View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Qucosa - Publikationsserver der Universität Leipzig

Universität Leipzig

Fakultät für Mathematik und Informatik

Institut für Informatik

Separierung mit FindLinks gecrawlter Texte nach Sprachen

Bachelorarbeit

Leipzig, Februar 2011

vorgelegt von

Pollmächer, Johannes
geb. am 15.12.1984
Studiengang Bachelor of Science

Betreuender Hochschullehrer: Prof. Dr. Quasthoff, Uwe
Fakultät für Mathematik und Informatik
Institut für Informatik
Abteilung Automatische Sprachverarbeitung

Inhaltsverzeichnis

In dieser Arbeit wird ein Programm zur Sprachidentifikation von Web-Dokumenten vorgestellt. Das Verfahren nutzt Worthäufigkeitslisten als Trainingsdaten, um anhand dieser Dokumentenklassifikation in Sprachen vorzunehmen. Somit gehört dieses Werkzeug zu den *supervised-learning*-Systemen. Die zu klassifizierenden Web-Dokumente wurden mittels des von der Abteilung für Automatische Sprachverarbeitung entwickelten Tools „FindLinks" heruntergeladen. Das Programm ist somit in die Nachverarbeitung bestehender Rohdaten einzuordnen.

1. Einleitung

Diese Arbeit bearbeitet das Problem der automatisierten Sprachklassifikation von maschinenlesbaren Dokumenten aus dem Web. Sprachidentifikation ist ein gut erforschtes Gebiet. Methoden und Heurismen sind in einer Vielzahl verfügbar und evaluiert worden. Die Aufgaben der vorliegenden Arbeit bestehen somit in der geeigneten Wahl der Methoden, deren Implementierung und deren Evaluierung auf vorliegenden Referenzdaten.

1.1. Zielstellung

Ziel dieser Arbeit ist es, einen funktionsfähigen Sprachidentifizierer zu entwickeln. Neben der allgemeinen Funktion, Sprachen von Dokumenten zu ermitteln, sollte es möglich sein, das Programm als aktives Glied in den FindLinks-Prozess zur Nachverarbeitung der gecrawlten Dokumente einzubetten. Hierfür soll das Programm regelmäßig zur Sprachseparierung zum Einsatz kommen. Auf Basis von monolingualen Dokumetensammlungen können in einem späteren Prozessschritt einsprachige Sprachkorpora generiert werden. Die Erstellung und der Einsatz des Sprachidentifizierers stellt somit ein Zwischenziel auf dem Weg zur Korpusanalyse dar.

Als Trainingsdaten sind Worthäufigkeitslisten von ca. 350 Sprachen gegeben. Diese sollen möglichst gewinnbringend in den Klassifikationsprozess eingebracht werden. Hierzu nutze ich bestehende Verfahren und deren Evaluationen als auch die Struktur der zu klassifizierenden Web-Dokumente. Da bei einem Einzelnen der betrachteten Dokumente keine Einsprachigkeit vorausgesetzt werden kann, muss eine mehrsprachige Klassifikation pro Dokument, als auch eine Nullklassifikation, im Falle von unbrauchbaren Material, möglich sein.

1.2. Aufbau der Arbeit

Der Aufbau der Arbeit folgt den Phasen des Entwicklungsprozesses des Werkzeugs *LangSepa*.

Im Kapitel 2 werden zunächst die zentralen Begriffe und einige grundlegende wissenschaftliche Erkenntnisse aus dem Feld der Sprachidentifikation vorgestellt. Anschließend erfolgt ein Ein- und Überblick in die Einbettung des Sprachidentifizierers und seinem Umfeld. Danach werden die zentralen mathematischen Formalismen eingeführt, auf denen im Klassifikationsprozess aufgebaut wird. Der ISO 639, der die Kodierung von Sprachen in Sprachcodes in dieser Arbeit als auch der Implementierung regelt, schließt dieses Kapitel ab.

Nachdem im darauffolgenden Kapitel eine erste Begutachtung der FindLinks-Dokumente und Trainingssprachen erfolgt ist, wird im Kapitel 4 die Implementierung des Werkzeugs ausführlich beschrieben. Hierzu zählen die Architektur, verwendete Datenstrukturen und der Klassifikationsalgorithmus im Detail.

Einige Empfehlungen zur Konfiguration und erste Evaluationsergebnisse können in Kapitel 5 eingesehen werden. Zum Abschluss der Arbeit werden Ergebnisse einer Auswahl von FindLinks-Dokumenten präsentiert.

1.3. Eingrenzung der Verfahren

Zu Beginn der Arbeit standen mehrere besondere Merkmale zur Wahl, aus denen sich Sprachen ableiten lassen konnten. Zu diesen gehörten neben den bereits erwähnten Worthäufigkeitslisten, die *Top-Level-Domain*[1] als auch die Originalkodierung der Dokumente. Diese beiden Möglichkeiten wurden fallen gelassen, da es nicht selten vorkommt, dass falsche Kodierungen gesetzt werden und diese somit eine korrekte Klassifikation, aufgrund von Ausschluss der vermeindlich korrekten Sprache, unmöglich wird [7, vgl., S. 357]. Die *Top-Level-Domain* gibt lediglich Aufschluss über die Zugehörigkeit einer Seite, jedoch nicht über seinen sprachlichen Inhalt. So sind viele Seiten unter der `de`-Superdomäne veröffentlicht, aber ein gewisser Anteil dieser Seiten nicht in deutscher Sprache verfasst.

Die Worthäufigkeitslisten bilden somit die Basisdaten, aus denen die besonderen Merkmale zur Erstellung des Sprachprofils herangezogen werden. Da für ca. 300 der Sprachen keine stark trainierten Korpora[2] zur Verfügung stehen, werden Sprachmerkmale, die über einzelne Wörter hinausgehen[3] nicht betrachtet. Zwecks Vermeidung einer Überbewertung der Trainingsdaten, die zu einer potentiellen Überanpassung[4] des Sprachmodells führen würden [12, vgl., S. 271–272, 308–314], sind die gewählten besonderen Merkmale in dieser Arbeit Stopworte und Buchstaben-n-Gramme.

---

[1] Das ist die oberste Zugehörigkeitsebene einer Adresse im Internet. Für `http://www.uni-leipzig.de` ist die zugehörige Top-Level-Domain `de`.

[2] Damit sind Korpora gemeint, bei denen die Gütekriterien wie Balanciertheit und Repräsentativität nicht gewährleistet sind. Die hier verwendeten Korpora besitzen mit bis zu 2000 Wörtern pro Sprache nur einen äußerst geringen Umfang.

[3] Bspw. Wort-n-Gramme oder Kookkurrenzen zwischen Wörtern sind hiermit gemeint.

[4] Bzw. im Englischen: *overfitting*.

2. Grundlagen

In diesem Kapitel sollen zunächst die besonderen Merkmale für die Implementierung definiert und das Thema Sprachidentifikation bzw. Sprachseparierung theoretisch untermauert werden. Anschließend wird ein Überblick über die verschiedenen Teilprozesse und Komponenten, an die das zu entwickelnde Programm gekoppelt ist, gegeben. Hierauf folgt die theoretische Basis für das Programm. Zum Abschluß dieses Kapitels erfolgt eine Darstellung des verwendeten Sprachcode-Standards „ISO 639-3".

2.1. Betrachtete Zeichenstrukturen

Für die Identifikation von Sprache werden in dieser Arbeit ausschließlich Zeichenketten verwendet, die Bestandteil der zu untersuchenden Textdokumente sind. Dieser Abschnitt soll Klarheit darüber verschaffen, welche Art Zeichenketten benutzt werden und wie diese in der Implementierung von *LangSepa* definiert sind.

2.1.1. Wörter

Wörter sind die kleinsten Bedeutungsträger der meisten Sprachen. Um einzelne Wörter voneinander zu trennen, genügt es die endliche Menge aller Symbole einer Sprache in zwei Klassen zu unterteilen. Die eine Klasse besteht aus allen Symbolen, die Wörter voneinander trennen. Im Deutschen sind dies bspw. die Satzzeichen und das Leerzeichen. Ziffern werden ebenso als Trennsymbole gehandhabt. Alle übrigen Symbole sind Wortsymbole[5] und somit der anderen Klasse zuzuordnen [2, vgl. S. 145]. Ein Wort ist somit eine größtmögliche Zeichenfolge, die ausschließlich aus Wortsymbolen bzw. Buchstaben besteht[6]. In dieser Art und Weise werden Wörter in der vorliegenden Arbeit gehandhabt. Folgender Textausschnitt wird unter Verwendung dieses Ansatzes in folgende Wörter zerlegt:

`letzter Mann: „Ich verlasse das Deck nicht!"`

Wörter: `letzter,Mann,Ich,verlasse,das,Deck,nicht`

Worttrennersymbolen kommt somit eine wichtige Bedeutung zu. Leider besitzt nicht jede Sprache Trennsymbole für die Abgrenzung einzelner Wörter. Prominente Beispiele hierfür sind Chinesisch und Japanisch. Aus diesem Grund ist es sinnvoll Buchstaben-n-Gramme als zusätzliche Zeichenstrukturen mit in Betracht zu ziehen.

---

[5] Wortsymbole sind in den meisten Sprachen die Buchstaben. Für eine Überprüfung, ob ein Zeichen ein Wortsymbol ist, werden im Programm die utf-8-Symbolauszeichnungen herangezogen. Diese sind u.a. `Letter`, `Digit` und `Sign`.

[6] Vorausgesetzt die betrachtete Sprache besitzt Trennsymbole.

2.1.2. Buchstaben-n-Gramme

Für die Erkennung von Sprache haben sich viele n-Gramm Ansätze etabliert. Viele Autoren haben diese Technik untersucht und halten diese für äußerst erfolgreich [12, S. 46]. In Unterabschnitt 2.2.2 wird darauf näher eingegangen.

In dieser Arbeit werden ausschließlich n-Gramme auf Zeichenebene untersucht und implementiert. Ein Buchstaben-n-Gramm ist dabei eine Buchstabenfolge der Länge n [2, S. 447]. Ein Wort mit p Zeichen Länge kann somit in $\max\{(p-n)+1, 0\}$ Buchstaben-n-Gramme zerlegt werden. Da die Anfänge und Enden von Wörtern charakteristische Merkmale vieler Sprachen sind, kann diese Information in den n-Grammen mit $n > 1$, durch einen Unterstrich („_”)[7] vor dem ersten Buchstaben und nach dem letzten Buchstaben jedes Wortes eingefügt werden. Mit diesem Zusatz lassen sich für eben betrachtetes Wort $\max\{((p+2)-n)+1, 0\}$ Buchstaben-n-Gramme bilden. Alle Buchstaben-n-Gramme für $n < 6$ werden am Beispiel des Wortes „Traum” in Tab. 2.1 aufgeführt. Wenn im Folgenden von n-Grammen die Rede ist, so sind damit immer Zeichen- bzw. Buchstaben-n-Gramme gemeint.

n	Bezeichnung	alle n-Gramme
1	Uni- bzw. Monogramme	`T,r,a,u,m`
2	Bigramme	`_T,Tr,ra,au,um,m_`
3	Trigramme	`_Tr,Tra,rau,aum,um_`
4	Quadgramme	`_Tra,Trau,raum,aum_`
5	Quintgramme	`_Trau,Traum,raum_`

Tabelle 2.1: Bezeichnung und kommaseparierte Auflistung aller Buchstaben-n-Gramme für $n < 6$ am Beispiel des Wortes „Traum”. Die Darstellung entspricht der Zerlegung wie diese in *LangSepa* durchgeführt wird.

2.2. Sprachidentifikation

Mit der schnellen Entwicklung und Expansion des Internets und der damit korrespondierenden zunehmenden Verfügbarkeit an digital abrufbaren Texten, entwickelten sich erste effektive Werkzeuge zur automatischen Sprachidentifikation am Anfang der 1990er Jahre. Methoden der Automatischen Sprachverarbeitung, um automatische Sprachidentifikation zu betreiben, sind sehr gefragt, um bspw. Dokumentenselektion zu betreiben oder weitergehenden sprachspezifische Fragestellungen nachzugehen. Demnach existieren viele Ansätze zur Bewältigung dieser Problemstellung. Einige dieser Ansätze und deren Ergebnisse sollen in diesem Abschnitt präsentiert werden. Die formalen Grundlagen zum Problem der Sprachidentifikation von Textdokumenten werden im folgenden Unterabschnitt dargestellt. Des Weiteren werden die grundlegenden Besonderheiten der Sprachseparierung bei Webseitentexten zusammengefasst.

---

[7] Der Unterstrich fungiert dann als „virtueller” Buchstabe.

2.2.1. Formalisierung des Problems

Die textuellen Rohdaten, mit denen sich *LangSepa* hauptsächlich beschäftigen soll, sind Dokumente, die aus HTML-Webseitentexten generiert worden sind[8].

Die Identifikation der Dokumentensprache(n) wird auf einer Menge von Dokumenten $\mathfrak{D}$ durchgeführt. Diese sei mit

$$\mathfrak{D} = \{d_1, d_2, ..., d_N\} \tag{2.1}$$

gegeben. Wobei $N = |\mathfrak{D}|$, die Anzahl der Dokumente ist. Für die Klassifikation der Dokumente stehen mehrere Sprachen zur Verfügung. Die Menge aller uns bekannten Sprachen sei

$$\mathfrak{L} = \{L_1, L_2, ..., L_M\} \tag{2.2}$$

und deren Kardinalität M.

Die Aufgabe von *LangSepa* ist es, jedem der vorhandenen Dokumente eine Teilmenge von $\mathfrak{L}$ zuzuweisen. Unter mathematischen Aspekten beschreibt den benannten Sachverhalt die folgende Funktion:

$$\ell : \mathfrak{D} \longrightarrow \wp(\mathfrak{L}) \quad . \tag{2.3}$$

Diese Funktion wird im Folgenden als *Labelfunktion* bezeichnet. Diese erlaubt es jeweils einem Dokument keine, eine oder auch mehrere Sprachen zuzuordnen. Die Konstruktion dieser Funktion ist die Hauptaufgabe des Programms.

2.2.2. Methoden

Die meisten der bisher entwickelten automatischen Sprachidentifizierer aus der Literatur lassen sich in die Klasse der „Überwachten Lernsysteme"[9] einordnen. D.h., dass das System mit speziellen Sprachmerkmalen trainiert wird, um anhand dieser Merkmale die Sprache(n)klassifikation eines Textes durchführen zu können. Dies erfordert jedoch ebenso, dass alle Sprachen bekannt sind, die klassifiziert werden sollen. Die bisher am häufigsten verwendeten Sprachmerkmale und deren Erfolg bei der Sprachenklassifikation sollen an dieser Stelle vorgestellt werden.

Die Merkmale zur Spracherkennung basieren zunächst auf

- wortbasierten Ansätzen und/oder
- n-Gramm Ansätzen.

Bei den wortbasierten Ansätzen sind die Benutzung der häufigsten Wörter einer Sprache, den sogenannten Stopwörtern, und kurzen Wörtern, also solchen mit weniger als 6 Zeichen, äußerst populär. n-Gramm Ansätze arbeiten mit Buchstabenfolgen einer oder mehreren Längen n. Je nach Autor variiert das maximale n zwischen 4 und 5.

---

[8] Näheres hierzu findet sich im Unterabschnitt 2.3.2

[9] *supervised learning systems.*

Wortbasierte Ansätze SOUTER ET AL. haben 1994 [16] ein simples Stopwort-Verfahren entwickelt, welches für jede Sprache aus entsprechenden Trainingstexten die 100 häufigsten Wörter ermittelt. Um die Zielsprache eines Dokuments zu bestimmen, führen die Autoren für jede Sprache einen Wortzähler ein. Pro auftretendem Wort erhöhen sich dabei die Zähler derjenigen Sprachen um eins, in denen das Wort vorkommt. Die Sprache mit dem größten Zählerstand wird als Zielsprache des Dokuments gewählt.

Im darauffolgenden Jahr stellte GREFENSTETTE [8] eine Technik basierend auf kurzen Wörtern mit der Maximallänge 5 Zeichen vor. Für jede Sprache wurden aus dem tokenisierten Trainingskorpus die Wörter mit den größten relativen Häufigkeiten ermittelt und diese als Auftretenswahrscheinlichkeit des Wortes gehandhabt. Ein gegebener Satz kann anschließend mit den verketteten Wahrscheinlichkeiten jeder Sprache, die als Produkt der Einzelwahrscheinlichkeiten der Wörter entstehen, der Sprache zugeordnet werden, welche die größte Wahrscheinlichkeit besitzt.

n-Gramm Ansätze Ein Ansatz der auf Uni-, Bi-, Tri- und Quadgrammen aufbaut, wurde 1994 von CAVNAR und TRENKLE vorgestellt [6]. Aus jeder Sprache, die in den Trainingsdaten vorkommt, wurden die n-Gramme in ihrer Auftretens-Häufigkeit bestimmt. Anschließend sortierten sie die Strukturen unabhängig von der Zeichenlänge nach ihrer inversen Häufigkeit und erhielten somit eine Rangfolge. Diese Rangfolge wurde als Sprachprofil bezeichnet. Zu klassifizierende Dokumente erhielten je ein Dokumentenprofil, genau wie diese zu den Sprachen erzeugt worden sind. Dieses Profil kann nun mit den vorhandenen Sprachprofilen durch eine sogenannte „out-of-place" Messung verglichen werden. Hierbei wird die minimale Summe der Ranking-Abstände von Dokumentprofil und den vorhandenen Sprachprofilen ermittelt und die zur minimalen Summe gehörende Sprache dem Dokument zugeordnet.

GREFENSTETTE stellte in [8] ebenso ein Trigramm-Verfahren vor, welches den häufigsten Trigrammen jeder Sprache, in derselben Art wie den kurzen Wörtern, Auftretenswahrscheinlichkeiten zuordnet. Die Wahl der Zielsprache eines Satzes erfolgt auf gleiche Art und Weise durch verkettete Wahrscheinlichkeiten, wie dies auch bei den kurzen Wörtern erfolgte.

Neben den Überwachten Lernsystemen führten BIEMANN und TERESNIAK ein Unüberwachtes Lernsystem zur Sprachidentifikation im Jahr 2005 vor [3], welches hier lediglich kurz vorgestellt werden soll. In dieser Arbeit wurde auf Basis von signifikanten Satzkookkurrenzen ein Clustering durchgeführt, welches als Ergebnis bei geeigneter Parametrisierung Cluster erzeugte, die je einer Sprache zuzuordnen waren. Dieses besitzt zwar den Nachteil, dass die Cluster in einer Nachverarbeitung noch mit den Sprachen versehen werden müssen, aber dafür entstehen keine Fehlklassifikationen aufgrund von fehlenden Sprachen in den Trainingsdaten, da sich aus den zu überprüfenden Texten eine beliebige Anzahl von Clustern bilden kann.

Ergebnisse Zunächst ist festzuhalten, dass die qualitativen Ergebnisse beider Identifiziermethoden mit zunehmender Länge der Dokumente bzw. Sätze bessere Ergebnisse erzielen. Aus diesem Grund legen die Autoren oftmals Wert auf die Darstellung ihrer Genauigkeit in Abhängigkeit von der Wortanzahl oder der Größe der Dokumente in bytes. Des Weiteren wurden die vorgestellten Verfahren auf niemals mehr als zwanzig Sprachen gleichzeitig evaluiert. Die untersuchten Sprachen stammten hauptsächlich aus dem europäischen Sprachraum. Nun zu den Ergebnissen im Detail.

SOUTER erzielte mit den 100 besten Wörtern jeder Sprache eine Genauigkeit von bis zu 91 %. Die Dokumente enthielten dabei zwischen 60 und 420 Zeichen. GREFENSTETTE konnte mit seinem Kurzwort-Ansatz ab 16 Wörtern pro Satz eine Genauigkeit von mindestens 99 % pro Sprache erzielen. Beim n-Gramm Ansatz von CAVNAR und TRENKLE wurde neben der Abhängigkeit von der Dokumentenlänge zusätzlich die Qualität nach der Anzahl der verwendeten n-Gramme bestimmt. Es zeigte sich, dass bei 300 n-Grammen eine optimale gemittelte Genauigkeit von 99,8 % erzielt werden konnte. Den Grund hierfür sahen die Autoren darin, dass diese 300 n-Gramme die notwendigsten Informationen über die Sprache enthielten. Der Trigramm-Ansatz von GREFENSTETTE zeigte, im Vergleich zu seinem wortbasierten Gegenstück, bessere Genauigkeit bei kürzeren Sätzen. Bereits ab 11 Wörtern konnte in jeder Sprache eine Mindestgenauigkeit von 99 % ermittelt werden. Zu diesem Ergebnis gelang auch LANGER in seinen vergleichenden Untersuchungen zu wort- und n-Gramm-basierten Ansätzen im Jahr 2002 [10]. Dies ist der Tatsache geschuldet, dass bereits in einem Wort mehrere Trigramme enthalten sein können und somit ein Treffer in der Bestenliste jeder Sprache sich potenziell eher ereignet. Dies hat jedoch im Gegenzug dazu den Effekt, dass mehr Rechenoperationen erforderlich werden [8]. LANGER stellte außerdem fest, dass seltenes Vokabular besser von n-Grammen gehandhabt wird. Im Gegenzug haben wortbasierte Ansätze Vorteile gegenüber der n-Gramm Methode in der Beseitigung von Fehlerquellen in Häufigkeitslisten.

Die eben beschriebenen Ansätze gehen alle von der Vorraussetzung aus, dass die Dokumente bzw. Sätze jeweils genau einer Sprache zuzuordnen sind. Dennoch stellen diese zumeist Basistechniken für aktuellere Sprachidentifizierer dar. Diese verwenden zunehmend hybride Verfahren, um die Nachteile der einzelnen Methoden gegeneinander aufzuwiegen. Die Wahl und der Einsatz von Techniken ist dabei von der Problemstellung abzuwägen. So auch in der vorliegenden Arbeit, die sich inbesondere mit der Sprachidentifikation von Webtexten befasst.

2.2.3. Herausforderungen bei Webseitentexten

Maschinenlesbare Texte von Dokumenten bilden für diese Arbeit die Grundlage, um Sprachklassifikation vornehmen zu können. Im Folgenden sollen einige Besonderheiten von Dokumenten, die aus Webseitentexten erzeugt wurden, also solche, die durch Entfernung sämtlicher HTML-Tags aus Webseiten hervorgehen, vorgestellt werden. Die Besonderheiten dieser Dokumente sollen gegenüber Zeitungsartikeln oder Büchern wahrgenommen werden.

Mehrsprachigkeit: Aufgrund von Formfreiheit bzgl. der Erstellung und Veröffentlichung von Webseiten, kommt es vor, dass in einem Dokument verschiedene Sprachen erscheinen. Dies können bspw. Übersetzungen der Inhalte oder Einbindung von anderssprachigen Webseiten in die eigene sein.

Störende Elemente: Die Struktur von Webseiten sieht u.a. ein Menü zur Navigation innerhalb der Seite vor. Des Weiteren benötigen viele Seiten, um diese zu unterhalten oder um damit Geld zu verdienen, Werbetexte. Diese stören den Textfluss, und verändern je nach Häufigkeit des Erscheinens die statistischen Eigenschaften, die letztendlich in dieser Arbeit zur Identifikation einer Sprache herangezogen werden.

Nicht natürlichsprachiger Text: Dieser Punkt beinhaltet die Seiten, die künstliche Sprachen, wie z.B. Programmiersprachen oder computergenerierte Texte (z.B. Logdateien) im Dokument beinhalten.

Diesen speziellen Eigenschaften ist mit der Anwendung von speziellen Techniken zu begegnen, um eine erfolgreiche Identifikation durchführen zu können. In der vorliegenden Arbeit werden die vorgestellten Probleme mit kleinere Bedingungen an den Text teilweise umgangen. Dazu gehören u.a. die Forderung einer Mindestzeichenlänge pro Zeile, sowie eine Mindestmenge unterschiedlicher vorkommender Stopworte pro Sprache im Text.

2.3. Überblick der beteiligten Komponenten

In diesem Abschnitt erläutere ich den grundlegenden Aufbau und die Arbeitsweise des Sprachidentifizierers *LangSepa*. Anschließend möchte ich die beteiligten Komponenten kurz einführen und deren Funktionen in Bezug auf das Tool *LangSepa* beschreiben.

2.3.1. LangSepa

Die Hauptkomponente, welche in dieser Arbeit vorgestellt und eingeführt wird, hat die Bezeichnung *LangSepa* erhalten. Dies wurde in Anlehnung an den Titel dieser Arbeit als Kurzwort für „**Lang**uage **Sepa**ration"[10] gewählt und spiegelt die Aufgabe wieder, die das Werkzeug erfüllen soll.

Das System ist dem Überwachten Lernen zuzuordnen, da das Wissen, welches zur Sprachklassifikation verwendet wird, aus Trainingsdaten einer Vielzahl von Sprachen stammt. Die eingesetzten Techniken basieren auf den in Unterabschnitt 2.2.2 vorgestellten Methoden und sind somit ausschließlich auf die Zeichen- bzw. Buchstabenebene der Dokumente beschränkt.

In Abb. 2.1 ist das Zusammenspiel der beteiligten Komponenten am Sprachidentifikations-Prozess schematisch festgehalten. Die Trainingsbasis für *LangSepa* bilden Worthäufigkeitslisten von ca. 350 Sprachen. Diese werden im Unterabschnitt 2.3.3 näher beschrieben. Das Information-Retrieval-System FindLinks hält die zur Sprachklassifikation ausstehenden Dokumente bereit. Die Details hierfür

---

[10] Im Deutschen: „Sprachseparierung".

