten Suchtreffer von Hand klassifiziert werden. Um den Zeitaufwand zu begrenzen, wurden zu jeder Klasse lediglich 200 positive und 200 negative Beispiele gesammelt. Interessant sind weitere Versuche mit mehr Beispielen, um zu prüfen, ob durch mehr Trainingsbeispiele eine bessere Klassifikation erreicht werden kann.

Desweiteren ist es interessant, eine Klassifikation nach weiteren Kategorien zu untersuchen, beispielsweise nach Webseiten mit Jobangeboten. Viele Firmen schreiben offene Stellen auf ihren Webseiten aus, so dass eine Suche nach einer bestimmten Tätigkeit eine Liste mit entsprechenden Webseiten liefern könnte.

Ebenfalls untersuchbar ist, mit welcher minimalen Anzahl von Trainingsbeispielen noch eine gewisse Zuverlässigkeit in der Klassifikation erreicht werden kann. Ein System, das nur zehn Beispiele benötigt, um eine Trefferliste in zwei Klassen zu teilen, ist für einen Anwender attraktiver als ein System, dass mehrere hundert Beispiele braucht, um Webseiten sicher klassifizieren zu können.

Für die durchgeführten Versuche wurden ausschließlich Webseiten in deutscher Sprache verwendet. Fraglich ist, ob die hier verwendeten Attribute die Klassifikation der Webseiten in anderen Sprachen ebenfalls verbessert. Vielleicht lässt sich für bestimmte Kategorien auch ein Modell lernen, das zur Klassifikation von Webseiten in verschiedenen Sprachen verwendet werden kann.

8.3.2. Weitere Ideen für zusätzliche Attribute

Die zusätzlich verwendeten Attribute führen bereits zu einer Verbesserung der Klassifikation der getesteten Kategorien gegenüber dem wortbasierten Ansatz. Weitere Ideen, mit denen sich die Klassifikation je nach Thema eventuell noch weiter verbessern lässt, könnten wie folgt aussehen:

- Ein gängiger Trick von Webseitenbetreibern ist das sogenannte *User Agent Cloaking*, bei dem Google zum Indizieren eine andere Webseite geliefert wird, als der normale Besucher der Webseite zu sehen bekommt. So kann Google vorgetäuscht werden, dass die Webseite bestimmte Begriffe enthält, ohne dem Nutzer eine sinnlose Liste von Begriffs-Auflistungen (vgl. Abbildung 2.12 auf Seite 14) zu präsentieren.

  Zur Erkennung unerwünschter Webseiten bietet sich daher eine Prüfung an, ob der von Google gelieferte Ausschnitt tatsächlich in der Webseite enthalten ist.

- Nachrichtenseiten enthalten in der Regel zu jeder Nachricht mindestens das entsprechende Datum, in vielen Fällen auch eine Uhrzeit. Auf Forenseiten wird meist jeder Beitrag mit Datum und Uhrzeit versehen, so dass hier meist mehrere Zeitangaben auf einer Webseite enthalten sind.

  Die Erkennung von Nachrichten- und Forenseiten könnte durch Attribute zur Anzahl der Uhrzeit- und Datumsangaben weiter verbessert werden.

- Die Einbeziehung des Layouts der Webseiten in die Klassifikation ist sicher aufwändig, es ist aber durchaus denkbar, dass sich dadurch weitere Verbesserungen erzielen lassen. In Foren werden die einzelnen Beiträge meist durch Rahmen oder Linien abgetrennt, Nachrichtenseiten enthalten oft eine Überschrift in etwas größerer Schrift. Weitere Merkmale könnten beispielsweise aus der verwendeten Schrift- und Hintergrundfarbe oder der Anordnung und Größe der Bilder generiert werden.