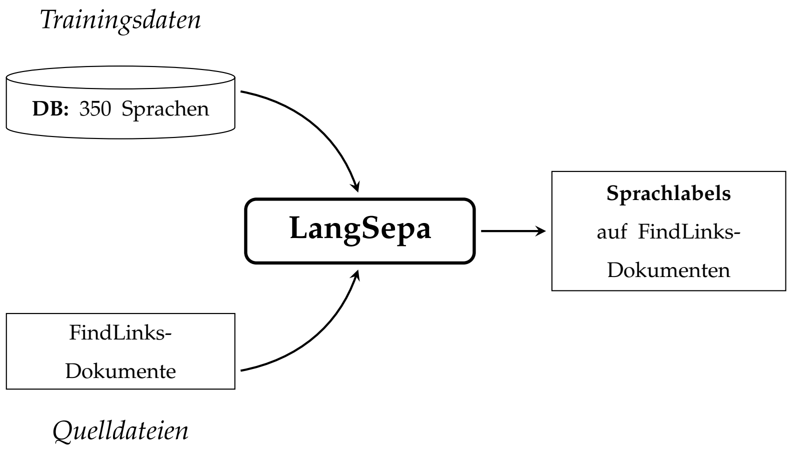


Abbildung 2.1: Grobüberblick der beteiligten Komponenten und den zugehörigen Datenflüssen von *LangSepa*

möchte ich im hierauffolgenden Unterabschnitt präsentieren. Als Ergebnis des Identifikationsschrittes sollen sämtliche Dokumente mit möglichst der Sprachkennung bzw. den Sprachkennungen[11] versehen werden, deren zugehörige Sprache(n) tatsächlich im Textinhalt erscheint bzw. erscheinen. Ebenso soll es die Möglichkeit geben, vorhandene Dokumente als untauglich, d.h. ohne Sprache zu kennzeichnen. Dies müsste u.a. die Dokumente treffen, die wie in 2.2.3 beschrieben, keinen natürlichsprachlichen Text beinhalten oder ausschließlich aus störenden Webseitenelementen bestehen.

2.3.2. FindLinks

FindLinks ist ein Webcrawler, dessen erste Version Ende 2003, als Teil des Projektes „Deutscher Wortschatz" in der Abteilung für Automatische Sprachverarbeitung des Instituts für Informatik der Universität Leipzig entwickelt worden ist. Das verteilte System ist durch eine Client-Server-Architektur realisiert. Das Ausgangsziel dieser Implementierung war es mit Hilfe verteilter Lasten und somit unter der Verwendung ungenutzter Rechenkapazitäten der Clients, die Linkstruktur des Webs zu analysieren [22]. Der FindLinks-Server verteilt hierbei Web-Linklisten an die Clients, welche die verlinkten Seiten u.a. auf weiterführende Links untersuchen, um anschließend die Resultate an den Server zurückzuschicken.

Mittlerweile ist die dritte Implementierung von FindLinks in ständiger Weiterentwicklung und der zugehörige Client ist für jeden zugänglich und verwendbar. Über das Plugin-Konzept des neuen, in Java implementierten Clients, entstanden weiterführende Werkzeuge zur Verarbeitung der Webseiten. Dies hat die Komplexität und Leistungsfähigkeit des Systems beträchtlich erhöht. Neben Linkanalyse, der Suche neuer Sprachen via Trigramm-basierten Verfahren und verschiedenster Werkzeuge zur statistischen Auswertung der Daten, sind für diese Arbeit

---

[11] Hiermit ist die Repräsentation einer Sprache in Form eines Sprachcodes gemeint. Der in dieser Arbeit verwendete Sprachcode ISO 639-3 wird in Abschnitt 2.5 ausführlich vorgestellt.

das CharsetTextCollector-Plugin und das SendHome-Plugin von Interesse. Diese stellen die Dokumente für *LangSepa* zur Verfügung und werden im Folgenden etwas näher erläutert. [20]

CharsetTextCollector-Plugin: Über diese Erweiterung steuert FindLinks die Dokumentensammlung und -speicherung lokal beim Client. Dabei werden jene HTML-Seiten gespeichert, deren charset-Metatag in einer entsprechenden Liste auftaucht. Die zugehörigen, vollständigen Seiten werden in Dateien separiert nach deren Zeichenkodierung abgelegt. Metainformationen zu Quell-URL, FindLinks-User und FindLinks-Version werden zusätzlich im Dokumentenkopf in Form von HTML-Tags gespeichert. [20]

SendHome-Plugin: Die vom CharsetTextCollector-Plugin gesammelten HTML-Texte können mit diesem Zusatzwerkzeug automatisch an die Server der Abteilung für Automatische Sprachverarbeitung in regelmäßigen Abständen geschickt werden. Die heimgeschickten Dateien werden anschließend clientseitig entfernt.

In einem ersten serverseitigen Nachverarbeitungsschritt wandelt ein Werkzeug mit der Bezeichnung HTML2Text[12] die vom Client gecrawlten Dokumente in gewöhnliche Textdokumente. Darüber hinaus konvertiert HTML2Text die Dokumente unterschiedlichster Originalkodierungen nach utf-8. Eine Darstellung der eben genannten Prozesse wird in Abb. 2.2 gegeben.

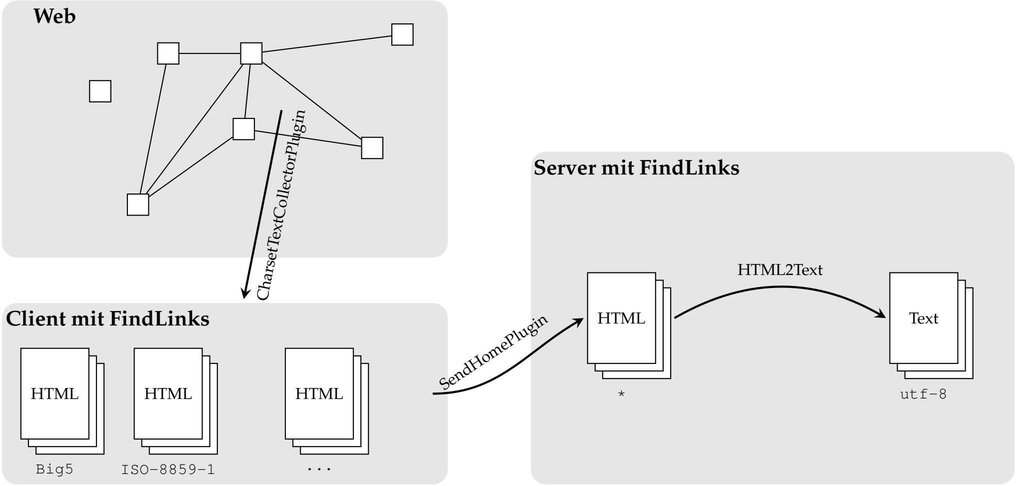


Abbildung 2.2: Die für diese Arbeit relevanten FindLinks-Komponenten und ihr Zusammenspiel im Überblick. Pfeile stellen grob die Datenflüsse dar. Die zugehörigen Bezeichner sind die Realisatoren dieser Datenflüsse.

2.3.3. Trainingsdaten

Für das Training von *LangSepa* werden Worthäufigkeitslisten von 351 Sprachen verwendet. Ungefähr 60 der Sprachen können von Datenbanken der Abteilung

---

[12] HTML2Text wurde ebenfalls in der Abteilung für Automatische Sprachverarbeitung implementiert.

für Automatische Sprachverarbeitung der Universität Leipzig bezogen werden. Alle Weiteren entstammen einer weiteren Datenbank, im Folgenden aspra10, deren Wortlisten auf der Menschenrechtscharta der Vereinten Nationen [23] basieren. Da die Charta pro Sprache aus lediglich 2000 Wörtern seine Daten bezieht, werden die Wortlisten der Abteilungsdatenbanken für das Training der Stopwörter bevorzugt. Aufgrund der stärkeren Variation der Wörter, in puncto Länge und erscheinenden Buchstaben, in den Worthäufigkeitslisten von aspra10, werden diese für das n-Gramm-Training herangezogen. In den Tab. 2.2 und 2.3 sind die Topwörter einiger Sprachen der Trainingsquellen aufgeführt. Eine Unicode-Kodierung ist in beiden Quellen durch utf-8 gegeben.

deu		sco		fri		ndo	
Wort	Freq.	Wort	Freq.	Wort	Freq.	Wort	Freq.
und	90	the	132	en	106	na	69
der	66	and	110	fan	103	uuthemba	48
die	48	o	98	de	69	kehe	36
zu	37	tae	53	it	60	nenge	31
auf	35	in	49	yn	42	oku	30
Recht	32	hes	35	te	41	Okatopolwa	30
...	...	...	..	...	..	...	..
$\sum$	1644	$\sum$	1862	$\sum$	2016	$\sum$	1382

Tabelle 2.2: Top-6 Wörter der Sprachen Deutsch, Schottisch, Westfriesisch und Ndonga auf der aspra10-Datenbank. Neben den Wörtern sind ebenso die jeweiligen absoluten Häufigkeiten gesetzt. Die dargestellte Summe aller absoluten Häufigkeiten entspricht der Gesamtzahl der Wörter der UN-Menschenrechtscharta je Sprache.

deu		eng		fra		fin	
Wort	Freq. $[10^6]$	Wort	Freq. $[10^6]$	Wort	Freq. $[10^6]$	Wort	Freq. $[10^6]$
und	4,321	the	10,03	de	10,112	ja	1,436
der	4,167	and	5,569	la	4,752	on	0,979
die	4,046	of	5,532	et	4,148	että	0,226
in	2,244	to	5,25	des	3,432	ei	0,218
den	1,667	a	3,854	à	3,428	myös	0,185
zu	1,581	in	3,194	le	3,12	tai	0,173
...	...	...	...	...	...	...	...
$\sum$	157,709	$\sum$	189,413	$\sum$	181,544	$\sum$	34,949

Tabelle 2.3: Top-6 Wörter der Sprachen Deutsch, Englisch, Französisch und Finnisch auf der aspra20-Datenbank. Strukturelle Darstellung wie in Tab. 2.2.

Es ist unschwer zu erkennen, dass die aspra10-Datenquelle schwach trainiert ist. Als Referenz hierfür ist Deutsch in beiden Tabellen aufgeführt. Die wenigen Wörter, die die Menschenrechtscharta enthält, führen dazu, dass Fachvokabular bereits in den Top-6 der Wortlisten Einschlag findet.

2.4. Sprachstatistik

Der automatische Sprachidentifizierer *LangSepa* verwendet zur Klassifikation der Dokumente statistische Eigenschaften von Sprache. Die Merkmalsträger sind Zeichen, Zeichenfolgen und Wörter.

Das ZIPFsche Gesetz gibt Aufschluß über den Zusammenhang zwischen Auftretenshäufigkeit von Wörtern bzw. Wortformen und deren Rang in einer Sprache. Die Darstellung dieses Gesetzes aus der quantitativen Linguistik, soll zunächst erfolgen. Danach werden die statistischen Größen, die im Programm Anwendung finden, eingeführt.

2.4.1. Das ZIPFsche Gesetz

Der amerikanische Sprachwissenschaftler ZIPF verfasste 1935 erstmals seine Untersuchungen zur „Psycho-Biologie" von Sprachen. Darin zeigte er anhand der Auftretenshäufigkeit von Wortformen, dass Sprachen bzw. deren Sprecher, dem Prinzip der geringsten Anstrengung folgen [9, S. 87]. Diese Sprachökonomie manifestiert sich u.a. durch die hochfrequente Verwendung von inhaltslosen Funktionswörtern kurzer Länge. Einige dieser sogenannten Stopwörter sind den Tab. 2.2 und 2.3 zu entnehmen. [19, vgl.]

Überdies hinaus formulierte er eine, auf eben genanntem aufbauende Regelmässigkeit, der jede natürliche Sprache mehr oder minder folgt. Die Bezeichnung als „das ZIPFsche Gesetz" wird in einem Großteil der Literatur vorgenommen, weil es das bedeutendste Zeugnis des Schaffens von ZIPF repräsentiert. Von ihm wurden jedoch eine Reihe von Gesetzen und Regelmäßigkeiten festgellt, die bspw. in [9, vgl. S. 87–94] dargestellt werden.

Das Gesetz trifft folgende Aussage:

$$r \cdot n \approx c \quad , \tag{2.4}$$

wobei r der Rang eines bestimmten Wortes in der Worthäufigkeitsliste einer Sprache $L \in \mathfrak{L}$ ist, n die absolute Häufigkeit ($H_{\text{abs}}(w, L) = n$) des Auftretens und c eine Konstante darstellt. ZIPF zeigte, dass c innerhalb einer Sprache für alle darin auftretenden Wörter relativ konstant ist [9, S. 88]. Die Genauigkeit seiner erkannten Regelmäßigkeit ist an der Linearität der doppeltlogarithmischen Koordinatendarstellung in Abb. 2.3 abzulesen. Aus dieser ist ersichtlich, dass eine Linearisierung, insbesondere im mittleren Bereich ($4 \leq \log(r) \leq 7$) der jeweiligen Sprachen gegeben ist. Die Randbereiche, welche auf der einen Seite die Top-Wörter ($\log(r) < 4$) und die seltensten Wörter auf der anderen Seite ($\log(r) > 7$) repräsentieren, besitzen je Sprache unterschiedliche Ausprägung und Abweichung von ihrer Linearisierung. Das Gesetz gilt demnach lediglich approximativ und ist als Idealisierung anzusehen [9, vgl. S. 88–90].

Da die Konstante c abhängig von der Größe des betrachteten Wortkorpus ist, kann eine allgemeinere Sprachkonstante bestimmt werden, indem beide Seiten des Gesetzes durch die Summe aller absoluten Worthäufigkeiten s geteilt werden.

$$\frac{r \cdot n}{s} \approx \frac{c}{s} \quad \Longleftrightarrow \quad r \cdot H_{\text{rel}}(w, L) \approx \frac{c}{s} \quad . \tag{2.5}$$

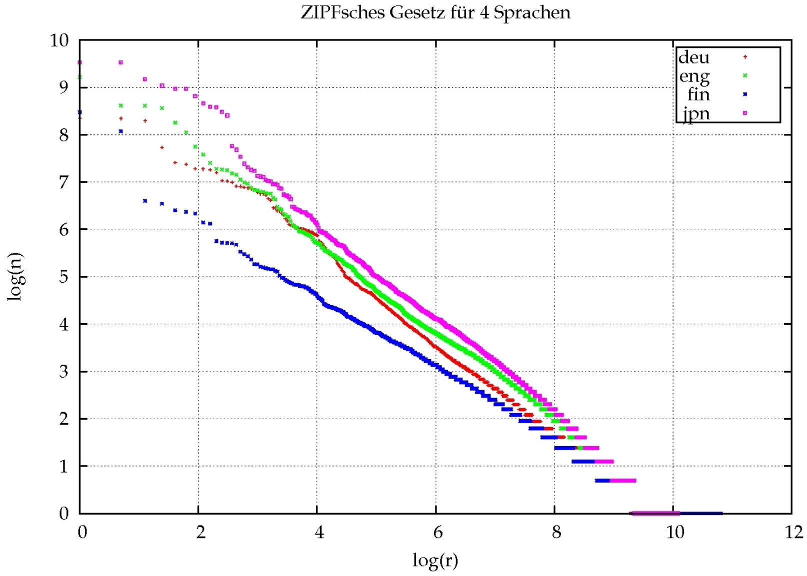


Abbildung 2.3: Doppeltlogarithmische Darstellung zur Visualisierung der Genauigkeit des ZIPFschen Gesetzes. Als Datenbasis dienten die Wortlisten der Sprachen Deutsch, Englisch, Finnisch und Japanisch der aspra20-Datenbank. Es wurden jeweils 10.000 Sätze verwendet.

$H_{\text{rel}}(w, L)$ ist dabei die relative Häufigkeit des Wortes w im Korpus zur Sprache $L \in \mathfrak{L}$. Diese, von der Wortanzahl unabhängige Konstante, ist dennoch abhängig von der Größe des verwendeten Korpus [21, Vl. 2]. Um verschiedene Sprachen anhand dieser Konstante vergleichen zu können, wurden für die Tab. 2.4 annähernd gleiche Werte für s gewählt. Die Werte $\bar{c}/s$ treffen einige Aussagen über die strukturellen Eigenschaften der Sprache. So besitzen die Sprachen mit einer höheren Konstante $\bar{c}/s$ mehr Wortformen. Das steht im Fall Finnisch in engem Zusammenhang mit deren hochflektierenden Wörtern. Wenn neben dieser Konstante zusätzlich die Wortlisten aus Tab. 2.3 zum Vergleich von Finnisch, Englisch und Deutsch herangezogen werden, so fällt zusätzlich die ungleiche Verteilung der Stopwort-Häufigkeiten ins Gewicht dieser Konstanten. Diese Resultate haben zur Folge, dass die relative Überdeckung eines gegebenen Textes durch eine vorgegebene Anzahl an Stopwörtern, unterschiedliche Werte annehmen wird. Aufgrund dessen wird im nächsten Abschnitt u.a. die sprachabhängige relative Überdeckung allgemein eingeführt.

Sprache	$\bar{c}/s$
Deutsch	0,139
Englisch	0,104
Finnisch	0,232
Japanisch	0,073

Tabelle 2.4: Die normierten ZIPFschen Konstanten einiger Sprachen. Als Grundlage dienten die Wortlisten, welche bereits in Abb. 2.3 visualisiert wurden. $\bar{c}$ stellt den Mittelwert von c über alle Wörter des jeweiligen Korpus dar.

2.4.2. Relative Textüberdeckung

Aus den Worthäufigkeitslisten der vorhandenen Sprachen können absolute Häufigkeiten abgelesen und relative Häufigkeit von Worten einfach errechnet werden. Aus diesen Einzelwerten kann nun die sprachabhängige Textabdeckung bestimmt werden, die für ein Textdokument entsprechender Sprache erwartet wird. Die Herleitung und Formalisierung der sprachabhängigen relativen Textüberdeckung für eine Zusammenstellung von Worten soll nun erfolgen.

Im Programm werden aus Effizienzgründen nicht die gesamte Wortliste, sondern lediglich die besten k Wörter jeder Sprache importiert. Die Funktionen B und B_k bilden Sprachen auf deren zugehörige Wortmenge aller Wortlisteneinträge, bzw. die Wortmenge der k höchstfrequenten Wörter ab.

$$B, B_k : \mathfrak{L} \longrightarrow \wp(W) \tag{2.6}$$

W ist hierbei die Menge aller Wörter. Wenn nun $H_{\text{abs}}(w, L)$ wiederum die absolute Häufigkeit des Wortes $w \in W$ in der Wortliste der Sprache $L \in \mathfrak{L}$ beschreibt, so ist die „optimale"[13] relative Textüberdeckung der Sprache L durch die besten k Worte gegeben durch

$$u_{\text{opt}}(L, k) = \frac{\sum\limits_{w \in B_k(L)} H_{\text{abs}}(w, L)}{\sum\limits_{\bar{w} \in B(L)} H_{\text{abs}}(\bar{w}, L)} \quad . \tag{2.7}$$

Diese gibt demnach den Anteil der besten k Worte im gesamten jeweiligen Wortkorpus wieder. Im Verfahren wird sie zum Vergleich mit der im Textdokument gemessenen Überdeckung herangezogen. Darauf aufbauend können Entscheidungen für die Zuordnung einer, mehrerer oder keiner Sprache getroffen werden.

2.4.3. n-Gramm-Wahrscheinlichkeiten

Im Programm werden beim Import der Wörter die vorhandenen n-Gramme für $n \leq 3$ aus den Wörtern extrahiert und deren absolute Häufigkeiten entsprechend bestimmt. Da die Erzeugung unterschiedlicher n-Gramme aufgrund ihrer fixierten Länge kombinatorisch beschränkter ist als bei Wörtern, wird die relative Textüberdeckung durch n-Gramme verschiedener Länge nicht berücksichtigt. Ähnliche Sprachen sind sich demnach in den verschiedenen n-Grammen noch ähnlicher, da diese auf Basis der Stopwörter gebildet werden. Aus diesem Grund werden Auftretenswahrscheinlichkeiten für n-Gramme verwendet. Neben dem bloßen Auftreten eines n-Gramms wird hiermit zusätzlich der Rang, also auch die Häufigkeit des Auftretens mit berücksichtigt.

Die Wahrscheinlichkeiten werden nun am Beispiel von Trigrammen eingeführt. Sei T zunächst die Menge aller Trigramme und $B_{T,k} : \mathfrak{L} \rightarrow \wp(T)$ eine Funktion, die die Trigramme einer Sprache auf Basis derer besten k Wörter bestimmt. Des

---

[13] Die Bezeichnung „optimal" wird in dem Sinne verwendet, dass die Wortlisten als Referenz für die zu klassifizierende Textdokumente fungieren.

Weiteren sei $H_{\text{abs}}(t, L)$ die absolute Häufigkeit des Trigramms $t \in T$ in der Trigrammliste der Sprache L. Somit lässt sich die zugehörige relative Häufigkeit $H_{\text{rel}}(t, L)$ in dieser Liste durch

$$H_{\text{rel}}(t, L) = \frac{H_{\text{abs}}(t, L)}{\sum\limits_{\bar{t} \in B_{T,k}(L)} H_{\text{abs}}(\bar{t}, L)} \tag{2.8}$$

berechnen. Dies entspricht der Auftretenswahrscheinlichkeit P unter der Bedingung, dass die Sprache L vorliegt:

$$P(t|L) := H_{\text{rel}}(t, L) \quad . \tag{2.9}$$

Die Wahrscheinlichkeit $P(t)$ lässt sich unter Betrachtung aller Sprachen folgendermaßen errechnen:

$$P(t) := \frac{\sum\limits_{L \in \mathfrak{L}} H_{\text{rel}}(t, L)}{M} \quad . \tag{2.10}$$

Das entspricht der gemittelten relativen Häufigkeit über alle Sprachen und gilt unter der Annahme, dass alle Sprachen gleichwahrscheinlich sind. D.h. $P(L) = 1/M$ für alle $L \in \mathfrak{L}$. Da im Dokument die Spracherkennung gefordert ist, benötigt man die Wahrscheinlichkeit der Sprache L unter der Bedingung, dass ein Trigramm t erschienen ist. Hierfür hilft der Satz von Bayes:

$$P(L|t) = \frac{P(t|L) \cdot P(L)}{P(t)} \quad . \tag{2.11}$$

Nach einigen Umformungen folgt

$$P(L|t) = \frac{P(t|L)}{\sum\limits_{\bar{L} \in \mathfrak{L}} P(t|\bar{L})} \quad . \tag{2.12}$$

Wahrscheinlichkeit von n-Gramm-Folgen Für die Sprachklassifikation eines Textdokumentes sind die eben hergeleiteten Wahrscheinlichkeiten die Basis für die Zuordnung einer Sprache zu einem Ausschnitt des Dokumentes. Ein solcher Ausschnitt ist eine Folge von Trigrammen. Diese sei durch $S = t_1 t_2 ... t_h$ benannt. Die Wahrscheinlichkeit $P(L|S)$, dass diese Folge der Sprache L angehört ist durch

$$P(L|S) = \frac{P(L) \cdot \prod\limits_{j=1}^{h} P(t_j|L)}{P(S)} \tag{2.13}$$

gegeben ([12, S. 258] und Gl. (2.11)). Da $P(L)$ und $P(S)$ für jede Sprache $L \in \mathfrak{L}$ in Gl. (2.13) die gleichen Werte haben, reicht es

$$P(L|S) \sim \prod_{j=1}^{h} P(t_j|L) \tag{2.14}$$

zu betrachten. Durch das fortlaufende Produkt von Einzelwahrscheinlichkeiten kann es einerseits zu Fließkommaunterläufen kommen und andererseits zu Nullungen [12, S. 258–259], wenn eine Einzelwahrscheinlichkeit den Wert 0 ergibt. Ersterem Problem kann mit Logarithmierung von Gl. (2.13) begegnet werden oder mit einer Begrenzung der Verkettungslänge. In dieser Arbeit finden beide Varianten Anwendung.

$$\begin{aligned} \log_{10} P(L|S) &= (\log_{10} P(L) - \log_{10} P(S)) + \sum_{j=1}^{h} \log_{10} P(t_j|L) \qquad (2.15) \\ &= Z + \sum_{j=1}^{h} \log_{10} P(t_j|L) \quad , \qquad (2.16) \end{aligned}$$

wobei Z für alle Sprachen den gleichen Wert annimmt. Auf die Berechnung von Z kann somit für die Bestimmung der wahrscheinlichsten Sprache verzichtet werden. Nullwahrscheinlichkeiten treten jedesmal auf, wenn ein erkanntes Trigramm in S nicht in der Trigrammliste der gerade betrachteten Sprache auftaucht. Dieses Problem kann durch eine Glättung umgangen werden. Diese wird folgendermaßen gewählt und ersetzt ebenso Gl. (2.9):

$$P(t|L) = \begin{cases} H_{\mathrm{rel}}(t, L) & \text{für } t \in B_{T,k}(L) \\ \frac{1}{10} \cdot \left(\min\limits_{L \in D(t)} H_{\mathrm{rel}}(t, L) \right) & \text{sonst.} \end{cases} \tag{2.17}$$

$D(t)$ ist hierbei die Menge aller Sprachen aus $\mathfrak{L}$, in der $H_{\mathrm{rel}}(t, L) > 0$ ist.

Nach der Berechnung der Wahrscheinlichkeiten $\log_{10} P(L|S)$ gemäß Gl. (2.15), ist es möglich die wahrscheinlichste Sprache zu bestimmen:

$$L_{\mathrm{I}} = \arg\max_{L \in \mathfrak{L}} (\log_{10} P(L|S)) \quad . \tag{2.18}$$

Diese kann nun der Trigrammsequenz als Sprache zugeordnet werden.

2.5. ISO 639 Sprachcode

Für die Kennung einer Sprache werden in *LangSepa* Sprachcodes als Bezeichner einer Sprache verwendet. Die hier verwendeten Sprachcodes entstammen den Richtlinien der standardisierten Kodierung nach ISO 639 in der Teilnorm ISO 639-3.

Der Herausgabe der ISO 639-3 Teilnorm gingen die zwei vorherigen Teilnormen ISO 639-1 und -2, sowie einige weitere Normierungsbestrebungen voraus. Für das Bibliothekswesen und den Austausch von Daten bildet die ISO 639 ein wichtiges und grundlegendes Werkzeug zur Auszeichnung von Texten. Die Entwicklung des Standards der ersten beiden Teilnormen wurde im Wesentlichen durch Institutionen vorangetrieben, die geschriebene Sprache auszeichnen müssen. Eine Darstellung aller gesprochenen Sprachen war somit durch oftmals zu geringe Mengen an geeigneten Text-Repräsentanten durch diesen Standard nicht möglich. Die Teilnorm 3 erweitert sich deshalb um die gesprochenen Sprachen.

Der Standard existiert seit 2002 und unterliegt seitdem einem stetigen Prozess der sukzessiven Ausdifferenzierung und Erweiterung um und von Sprachen. [24, vgl.]

Die Sprachen, die in den Kodierungstabellen eingetragen sind, erfüllen je Teilnorm bestimmte Bedingungen. So erscheinen in der ISO 639-1 Norm die häufigsten Literatursprachen und die am weitesten entwickelten Sprachen mit spezialisierter Terminologie. Mit dieser Einschränkung reicht ein Sprachcode, der aus lediglich zwei Buchstaben des lateinischen Alphabets besteht. Mit den sechsundzwanzig Buchstaben sind demnach $26^2 = 676$ Codes möglich. Eine Erweiterung dessen erfolgte in der darauffolgenden Teilnorm. In dieser wurden jene Sprachen mit einem Code der Form[14]

$$\text{[a-zA-Z]}\{3\}$$

versehen, für die eine Mindestmenge entsprechender Literatur vorhanden war. Außerdem wurden aufgrund der weniger restriktiven Form, Sprachcodes vergeben, die für Zusammenfassungen von ähnlichen Sprachen stehen, mit aufgenommen. Die erweiterte Form gestattet es $26^3 = 17\,576$ Identifikatoren zu vergeben [24, vgl.]. Die Norm definiert keine Unterscheidung zwischen Groß- und Kleinschreibung der Kennzeichnungen, d.h. der Sprachcode `deu` ist äquivalent zu `Deu` und `DEU`. Für die Anwendung und Vergabe der Sprachcodes zur Auszeichnung von Dokumenten bietet sich jedoch eine Vereinheitlichung an, um keine Verwirrung zu verursachen. In dieser Arbeit als auch in *LangSepa* werden Sprachen deshalb ausschließlich mit lateinischen Kleinbuchstaben kodiert.

Aufgrund der großen Differenzierung von Sprachen und der Verwendung des ISO 639-3, insbesondere auf den Datenbanken, der Abteilung für Automatische Sprachverarbeitung, findet dieser Standard hier Anwendung. Dieser verwendet die Kodierrichtlinie bzw. -form der vorherigen Teilnorm. Allerdings unterscheidet dieser zwischen Makrosprachen und Einzelsprachen [11, vgl. S. 8–9]:

Makrosprachen sind Zusammenfassungen genetisch nah verwandter Einzelsprachen. Eine Makrosprache enthält somit mehrere Einzelsprachen. Beispiele:

`nor` (Norwegisch) enthält die Einzelsprachen `nno` (Nynorsk) und `nbo` (Bokmål)

`zho` (Chinesisch) enthält 14 Einzelsprachen, u.a. `cmn` (Mandarin Chinesisch) und `czo` (Min Zhong Chinesisch)

Einzelsprachen sind in erster Linie natürliche Sprachen und deren Dialekte. Aber auch historische Sprachen, Plansprachen, Zeichensprachen als auch konstruierte Sprachen finden sich in den Kodierungen wieder. Programmiersprachen werden in den Listen nicht geführt [24, vgl.]. Der ISO 639-3 hat fixe Kriterien um Sprachen und Dialekte zu trennen. Näheres hierzu findet sich in [11, vgl. S. 9]. Die Anwendung dieser Kriterien, sowie die Zuordnung und Vergabe der Kodierungen obliegt der SIL[15] International.

Eine vollständige Liste der Makrosprachen und deren zugehörigen Einzelsprachen kann unter [25] eingesehen werden. Eine Einzelsprachenliste ist unter [26]

---

[14] Hier in Form eines regulären Ausdrucks.

[15] Summer Institute of Linguistics.

abrufbar. In der ISO 639-3 sind derzeit 7 918 der 17 576 theoretisch möglichen Kodierungen vergeben.

Eine Auszeichnung erfolgt in *LangSepa* durch Einzelsprachen, wenn dies möglich ist. Makrosprachen werden nur dann verwendet, wenn die zugehörige speziellere Wortliste der Einzelsprache nicht in den Datenbanken enthalten ist.

3. Experimentelle Voruntersuchungen

Bevor das Verfahren und die Implementierungsdetails vorgestellt werden, soll dieses Kapitel der Darstellung einiger experimentell ermittelter Basisdaten dienen. Dabei sind die FindLinks-Dokumente, die trainierten Sprachen und die im Abschnitt 2.4.2 eingeführten relativen Textüberdeckung Gegenstand der Untersuchungen. Die vorgestellten Ergebnisse werden der Anpassung des Verfahrens dienlich sein. Wenn sich direkte Konsequenzen ableiten lassen, so werden diese hier ebenfalls genannt.

3.1. Dokumente

Im Unterabschnitt 2.2.2 ist die Güte der Ergebnisse bei vielen Ansätzen in Abhängigkeit von der Dokumentenlänge dargestellt worden. Ein längeres Dokument ging dabei mit einer besseren Spracherkennung einher. Die hier betrachteten FindLinks-Dokumente beinhalten Texte, welche aus Webseiten generiert worden sind. In Abb. 3.1 sind die FindLinks-Dokumentlängen grafisch dargestellt. Aus dieser ist ersichtlich, dass gecrawlte Seiten mit Zeichenlänge < 100 bereits vorher aussortiert werden. Dies realisiert das CharsetTextCollector-Plugin (siehe Unterabschnitt 2.3.2). Es ist also bereits eine Mindestlänge gegeben, die möglicherweise relevante Informationen beinhaltet, um die Sprache klassifizieren zu können. Demnach ist im Verfahren kein Vorverarbeitungsschritt nötig, der eine Mindestlänge für Dokumente fordert. Für die Tests auf Referenzdokumenten wird aufgrund dieses Befundes eine Mindestlänge von 100 Zeichen festgelegt.

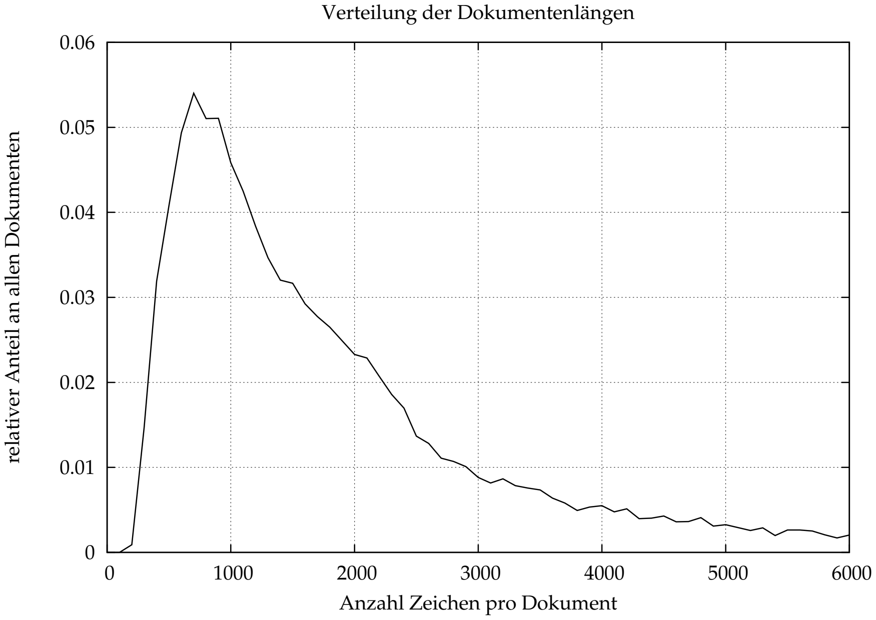


Abbildung 3.1: Verteilung der Dokumentengrößen gemessen in Anzahl Zeichen pro Dokument. Für diese Erhebung wurden 32 000 zufällig ausgewählte FindLinks-Dokumente herangezogen.

Ein weiterer Untersuchungsgegenstand der Dokumente baut auf den Besonderheiten von Webseiten auf, welche bereits in 2.2.3 vorgestellt worden. Die Elemente, die zur Störung des Textflusses beitragen, werden vom Werkzeug HTML2-Text auf Grund ihrer ursprünglichen Einbettung in den HTML-Code in einzelne Zeilen des resultierenden Dokuments geschrieben. Dies hat sehr viele Zeilen kurzer Länge zur Folge. Diese sollten ignoriert werden, da sich in ihnen mit hoher Wahrscheinlichkeit kein reiner Fließtext befindet, sondern vermutlich Schlagworte oder -zeilen. Diese sind dem Identifikationsprozess nicht zuträglich, da die verwendeten Worthäufigkeitslisten aus mehreren Millionen Sätzen bzw. Fließtexten erzeugt wurden. Kurze Zeilen verfälschen die Identifikation insbesondere dann, wenn Stopworte herangezogen werden, da diese dann deutlich unterrepräsentiert auftreten. Abb. 3.2 stellt die Zeichenlängenverteilung dar. Es zeigt sich, dass ein Drittel aller Zeilen nur sehr wenige Zeichen (0–5) beinhalten. Der von diesen Zeilen transportierte Inhalt ist entsprechend gering, wie an dem Anteil an allen Zeichen abzulesen ist. Aus diesem Grund wird im Verfahren ein Parameter bereitgestellt, der eine Zeilenselektion für den Identifikationsprozeß über eine Mindestzeilenlänge regelt. Aus der Abbildung wird ebenso der Hauptanteil der Zeichen im Bereich 70 Zeichen pro Zeile deutlich. Diese Zeilen können bereits Sätze kurzer Länge sein. Der Parameter sollte demnach nicht zu hoch angesetzt werden, um nicht zu viele Dokumentzeilen, also relevante Informationen, auszublenden.

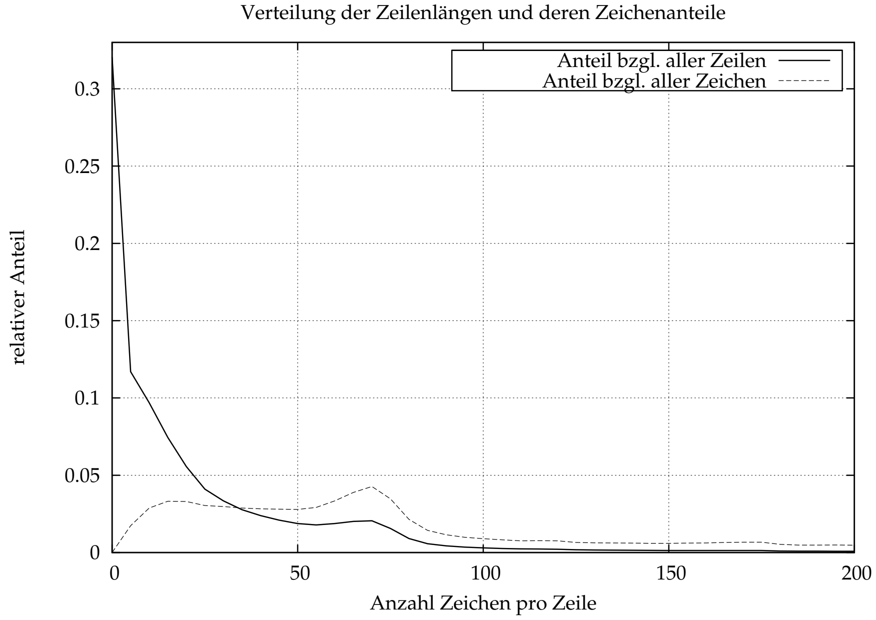


Abbildung 3.2: Verteilung der Zeilenlängen und deren jeweilige Anteile an allen vorhandenen Zeichen. Darstellung basiert auf 32 000 zufällig ausgewählten FindLinks-Dokumenten.

3.2. Sprachen

Für den Trainingsprozeß fungieren die Wortlisten als Basismerkmale der Sprachen. Deswegen sollte diesen besondere Aufmerksamkeit zukommen. Es fällt zunächst auf, dass einige der verwendeten Stopworte in syntaktisch gleicher Form mehreren der Worthäufigkeitslisten angehören. Das liegt zum Einen an ihrer Kürze, also den begrenzten kombinatorischen Möglichkeiten Wörter kurzer Länge zu bilden und zum Anderen an den verwandtschaftlichen Beziehungen einiger Sprachen zueinander. In Tab. 3.1 sind die Stopwörter aufgeführt, die in den Trainingsdaten den meisten Sprachen zugeschrieben werden. Diese Wörter scheinen demnach keine geeigneten Merkmalsträger für die Sprachklassifikation zu sein. Problematisch wird es insbesondere dann, wenn sich nahe Sprachen viele Stopwörter teilen. Die Trennung dieser nahen Sprachen ist ein wichtiges Teilziel, um die Dokumente sprachlich eindeutig zu klassifizieren. Dies ist notwendig, da nicht mehrere ähnliche Sprachen dem gleichen Dokument zugeschrieben werden sollten. Für die Feststellung der Nähe jeweils zweier Sprachen findet hier ein Ähnlichkeitsmaß f_{L_i,L_j} zwischen den zwei entsprechenden Wortlisten Anwendung:

Wort	Sprachenanzahl	einige Sprachvertreter mit Sprachcode
`a`	130	`eng,ita,hrv,pol,por,ron,spa,tur`
`na`	116	`aka,bos,ces,hrv,pol,xho`
`o`	95	`ceb,hsb,por,ron,sco,spa`
`e`	80	`csa,fri,ita,lat,por,spa`
`i`	77	`bos,cat,dan,nno,nob,swe`

Tabelle 3.1: Liste der Stopwörter, die in den meisten Sprachen vertreten sind mit einigen Sprachvertretern. Die Liste basiert auf den Trainingsdaten. Für jede Sprache wurden die häufigsten 100 Worte in Betracht gezogen.

$$f_{L_i,L_j} : \mathbb{N}^+ \longrightarrow [0,1], \quad k \longmapsto f_{L_i,L_j}(k) = \frac{\sum\limits_{w \in (B_k(L_i) \cap B_k(L_j))} H_{\text{abs}}(w, L_i)}{\sum\limits_{\bar{w} \in B_k(L_i)} H_{\text{abs}}(\bar{w}, L_i)} \tag{3.1}$$

Das Maß gibt darüber Aufschluß in welchem Anteil sich Worte der Sprache L_j in der eigenen Sprachliste, also L_i, niederschlagen. Ein Wert von 0 bedeutet dabei, dass es keine Übereinstimmungen zwischen den beiden Listen gibt, währenddessen der Wert 1 für vollständige Wortgleichheit steht, d.h. alle Worte von L_j kommen auch in L_i vor. Dieser Wert ist insbesondere dann relevant, wenn er hoch ist, denn dann besteht die Gefahr ungewollte zusätzliche Klassifikationen vorzunehmen. Aus der Tab. 3.2 können die Werte für f einiger ausgewählter Sprachpaare entnommen werden. Der Wert 0.577 zwischen `eng` und `sco` entspricht dabei der Wahrscheinlichkeit, dass ein Wort, welches einen Eintrag in der englischen Wortliste besitzt auch einen solchen in der schottischen hat. Vorraussetzung ist dabei, dass ein englischer Text vorliegt, der idealerweise dieselben relativen Worthäufigkeiten besitzt wie der Trainingstext. Umgekehrt gilt für einen

$f_{L_i,L_j}(30)$	eng	sco	deu	afr	nld	fra	spa	por
eng	1.0	**0.577**	0.0319	0.0956	0.071	0.069	0.096	0.119
sco	**0.517**	1.0	0.031	0.059	0.035	0.034	0.119	0.161
deu	0.062	0.072	1.0	**0.270**	0.110	0.0	0.0	0.015
afr	0.184	0.095	0.125	1.0	**0.557**	0.024	0.055	0.016
nld	0.136	0.05	0.108	**0.518**	1.0	0.155	0.189	0.089
fra	0.055	0.024	0.0	0.103	0.245	1.0	**0.450**	0.292
spa	0.020	0.123	0.0	0.095	0.225	0.304	1.0	**0.312**
por	0.037	0.168	0.038	0.011	0.142	0.330	**0.421**	1.0

Tabelle 3.2: Sprachüberlappungen gemäß dem Ähnlichkeitsmaß f_{L_i,L_j} einiger verwandter Sprachen bei $k = 30$ Stopwörtern. Die Maximalwerte jeder Zeile sind jeweils zur Hervorhebung *fett* markiert.

idealen schottischen Text, dass mit einer Wahrscheinlichkeit von 0.517 ein Stopwort in beiden Listen (`eng` und `sco`) auftaucht. Bei hohen Werten dieses Ähnlichkeitsmaßes kann somit durch die Sprachgemeinsamkeiten eine hohe relative Textabdeckung der zur Originalsprache des Textes ähnlichen Sprache erwartet werden. Da dies eine zusätzliche Klassifikation mit der ähnlichen Sprache `eng` für das Dokument bedeuten könnte, muss noch ein zusätzliches Kriterium gefunden werden, welches auf die Unterschiede verwandter Sprachen abzielt. Dies wird jedoch erst im nächsten Kapitel, im Unterbschnitt 4.3.1 vorgestellt.

Analog zum Ähnlichkeitsmaß f für Wörter, können die Ähnlichkeitswerte zweier Sprachen für n-Gramme ermittelt werden. Im Anhang A.2 sind diese Werte ebenso für Tri-, Bigramme aufgeführt. Des Weiteren wird der Parameter k variiert, um diese Dimension ebenfalls mit in Betracht zu ziehen. Aus den Listen ist ersichtlich, dass sich insbesondere Bi- und Trigramme für eine Überdeckungsbetrachtung, wie diese bei Stopworten benutzt wird, nicht eignen, da sich die Einträge der Listen zu ähnlich sind. Für Unigramme ist dies ebenfalls zu erwarten und aufgrunddessen könnten stattdessen einzigartige[16] Unigramme als besonderes Merkmal in die Klassifikation einfließen. Um den Rang der Bi- und Trigramme in den jeweiligen Sprachlisten mit einfließen zu lassen, werden hierfür verkettete Wahrscheinlichkeiten verwendet. Die Konkretisierung dieser Ansätze erfolgt ebenso im nächsten Kapitel.

3.3. Relative Textüberdeckung

Im Unterabschnitt 2.4.2 wurde die optimale relative Textüberdeckung u_{opt} auf Basis der Trainingsdaten eingeführt. Eine Möglichkeit Sprachklassifikation auf einem Dokument $d \in \mathfrak{D}$ durchzuführen, ist die optimale Überdeckung mit der im Dokument ermittelten Überdeckung für alle trainierten Sprachen zu vergleichen,

---

[16] Einzigartigkeit ist hier relativ zu sehen, da für die Feststellung lediglich die Worthäufigkeitslisten der Trainingssprachen herangezogen wurden. Es wird demnach angenommen, dass die Trainingsdaten gepflegt werden und alle Sprachen enthalten, die für eine Klassifikation in Frage kommen. Eine Liste der Sprachen, für die einzigartige Zeichen ermittelt werden konnten, ist im Anhang A.3 einsehbar.

um auf Grundlage dessen eine Sprachwahl zu treffen. $u_d(L, k)$ beschreibt nun die gemessene relative Textüberdeckung durch die Sprache $L \in \mathfrak{L}$ des Dokuments d bei Betrachtung der k höchstfrequenten Wörter der entsprechenden Wortliste. Berechnungsvorschrift:

$$u_d(L, k) = \frac{\sum\limits_{w \in B_k(L)} H_{\text{abs}}(w, d)}{\sum\limits_{\bar{w} \in A(d)} H_{\text{abs}}(\bar{w}, d)} \quad , \tag{3.2}$$

wobei $H_{\text{abs}}(w, d)$ die gemessene absolute Häufigkeit des Wortes w im Dokument d ist und $A(d)$ die Menge aller im Dokument erscheinenden Wortformen beschreibt. In Abb. 3.3 ist die Verteilung von u_d zweier Testquellen[17] mit je 13 000 Dokumenten für die Sprache `eng` visualisiert. Die Abbildung vermittelt sehr deutlich, dass die englischen Webseiten im Schnitt lediglich die Hälfte der optimalen Abdeckung erreichen. Die zur Wortliste gehörenden Trainingssätze der aspra20-Datenbank erreichen im Durchschnitt die erwartete optimale Überdeckung. Als Ursache für die starken Abweichungen der „Newspaperlinks" können die in Unterabschnitt 2.2.3 angesprochenen Herausforderungen und Schwierigkeiten bei Webseiten angebracht werden und im Zusammenhang damit, die in diesem Kapitel untersuchten hochfrequenten Dokumentenzeilen kurzer Länge. Zum Ande-

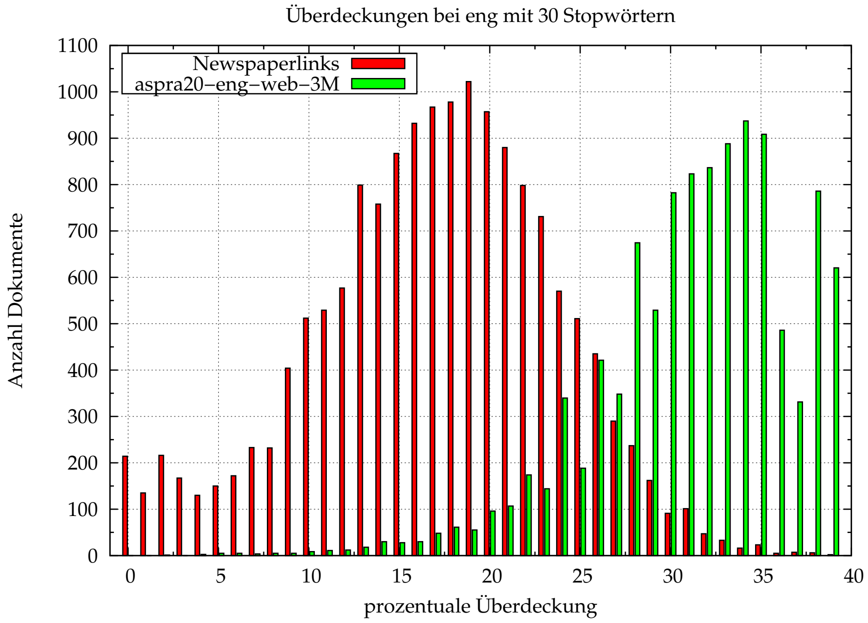


Abbildung 3.3: Verteilung der Stopwortabdeckung u_d für Englisch bei englischen Dokumenten. Es wurden je 13 000 Dokumente zweier Testquellen verwendet. $k = 30$ Stopwörter wurden gewählt und die zugehörige optimale Abdeckung $u_{\text{opt}}(\text{eng}, 30) = 0.317 \mathrel{\widehat{=}} 31.7\,\%$.

---

[17] Die beiden Testquellen sind zu Dokumenten zusammengefügte Sätze der aspra20-Datenbank und gecrawlte Webseiten der „Newspaperlinks". Eine nähere Beschreibung findet sich in Unterabschnitt 5.1.

ren besitzt die Testquelle relativ viele falsch ausgezeichnete Dokumente[18]. Dies hat zur Folge, dass sich viele Dokumente im Bereich 0–5 % finden.

---

[18] Dies konnte durch mehrere Stichproben festgestellt werden. Bspw. sind sehr viele Webseiten mit spanischem Inhalt als Englische ausgezeichnet.

4. Implementierung

In diesem Kapitel soll *LangSepa* als Sprachidentifizierer und dessen Implementierung vorgestellt werden. Hierzu gehe ich auf den Aufbau, die Arbeitsweise und die verwendeten Datenstrukturen ein. Der Kern des Werkzeugs ist das Verfahren zur Klassifikation der Textdokumente. Dieses wird im letzten Abschnitt ausführlich behandelt.

LangSepa wurde in Java 1.6 implementiert. Dies stellte die Kompatibilität zu einigen hilfreichen Java-Paketen her, die von Mitarbeitern der Abteilung für Automatische Sprachverarbeitung entwickelt wurden. Des Weiteren ist damit die Wiedervendbarkeit, als auch die Möglichkeit das Tool auszubauen, gesichert. Zusätzlich wurde hierfür detailliert nach den Konventionen von `JavaDoc` kommentiert. Als Entwicklungsumgebung wurde `eclipse` verwendet.