Literaturverzeichnis

[1] Bayes, Thomas: An Essay towards solving a Problem in the Doctrine of Chances. (1763)

[2] Blankenhorn, Kai: *Spam-Filterung mittels maschinellem Lernen*, Fachhochschule Furtwangen, Diplomarbeit, 2002

[3] Burges, Christopher J. C.: A Tutorial on Support Vector Machines for Pattern Recognition. In: *Data Mining and Knowledge Discovery* 2 (1998), Nr. 2, S. 121–167

[4] Cortes, Corinna ; Vapnik, Vladimir: Support-Vector Networks. In: *Machine Learning* 20 (1995), Nr. 3, S. 273–297

[5] Euler, Timm: *Informationsextraktion durch Zusammenfassung maschinell selektierter Textsegmente*, Universität Dortmund, Diplomarbeit, Oktober 2001

[6] Ferber, Reginald: *Information Retrieval.* Heidelberg : dpunkt.verlag, 2003

[7] Granka, Laura A. ; Joachims, Thorsten ; Gay, Geri: Eye-tracking analysis of user behavior in WWW search. In: *Proceedings of the Conference on Research and Development in Information Retrieval*, ACM Press, 2004

[8] Iwayama, Makoto ; Tokunaga, Takenobu: Cluster-Based Text Categorization: a Comparison of Category Search Strategies / Department of Computer Science, Tokyo Institute of Technology. 1995 (95-TR0016). – Forschungsbericht

[9] Joachims, Thorsten: Text Categorization with Support Vector Machines: Learning with Many Relevant Features / Universität Dortmund. 1997 (LS-8 Report 23). – LS-8 Report

[10] Joachims, Thorsten: Optimizing Search Engines Using Clickthrough Data. In: *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, ACM Press, 2002

[11] Lovins, Janet B.: Development of a Stemming Algorithm. In: *Mechanical Translation and Computational Linguistics* 11 (1968), Nr. 1-2, S. 22–31

[12] Mierswa, Ingo ; Wurst, Michael ; Klinkenberg, Ralf ; Scholz, Martin ; Euler, Timm: YALE: Rapid Prototyping for Complex Data Mining Tasks. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*, ACM Press, 2006

[13] Mitchell, Tom: *Machine Learning*. McGraw Hill, 1997

[14] Morik, Katharina: *Maschinelles Lernen*. 1999. – Vorlesungsskript

[15] Porter, M. F.: An Algorithm for Suffix Stripping. In: *Program* 14 (1980), Juli, Nr. 3, S. 130–137

[16] Rumelhart, David E. ; Hinton, Geoffrey E. ; Williams, Ronald J.: Learning internal representations by error propagation. (1986), S. 318–362

[17] Sahami, Mehran ; Dumais, Susan ; Heckerman, David ; Horvitz, Eric: A Bayesian Approach to Filtering Junk E-Mail. In: *Learning for Text Categorization: Papers from the 1998 Workshop*. Madison, Wisconsin : AAAI Technical Report WS-98-05, 1998

[18] Salton, Gerard: *Automatic Text Processing*. Addison-Wesley, 1989

[19] Salton, Gerard ; Buckley, Christopher: Term-weighting approaches in automatic text retrieval. In: *Information Processing and Management* 24 (1988), Nr. 5, S. 513–523

[20] *Einführung in Neuronale Netze*. http://wwwmath.uni-muenster.de/SoftComputing/lehre/material/wwwnnscript/startseite.html

[21] Vapnik, Vladimir N.: *The nature of statistical learning theory*. New York : Springer-Verlag New York, Inc., 1995

[22] Yang, Yiming: An Evaluation of Statistical Approaches to Text Categorization. In: *Information Retrieval* 1 (1999), Nr. 1/2, S. 69–90

Index

Erklärung

Hiermit erkläre ich, Norbert Basmaci, die vorliegende Diplomarbeit mit dem Titel *Filterung von Ergebnislisten von Suchmaschinen* selbständig verfasst und keine anderen als die hier angegebenen Hilfsmittel verwendet, sowie Zitate kenntlich gemacht zu haben.

Dortmund, 31. Oktober 2006