4.1. Aufbau und Ablauf des Programms

Der Aufbau von *LangSepa* folgt dem Schema der Klassifikation auf Grundlage von überwachtem maschinellem Lernen. Der Ablauf des Programms lässt sich somit grob in drei Phasen gliedern:

1. **Trainingsphase**: Import der Worthäufigkeitslisten aller vorhandenen Sprachen und erzeugen des Sprachmodells.
2. **Arbeitsphase**: Abarbeiten der Textdokumente (Laden, Textanalyse, Klassifikation, Ausgabe).
3. **Auswertungsphase**: Zusammenfassung der Resultate des Klassifikationsprozesses der Arbeitsphase.

Diese Phasen werden in den folgenden Unterabschnitten konkretisiert.

4.1.1. Trainingsphase

In einem ersten Schritt werden die Trainingsdaten, welche bereits in 2.3.3 vorgestellt wurden, von den entsprechenden Datenbanken geladen. Für jede Sprache werden die k höchstfrequenten Wörter aus der zugehörigen Wortliste ermittelt und in entsprechenden Klassen gespeichert.

Vor jedem Durchlauf des Programms findet ein neuer Import statt. Diese Variante wurde der lokalen Wortlistenspeicherung in Dateien vorgezogen, um aktuelle Änderungen und Korrekturen in den Datenbanken sofort in die Klassifikation mit einfließen zu lassen. Außerdem lassen sich Datenbanken neu hinzugefügter Sprachen oder Wortlisten verbesserter Datenbanken sofort für die Klassifikation verwenden, indem die Konfigurationsdatei `db-sources.ini` im Programmverzeichnis mit den entsprechenden zusätzlichen Einträgen versehen wird. Vorraussetzung hierfür ist, dass es sich wie bei den Trainingsdatenbanken um MySQL handelt und diese dasselbe Tabellenschema wie die aspra20-Datenbanken haben. Wenn dies nicht der Fall ist, müssten im Quellcode zusätzliche Schemata und/oder Schnittstellen definiert werden.

Beim Import der Wörter wird zunächst jedes Wort in alle möglichen n-Gramme der Größe $n \leq 3$ gemäß dem Beispiel aus Abschnitt 2.1.2 zerlegt. Dabei werden die absoluten Häufigkeiten der n-Gramme gesetzt und fortlaufend sprachabhängig inkrementiert. Anschließend errechnet das Programm die relativen Häufigkeiten der Wörter und generiert Suchstrukturen[19], um effizient Wörter und n-Gramme sprachun- und sprachabhängig finden zu können. Ein Sprachmodell jeder Sprache ist somit durch deren importierte Wörter und den generierten n-Gramme mit deren relativen bzw. absoluten Häufigkeiten gegeben.

4.1.2. Arbeitsphase

In dieser Phase wird die Hauptaufgabe des Programms realisiert. Alle Dokumente der Menge $\mathfrak{D}$ werden hier abgearbeitet. Jedes Dokument $d \in \mathfrak{D}$ durchläuft dabei die folgenden Teilprozesse in angegebener Reihenfolge:

1. d aus Datei **laden** und Text im Hauptspeicher ablegen
2. **Tokenisierung** des Textes in Worte
3. Sprach-**Analyse** der Tokens
4. Sprach-**Klassifikation** auf Basis der Analyse
5. d mit ermittelten Sprachcode(s) in Datei **ausgeben**

Die Umsetzung in *LangSepa* erfolgt durch das Architekturmuster „Pipes-and-Filters" [4, vgl. S. 54–71]. Filter sind die Komponenten, die Daten lesen, liefern, transformieren, ergänzen und verfeinern. Pipes stellen dabei die Datenkanäle zwischen den Filtern dar. Als Datenquelle fungieren hier die FindLinks-Dateien. Diese liefert die Daten dem ersten Filter. Pro Filter werden im Ablauf jedem Dokument zusätzliche Informationen aus den jeweils vorher verfügbaren bzw. generierten angefügt. Diese schrittweise Ergänzung erfolgt, bis alle nötigen Informationen in die Sprachklassifikation münden, nach der die zentrale Aufgabe abgeschlossen ist. Die Datensenke ist wiederum durch Dateien gegeben. [4, vgl. S. 56]

Jeder Teilprozess bzw. Filter ist in dieser Implementierung als Thread realisiert. Diese Threads agieren im Programm als aktive Filter, da sie sich selbständig die Dokumente aus der Pipe entnehmen und sie anschließend der Pipe wieder zur Weiterverarbeitung hinzufügen [4, S. 56]. Dadurch wird die parallele Verarbeitung mehrerer Dokumente ermöglicht. Insbesondere verstreicht keine CPU-Zeit durch den aufwändigen Prozeß beim Laden und Schreiben der Dateien. Des Weiteren ist eine Überlastung des Hauptspeichers ausgeschlossen, da die Dokumente direkt nach Fertigstellung in die Datensenke münden und somit nicht brach liegen und diesen unnötig füllen. Aus Entwicklerperspektive besitzt diese Architektur hohe Flexibilität, da einzelne Filterkomponenten einfach gegen andere ausgetauscht werden können. Wenn zusätzliche Filter zwischengeschaltet werden, so ist auch eine Wiederverwendbarkeit der vorhandenen Komponenten in einer neuen Filteranordnung gegeben [4, S. 69–70].

---

[19] Näheres hierzu ist in 4.2 erläutert.

Die Umsetzung der Tokenisierungsaufgabe ist durch einen umgebauten Satzsegmentierer der Abteilung für Automatische Sprachverarbeitung im Programm realisiert. Hierfür wurden mir Java-Pakete zur Verfügung gestellt. Eine Anpassung ist durch Hinzufügen entsprechender Trennungssymbole, die nicht Teil der Tokens, also der Wörter, sein sollen, durchgeführt worden. Die Bearbeitung und Speicherung der Wort-Tokens erfolgt über die javainternen InMemory-Streams.

Der Teilprozess Sprachanalyse erfolgt auf Basis der generierten Tokens. Diese werden an dieser Stelle vollständig und sequentiell überprüft. Dabei testet das Programm das Vorkommen jedes Wortes in der kompletten Wortliste, die die Wörter aller Sprachen enthält. Jedem der bekannten Worte wird je ein Zähler zugewiesen, der die absoluten Häufigkeiten der Auftreten im Text repräsentiert. Parallel zum Zählen werden Trigramm und Unigrammsequenzen generiert. Auf Grundlage dieser Informationen lassen sich die in 2.4.2 und 2.4.3 vorgestellten Maße ermitteln, die für das Sprachklassifikationsverfahren dieser Arbeit relevant sind.

Aufgrund der inhaltlichen Fülle des Klassifikationsverfahrens, wird dies in dem selbständigen Abschnitt 4.3 dargestellt.

4.1.3. Auswertungsphase

Zum Abschluss eines Programmdurchlaufs fasst *LangSepa* die Ergebnisse der Klassifikation zusammen. Zu diesen zählen die absoluten und relativen Anteile der zugeordneten Sprachen, die Sprachklassifikation in Abhängigkeit von der Dokumentenkodierung, dem FindLinks-Nutzer und dem jeweiligen Crawlzeitpunkt.

4.2. Datenstrukturen

Zu den verschiedenen Aufgaben der Phasen und Teilprozesse sind spezielle Datenstrukturen zur Verarbeitung der Daten gewählt worden. Einige von diesen werden eher selten angewandt. Aus diesem Grund möchte ich an dieser Stelle die wichtigsten und im Programm häufig aufgerufenen und gebrauchten vorstellen und deren Einsatz begründen.

In der Trainingsphase werden die Sprachen und deren zugehörige Stopwörter aus den Wortlisten ins Programm geladen. Für die spätere Verarbeitung der Dokumente müssen die Wörter und n-Gramme effizient gefunden werden können. Außerdem müssen im Klassifikationsverfahren die zugehörigen Sprachen, zugehörige Ränge, sowie die absoluten und relativen Häufigkeiten erreichbar sein. Da es sich lediglich um Einfüge- und Suchoperationen handelt, fiel die Wahl auf die javainternen Hashverfahren. Es sind maximal $k \cdot M$ Wörter[20]. Aufgrund der Tatsache, dass sich Hashverfahren im Durchschnitt effizienter als Schlüsselvergleichsverfahren verhalten, da sie eine Zeitkomplexität für die Suche haben, die unabhängig von der Anzahl der vorhandenen Schlüssel ist [14, vgl. S .183–184], wurde diese Wahl getroffen. Die aufwändigen Operationen, die entstehen, wenn Hashtabellen aufgrund des Füllfaktors vergrößert werden müssen [15, vgl.

---

[20] Bei $k = 100$ Stopwörtern pro Sprache und $M = 370$ Sprachen sind dies schon 37 000 mögliche Wörter, ohne Berücksichtigung identischer Stopwörter verschiedener Sprachen.

S. 73], können umgangen werden, da die maximale Größe aufgrund der gegebenen Sprachen und der begrenzten Anzahl der Stopworte pro Sprache bekannt ist. Die Kapizität der Hashtabelle kann somit zu Beginn geeignet[21] gewählt werden. In *LangSepa* wurden die genannten Suchstrukturen über die Klassen `HashSet` und `HashMap` implementiert. Wörter und n-Gramme mit $n \leq 3$ können auf Vorkommen in den importierten Listen durch die `HashSet`s überprüft werden. Die `HashMap`s realisieren die Zuordnung der Zeichenketten zu den entsprechenden Objekten, in denen die Eigenschaften wie Rang und Häufigkeit der zugehörigen Sprachen abgerufen werden können.

In der Dokumentenanalyse werden Zähler verwandt, die über Baumstrukturen im Programm verankert sind. Diese erhielten an dieser Stelle den Vorzug gegenüber Hashing, da neben dem Suchen und Einfügen auch das effiziente Sortieren eine Teilaufgabe des Analyseprozesses ist. Hier wurde somit ein Kompromiss zwischen hinreichend schnellem Einfügen neuer Zeichenketten und Aufwand für die Sortierung getroffen [18, vgl. S. 684–686]. Mit den sortierten Dokumentzähllisten für Wörter und n-Gramme kann die Möglichkeit offen gehalten werden, die dokumenteninternen Ränge dieser Strukturen im Klassifikationsverfahren mit einfließen zu lassen. Die javainterne Klasse `TreeMap` wurde für die Wortzählung verwendet.

Die Datenkanäle zwischen den Threads, also die Pipes, halten die Dokumente vor, auf die die Filter bzw. Threads zugreifen. Die Pipes sind hier durch Warteschlangen der Klasse `ConcurrentLinkedQueue` umgesetzt. Durch eine Datenstruktur basierend auf einer verketteten Liste ist es somit sehr schnell möglich ein Element am Ende der Liste anzufügen und der Liste das erste Element zu entnehmen. Des Weiteren verhält sich diese Klasse javaseitig threadsicher durch eine entsprechende Handhabung der Zugriffe durch mehrere Threads [18, vgl. S. 683–684]. Um die Möglichkeit auszuschließen, dass der Hauptspeicher von zu vielen Dokumenten überflutet wird, wurde eine maximale Anzahl an Dokumenten festgesetzt, die pro Warteschlange vorgehalten werden darf. Bei maximaler Füllung muss der vorhergehende Thread pausieren bis wieder mindestens ein freier Platz zur Verfügung steht. Ist umgekehrt kein Dokument in der Warteschlange, so muss der darauffolgende Thread pausieren und abwarten bis ein neues Dokument in der Warteschlange zur Verfügung gestellt wird.

4.3. Klassifikationsverfahren

Der Entwicklung des Algorithmus', der jedem Dokument $d \in \mathfrak{D}$ keine, eine oder mehrere Sprachen zuweist, gingen die Beobachtungen aus Kapitel 3 als auch Ergebnisse, die auf vielen kleinen Testdatensätzen im Verlauf der Implementierung

---

[21] Eine geeignete Wahl ist gegeben, wenn ein guter Kompromiss zwischen benötigtem Speicherplatz und Suchdauer für ein Element gegeben ist. Das wird javaintern über einen maximalen Füllfaktor der Hashtabelle von $0{,}75$ definiert [18, vgl. S. 694–697] und im Programm entsprechend auf die maximale Größe der Hashtabelle umgemünzt. Die maximale Größe für die Suchliste für Wörter ist demnach $k \cdot M/0{,}75$. Für n-Gramme muss diese Größe abgeschätzt werden. Hierbei wird mit einer durchschnittlichen Länge von 4 Buchstaben pro Wort gerechnet. Somit 4 Trigrammen und 5 Bigrammen pro Wort, was eine Vervierfachung bzw. Verfünffachung der Maximalgröße der Wortsuchliste für die Tri- bzw. Bigrammlisten heißen würde.

ermittelt worden sind. Da eine Auflistung aller experimentellen Vortests im Detail zu einer Übersättigung dieses Abschnitts führen würde, werden stattdessen an Ort und Stelle gemachte Erfahrungen kurz dargestellt und zur Argumentation für eingesetzte Techniken herangezogen.

Ein erster Überblick über den gesamten Prozess der Klassifikation ist in Abb. 4.1 durch eine Pseudocode-Darstellung vom Algorithmus gegeben. Es ist zunächst zu beachten, dass noch nicht alle Parameter, welche im Aufruf der Hauptfunktion vorkommen, eingeführt worden sind. Des Weiteren sei darauf hingewiesen, dass die drei aufgerufenen Funktionen StopwortKlassifikation-, Trigramm- und UnigrammKlassifikation jeweils Teilmengen von $\mathfrak{L}$ bestimmen und zurückgeben. Eine Darstellung der Parameter und ausführliche Erklärungen der eben genannten Funktionen können in den nachfolgenden Unterabschnitten eingesehen werden.

```
SprachKlassifikation($d, k, \alpha, \beta, \gamma, \delta, w, \mu, \lambda, \kappa$)
$\ell(d) \leftarrow \{\}$
$\ell_S(d) \leftarrow$ StopwortKlassifikation($d, k, \alpha, \beta, \gamma, \delta$)
if $\ell_S(d) = \{\}$ then
    $\ell_T(d) \leftarrow$ TrigrammKlassifikation($d, k, w, \mu, \lambda$)
    $\ell(d) \leftarrow \ell(d) \cup \ell_T(d)$
else
    $\ell(d) \leftarrow \ell(d) \cup \ell_S(d)$
end if
$\ell_U(d) \leftarrow$ UnigrammKlassifikation($d, k, w, \mu, \lambda, \kappa$)
$\ell(d) \leftarrow \ell(d) \cup \ell_U(d)$
return $\ell(d)$
```

Abbildung 4.1: Pseudocode des Klassifikationsalgorithmus, der die Labelfunktion ℓ berechnet und ausgibt. Ein Dokument d wird unter der Verwendung weiterer Parameter einer Teilmenge von $\mathfrak{L}$ zugeordnet.

Der Aufbau des Algorithmus ist folgendermaßen zu erklären: Zunächst kommt ein Kriterium das Stopwortüberdeckungen im Text mit den optimalen der jeweiligen Sprachen vergleicht, zum Tragen. Wenn keine Sprache gegebene Kriterien erfüllt, so handelt es sich um einen Text vieler Sprachen, einen Text mit wenigen Stopwörtern oder tatsächlich um unbrauchbares Material, dem keine Sprache zugeordnet werden sollte. In diesem Fall wird das Trigrammverfahren, welches auf Wahrscheinlichkeitsberechnungen basiert, initiiert und die herausgefundenen Sprachen dem Dokument zugeordnet. Im anderen Falle werden lediglich die Sprachen der Stopwort-Klassifikation verwendet. Somit ist das rechenaufwändigere Trigrammverfahren nur im Falle der Nullklassifikation durch die Stopwortmethode notwendig. Des Weiteren ist es erforderlich auf jedem Dokument eine Unigrammuntersuchung durchzuführen, da für 9 der 351 Sprachen lediglich „Wörter" mit einzelnen Zeichen und sehr wenige mit zwei oder mehr Zeichen in den Wortlisten zur Verfügung stehen[22]. Das sind die Sprachen, in denen keine

---

[22] Die Funktionen Stopwort- und TrigrammKlassifikation würden dadurch in jedem Dokument einer dieser 9 Sprachen die leere Menge zurückgeben.

Worttrennungssymbole benutzt werden[23]. Eine Überprüfung der Unigramme erfolgt nur für die genannten 9 Sprachen, um die Zeitkomplexität nicht unnötig zu erhöhen.

4.3.1. Stopwortüberdeckung

Die ersten Merkmale, die nach der Textanalyse überprüft werden, sind die Stopwörter. Der Text wird auf Übereinstimmung mit der erwarteten relativen Textüberdeckungen aller Sprachen abgeglichen. Ein erster Überblick über das Vorgehen wird in Abb. 4.2 gegeben. Die Funktion SprachKandidaten(d, k, α) gibt dabei alle Sprachen $L_j \in \mathfrak{L}$ aus, die das folgende Kriterium erfüllen:

```
StopwortKlassifikation($d, k, \alpha, \beta, \gamma, \delta$)
$\ell_{\mathrm{S}}(d) \leftarrow \{\}$
$\ell_{\mathrm{K1}} \leftarrow$ SprachKandidaten($d, k, \alpha$)
if $\ell_{\mathrm{K1}} = \{\}$ then
    return $\ell_{\mathrm{S}}(d)$
else
    $\ell_{\mathrm{S}}(d) \leftarrow$ WähleAusgeprägteSprachen($d, k, \{\}, \ell_{\mathrm{K1}}, \gamma, \delta$)
end if
$\ell_{\mathrm{K2}} \leftarrow$ SprachenImIntervall($d, k, \alpha, \beta$)
$\ell_{\mathrm{S}}(d) \leftarrow \ell_{\mathrm{S}}(d) \cup$ WähleAusgeprägteSprachen($d, k, \ell_{\mathrm{S}}(d), \ell_{\mathrm{K2}}, \gamma, \delta$)
return $\ell_{\mathrm{S}}(d)$
```

Abbildung 4.2: Pseudocode für die Sprachklassifikation durch das Kriterium der relativen Textüberdeckung durch Stopworte.

$$\frac{|u_{\mathrm{opt}}(L_j, k) - u_d(L_j, k)|}{u_{\mathrm{opt}}(L_j, k)} \leq \alpha \qquad j \in \{1, ..., M\} \quad , \tag{4.1}$$

wobei $\alpha \in [0, 1]$ die relative Überdeckungsabweichung darstellt. Es kommen demnach die Sprachen in die erste nähere Auswahl, die keine stärkere relative Abweichung als α von ihrer Optimalüberdeckung für den betrachteten Text besitzen. Durch diese Konstruktion der Auswahl werden keine Sprachen benachteiligt, die durch starke Flexion o.ä. eine relativ geringe optimale Überdeckung besitzen.

Etwas später wird die Funktion SprachenImIntervall(d, k, α, β) aufgerufen. Diese verhält sich prinzipiell wie die eben vorgestellte, mit dem Unterschied, dass nun alle Sprachen ausgewählt werden, die sich im relativen Abweichungsintervall $(\alpha, \beta]$ befinden. Etwas formaler müssen diese Sprachen das Kriterium

$$\alpha < \frac{|u_{\mathrm{opt}}(L_j, k) - u_d(L_j, k)|}{u_{\mathrm{opt}}(L_j, k)} \leq \beta \qquad j \in \{1, ..., M\} \tag{4.2}$$

erfüllen. Mit $\beta \in (\alpha, 1]$. Dies wurde eingeführt, damit Sprachen, die Fließtext in einem mehrsprachigen Dokument enthalten, berücksichtigt werden, da sie evtl.

---

[23] Die erwähnten Sprachen sind: `amh`, `bod`, `cmn`, `dzo`, `iii`, `jpn`, `lao`, `tha`, `zho`.

aufgrund des geringen Anteils am Gesamttext a-priori keine hinreichend große Überdeckung erzielen können.

Für den Fall, dass mehrere Sprachen als Kandidaten für das Dokument vorgeschlagen werden, wird eine Sprachauswahl getroffen. WÄHLEAUSGEPRÄGTESPRACHEN$(d, k, \{\}, L_{\mathrm{K1}}, \gamma, \delta)$ wählt insbesondere zwischen ähnlichen Sprachen diejenige, die die stärkste Ausprägung im Dokument besitzt. Wichtigste Entscheidungsgrundlage liefert die Funktion $g(d, k, L_i, L_j)$, die folgendermaßen berechnet wird:

$$g(d, k, L_i, L_j) = \frac{\sum\limits_{w \in A(d) \cap B_k(L_i) \cap B_k(L_j)} H_{\mathrm{abs}}(w, d)}{\sum\limits_{\bar{w} \in A(d) \cap B_k(L_i)} H_{\mathrm{abs}}(\bar{w}, d)} \in [0, 1] \quad . \tag{4.3}$$

Diese gibt an, welcher Anteil der gemessenen relativen Stopwortüberdeckung der Sprache L_i, durch gemeinsame Wörter der Sprachen L_i und L_j in L_i induziert wird. Ist dieser Wert nahe oder gleich 1, so sind fast alle oder alle Wörter, die zur gemessenen Überdeckung $u_d(L_i, k)$ führen, aus L_j. In diesem Fall müsste L_i aus der Kandidatenliste entfernt werden, da lediglich die Ähnlichkeit zur anderen Sprache, diese in die Kandidatenliste gebracht hat. Der Algorithmus, der zur Auswahl führt, ist in Abb. 4.3 dargestellt. Das Auswahlverfahren baut sich darauf auf, dass jeweils eine Menge ℓ_{S1} zur Verfügung steht, in der die aktuell ausgewählten Sprachen vorliegen. Der Kandidatensprachmenge ℓ_{K} wird sukzessive immer ein Element L entnommen, welches gegen alle Sprachen der Menge ℓ_{S1} antreten muss. Bei eindeutigem Ergebnis zu Gunsten von L wird dieses Element der Auswahl hinzugefügt und die Verlierersprache L' aus ℓ_{S1}, wird aus dieser entfernt. Für den Fall, dass bei allen Sprachen aus ℓ_{S1} jeweils die Sprache L und die Sprache L' der Auswahlmenge ℓ_{S1} gewählt werden, wird L der Auswahlliste ebenso hinzugefügt. Als zusätzlich Bedingung fungiert hierbei der Parameter $\delta \in \mathbb{N}^+$. Dieser gibt die mindestens notwendige absolute Zahl der eigenen Wörter, die die Sprache L gegenüber jeder Sprache der Auswahlmenge haben muss. Dies verhindert, dass Sprachen, die sehr wenige eigene Wörter zur Textüberdeckung einbringen klassifiziert werden.

Wie ein Sprachduell ausgetragen wird, kann in Abb. 4.4 anhand der Funktionsdarstellung von SPRACHWAHL eingesehen werden. Mit dem Parameter $\gamma \in [0, 1]$ ist es möglich festzulegen, ab welchem Anteil fremdsprachlich induzierter Überdeckung die Sprache nicht ausgewählt werden sollte. Unterschreitet $g(d, k, L_i, L_j)$ den Grenzwert γ, so wird L_i der Auswahlmenge hinzugefügt, vorausgesetzt, dass das δ-Kriterium ebenfalls erfüllt ist. Mit γ kann somit geregelt werden, wie stark ähnliche Sprachen voneinander getrennt werden. Ein niedrigerer Wert für γ sorgt für ein schärferes Vorgehen gegen ähnliche Sprachen. Der Wert darf jedoch auch nicht zu niedrig gewählt werden, damit nicht geringfügige Ähnlichkeiten zum Sprachenausschluss führen. Wird ein zu hoher Wert gewählt, so besteht die Gefahr verwandte Sprachen dem gleichen Dokument zuzuschreiben.

4.3.2. Trigrammwahrscheinlichkeit

Mit dem Trigrammverfahren wird ein zusätzliches Spracherkennungsverfahren implementiert. Hierbei ist eine Fenstertechnik implementiert worden, die den

```
WÄHLEAUSGEPRÄGTESPRACHEN($d, k, \ell_{\mathrm{S}}(d), \ell_{\mathrm{K}}, \gamma, \delta$)
$\ell_{\mathrm{S1}} \leftarrow \ell_{\mathrm{S}}(d)$
if $|\ell_{\mathrm{K}}| > 1$ then
    while $\ell_{\mathrm{K}} \neq \{\}$ do
        if $\ell_{\mathrm{S1}} = \{\}$ then
            $L \leftarrow$ wähle $\bar{L} \in \ell_{\mathrm{K}}$ beliebig
            $L' \leftarrow$ wähle $\bar{L} \in \ell_{\mathrm{K}} \setminus \{L\}$ beliebig
            $\ell_{\mathrm{S1}} \leftarrow \ell_{\mathrm{S1}} \cup$ SPRACHWAHL($d, k, L, L', \gamma$)
            $\ell_{\mathrm{K}} \leftarrow \ell_{\mathrm{K}} \setminus \{L, L'\}$
        else
            $L \leftarrow$ wähle $\bar{L} \in \ell_{\mathrm{K}}$ beliebig
            for each $L' \in \ell_{\mathrm{S1}}$ do
                if $L \notin$ SPRACHWAHL($d, k, L, L', \gamma$) then
                    break
                end if
                if $L' \notin$ SPRACHWAHL($d, k, L, L', \gamma$) then
                    $\ell_{\mathrm{S1}} \leftarrow (\ell_{\mathrm{S1}} \setminus \{L'\}) \cup \{L\}$
                    break
                end if
                if EIGENEWORTE($d, k, L, L'$)$\geq \delta$ then
                    if ISTLETZTESINFOREACHSCHLEIFE($L'$) then
                        $\ell_{\mathrm{S1}} \leftarrow \ell_{\mathrm{S1}} \cup \{L\}$
                    end if
                else
                    break
                end if
            end for
            $\ell_{\mathrm{K}} \leftarrow \ell_{\mathrm{K}} \setminus \{L\}$
        end if
    end while
else
    $\ell_{\mathrm{S1}} \leftarrow \ell_{\mathrm{K}}$
end if
return $\ell_{\mathrm{S1}}$
```

Abbildung 4.3: Pseudocode der Funktion WÄHLEAUSGEPRÄGTESPRACHEN.

```
SPRACHWAHL($d, k, L, L', \gamma$)
$g_L \leftarrow g(d, k, L, L')$
$g_{L'} \leftarrow g(d, k, L', L)$
if $(g_L \geq \gamma) \wedge (g_{L'} \geq \gamma)$ then
    if $g_L \leq g_{L'}$ then
        return $\{L'\}$
    else
        return $\{L\}$
    end if
else
    if $(g_L \geq \gamma) \wedge (g_{L'} < \gamma)$ then
        return $\{L'\}$
    end if
    if $(g_L < \gamma) \wedge (g_{L'} \geq \gamma)$ then
        return $\{L\}$
    end if
    if $(g_L < \gamma) \wedge (g_{L'} < \gamma)$ then
        return $\{L, L'\}$
    end if
end if
```

Abbildung 4.4: Pseudocode der Funktion SPRACHWAHL, die eine oder beide Sprachen des Sprachduells zurückgibt. $\gamma \in [0, 1]$ stellt den Grenzwert dar, ab dem fremdsprachlich induzierter Anteil der Überdeckung zum Ausschluss der betrachteten Sprache führt.

Text schrittweise auf Sprachen analysiert[24]. Da die Trigramm-Klassifikation nur dann aufgerufen wird, wenn das Stopwortverfahren fehl schlägt, muss wie bereits erwähnt, ein mehrsprachiges, ein nicht klassifizierbares oder ein stopwortarmes Dokument vorliegen.

Das Vorgehen ist schematisch anhand von Pseudocode in Abb. 4.5 nachvollziehbar. Das Dokument d besteht aus einer Folge von Trigrammen $t_1 t_2 ... t_e$. In jedem Schritt betrachtet *LangSepa* eine Teilfolge von w Trigrammen. Diese werden für je eine Sprachanalyse herangezogen. Es wird angenommen, dass diese Folge einsprachig ist oder ungeeignetes Material für eine spätere Weiterverarbeitung enthält. Als Kriterien für die Sprachwahl für einen Trigramm Bereich der Länge w wird zunächst die Parameter μ herangezogen. Der Parameter μ stellt eine Bedingung an L_{I}. Diese besagt, dass mindestens μ der vorkommenden Trigramme in der Trigrammliste von L_{I} vorkommen müssen, um sie als Sprachkandidat geltend machen zu können.

```
TRIGRAMMKLASSIFIKATION($d, k, w, \mu, \lambda$)
$i \leftarrow 1$
$count[?] \leftarrow 0$
for each $L \in \mathfrak{L}$ do
    $count[L] \leftarrow 0$
end for
while $i + w \leq e$ do
    $S \leftarrow t_i t_{i+1} ... t_{i+w}$
    $L_{\mathrm{I}} \leftarrow$ BERECHNEWAHRSCHEINLICHKEITEN($S, k$)
    if EIGENETRIGRAMME($L_{\mathrm{I}}, S$) $\geq \mu$ then
        $count[L_{\mathrm{I}}] \leftarrow count[L_{\mathrm{I}}] + 1$
    else
        $count[?] \leftarrow count[?] + 1$
    end if
    $i \leftarrow i + w + 1$
end while
return BESTIMMESPRACHEN($count[\,], \lambda$)
```

Abbildung 4.5: Pseudecode der Klassifikation durch Trigramme. Parameter w ist die Fensterbreite pro Iterationsschritt, μ die Mindestzahl eigener Trigramme, um in einem Schritt in der Kandidatenliste den Zähler hochgesetzt zu bekommen und λ der minimale Zählerstand, den eine Sprache haben muss, um klassifiziert zu werden. Die Sprache „?" entspricht der Zuordnung keiner Sprache bzw. unbrauchbarem Material.

Durch die Funktion BESTIMMESPRACHEN, einsehbar in Abb. 4.6, werden aus den wahrscheinlichsten Sprachen der Textfenster diejenigen ausgewählt, die mindestens λ mal als wahrscheinlichste Sprache aus den einzelnen Bereichen hervorgegangen sind. Dies macht es möglich Sprachen zu entfernen, die sehr selten im Gesamttext als wahrscheinlichste Sprache gewählt wurden.

---

[24] Die Idee hierfür stammt aus [13].

```
BESTIMMESPRACHEN($count[\,], \lambda$)
$\ell_{\mathrm{T}}(d) \leftarrow \{\}$
for each $L \in \mathfrak{L} \cup \{?\}$ do
    if $count(L) \geq \lambda$ then
        $\ell_{\mathrm{T}}(d) \leftarrow \ell_{\mathrm{T}}(d) \cup \{L\}$
    end if
end for
return $\ell_{\mathrm{T}}(d)$
```

Abbildung 4.6: Pseudocode der Klassifikationsentscheidung auf Basis der Trigrammiteration über den Text. Parameter λ entspricht dem Mindestzählerstand einer Sprache, um klassifiziert zu werden.

4.3.3. Unigrammwahrscheinlichkeiten ausgewählter Sprachen

Das Unigrammverfahren wurde als ein erster Versuch zur Identifikation von Sprachen, die keine Worttrennersymbole besitzen, in *LangSepa* eingebaut. Zurückzuführen ist dies auf die Wortlisten dieser Sprachen. Da die zugehörigen Topwörter aus einzelnen oder maximal drei Zeichen bestehen, ist ein Verfahren auf Bi- oder Trigrammbasis zum Scheitern verurteilt. Insbesondere dadurch, dass in den n-Grammen mit $n > 1$ die Anfänge und Enden durch Unterstriche ausgezeichnet sind, ist die Erkennung von langen Zeichenfolgen stark eingeschränkt. Als Konsequenz wurde somit ein Unigrammverfahren implementiert, welches so wie das Trigrammverfahren auf der Berechnung von verketteten Wahrscheinlichkeiten von Teilfolgen der Zeichenfolge $u_1 u_2 ... u_b$ des Dokuments d aufbaut. In der Auswertung werden jedoch nur Sprachen berücksichtigt, die durch die vorherigen Verfahren nicht abgedeckt werden konnten. Das sind gerade die Sprachen, die keine Worttrenner haben. $\tilde{\mathfrak{L}}$ bezeichne die Menge dieser Sprachen.

Abb. 4.7 gibt einen Überblick zum Unigrammverfahren. Die Ideen hierfür stammen alle aus dem Algorithmus zur Klassifikation durch Trigramme aus dem vorherigen Abschnitt. Zunächst wird das Dokument auf Vorkommen von Unigrammen aus mindestens einer der Sprachen aus $\tilde{\mathfrak{L}}$ überprüft. Das geschieht durch die Funktion ENTHÄLTSPRACHENZEICHEN($d, \tilde{\mathfrak{L}}, \kappa$). Wenn mindestens κ Zeichen der Trainingsdaten zu $\tilde{\mathfrak{L}}$ auch im Dokumententext erscheinen, so liefert diese Funktion ein `true` zurück und die Fenstertechnik aus 4.3.2 wird nun auf die Unigramme angewandt. Die Parameter für Fensterbreite w, Minimum eigener Unigramme μ pro Fenster und Mindestanzahl λ, die eine Sprache als Wahrscheinlichste von Folgen bestimmt werden muss, sind die gleichen wie im Trigrammalgorithmus (siehe Abb. 4.5). Die Wahrscheinlichkeiten P_u für Unigrammfolgen errechnen sich analog zu den Trigrammwahrscheinlichkeiten P, welche ausführlich in 2.4.3 eingeführt wurden. Nach der Überprüfung des gesamten Dokuments extrahiert BESTIMMESPRACHEN($count[\,], \lambda$) (siehe Abb. 4.6) die Sprachen aus $\tilde{\mathfrak{L}}$, die dem Dokument zugeordnet werden.

Dieses Verfahren ist eine erste Lösung zur Begegnung des angesprochenen Problems. Es könnte durch verbesserte Wortlisten dieser Sprachen oder ein spezielles n-Gramm-Verfahren abgelöst werden, welches Wortanfänge und -enden keine Bedeutung zuordnet. In [12, S. 45–46] wird angesprochen, dass Bigrammverfahren für derartige Sprachen sehr erfolgreich eingesetzt werden.

```
UnigrammKlassifikation($d, k, w, \mu, \lambda, \kappa$)
$i \leftarrow 1$
for each $L \in \tilde{\mathfrak{L}}$ do
    $count[L] \leftarrow 0$
end for
if EnthältSprachenZeichen($d, \tilde{\mathfrak{L}}, \kappa$) then
    while $i + w \leq b$ do
        $S \leftarrow u_i u_{i+1} ... u_{i+w}$
        $L_{\mathrm{I}} \leftarrow$ BerechneWahrscheinlichkeiten($S, k$)
        if $L_{\mathrm{I}} \in \tilde{\mathfrak{L}}$ then
            if EigeneUnigramme($L_{\mathrm{I}}, S$) $\geq \mu$ then
                $count[L_{\mathrm{I}}] \leftarrow count[L_{\mathrm{I}}] + 1$
            end if
        end if
        $i \leftarrow i + w + 1$
    end while
end if
return BestimmeSprachen($count[\,], \lambda$)
```

Abbildung 4.7: Pseudocode der Klassifikation durch Unigrammwahrscheinlichkeiten für alle Sprachen $\tilde{\mathfrak{L}}$, die das Leerzeichen nicht als Worttrenner benutzen. Die Parameter w, μ, λ sind dem Trigrammverfahren entnommen. κ ist die benötigte Mindestzahl an Zeichen der Sprachen aus $\tilde{\mathfrak{L}}$ pro Dokument, um diesen Algorithmus zu initiieren.

5. Ergebnisse

In diesem Kapitel soll das entwickelte Sprachidentifikationswerkzeug *LangSepa* hauptsächlich evaluiert werden. Hierzu werden zunächst die Quellen und deren zugehörige Dokumente vorgestellt, auf denen die Qualität des Verfahrens getestet wird. Die verwendeten Qualitätsmaße, die aus dem Information Retrieval stammen, schließen sich an. Eine Evaluation erfolgt über allen Sprachen. Erste Ergebnisse zu FindLinks-Dokumenten werden abschließend präsentiert.

5.1. Testquellen

Zur Überprüfung der Güte des Verfahrens wurden zwei Testquellen verwendet. Die Sprachen der Dokumente dieser Quellen sind als Referenz gegeben. Diese dient als Vergleichswert zu der Sprache bzw. den Sprachen, die auf Basis des Klassifikationsergebnisses $\ell(d)$ ermittelt wurden.

aspra-20 Dokumente Die erste Testquelle besteht aus Dokumenten, die durch aneinanderfügen von Sätzen der bereits in 2.3.3 erwähnten aspra-20 Datenbank der Abteilung für Automatische Sprachverarbeitung gebildet werden. Hierfür stehen Sätze der 60 Sprachen zur Verfügung, für die auch die zugehörigen Wortlisten importiert wurden. Die Trainingsdaten, also die Worthäufigkeitslisten, wurden auf Basis der zur Verfügung stehenden Sätze erstellt. Für einige der Sprachen können die Herkunftsorte der Sätze gewählt werden. Zur Auswahl stehen u.a. Wikipedia-Dumps, Zeitschriften- und Webseitentexte. Um die Nähe zur Aufgabe zu wahren, wurden bevorzugt die Quellen der Sprachen gewählt, die aus dem Web extrahiert worden sind. Allerdings muss festgehalten werden, dass die angesprochenen Charakteristika und Probleme bei Webseiten 2.2.3 durch Datensäuberung[25] in den Datenbanken nur sehr rudimentär ausgebildet sind.

Um das Verfahren in Abhängigkeit von Dokumentenlängen zu testen, ist es möglich die Satzanzahl pro Dokument frei zu wählen.

Nachrichtenseiten Für einen Test über realen Webseiten wurde eine Liste von Links mit zugehörigen Sprachen über die Seite `http://abyznewslinks.com/` bezogen. Die Liste enthielt ca. 33 000 Seiten, die in einem Prozess, der dem FindLinks-Schema (siehe 2.3.2) zum Crawling ähnelt, heruntergeladen und lokal abgespeichert worden. Um dies zu realisieren, wurde eine Kodierungsdetektion durchgeführt und eine anschließende Entfernung von HTML-Code durch HTML-2Text. Diese Textdokumente repräsentieren die Webseiten, die durch FindLinks gecrawlt werden, um einiges besser als die der Datenbanken. Leider musste in mehreren Stichproben festgestellt werden, dass die Sprachreferenzwerte oftmals nicht mit dem tatsächlichen Inhalt übereinstimmten. Das führt demnach zu einer geringeren Aussagekraft der ermittelten Qualitätsmaße. Die Anteile der einzelnen Sprachen an allen Seiten sind unterschiedlich groß. In der nachfolgenden

---

[25] Hierbei werden u.a. Sätze entfernt, die Sonderzeichen, aufeinanderfolgende Leerzeichen, zu viele gleiche Zeichen oder zu viele Satzendezeichen enthalten.

Tabelle sind die meisten Sprachen und deren Dokumentenanzahl in dieser Dokumentenkollektion aufgeführt.

Code	Doks.	Code	Doks.	Code	Doks.
`eng`	15 114	`nor`	250	`ita`	139
`spa`	2 357	`swe`	220	`ron`	133
`fra`	1 095	`ara`	210	`ces`	121
`por`	1 071	`nld`	172	`dan`	109
`deu`	636	`fin`	148	`srp`	103
`rus`	549	`zho`	142	`...`	...

Tabelle 5.1: Auflistung der Sprachen mit den meisten Dokumenten der heruntergeladenen Nachrichtenseiten, die nachfolgend als Referenzdokumente fungieren.

5.2. Qualitätsmaße

Zur Evaluierung des Verfahrens wurden Maße verwendet, die ursprünglich im Information Retrieval für die Auswertung der Güte der Ergebnisse einer Anfrage eingeführt worden sind. Hiermit sind *precision*, *recall* und das darauf aufbauende *F-Measure* gemeint. Für die hier anzutreffenden Gegebenheiten wurden diese Maße entsprechend angepasst. Auf welche Art und Weise dies geschehen ist, soll nun dargestellt werden.

Zunächst betrachten wir eine Dokumentenkollektion $\mathfrak{D}_{\text{test}}$, für deren Dokumente die zugehörigen Sprachen bekannt sind. D.h. für jedes $d \in \mathfrak{D}_{\text{test}}$ ist eine Referenzsprachenmenge $\ell_{\text{ref}}(d)$ bekannt. Klassifiziert *LangSepa* nun alle diese Dokumente nach Sprachen mit $\ell(d)$, so können die Dokumente ergebnisbezogen durch eine Partitionierung von $\mathfrak{D}_{\text{test}}$ in die folgenden drei Teilmengen eingeteilt werden:

$$\begin{aligned}
\mathfrak{D}^{K}_{\text{test}} &= \{d \in \mathfrak{D}_{\text{test}} | \ell(d) = \ell_{\text{ref}}(d)\} && (5.1)\\
\mathfrak{D}^{W}_{\text{test}} &= \{d \in \mathfrak{D}_{\text{test}} | (\ell(d) \neq \ell_{\text{ref}}(d)) \wedge (\ell(d) \neq \{\} \vee \{?\})\} && (5.2)\\
\mathfrak{D}^{U}_{\text{test}} &= \{d \in \mathfrak{D}_{\text{test}} | \ell(d) = \{\} \vee \{?\}\} \quad . && (5.3)
\end{aligned}$$

$\mathfrak{D}^{K}_{\text{test}}$ enthält alle Dokumente, die vollständig korrekt klassifiziert worden, $\mathfrak{D}^{W}_{\text{test}}$ jene, denen mindestens eine falsche Sprache zugeordnet wurde und $\mathfrak{D}^{U}_{\text{test}}$ die Dokumente, die sich durch *LangSepa* als ungeeignet oder nicht klassifizierbar erwiesen haben. Diese drei Mengen decken alle Dokumente ab, d.h.

$$\mathfrak{D}^{K}_{\text{test}} \cup \mathfrak{D}^{W}_{\text{test}} \cup \mathfrak{D}^{U}_{\text{test}} = \mathfrak{D}_{\text{test}} \quad . \tag{5.4}$$

Insbesondere die Einteilung in $\mathfrak{D}^{U}_{\text{test}}$ konnte gewählt werden, da für den Test keine Dokumentenmenge zur Verfügung stand, die zur Überprüfung ungeeigneten Inhalts[26] diente.

Durch diese Definitionen sind die bereits genannten Maße in Analogie zu ihrer ursprünglichen Beschreibung (siehe [2, S. 75, 82]) folgendermaßen anpassbar:

---

[26] Hiermit sind Log-Dateien, Programmiercode und weitere, von natürlichen Sprachen losgelöste Inhalte gemeint.

precision: Der Anteil korrekt klassifizierter Dokumente an allen Dokumenten, denen mindestens eine Sprache aus $\mathfrak{L}$ zugeordnet wurde.

$$p = \frac{|\mathfrak{D}_{\text{test}}^{K}|}{|\mathfrak{D}_{\text{test}}^{K}| + |\mathfrak{D}_{\text{test}}^{W}|} \tag{5.5}$$

recall: Der Anteil korrekt klassifizierter Dokumente an allen Dokumenten, die korrekt oder nicht klassifiziert werden konnten.

$$r = \frac{|\mathfrak{D}_{\text{test}}^{K}|}{|\mathfrak{D}_{\text{test}}^{K}| + |\mathfrak{D}_{\text{test}}^{U}|} \tag{5.6}$$

F-Measure: Das harmonische Mittel von p und r.

$$F = \frac{2pr}{p + r} \tag{5.7}$$

Der Wert für *precision* beschreibt gewissermaßen die Genauigkeit, mit der das Verfahren Sprachklassifikationen vornimmt. *Recall* gibt Aufschluss über die Empfindlichkeit[27], mit der die Merkmale eines Dokuments zur Wahl der korrekten Sprache(n) führt . Erstrebenswert sind hierbei p und r nahe oder gleich 1. Um einen Kompromiss zwischen der Optimierung beider Werte zu finden wurde das *F-Measure* eingeführt. Falls beide Maße gleichberechtigt in eine Evaluation einfließen sollen, so ist die beste Kombination zwischen p und r gegeben, wenn F maximal wird. Problembezogen können an Stelle von F auch andere Maße verwendet werden.

5.3. Parameteranalyse

Die vorgestellten Algorithmen zur Klassifikation besitzen jeweils einige Parameter, die vom Anwender des Werkzeugs gesetzt werden müssen. Eine geeignete Wahl der Parameter kann die Güte der Ergebnisse maßgeblich erhöhen. Aus diesem Grund wird in diesem Abschnitt für jeden der drei Teilalgorithmen zur Sprachklassifikation die Güte nach den eben vorgestellten Maßen in Abhängigkeit von einzelnen Parametern des Verfahrens bestimmt. Für jeden der Tests wurde $k = 400$ als globaler Parameter gewählt.

5.3.1. Stopwort-Klassifikation

Im Stopwortverfahren spielen die Parameter α und γ die größte Rolle. α gibt die maximale relative Überdeckungsabweichung von der Idealüberdeckung der betrachteten Sprache an und über γ regelt man die Härte, mit denen nahe Sprachen, die sich durch gleiche Stopwörter gegenseitig Überdeckung induzieren, aus der Klassifikation ausgeschlossen werden.

---

[27] Eine hohe Empfindlichkeit ist dabei gegeben, wenn bereits eine geringe Ausprägung von Merkmalen zur korrekten Sprachwahl führen. Geringe Empfindlichkeit dann, wenn die Klassifikation erst bei starker Merkmalsausprägung korrekt vorgenommen wird.

Zunächst ist in Abb. 5.1 die Qualität der Klassifikation ausschließlich durch das Stopwortkriterium in Abhängigkeit von α dargestellt. Die Werte für precision sind über den gesamten Bereich größer als $0{,}97$. Das Maximum ist bei $\alpha = 0{,}25$ mit $0{,}998$ zu finden. Der relativ schwache recall im Bereich $0 < \alpha < 0{,}3$ ist mit Dokumenten zu begründen, die entweder zu viele oder zu wenige Stopwörter enthalten. Der zunehmende recall mit steigendem α muss für eine geeignete Wahl α's mit der zunehmend sinkenden precision abgewogen werden. Für die Evaluation auf den aspra20-Dokumenten wird $\alpha = 0{,}4$ gewählt, um nicht zu viel precision einzubüßen.

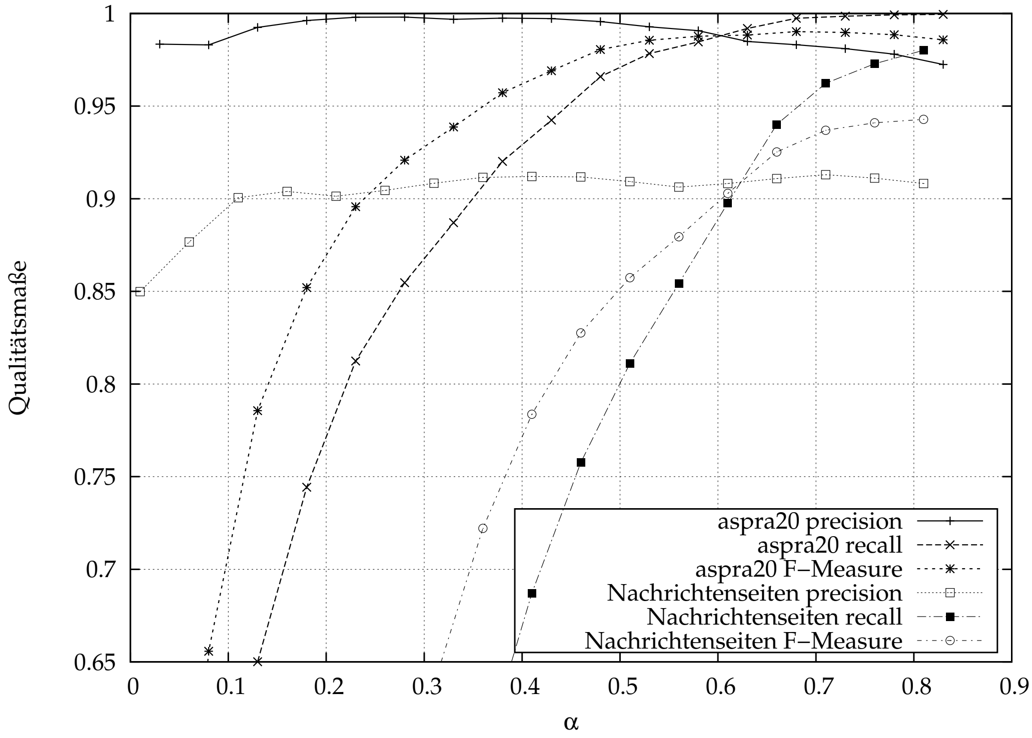


Abbildung 5.1: Die Qualitätsmaße für den Parameter α bei Stopwortüberdeckung als Klassifikationskriterium. Für aspra20-Dokumente wurden alle Sprachen verwendet, die keine Worttrenner enthalten (54 Sprachen). Für jede Sprache wurden randomisiert je 100 Dokumente generiert, die jeweils maximal 20 Sätze enthielten. Die Auswertung der Nachrichtenseiten erfolgte auf allen heruntergeladenen Dokumenten, die mehr als 250 Zeichen beinhalteten. Die weiteren Parameter zur Stopwortklassifikation wurden folgendermaßen fixiert: $\beta = 0{,}9$, $\gamma = 0{,}5$ und $\delta = 3$.

Um eine Vorstellung zu bekommen, wie eine Wahl von α erfolgen sollte, um erfolgreich die Sprache(n) von Webseiten zu bestimmen, wurde ebenso eine Auswertung über den Nachrichtenseiten vorgenommen. Hierbei ist anzumerken, dass aufgrund der ungleichen Verteilung der Seitensprachen insbesondere die häufigen Sprachen stark ins Gewicht fallen. Das sind Englisch, Spanisch, Franzöische und Portugiesisch. Es fällt auf, dass die zugehörige precision deutlich schwächer ist, als die der aspra20 Dokumente. Das liegt hauptsächlich an den falsch gesetzten Referenzsprachen auf einem Teil dieser Dokumente. Anhand der Recallwerte, die eine deutliche Verschiebung nach rechts gegenüber denen der aspra20-

Dokumente aufweisen, kann eine gewisse Stopwortreduktion in den Nachrichten-Webseiten abgelesen werden. Die Verschiebung der Überdeckung wurde bereits in Unterabschnitt 3.3 durch Abb. 3.3 für die Sprache Englisch visualisiert. Die Tatsache, dass die aspra20-Dokumente aus reinen Fließtexten bestehen und keine störenden Elemente, wie die angesprochenen in 2.2.3, enthalten, ist hierfür verantwortlich. Um nicht zu viele Webseiten unklassifiziert zu belassen, sollte α tendenziell größer gewählt werden als der Wert, welcher für aspra20 zum Einsatz kommt.

In Abb. 5.2 wird der Einfluss des Parameters γ auf die precision dargestellt. Da dieser Parameter zur Trennung naher Sprachen dient, wurden Afrikaans und Niederländisch exemplarisch aus dem Testdatensatz herausgegriffen. Als grobe Wirkung zeigt sich zunächst, dass ein zu hoher Wert für γ zur Klassifikationen mehrerer Sprachen und somit zu Precisionverlust auf den betrachteten einsprachigen Texten führt. Vorausgesesetzt die betrachtete Sprache besitzt mindestens eine sehr nahe Sprache. Im Falle der betrachteten Abbildung, werden Niederländisch und Afrikaans, für γ nahe 1, teils gemeinsam als Sprache für Dokumente gewählt. Dabei fällt ebenso auf, dass Afrikaans eher mit Niederländisch gleichzeitig klassifiziert wird als umgekehrt. Die Begründung hierfür liegt in der gegenseitigen Induzierung von Überdeckungen durch Stopwörter der jeweils nahen Sprache oder Sprachen. Aus Tab. 3.2 kann bereits die Tendenz abgelesen werden, dass Niederländisch mehr von den Stopwörtern von Afrikaans profitiert, als dies umgekehrt der Fall ist. Ein zu niedriger Wert für γ könnte zu starken Ausschlüssen vermeintlich naher Sprachen in mehrsprachigen Dokumenten führen. Da für die ersten Untersuchungen jedoch nur einsprachige Dokumente betrachtet wurden, kann dieser Effekt an dieser Stelle nicht näher untersucht werden.

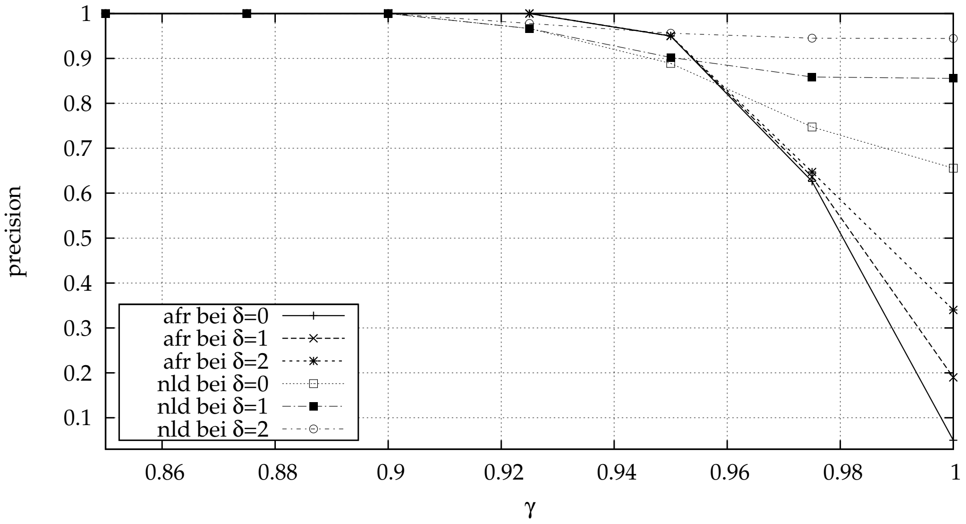


Abbildung 5.2: Die Genauigkeit bzw. *precision* für den Parameter γ bei Stopwortüberdeckung als Klassifikationskriterium auf den aspra-20-Dokumenten. Die Dokumente wurden in der Art wie in Abb. 5.1 genutzt. Für die Gegenüberstellung der nahen Sprachen, Afrikaans und Niederländisch, wurde ebenso der Parameter δ mit in Betracht gezogen. Die weiteren Parameter wurden konstant gesetzt: $\alpha = 0{,}4$ und $\beta = 0{,}9$

Zusätzlich zu γ wurde der Einfluss des Parameters δ ebenfalls untersucht. δ lässt bei höherem Wert die precision weniger stark abfallen. Je größer δ gewählt wird, desto mehr eigene Stopwörter[28] müssen durch die konkurrierende nahe Sprache im betrachteten Dokument aufgebracht werden, um Klassifikationschancen zu wahren. Für die späteren Untersuchungen wird $\gamma = 0{,}8$ und $\delta = 3$ gesetzt.

5.3.2. Trigramm-Klassifikation

Die Trigramm-Klassifikation dient im Verfahren, welches letztendlich eingesetzt werden soll, der Erkennung mehrerer Sprachen und ungeeignetem Material für die Weiterverarbeitung. In diesem Abschnitt sollen die Fensterbreite w und die minimale Anzahl benötigter Trigramme pro Sprache und pro Trigrammfolge untersucht werden. Diese Untersuchung erfolgt wieder auf den Referenzdokumenten der aspra20-Datenbank.

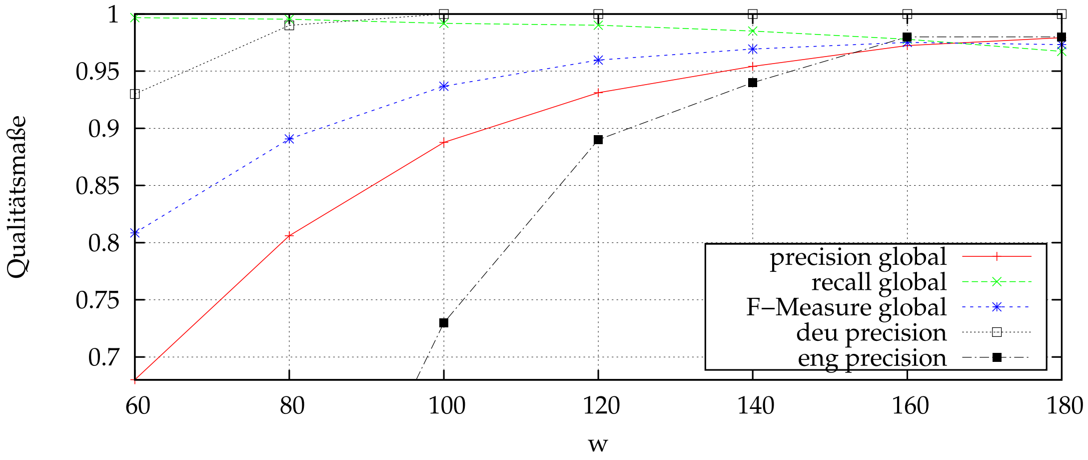


Abbildung 5.3: Qualitätsmaße in Abhängigkeit von der Fenstergröße w bei der Trigrammklassifikation auf aspra20-Dokumenten. Hierfür wurden sämtliche Sprachen ausgewertet, für die Trigrammlisten aus der aspra10-Datenbank und Referenzdokumente aus aspra20 generiert werden konnten. Precision wurde exemplarisch an den Sprachen Deutsch und Englisch aufgetragen. Die weiteren Parameter wurden konstant gesetzt: $\mu = 15$ und $\lambda = 2$.

Abb. 5.3 zeigt die Auswirkungen unterschiedlicher Fensterbreiten w auf die Gütemaße. Über alle Sprachen kann beobachtet werden, dass mit zunehmender Fensterbreite die Genauigkeit zunimmt, bis diese bei ca. $w = 160$ auf konstantem Niveau bei $p = 0{,}97$ bleibt. Die Zunahme verläuft bei einzelnen Sprachen sehr unterschiedlich. Die zwei typischen Muster sind hier durch die Sprachen Deutsch und Englisch widergespiegelt. Deutsch kann bereits bei 100 Zeichen Fensterbreite mit perfekter Genauigkeit erkannt werden. Englisch findet sich erst bei 160 Trigrammen im Bereich nahe 1. Als Erklärung für diesen Effekt können die nahen Sprachen wieder angebracht werden. Schottisch und Englisch teilen sich viele Stopwörter und somit auch viele Trigramme. Eine klare Trennung erfordert somit

---

[28] Eigene Stopwörter sind an dieser Stelle bzgl. zweier Sprachen L_i und L_j die Stopwörter der Sprache L_i, die nicht identisch sind mit Stopwörtern aus L_j.

eine längere Sequenz an Zeichen, um Englisch klar von Schottisch unterscheiden zu können. Im Gegensatz dazu verhält sich Deutsch. Hierfür kann wieder die Tab. 3.2 angebracht werden, aus der ersichtlich wird, dass Deutsch keine sehr nahe Sprache in den Trainingsdaten besitzt. Um nicht zu grob an die Texte heranzugehen wird $w = 160$ gewählt. Dies stellt einen Kompromiss zwischen möglichst kleinem w und guter precision dar.

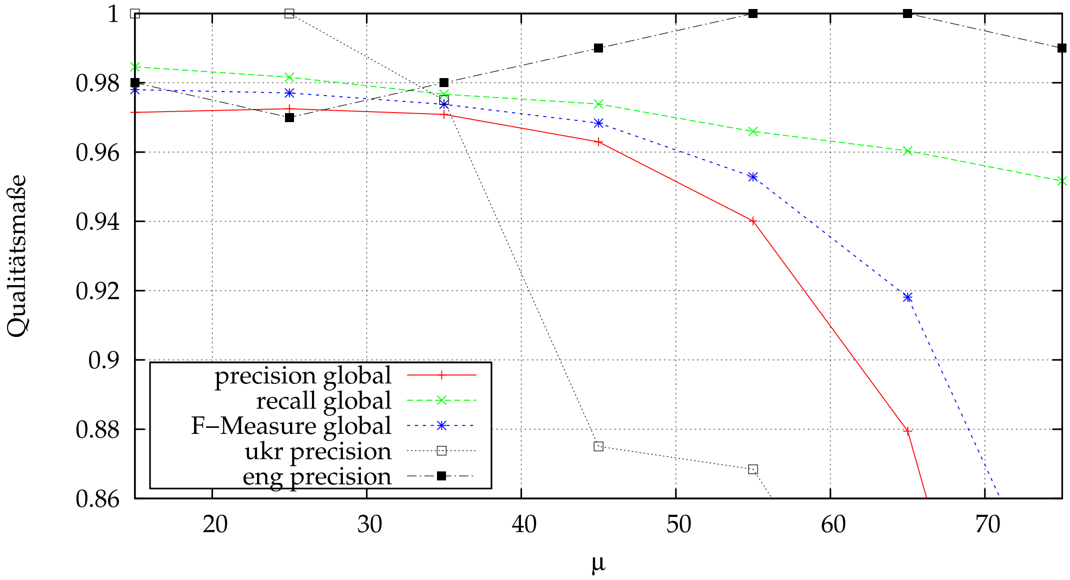


Abbildung 5.4: Qualitätsmaße in Abhängigkeit von der Mindestanzahl μ pro Trigrammsequenz auf aspra20-Dokumenten. Gleiche Dokumentenkollektion wie in Abb. 5.3. Precision wurde exemplarisch an den Sprachen Ukrainisch und Englisch aufgetragen. Die weiteren Parameter wurden konstant gesetzt: $w = 160$ und $\lambda = 2$.

Für eine Anpassung des Parameters μ hilft ein Blick in die Abb. 5.4. Aus dieser ist ersichtlich, dass es äußerst problematisch ist eine geeignete Wahl hierfür zu finden. In der globalen Kurve für precision ist $\mu = 25$ optimal. Bei Betrachtung einzelner Sprachen fällt jedoch auf, dass nahe Sprachen durch ein optimales μ im höheren Bereich voneinander zu trennen sind. Das hängt im Übrigen mit der Überdeckung des Textes durch Trigramme einer bestimmten Sprache zusammen. Im Fall von Englisch ergibt sich die Loslösung von Schottisch im Bereich $55 \leq \mu \leq 65$. Ukrainisch stellt das Gegenbeispiel dar. Mit sehr geringem Trigrammanteil pro Fenster kann die Sprache perfekt identifiziert werden. Bei zu großen μ reichen die Trainingsdaten nicht aus, um mehr Trigrammüberdeckung hervorzurufen, was zum Ausschluss von Ukrainisch aus der Trigrammsequenz führt, somit zur Nicht- oder Falschklassifikation und deshalb wiederum zu schlechterer precision und recall. Um keine Sprache durch zu geringe Überdeckung durch Trigramme auszuschließen, wird $\mu = 25$ in späteren Ergebnissen gesetzt.

5.3.3. Unigramm-Klassifikation

Bei der Klassifikation durch Unigramme auf den Sprachen, die keine Worttrennsymbole zwischen Wörtern verwenden[29], konnten sehr gute Ergebnisse bereits bei kleinen Fenstergrößen w erzielt werden. Um eine Klassifikation unter gleichen Voraussetzungen, wie im Falle der Klassifikation durch Trigramme zu garantieren, wird die Fenstergröße auf jene, die im Trigrammverfahren ermittelt wurde, auch hier gesetzt. Damit kann zwar die Stärke dieser Teilklassifikation nicht im vollen Umfang ausgeschöpft werden, aber dafür bleibt die Chancengleichheit jeder Sprache gewahrt, dem Dokument zugeordnet zu werden.

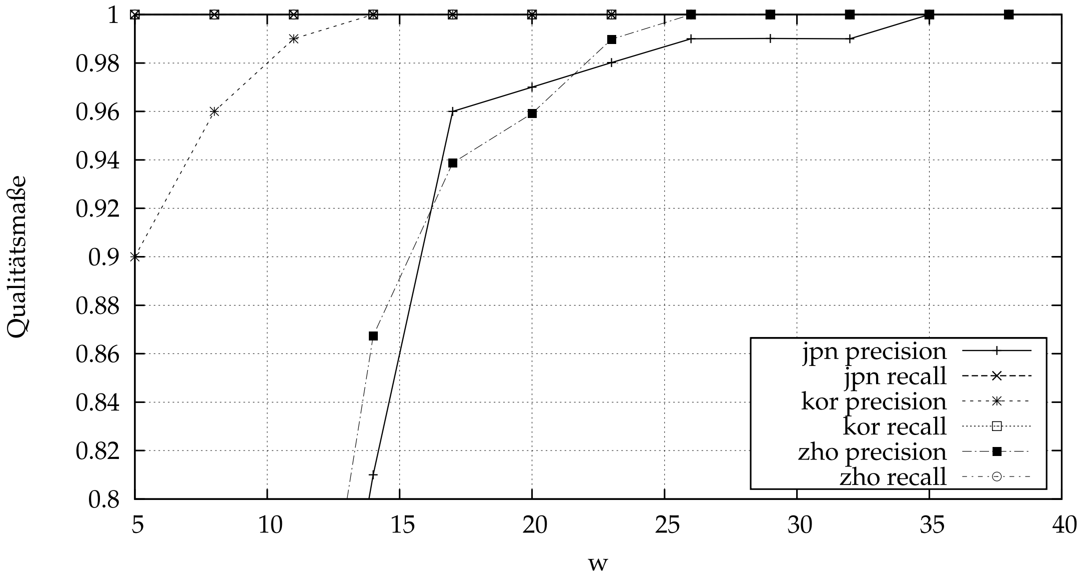


Abbildung 5.5: Qualitätsmaße in Abhängigkeit von der Fenstergröße w bei der Unigrammklassifikation auf aspra20-Dokumenten. Die Sprachen Japanisch, Koreanisch und Chinesisch wurden unter Benutzung von aspra20-Dokumenten überprüft. Die weiteren Parameter wurden konstant gesetzt: $\mu = 1$ und $\lambda = 2$.

5.4. Evaluation

Für die Evaluation wurden precision und recall auf allen Sprachen der aspra20-Datenbank mit je 100 zufällig gewählten Dokumenten gemessen. Die Auswertung erfolgte nach Anzahl der importierten Stopwörter k und der Dokumentenlänge in der Anzahl der bezogenen Sätze. Die Ergebnisse des vollständigen Verfahrens aus Abb. 4.1 sind in nachfolgenden Tabellen, nach Sprachcode sortiert, aufgelistet.

---

[29] Einzige Ausnahme hierbeit ist Koreanisch. Diese ist zwar eine Sprache, die Worttrenner verwendet, sticht aber in den Trainingsdaten durch eine vergleichbar hohe Zahl an einzigartigen Zeichen hervor, wie dies bei Japanisch und Chinesisch der Fall ist. Tests, die auf Basis der Klassifikation durch Stopwortüberdeckung und Trigrammwahrscheinlichkeit aufbauten, resultierten in schlechten Qualitätsmaßen. Aus diesem Grund wurde für die Klassifikation von Koreanisch die Unigrammvariante gewählt.

	k						Dokumentenlänge in Sätzen					
	50		200		400		2		5		15	
Code	p	r	p	r	p	r	p	r	p	r	p	r
`afr`	1	1	1	1	1	1	1	1	1	1	1	1
`als`	1	,963	1	,976	,988	,988	1	1	1	,944	,984	,984
`ara`	1	1	1	1	1	1	,966	1	,990	1	1	1
`azj`	,949	,974	,988	1	,987	,987	1	1	,973	,947	,973	1
`bre`	1	,909	,984	,923	1	,909	1	1	,857	1	1	,895
`bul`	,926	,980	,945	1	,981	1	1	1	1	1	1	1
`ceb`	1	1	1	1	1	1	1	1	1	1	1	1
`ces`	1	1	1	1	1	1	,929	1	1	1	1	1
`cym`	,901	,924	,988	,977	,976	,976	1	1	1	,926	,987	,961
`dan`	1	1	1	1	1	1	1	1	1	1	1	1
`deu`	1	1	1	1	1	1	1	1	1	1	1	1
`ell`	,862	,862	,990	,990	,990	,990	1	1	1	1	1	,979
`eng`	1	1	1	1	1	1	1	1	1	1	1	1
`epo`	,856	,975	,910	,964	,921	,965	1	1	1	1	1	1
`est`	,909	1	,990	1	,990	1	,946	1	,947	1	1	1
`fin`	,963	,981	1	1	1	1	1	1	1	1	1	1
`fra`	1	1	1	1	1	1	1	1	1	1	1	1
`gle`	,947	,989	,958	1	,958	,989	1	1	,940	,963	,927	,988
`hin`	1	,360	1	1	1	1	1	,959	1	,990	1	1
`hsb`	,990	1	1	1	1	1	1	1	1	1	1	1
`hun`	1	,938	1	,969	1	1	1	1	1	1	1	1
`ido`	1	1	1	1	1	1	1	1	1	1	1	1
`ind`	,932	1	,977	1	,989	1	,850	1	,943	,980	,977	1
`ita`	1	1	1	1	1	1	1	1	1	1	1	1
`jpn`	1	1	1	1	1	1	1	1	1	,874	1	1
`kor`	1	1	1	1	,990	1	1	1	1	1	,990	1
`lat`	,947	,866	,986	,859	1	,872	,929	1	1	,980	1	,936
`lav`	1	1	1	1	1	1	,975	1	1	1	1	1
`lit`	,960	1	1	1	1	1	,984	1	1	1	1	1

Tabelle 5.2: Evaluation des vollständigen vorgestellten Verfahrens anhand von Referenzdokumenten aus dem Satzkorpus der aspra20-Datenbanken (Teil 1). Für die Untersuchung nach dem globalen Parameter k wurde die Satzanzahl pro Dokument auf 20 festgesetzt. Bei der Analyse der Qualität nach der Satzanzahl pro Dokument wurde $k = 400$ fest gewählt. Die weiteren Parameter wurden entsprechend der Analyse folgendermaßen gewählt: $\alpha = 0{,}4$, $\beta = 0{,}9$, $\gamma = 0{,}5$, $\delta = 3$, $w = 160$, $\mu = 25$, $\lambda = 2$ und $\kappa = 30$. (Fortsetzung dieser Tabelle folgt auf der nächsten Seite)

	k						Dokumentenlänge in Sätzen					
	50		200		400		2		5		15	
Code	p	r	p	r	p	r	p	r	p	r	p	r
`ltz`	1	1	1	1	1	1	1	1	1	1	1	1
`mar`	1	,515	1	,859	,909	1	1	1	1	,960	1	,901
`mkd`	1	,667	1	,653	1	,653	1	,100	1	,375	,982	,724
`msa`	,610	1	,860	1	,980	1	,700	1	,870	1	,970	1
`nep`	1	,980	1	1	1	1	1	,987	1	,990	1	1
`new`	1	,960	1	1	1	1	,860	,980	1	,910	1	1
`nld`	1	1	1	1	1	1	1	1	1	1	1	1
`nno`	1	1	1	1	1	1	,986	1	,990	1	1	1
`nob`	1	1	1	1	1	1	,900	1	,990	1	1	1
`oci`	,882	,957	,962	1	,981	1	,375	1	,750	,600	,952	,976
`pdc`	,903	1	,935	,989	,978	,989	,889	1	,909	,930	,965	,988
`pol`	1	1	1	1	1	1	,968	1	1	1	1	1
`rus`	,990	1	1	1	1	1	1	1	1	1	1	1
`slk`	,810	,944	,952	,952	,952	,952	1	1	1	1	,950	1
`srp`	,731	,919	,785	,939	,926	,974	,800	1	,981	,930	,973	,948
`sun`	1	,895	1	,895	1	1	1	1	1	1	1	1
`swe`	1	1	1	1	1	1	1	1	1	1	1	1
`tgl`	,971	,944	1	,973	1	,973	1	,926	1	,950	1	1
`tur`	,970	1	1	1	1	1	,958	1	,990	1	1	1
`ukr`	,925	,925	1	,953	1	,953	1	1	1	1	1	1
`urd`	1	,526	,983	,598	,986	,722	1	,763	1	,782	1	,656
`vie`	1	,569	,833	,577	,881	,698	,750	,600	,900	,474	,879	,604
`zho`	1	1	1	1	1	1	1	1	1	1	1	1
min	,610	,515	,785	,577	,881	,653	,375	,100	,750	,375	,879	,604
max	1	1	1	1	1	1	1	1	1	1	1	1
∅	**,960**	**,935**	**,983**	**,967**	**,991**	**,974**	**,966**	**,978**	**,988**	**,972**	**,993**	**,976**

Tabelle 5.3: Fortsetzung der Tab. 5.2 zur Sprachenevaluation. Minima, Maxima und Durchschnittswerte beziehen sich auf alle Sprachen, also beide Tabellen.

Über einem Großteil der Sprachen können beachtliche Erfolge erzielt werden. Bei den Sprachen `mar`, `mkd`, `msa`, `oci` und `vie` entstehen sehr starke Schwankungen in den gemessenen Werten und äußerst miserable Ergebnisse. Hierbei könnten verunreinigte bezogene Datenbanksätze eine Rolle spielen. Dies wiegt sich besonders stark auf, da aus Gründen der persönlichen Rechenkapazität pro Sprache und Analysekonfiguration lediglich 100 Sätze verwandt wurden. Eine nähere Untersuchung der zugehörigen Quellen und ein Test auf gesichert guten Referenzdaten wäre für eine tiefergehende Evaluation sinnvoll. Es zeigt sich ebenso, dass für einige der Sprachen perfekte precision und recall Werte auf allen Tests erzielt werden konnten.

In der Tabelle kann beobachtet werden, dass sich durch eine Erhöhung von k die Recall- und Precisionwerte verbessern. Eine weitere Erhöhung von k wurde nicht untersucht, da sich einerseits die Komplexität merklich durch verlängerte Laufzeiten des Programms sichtbar machte, andererseits aber auch aus dem Grund, da die meisten der Worthäufigkeitslisten, die auf der UN-Menschenrechtscharta basieren, nicht mehr als 500 Wortformen beinhalteten. Bessere Ergebnisse werden auch mit zunehmender Dokumentenlänge sichtbar. Im günstigsten Falle konnte eine durchschnittliche precision von $0{,}993$ mit dem zugehörgen recall $0{,}976$ erzielt werden. Diese Ergebnisse sind zu den vorgestellten Evaluationen in Unterabschnitt 2.2.2 vergleichbar gut.

5.5. FindLinks-Dokumente

Um einen ersten Sprachüberblick über die gecrawlten FindLinks-Dokumente zu erhalten, wurde ein vollständiger Programmdurchlauf auf den gesamten Dokumenten eines FindLinks-Client-Benutzers durchgeführt. Unter der Annahme, dass die Sprachanteile aller FindLinks-Dokumente gleichmäßig auf alle Clients verteilt werden, repräsentiert einer der Teilnehmer am FindLinks-Crawling die Sprachverteilung aller gecrawlten Texte.

Als Parameter fungierten jene, die für die Evaluation eingesetzt wurden, bis auf α. Aufgrund der besonderen Charakteristik von Webseitentexten wurde mit $\alpha = 0{,}7$ die Sprachidentifikation durchgeführt.

Auf Basis der gleichen Daten werden anschließend einige Messergebnisse zu den Originalkodierungen der Webseiten präsentiert.

5.5.1. Sprachen

In nachfolgender Tabelle (Tab. 5.4) kann die Verteilung der Sprachen auf die Dokumente eingesehen werden. Als zusätzliche Kategorien fungieren mehrsprachige Dokumente und jene mit unbrauchbaren Material oder nichtidentifizierbarer Sprache.

5.5.2. Kodierungen

Ein Überblick über die Verteilung der Originalkodierungen der gecrawlten Webseiten ist in Tab. 5.5 gegeben. Um auf die in der Einleitung angesprochene Problematik zurückzukommen, die mit der Spracherkennung anhand der Kodierung

Code	Doks.	Anteil [%]	Code	Doks	Anteil [%]
deu	70 205	29,16	slk	2 996	1,24
eng	66 356	27,56	fin	2 981	1,24
???	33 268	13,82	por	2 975	1,24
spa	10 292	4,28	est	2 727	1,13
fra	5 929	2,46	pdc	2 448	1,02
ita	3 102	1,29	...	...	

Tabelle 5.4: Sprachverteilung der gecrawlten FindLinks-Dokumente eines FindLinks-Client Nutzers. Insgesamte Anzahl der untersuchten Dokumente: 240 741. Es wurden 5 351 Dokumente mehrsprachig klassifiziert.

Kodierung	Doks.	Anteil [%]	Kodierung	Doks	Anteil [%]
iso-8859-1	164 377	68,28	euc-kr	2 688	1,12
utf-8	26 336	10,94	windows-1257	2 635	1,09
windows-1251	16 664	6,92	big5	2 449	1,02
windows-1250	6 821	2,83	iso-8859-7	1 984	0,82
gb-2312	4 640	1,93	windows-1253	1 902	0,79
iso-8859-2	3 433	1,43	...	...	...

Tabelle 5.5: Kodierungsverteilung gecrawlter FindLinks-Dokumente eines FindLinks-Client Nutzers. Insgesamte Anzahl der untersuchten Dokumente: 240 741.

eines Dokuments einherging, sei auf die Tab. 5.6 verwiesen. In dieser kann eingesehen werden, dass eine Kodierung durchaus falsch gesetzt werden kann und auch wird. Im Falle der `iso-8859` und `windows-code-pages` sind nur für einige Sprachen deren Sprachsymbole durch die Kodierungen vollständig abgedeckt. Die Tabelle illustriert dies an den Kodierungen, die primär für die Sprache Griechisch angedacht sind.

iso-8859-7			windows-1253		
Code	Doks.	Anteil [%]	Code	Doks	Anteil [%]
???	1 394	70,26	???	1 274	66,98
eng	440	22,18	eng	473	24,88
ell	18	0,91	ell	45	2,37
cat	7	0,35	ita	10	0,53
deu	6	0,30	deu	10	0,53
ita	5	0,25	lit	4	0,21

Tabelle 5.6: Sprachverteilung bei den Kodierungen `iso-8859-7` und `windows-1253`, die eigentlich für die Kodierung griechischer Zeichen vorgesehen sind. Als Grundlage dienten die Daten, die bereits in Tab. 5.4 und Tab. 5.5 verwendet wurden.

6. Abschluss

6.1. Zusammenfassung

Der in dieser Arbeit vorgestellte automatische Sprachidentifizierer *LangSepa* klassifiziert Webseitendokumente, die durch das Verteilte System FindLinks gecrawlt wurden. Zur Klassifikation werden Stopworte, Trigramme und Unigramme herangezogen. Bei der Klassifikation durch Stopworte kommt ein Überdeckungskriterium zum Einsatz, welches die Sprachen klassifiziert, deren Dokumenten-Stopwortüberdeckung sich möglichst nahe an der optimalen Überdeckung der zugehörigen Trainingsdaten befindet. Für Trigramme und Unigramme ist ein probabilistisches Verfahren im Einsatz, das auf Basis von verketteten Wahrscheinlichkeiten die wahrscheinlichste Sprache jedes Textausschnitte des Dokuments ermittelt und daraus die Klassifikationssprachen extrahiert. Das Programm kann durch seine Parameter entsprechend zu seiner Verwendung konfiguriert werden. Hilfestellungen zur geeigneten Wahl der Parameter, um eine möglichst hohe Genauigkeit zu erzielen, konnten durch eine Parameteranalyse geliefert werden.

6.2. Fazit und Zukünftige Arbeiten

Das vorgestellte Werkzeug zur Sprachidentifikation von FindLinks kann sehr gute Ergebnisse über einsprachigen Dokumenten vorweisen. Eine Überprüfung multilingualer Dokumente lag außerhalb des zeitlichen und inhaltlichen Umfangs dieser Arbeit. Dies wäre jedoch in einer sich anschließenden Untersuchung wünschenswert, um ebenso gute Resultate auf solchen Dokumenten zu erhalten.

Durch die große Zahl von Sprachen, die klassifiziert werden können, war es unumgänglich Methoden zu implementieren, die nahe Sprachen voneinander trennen. Dies stellte sich als ein Schlüsselproblem zur exakten Identifikation der korrekten zweier oder mehrerer naher Sprachen dar. Ein erster Ansatz hierfür konnte im Klassifikationsverfahren anhand von Stopworten erfolgreich eingeführt werden. Im Trigrammverfahren wurde keine solche Heuristik angewandt, so dass eine sehr lange Sequenz von Trigrammen notwendig ist, bevor die korrekte Sprache identifiziert werden kann. Hier besteht demnach definitiv noch Optimierungsbedarf. Vielleicht ist es generell notwendig eine methodenübergreifende Strategie zu entwickeln, um nahe Sprachen voneinander trennen zu können. Die Unigrammklassifikation zeigte äußerst präzise Ergebnisse. Das hängt jedoch mit der geringen Zahl der zu betrachtenden Sprachen zusammen. Wenn zukünftig die Einzelsprachen der Makrosprache Chinesisch klassifiziert werden sollen, so wird ein Ausbau oder eine Ablösung dieser Methode notwendig.

Der Aufbau und die Implementierung von *LangSepa* ist simpel und überschaubar gehalten, so dass einer Weiterentwicklung an dieser Stelle nichts im Wege steht. Insbesondere die Pipes-and-Filters Architektur erleichtert den Austausch von einzelnen Komponenten, so dass im Falle von umfangreicheren Trainingsdaten komplexere Verfahren bei Bedarf eingebaut werden können.

Literatur

[1] ARTEMENKO, O. und SHRAMKO, M.: *Entwicklung eines Werkzeugs zur Sprachidentifikation in mono- und multilingualen Texten*. Magisterarbeit, Universität Hildesheim, Hildesheim 2005

[2] BAEZA-YATES, R. und RIBEIRO-NETO, B. (Hrsg.): *Modern Information Retrieval*.
Addison-Wesley, 1. Auflage, New York, Oxford 1999

[3] BIEMANN, C. und TERESNIAK, S.: Disentangling from Babylonian Confusion - Unsupervised Language Identification. In: GELBUKH, A. (Hrsg.): *CICLing 2005*, LNCS 3406, S. 773–784, Springer-Verlag, Berlin, Heidelberg 2005

[4] BUSCHMANN, F., MEUNIER, R., ROHNERT, H., SOMMERLAD, P. und STAL, M.: *Patternorientierte Softwarearchitektur*, Ein Pattern-System.
Addison-Wesley, 1. Auflage, Bonn 2000

[5] CAI, L.: *Language Identification on the WWW*. Master-Thesis, Acadia University, Wolfville 2009

[6] CAVNAR, W. B. und TRENKLE, J. M.: N-Gram-Based Text Categorization. In: *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval*, S. 161–175, UNLV Publications/Reprographics, Las Vegas 1994

[7] REHUREK, R. und KOLKUS, M.: Language Identification on the Web: Extending the Dictionary Method. In: GELBUKH, A. (Hrsg.): *CICLing 2009*, LNCS 5449, S. 357–368, Springer-Verlag, Berlin, Heidelberg 2009

[8] GREFENSTETTE, G.: Comparing two language identification schemes. In: *Proceedings of the 3rd International Conference on Statistical Analysis of Textual Data*, Rom, 1995

[9] HEYER, G., QUASTHOFF, U. und WITTIG, T.: *Text-Mining: Wissensrohstoff Text*, Konzepte, Algorithmen, Ergebnisse.
W3L GmbH, Herdecke, Bochum 2006

[10] LANGER, S.: Grenzen der Sprachenidentifizierung. In: *Tagungsband KONVENS 2002*, S. 99–106, Saarbrücken 2002

[11] LEWIS, M. P. (Hrsg.): *Ethnologue: Languages of the World*.
SIL International, 16. Auflage, Dallas 2009

[12] MANNING, C. D., RAGHAVAN, P. und SCHÜTZE, H.: *An Introduction to Information Retrieval*.
Cambridge University Press, Cambridge 2008

[13] MARTINO, M. J. und PAULSEN, R. C.: Determining a natural language shift in a computer document. U.S. Patent No. 5913185, Juni 1999

[14] OTTMANN, T. und WIDMAYER, P.: *Algorithmen und Datenstrukturen*. Spektrum, Akad. Verlag, 4. Auflage, Heidelberg, Berlin 2002

[15] SOLYMOSI, A. und GRUDE, U.: *Grundkurs Algorithmen und Datenstrukturen in JAVA*, Eine Einführung in die Praktische Informatik. Vieweg+Teubner, 4. Auflage, Wiesbaden 2008

[16] SOUTER, C., CHURCHER, G., HAYES, J., HUGHES, J. und JOHNSON, S.: Natural Language Identification using Corpus-Based Models. In: *Hermes Journal of Linguistics*, **13** (1994), S. 183–203.

[17] TERESNIAK, S.: *Statistikbasierte Sprachenidentifikation auf Satzbasis*. Bachelorarbeit, Universität Leipzig, Leipzig 2005

[18] ULLENBOOM, C.: *Java ist auch eine Insel*, Programmieren mit der Java Platform, Standard Edition 6. Galileo Press, 8. Auflage, Bonn 2009

[19] ZIPF, G. K.: *The Psycho-Biology of Languages.* An Introduction to Dynamic Philology Houghton-Mifflin, Boston 1935

Internetquellen

[20] QUASTHOFF, U.: Information Retrieval, Vorlesung, Sommersemester 2010, Universität Leipzig. `http://asv.informatik.uni-leipzig.de/courses/57` Letzter Zugriff: 04.01.2011

[21] QUASTHOFF, U.: Text Mining, Vorlesung, Wintersemester 2009/2010, Universität Leipzig `http://asv.informatik.uni-leipzig.de/courses/41` Letzter Zugriff: 08.01.2011

[22] Das Projekt Deutscher Wortschatz `http://wortschatz.uni-leipzig.de/` Letzter Zugriff: 04.01.2011

[23] Die Menschenrechtscharta der Vereinten Nationen in 370 Sprachen. `http://www.ohchr.org/EN/UDHR/Pages/SearchByLang.aspx` Letzter Zugriff: 05.01.2011

[24] Wikipedia: ISO 639 `http://de.wikipedia.org/wiki/ISO_639`

[25] Auflistung der Makrosprachen nach ISO 639-3 `http://www.sil.org/iso639-3/macrolanguages.asp` Letzter Zugriff: 10.01.2011

[26] Auflistung der Einzelsprachen nach ISO 639-3 `http://www.sil.org/iso639-3/codes.asp` Letzter Zugriff: 11.01.2011

A. Anhang

A.1. Sprachen für die Klassifikation

Code	Bezeichnung	Code	Bezeichnung
abk	Abkhaz	ace	Aceh
acu	Achuar-Shiwiar	ada	Dangme
afr	Afrikaans	agr	Aguaruna
aii	Assyrian Neo-Aramaic	ajg	Aja
aka	Akan	als	Albanian, Tosk
amc	Amahuaca	ame	Yanesha
amh	Amharic	amr	Amarakaeri
ara	Arabic	arb	Arabic, Standard
arl	Arabela	arn	Mapudungun
ast	Asturian	auv	Auvergnat
ayr	Aymara, Central	azj	Azerbaijani
bam	Bamanankan	ban	Bali
bba	Baatonum	bci	Baoulé
bcl	Bicolano, Central	bel	Belarusan
bem	Bemba	ben	Bengali
bho	Bhojpuri	bin	Edo
bis	Bislama	blu	Hmong Njua
boa	Bora	bod	Tibetan, Central
bos	Bosnian	bre	Breton
btb	Beti	bug	Bugis
bul	Bulgarian	cab	Garifuna
cak	Kaqchikel, Central	cat	Catalan-Valencian-Balear
cbr	Cashibo-Cacataibo	cbs	Cashinahua
cbt	Chayahuita	cbu	Candoshi-Shapra
ccx	Zhuang, Northern	ceb	Cebuano
ces	Czech	cha	Chamorro
chj	Chinantec, Ojitlán	chk	Chuukese
chr	Cherokee	cic	Chickasaw
cjk	Chokwe	ckb	Kurdish, Central
cmn	Chinese, Mandarin	cnh	Chin, Haka
cni	Asháninka	cof	Colorado
cos	Corsican	cot	Caquinte
cpu	Ashéninka, Pichis	crs	Seselwa Creole French
csa	Chinantec, Chiltepec	csw	Cree, Swampy
ctd	Chin, Tedim	cym	Welsh
dag	Dagbani	dan	Danish
ddn	Dendi	deu	German
dga	Dagaare, Southern	dip	Dinka, Northeastern
div	Maldivian	dyo	Jola-Fonyi

`dzo`	Dzongkha	`ell`	Greek
`emk`	Maninkakan, Eastern	`eml`	Emiliano-Romagnolo
`eng`	English	`epo`	Esperanto
`est`	Estonian	`eus`	Basque
`eve`	Even	`ewe`	Éwé
`fao`	Faroese	`fij`	Fijian
`fin`	Finnish	`flm`	Chin, Falam
`fon`	Fon	`fra`	French
`fri`	Frisian, Western	`fuc`	Pulaar
`fur`	Friulian	`gaa`	Ga
`gag`	Gagauz	`gax`	Oromo, Borana-Arsi-Guji
`gjn`	Gonja	`gkp`	Kpelle, Guinea
`gla`	Gaelic, Scottish	`gle`	Gaelic, Irish
`glg`	Galician	`guc`	Wayuu
`gug`	Guaraní, Paraguayan	`guj`	Gujarati
`gyr`	Guarayu	`hat`	Haitian Creole French
`hau`	Hausa	`haw`	Hawaiian
`hea`	Hmong, Northern Qiandong	`heb`	Hebrew
`hil`	Hiligaynon	`hin`	Hindi
`hms`	Hmong, Southern Qiandong	`hna`	Mina
`hni`	Hani	`hrv`	Croatian
`hsb`	Sorbian, Upper	`hun`	Hungarian
`hus`	Huastec, Veracruz	`huu`	Huitoto, Murui
`hva`	Huastec, San Luís Potosí	`hye`	Armenian
`ibb`	Ibibio	`ibo`	Igbo
`ido`	Ido	`iii`	Yi, Sichuan
`ike`	Inuktitut, Eastern Canadian	`ilo`	Ilocano
`ina`	Interlingua	`ind`	Indonesian
`isl`	Icelandic	`ita`	Italian
`jav`	Javanese	`jpn`	Japanese
`kal`	Inuktitut, Greenlandic	`kan`	Kannada
`kat`	Georgian	`kaz`	Kazakh
`kbp`	Kabiyé	`kde`	Makonde
`kea`	Kabuverdianu	`kek`	Q'eqchi'
`khk`	Mongolian, Halh	`khm`	Khmer, Central
`kin`	Rwanda	`kir`	Kirghiz
`kmb`	Mbundu	`knc`	Kanuri, Central
`kng`	Koongo	`koo`	Konjo
`kor`	Korean	`kqn`	Kaonde
`kri`	Krio	`ktu`	Kituba
`lao`	Lao	`lat`	Latin
`lav`	Latvian	`lia`	Limba, West-Central
`lin`	Lingala	`lit`	Lithuanian
`lnc`	Languedocien	`lns`	Lamnso'

`loz`	Lozi	`ltz`	Luxembourgeois
`lua`	Luba-Kasai	`lue`	Luvale
`lug`	Ganda	`lun`	Lunda
`mad`	Madura	`mag`	Magahi
`mah`	Marshallese	`mai`	Maithili
`mal`	Malayalam	`mam`	Mam, Northern
`mar`	Marathi	`maz`	Mazahua Central
`mcd`	Sharanahua	`mcf`	Matsés
`men`	Mende	`mic`	Micmac
`min`	Minangkabau	`miq`	Mískito
`mkd`	Macedonian	`mlt`	Maltese
`mly`	Malay	`mos`	Mòoré
`mri`	Maori	`msa`	msa
`mxi`	Mozarabic	`mxv`	Mixtec, Metlatónoc
`mya`	Burmese	`mzi`	Mazatec, Ixcatlán
`nav`	Navajo	`nba`	Nyemba
`nbl`	Ndebele	`ndo`	Ndonga
`nep`	Nepali	`new`	new
`nhn`	Nahuatl, Central	`nld`	Dutch
`nno`	Norwegian, Nynorsk	`nob`	Norwegian, Bokmål
`not`	Nomatsiguenga	`nso`	Sotho, Northern
`nya`	Nyanja	`nym`	Nyamwezi
`nyn`	Nyankore	`nzi`	Nzema
`oci`	Occitan	`ojb`	Ojibwa, Northwestern
`oss`	Osetin	`ote`	Otomi, Mezquital
`pam`	Pampangan	`pan`	Panjabi, Eastern
`pau`	Palauan	`pbb`	Páez
`pbu`	Pashto, Northern	`pcd`	Picard
`pcm`	Pidgin, Nigerian	`pdc`	German, Pennsylvania
`pes`	Farsi, Western	`pis`	Pijin
`plt`	Malagasy, Plateau	`pnb`	Panjabi, Western
`pol`	Polish	`pon`	Pohnpeian
`por`	Portuguese	`pov`	Crioulo, Upper Guinea
`ppl`	Pipil	`prq`	Ashéninka Perené
`prv`	Provençal	`quc`	K'iche', Central
`qud`	Quichua, Calderón Highland	`quy`	Quechua, Ayacucho
`quz`	Quechua, Cusco	`qva`	Quechua, Ambo-Pasco
`qvc`	Quechua, Cajamarca	`qvh`	Quechua, Huamalíes-Dos de Mayo Huánuco
`qvm`	Quechua, Margos-Yarowilca-Lauricocha	`qvn`	Quechua, North Junín
`qwh`	Quechua, Huaylas Ancash	`qxa`	Quechua, Chiquián Ancash
`qxn`	Quechua, Northern	`qxu`	Quechua, Arequipa-

	Conchucos Ancash		La Unión
`rar`	Rarotongan	`rmn`	Romani, Balkan
`rmy`	Romani, Vlax	`roh`	Romansch
`ron`	Romanian	`run`	Rundi
`rus`	Russian	`sag`	Sango
`san`	Sanskrit	`sco`	Scots
`shp`	Shipibo-Conibo	`skr`	Seraiki
		`slk`	Slovak
`slv`	Slovenian	`sme`	Saami, North
`smo`	Samoan	`sna`	Shona
`snk`	Soninke	`som`	Somali
`sot`	Sotho, Southern	`spa`	Spanish
`src`	Sardinian, Logudorese	`srp`	Serbian
`srr`	Serer-Sine	`ssw`	Swati
`suk`	Sukuma	`sun`	Sunda
`sus`	Susu	`swe`	Swedish
`swh`	Swahili	`tah`	Tahitian
`taj`	Tamang, Eastern	`tam`	Tamil
`tat`	Tatar	`tbz`	Ditammari
`tca`	Ticuna	`tem`	Themne
`tet`	Tetun	`tgk`	Tajiki
`tgl`	Tagalog	`tha`	Thai
`tir`	Tigrigna	`tiv`	Tiv
`tob`	Toba	`toi`	Tonga
`toj`	Tojolabal	`ton`	Tongan
`top`	Totonac, Papantla	`tpi`	Tok Pisin
`tsn`	Tswana	`tso`	Tsonga
`tsz`	Purepecha	`tuk`	Turkmen
`tur`	Turkish	`tzc`	Tzotzil, Chamula
`tzh`	Tzeltal, Oxchuc	`tzm`	Tamazight, Central Atlas
`uig`	Uyghur	`ukr`	Ukrainian
`umb`	Umbundu	`ura`	Urarina
`urd`	Urdu	`uzn`	Uzbek, Northern
`vai`	Vai	`ven`	Venda
`vie`	Vietnamese	`vmw`	Makhuwa
`war`	Waray-Waray	`wln`	Walloon
`wol`	Wolof	`wwa`	Waama
`xho`	Xhosa	`xsm`	Kasem
`yad`	Yagua	`yao`	Yao
`yap`	Yapese	`ydd`	Yiddish, Eastern
`ykg`	Yukaghir, Northern	`yor`	Yoruba
`yua`	Maya, Yucatán	`zam`	Zapotec, Miahuatlán
`zho`	Chinese	`ztu`	Zapotec, Güilá

A.2. Überschneidungen von Sprachen

$f_{L_i,L_j}(100)$ $f^{\mathrm{Bi}}_{L_i,L_j}(100)$ $f^{\mathrm{Tri}}_{L_i,L_j}(100)$	eng	sco	deu	afr	nld	fra	spa	por
eng	1.0	0.511	0.068	0.179	0.138	0.082	0.055	0.071
	1.0	0.832	0.48	0.653	0.677	0.389	0.503	0.575
	1.0	0.517	0.072	0.128	0.188	0.038	0.151	0.119
sco	0.411	1.0	0.042	0.071	0.051	0.023	0.095	0.116
	0.659	1.0	0.517	0.576	0.71	0.366	0.518	0.665
	0.253	1.0	0.086	0.086	0.286	0.047	0.098	0.276
deu	0.063	0.049	1.0	0.127	0.123	0.016	0.017	0.031
	0.546	0.851	1.0	0.691	0.82	0.528	0.633	0.746
	0.082	0.195	1.0	0.274	0.37	0.031	0.088	0.267
afr	0.153	0.128	0.281	1.0	0.578	0.128	0.076	0.037
	0.465	0.806	0.688	1.0	0.87	0.563	0.548	0.708
	0.059	0.057	0.276	1.0	0.65	0.055	0.06	0.065
nld	0.054	0.031	0.048	0.52	1.0	0.168	0.168	0.09
	0.527	0.787	0.565	0.734	1.0	0.457	0.536	0.645
	0.117	0.169	0.147	0.331	1.0	0.013	0.108	0.187
fra	0.022	0.014	0.042	0.167	0.154	1.0	0.35	0.216
	0.481	0.816	0.635	0.715	0.664	1.0	0.767	0.776
	0.059	0.11	0.133	0.119	0.12	1.0	0.252	0.268
spa	0.055	0.049	0.016	0.265	0.225	0.54	1.0	0.376
	0.427	0.79	0.494	0.69	0.634	0.762	1.0	0.914
	0.047	0.15	0.036	0.083	0.146	0.223	1.0	0.47
por	0.127	0.13	0.044	0.105	0.074	0.241	0.38	1.0
	0.479	0.735	0.538	0.6	0.674	0.55	0.665	1.0
	0.028	0.252	0.102	0.046	0.169	0.12	0.232	1.0

Tabelle A.2: Sprachüberlappungen gemäß dem Ähnlichkeitsmaß f_{L_i,L_j} einiger verwandter Sprachen bei $k = 100$ Stopwörtern. An dieser Stelle wurden ebenso die Ähnlichkeiten bei Bi- und Trigramme dargestellt.

$f_{L_i,L_j}(500)$ $f^{\mathrm{Bi}}_{L_i,L_j}(500)$ $f^{\mathrm{Tri}}_{L_i,L_j}(500)$	eng	sco	deu	afr	nld	fra	spa	por
eng	1.0	0.442	0.079	0.168	0.12	0.09	0.05	0.054
	1.0	0.977	0.917	0.917	0.921	0.91	0.823	0.818
	1.0	0.707	0.427	0.435	0.377	0.362	0.348	0.347
sco	0.341	1.0	0.038	0.061	0.057	0.034	0.074	0.09
	0.975	1.0	0.918	0.892	0.895	0.905	0.836	0.854
	0.62	1.0	0.376	0.386	0.448	0.433	0.407	0.467
deu	0.067	0.045	1.0	0.112	0.111	0.036	0.015	0.022
	0.924	0.918	1.0	0.884	0.895	0.872	0.862	0.847
	0.51	0.578	1.0	0.463	0.581	0.395	0.458	0.461
afr	0.131	0.112	0.241	1.0	0.545	0.114	0.073	0.03
	0.944	0.932	0.932	1.0	0.961	0.837	0.829	0.844
	0.384	0.526	0.457	1.0	0.773	0.439	0.344	0.279
nld	0.043	0.042	0.083	0.442	1.0	0.13	0.141	0.065
	0.898	0.887	0.911	0.905	1.0	0.798	0.826	0.835
	0.444	0.533	0.462	0.651	1.0	0.314	0.357	0.38
fra	0.041	0.025	0.071	0.178	0.131	1.0	0.3	0.169
	0.941	0.926	0.886	0.888	0.852	1.0	0.904	0.936
	0.558	0.584	0.335	0.36	0.368	1.0	0.471	0.53
spa	0.05	0.04	0.014	0.212	0.189	0.438	1.0	0.333
	0.943	0.931	0.9	0.861	0.856	0.941	1.0	0.959
	0.497	0.483	0.26	0.319	0.349	0.502	1.0	0.676
por	0.092	0.096	0.031	0.094	0.052	0.169	0.352	1.0
	0.902	0.892	0.859	0.84	0.833	0.916	0.905	1.0
	0.428	0.571	0.301	0.328	0.393	0.428	0.622	1.0

Tabelle A.3: Sprachüberlappungen gemäß dem Ähnlichkeitsmaß f_{L_i,L_j} einiger verwandter Sprachen bei $k = 500$ Stopwörtern. Ebenso mit der Betrachtung der Sprachähnlichkeiten bzgl. Bi- und Trigrammen.

A.3. Einzigartige Zeichen von Sprachen

In der nachfolgenden Darstellung sind alle Sprachen aufgelistet, die bezüglich aller Trainingssprachen einzigartige Zeichen besitzen. Aus der Liste ist gut ersichtlich, dass die Identifikation für Koreanisch, Chinesisch und Japanisch besonders gut mit dem Unigrammverfahren möglich ist, da diese die mit Abstand meisten einzigartigen Zeichen besitzen.

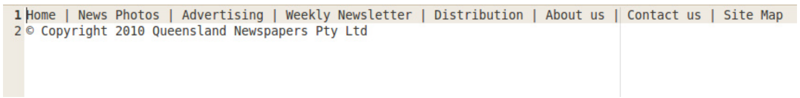
abk: ҿ, ҷ, ҵ, ҽ, ҟ, ҧ, Ӡ, Ҭ, ҩ,
aii: ܚ, ܗ, ܣ, ܠ, ܨ, ܢ, ܬ, ܓ̰, ܝ, ܘ, ܫ, ܦ݂, ܓ, ܟ, ܐ, ܡ, ܪ, ܒ, ܟ݂, ܦ, ܒ݂, ܩ,
arb: إ, ة,
azj: ҝ, ҹ,
bci: ɑ,
bel: ў,
ben: ঞ, ঢ, ষ, দ, ছ, ন, চ, স, এ, ড, থ, ত, ই, য, প, ব, অ, গ, শ, হ, য়, ঘ, ও, ভ, ট, ধ, আ, ক, ম, খ, ণ, ঠ, ঙ, ল, জ, ফ, র, উ,
boa: Ȉ,
bos: Њ, Ђ,
bre: Ù,
btb: Ń,
bul: Щ, ъ,
ces: ě, ů, ř,
chr: Ꮷ, Ꮼ, Ꮩ, Ꮄ, Ᏸ, Ꮝ, Ꮻ, Ꮿ, Ꮸ, Ꮃ, Ꮪ, Ꮜ, Ꮑ, Ꭰ, Ꮭ, Ꮒ, Ꮊ, Ꮦ, Ꮓ, Ꮞ, Ꮨ, Ꮎ, Ꮏ, Ꭻ, Ꮧ, Ꮥ, Ꮲ, Ꭱ, Ꮁ, Ꮬ, Ꭷ, Ꮯ, Ꮢ, Ꮹ, Ꮾ, Ꭸ, Ꭼ, Ꭲ,
csw: ᐡ, ᐧᓭ, ᐧᒉ, ᐣ, ᐧᒪ, ᔕ, ᑦ, ᐟ, ᐢ, ᐠ, ᐧᓇ, ᐧᔕ, ᑊ, ᐧᑌ,
deu: ß,
div: [illegible]
ell: ώ, ί, ό, ὑ, Ἡ, ἱ, ή, ῆ, ἐ, ά, ϊ, ὼ, ἢ, ἔ, ὔ, Ἐ, ί, Ὅ, ὺ, ώ, ἴ, ό, ὲ, ἄ, ἕ, ὥ, ἰ, έ, ὅ, ὁ, ῖ, ῦ, ὸ, ἑ, ψ, ὰ, ὶ, ὐ, ὡ, ἷ, ἡ, ῶ, ᾶ, ύ, ἂ, ὴ, ἀ,
epo: ĵ, Ĉ, ŭ, ŝ, ĉ, Ĝ, Ŝ,
eve: θ, Ӈ,
ewe: Ƒ, ƒ, ʋ,
fao: Ð, Ø,
fon: ǔ,
gag: ţ,
glg: ª,
guj: ડ, થ, ઝ, દ, ળ, એ, હ, ઞ, વ, ક, ન, ષ, ઉ, ભ, ઢ, સ, ઠ, ધ, ત, ણ, આ, ઊ, ફ, પ, ઘ, ચ, ખ, ય, શ, છ, અ, ગ, મ, જ, ઓ, લ, ટ, ઈ, ર, બ,
hin: ज़, क़,
hna: ǒ,
hun: ő,
huu: Ɨ,
hye: օ, Ա, Օ, Ի, Ն, ծ, մ, Գ, Պ, վ, տ, Դ, ն, ր, Զ, Լ, գ, զ, ե, ո, ժ, փ, յ, Տ, լ, կ, Ղ, ս, Կ, ղ, թ, ռ, Ջ, Ժ, է, ձ, ջ, ւ, ի, ը, ա, պ, Խ, Ո, Ը, ֆ, շ, Վ, Է, Ե, ց, Ք, Ս, բ, Բ, Թ, Մ, Ճ, ճ, Ռ, Ր, հ, խ, Յ, չ, Ձ, դ,
ibo: Ụ, ụ, ị, Ị, Ọ, ọ,
ike: ᒐ, ᖕ, ᓵ, ᑉ, ᓅ, ᕈ, ᕕ, ᙱ, ᓈ, ᖅ, ᐳ, ᓰ, ᖔ, ᔫ, ᔪ, ᑑ, ᐆ, ᒡ, ᖂ, ᑰ, ᕋ, ᖏ, ᕿ, ᓪ, ᓇ, ᑖ, ᙵ, ᓗ, ᓚ, ᖀ, ᓅ, ᓛ, ᒌ, ᕗ, ᒨ, ᕙ, ᐋ, ᖁ, ᐹ, ᕐ, ᒦ, ᒫ, ᔮ, ᖑ, ᓵ, ᒎ, ᖃ, ᖓ, ᒍ, ᓃ, ᓘ, ᑏ,
isl: Þ, þ,
jpn: 末, カ, 先, 万, 購, の, ー, 登, 配, る, ン, 件, 円, を, 介, 千, 確, ザ, だ, 戦, 台, 紹, か, 売, サ, き, 査, 説, な, 題, 料, 期, 環, く, 意, ば, 氏, 県, て, 明, 機, ケ, う, 版, ほ, リ, 男, ム, 資, 契, 業, 販, じ, 調, 女, 般, 追, つ, レ, 続, ロ, 阪, 広, み, 側, バ, が, 援, 応, デ, ユ, 場, 認, 索, 報, ウ, ち, で, そ, 連, び, ド, こ, ゲ, 術, ス, 導, 予, コ, 真, り, せ, 状, 代, 担, タ, 営, 検, モ, 録, あ, 約, セ, 問, 私, 容, 法, 製, ん, 際, し, 見, 現, 対, 係, ビ, 気, よ, 株, 動, け, 店, た, 度, ェ, べ, 実, 別, わ, メ, 携, 関, は, 東, 運, ら, す, 載, ズ, オ, に, 歳, ネ, ア, ッ, 思, 施, 端, 象, ね, い, 設, 昨, れ, さ, ご, 利, ソ, 帯, 考, 変, も, ワ, と, 車, 疑, っ, 電, 結, 商, 搭, 発, ず, 型, グ, 時, ろ, イ, ペ, 価, 感, ま, 間, ョ, 参, フ, 化, 米, 操, お, ブ, め, 従, ル, ト, ジ, 長, 記, 写, プ, 開, 画, テ, 響, ょ, 強, ツ, 催, 無, 億, ど, ァ, 採, へ, シ, ポ, ク, え, や, 後, 話, 況,
kan: ರ, ಜ, ಬ, ಭ, ಹ, ಸ, ಪ, ಇ, ಳ, ಶ, ಉ, ವ, ಣ, ಟ, ಧ, ದ, ಖ, ಠ, ಡ, ಕ, ನ, ಅ, ಮ, ಗ, ಫ, ಛ, ಆ, ಘ, ಚ, ಒ, ಈ, ಞ, ಥ, ತ, ಎ, ಷ, ಲ, ಢ, ಯ,

kat: ღ, ე, შ, ა, დ, ლ, ს, ჩ, რ, წ, ჭ, ი, ზ, პ, ბ, ძ, ქ, თ, ც, მ, ყ, ვ, ო, ფ, ჯ, ჰ, ხ, ნ, გ, კ, ტ, უ,
kaz: Ғ, ү, Ұ,
kbp: Ʊ, Ɣ,
khk: Ү,
kor: 법, 서, 던, 포, 장, 큰, 라, 차, 태, 섭, 년, 화, 발, 산, 비, 특, 내, 합, 제, 지, 해, 받, 같, 생, 높, 자, 했, 안, 쳐, 과, 아, 든, 끼, 려, 실, 글, 더, 쓰, 권, 였, 탕, 닌, 냐, 옛, 투, 인, 래, 등, 중, 행, 났, 름, 항, 칙, 키, 문, 덕, 듬, 멸, 함, 된, 으, 불, 출, 만, 령, 난, 익, 뒤, 릴, 위, 마, 민, 음, 범, 따, 리, 느, 응, 교, 립, 정, 수, 없, 그, 죄, 유, 짐, 박, 터, 당, 요, 적, 죽, 개, 작, 와, 명, 즉, 질, 에, 예, 굴, 떠, 까, 신, 필, 잘, 달, 랑, 회, 황, 이, 규, 히, 분, 트, 도, 성, 을, 류, 린, 올, 첫, 족, 토, 표, 놓, 시, 밀, 반, 엄, 변, 면, 니, 편, 쿄, 보, 완, 육, 었, 통, 욕, 것, 독, 데, 진, 넷, 런, 별, 았, 세, 사, 용, 섬, 각, 악, 대, 호, 다, 탈, 하, 란, 관, 울, 거, 뿐, 존, 바, 로, 물, 체, 때, 최, 의, 재, 설, 컴, 영, 계, 러, 널, 월, 업, 후, 직, 할, 또, 축, 르, 입, 며, 평, 두, 연, 날, 약, 되, 게, 미, 양, 방, 백, 강, 은, 타, 충, 뜻, 누, 는, 원, 배, 격, 및, 럽, 쟁, 경, 갖, 져, 전, 퓨, 공, 혹, 술, 김, 들, 속, 준, 께, 금, 동, 치, 떤, 가, 여, 나, 소, 녀, 간, 언, 롭, 담, 막, 초, 읽, 알, 일, 돌, 현, 야, 본, 한, 능, 택, 구, 프, 말, 결, 력, 핍, 무, 우, 복, 처, 를, 총, 선, 상, 군, 단, 고, 견, 매, 향, 학, 국, 철, 람, 남, 역, 른, 새, 념, 조, 크, 록, 망, 앞, 많, 모, 주, 활, 목, 건, 종, 노, 기, 스, 임, 운, 부, 접, 있, 근, 심, 률, 어, 않,
kri: ŋ̱,
lav: Ē, ģ, ķ, Ģ, ļ, ņ,
lit: ė, ų, Ų, Ė,
mal: ഷ, ള, ല, യ, ഭ, ഖ, ഊ, ഇ, ം, ന, റ, ഒ, സ, ഥ, ദ, ഉ, ശ, ഘ, ധ, ഈ, ച, ഴ, ക, മ, ണ, ഏ, ത, വ, ഗ, ബ, ങ, അ, ഐ, ആ, പ, ട, ഹ, ജ, ര, ഞ, എ,
mar: ळ,
mkd: ѓ, ќ,
mlt: ċ, Ħ, Ġ, ġ, ħ,
mly: ڽ, ڤ, ڬ, ڠ,
nav: ǫ́,
ojb: ᐧᔕ, ᓭ, ᐧᐱ, ᐧᑭ, ᐧᐹ, ᔥ,
oss: æ,
pan: ਛ, ਣ, ਧ, ਐ, ਸ, ੳ, ਫ਼, ਢ, ਲ, ਯ, ਜ, ਅ, ਘ, ਭ, ਕ, ਮ, ਟ, ਹ, ਝ, ੜ, ਜ਼, ਦ, ਡ, ਚ, ਬ, ਰ, ਉ, ਈ, ਔ, ਵ, ਗ, ਪ, ਖ, ਤ, ਊ, ਠ, ਏ, ਸ਼, ਨ, ਥ, ਫ, ਆ, ਇ,
pbu: ټ, ځ, ی, ړ,
pol: Ź,
rmn: ɵ,
sag: Ï,
san: ऽ,
skr: ڄ, ٻ, ٽ, ݙ, ڳ,
slk: Ž, ľ,
sme: ŧ,
srp: ћ,
tam: ங, ஏ, த, ல, ற, எ, உ, ன, ஐ, ந, இ, ஒ, ம, ச, அ, ள, ழ, ஊ, ஸ, ஆ, ப, ர, ஓ, ட, ண, ய, க, ஞ, வ,
tca: Ĩ, Ẽ, ẽ,
tem: ʌ,
tgk: ӣ, ӯ,
tir: ለ, የ, ዊ, ዕ, ያ, ቋ, ብ, ቴ, ሉ, ጠ, ጎ, ኢ, ፃ, ቀ, ቤ, ዋ, ደ, ባ, ሻ, ጋ, ስ, ኪ, ፁ, ኡ, ዖ, ም, ገ, ራ, ኸ, ኻ, ጣ, ከ, ኤ, ዮ, ሲ, ላ, ት, ኩ, ሽ, ሳ, ኝ, ፂ, ሄ, ሎ, ጭ, ቻ, ኹ, ጥ, ቶ, ጅ, ክ, ተ, ቓ, ቂ, ዜ, ዲ, ወ, ነ, ጃ, ዌ, ኾ, ው, ና, ፍ, ዪ, ቅ, መ, ዳ, ቐ, ቲ, ድ, ሂ, እ, ቄ, ኟ, ሮ, ኦ, ዚ, ሓ, ቹ, ፖ, ጊ, ዑ, ፅ, ዩ, ዛ, ኺ, ሚ, ቱ, ቢ, ኖ, ሃ, ዘ, ይ, ቡ, ል, ዓ, ታ, ካ, ሞ, ኮ, ሩ, ኛ, ዒ, ማ, ፈ, ን, ኽ, ዝ, ዶ, ሊ, ፀ, ሪ, ዎ, በ, ኑ, ቦ, ግ, ር, ሀ, ዘ, ሒ, ጢ, ሕ, ቾ, ጉ, ቁ, ኒ, ሰ,
tob: Ỹ, ỹ,
tuk: Ş, Ň,
uig: ە, ۋ, ۆ, ڭ, ۈ, ۇ,
ukr: ї, є,

```
urd: ۓ,
uzn_latn: ‘,
ven: ṱ, ḽ, ṅ, ṋ, ḓ,
vie: ơ, Đ, ư,
wln: Û, È,
ydd: ױ ,ײ ,װ,
yor: Ì,
zho: 也，不，际，问，需，甚，队，买，话，工，收，号，占，根，达，三，游，选，去，
整，是，望，外，括，其，究，被，济，打，带，决，等，太，已，股，机，持，让，诉，比，
司，海，门，跟，知，主，子，有，续，价，此，络，里，城，所，够，就，说，钟，们，及，
较，无，之，间，注，首，银，半，断，术，记，或，孩，发，只，销，但，还，结，我，教，
包，二，些，业，调，息，经，球，而，未，至，划，特，车，季，常，样，想，给，存，强，
才，析，该，赛，集，后，原，设，非，超，解，都，建，现，觉，产，售，然，户，牌，并，
和，增，程，关，她，低，虽，到，计，系，得，却，为，开，证，路，拿，题，场，认，平，
天，涨，仍，张，获，统，因，美，网，每，于，总，走，又，正，吃，钱，史，种，券，什，
团，生，研，继，况，连，汽，如，且，岁，基，条，直，水，求，你，将，曾，进，呢，务，
少，四，近，别，媒，那，看，由，好，北，时，了，己，动，么，把，资，功，安，过，再，
称，险，没，这，长，个，重，道，实，像，历，显，风，副，构，小，服，应，几，着，戏，
起，两，则，很，候，相，做，难，学，项，育，线，择，须，据，专，与，它，校，员，对，
环，响，希，从，
```

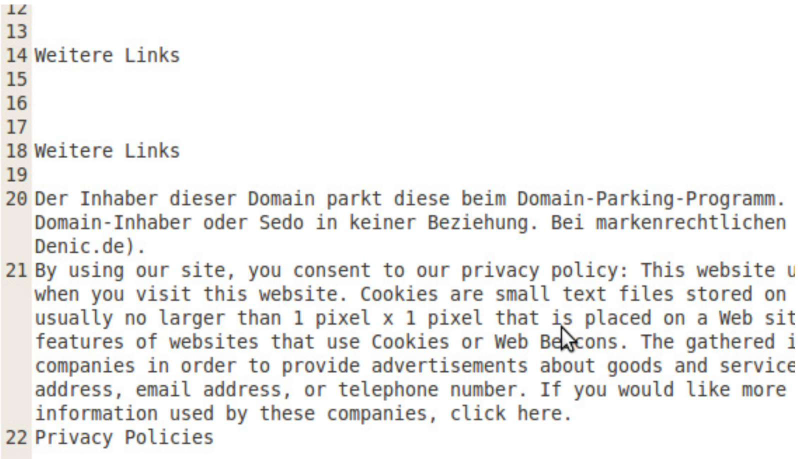
A.4. Beispieldokumente

In diesem Abschnitt werden einige wenige exemplarische Dokumente präsentiert, die im Laufe der Arbeit angesprochen wurden. Zum Einen sollen die zur Klassifikation bereitstehenden FindLinks-Dokumente und zum Anderen Referenzdokumente der Testquellen gezeigt werden, bei denen das Programm eine fehlerhafte Identifikation durchführte.



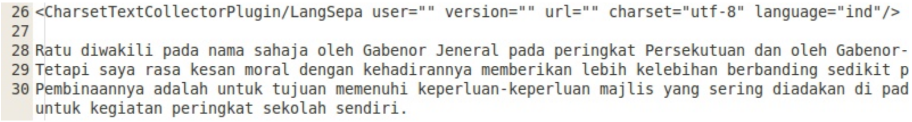
```
Home | News Photos | Advertising | Weekly Newsletter | Distribution | About us | Contact us | Site Map
© Copyright 2010 Queensland Newspapers Pty Ltd
```

Abbildung A.1: Ein englischsprachiges Dokument, das lediglich ein Menü und Copyrightinformationen enthält. Das Stopwortverfahren schlägt aufgrund von mangelnden trainierten Wortformen nicht an.



```


Weitere Links



Weitere Links

Der Inhaber dieser Domain parkt diese beim Domain-Parking-Programm.
Domain-Inhaber oder Sedo in keiner Beziehung. Bei markenrechtlichen
Denic.de).
By using our site, you consent to our privacy policy: This website u
when you visit this website. Cookies are small text files stored on
usually no larger than 1 pixel x 1 pixel that is placed on a Web sit
features of websites that use Cookies or Web Beacons. The gathered i
companies in order to provide advertisements about goods and service
address, email address, or telephone number. If you would like more
information used by these companies, click here.
Privacy Policies
```

Abbildung A.2: Für dieses Dokument wurde in den Nachrichtenseiten die Referenz Spanisch gesetzt. Unschwer zu erkennen, handelt es sich um ein mehrsprachiges Dokument mit deutschem und englischem Inhalt.



```
<CharsetTextCollectorPlugin/LangSepa user="" version="" url="" charset="utf-8" language="ind"/>

Ratu diwakili pada nama sahaja oleh Gabenor Jeneral pada peringkat Persekutuan dan oleh Gabenor-
Tetapi saya rasa kesan moral dengan kehadirannya memberikan lebih kelebihan berbanding sedikit p
Pembinaannya adalah untuk tujuan memenuhi keperluan-keperluan majlis yang sering diadakan di pad
untuk kegiatan peringkat sekolah sendiri.
```

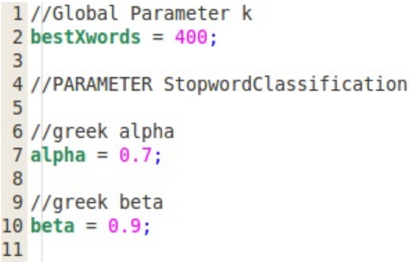
Abbildung A.3: Dieses Dokument stammt aus der Malaiischen Satzdatenbank von aspra20. Es wurde Indonesisch als Sprache identifiziert. Das liegt einerseits an der Nähe der beiden Sprachen und andererseits an der Kürze des vorliegenden Dokuments. Aufgrund der Dominanz von Stopworten des Indonesischen wird Indonesisch gewählt.

```
<CharsetTextCollectorPlugin/LangSepa user="Quix0r" version="
language="eng,msa"/>

        Price : RM 50.00 (plus RM 5.00 for normal postage or
 Orders may be placed by contacting :
 En. Ramlan Ramli at ramlan@nstp.com.my
 En. Mohd Isam at isam@nstp.com.my
 Telephone No : 603-2056 9318 or
Gambar-gambar yang dibeli dari laman web ini adalah
untuk kegunaan peribadi sahaja.
```

Abbildung A.4: Dieses Dokument entstammt der FindLinks-Untersuchung aus dem Ergebnisteil. Diesem Dokument wurden die Sprachen Englisch und Malaiisch zugeordnet, obwohl eine Tendenz zu nicht brauchbarem Material existiert.

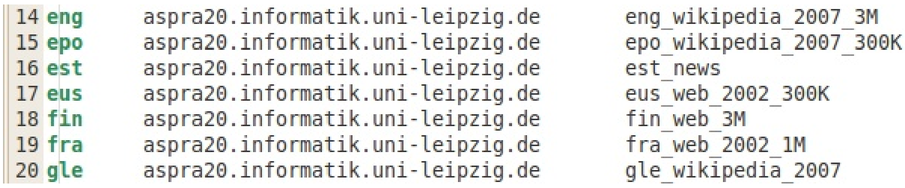
A.5. Konfiguration des Programms *LangSepa*

Die Konfiguration und Parametrisierung des Programms wurden aufgrund der Fülle an Optionen in Konfigurationsdateien ausgelagert. Dem Nutzer ist es möglich in der Datei `langsepa.ini` sämtliche Parameter zu setzen.
Die Datei `db-sources.ini` enthält alle Datenbanken mit den zugehörigen Sprachen und der Serveradressen, aus denen die Wortlisten bezogen werden. Das Programm ist durch den Kommandozeilenbefehl `java -jar LangSepa.jar` initiierbar.



```
//Global Parameter k
bestXwords = 400;

//PARAMETER StopwordClassification

//greek alpha
alpha = 0.7;

//greek beta
beta = 0.9;

```

Abbildung A.5: Ausschnitt aus der Datei `langsepa.ini`, in der die Konfiguration der Parameter des Programms vorgenommen werden kann.



```
eng      aspra20.informatik.uni-leipzig.de        eng_wikipedia_2007_3M
epo      aspra20.informatik.uni-leipzig.de        epo_wikipedia_2007_300K
est      aspra20.informatik.uni-leipzig.de        est_news
eus      aspra20.informatik.uni-leipzig.de        eus_web_2002_300K
fin      aspra20.informatik.uni-leipzig.de        fin_web_3M
fra      aspra20.informatik.uni-leipzig.de        fra_web_2002_1M
gle      aspra20.informatik.uni-leipzig.de        gle_wikipedia_2007
```

Abbildung A.6: Ausschnitt aus der Datei `db-sources.ini`, in der die zu importierenden Datenbanken von der aspra20 Datenbank konfiguriert werden kann.

Variablen- und Parameterverzeichnis

Parameter

k	Anzahl der zu importierenden höchstfrequenten Wörter pro Sprache
w	Fensterbreite in Anzahl Zeichen bei n-Gramm Klassifikation
α	maximale relative Abweichung einer Sprache von der Optimalüberdeckung u_{opt}, Hauptbedingung für die Durchführung der Stopwortklassifikation
β	maximale relative Abweichung einer Sprache von der Optimalüberdeckung u_{opt}, um in die Kandidatenliste der Stopwortklassifikation zu gelangen.
γ	maximaler Anteil fremdsprachlich induzierter Überdeckung, Grenzwert für den Ausschluss von der Klassifikation
δ	minimale Anzahl eigener Wörter gegenüber jeder anderen Sprache in einem Dokument, um klassifiziert zu werden
κ	Mindestanzahl von Buchstaben aus $\tilde{\mathfrak{L}}$, um Unigramm-Klassifikation zu initiieren
λ	Mindestanzahl von n-Gramm Sequenzen, in denen eine Sprache in n-Gramm Klassifikationen als Wahrscheinlichste hervorgegangen sein muss, um klassifiziert zu werden
μ	Mindestanzahl der zur Sprache gehörigen n-Gramme in einer n-Gramm-Sequenz der Länge w

Variablen

$A(d)$	Menge aller Wörter, die im Dokument d erscheinen
$B(L)$	Menge aller Wörter, der Worthäufigkeitsliste zur Sprache L
$B_k(L)$	Menge der höchstfrequenten k Wörter der Worthäufigkeitsliste zur Sprache L
$B_{T,k}(L)$	Menge aller Trigramme, die auf Basis der höchstfrequenten Wörter der Worthäufigkeitsliste zur Sprache L erzeugt worden
c	ZIPFsche Konstante
$\bar{c}$	Mittelwert über alle gemessenen c aus einer Worthäufigkeitsliste
$D(t)$	Menge aller Sprachen, die das Trigramm t enthalten
$\mathfrak{D}$	Menge aller Dokumente

F	F-Measure, das harmonische Mittel aus p und r
$f_{L_i,L_j}(k)$	Ähnlichkeitsmaß zwischen den Sprachen L_i und L_j bezüglich der höchstfrequenten k Wörter
$f^{\mathrm{Bi}}_{L_i,L_j}(k)$	Ähnlichkeitsmaß zwischen den Sprachen L_i und L_j bezüglich der Bigramme, welche auf Basis der höchstfrequenten k Wörter generiert worden
$f^{\mathrm{Tri}}_{L_i,L_j}(k)$	Ähnlichkeitsmaß zwischen den Sprachen L_i und L_j bezüglich der Trigramme, welche auf Basis der höchstfrequenten k Wörter generiert worden
$g(d,k,L_i,L_j)$	Anteil der Überdeckung $u_d(L_i,k)$, die durch Stopwörter von L_j induziert wurde
$H_{\mathrm{abs}}(w,L)$	absolute Häufigkeit des Wortes w in der Worthäufigkeitsliste zur Sprache L
$H_{\mathrm{abs}}(w,d)$	absolute Häufigkeit des Wortes w im Dokument d
$H_{\mathrm{rel}}(w,L)$	relative Häufigkeit des Wortes w in der Worthäufigkeitsliste zur Sprache L
$H_{\mathrm{rel}}(w,d)$	relative Häufigkeit des Wortes w im Dokument d
$\mathfrak{L}$	Menge aller Sprachen
$\tilde{\mathfrak{L}}$	Menge aller Sprachen ohne Worttrennersymbole
L_{I}	wahrscheinlichste Sprache einer n-Gramm-Sequenz
$\ell(d)$	Sprachmenge der Klassifikation zum Dokument d
M	Anzahl der Sprachen
N	Anzahl der Dokumente
$P(L\|S)$	Wahrscheinlichkeit der Sprache L zur n-Gramm-Sequenz S
p	precision
r	recall
S	n-Gramm-Sequenz
s	Summe aller absoluten Häufigkeiten der Wörter einer Sprache
$u_{\mathrm{opt}}(L,k)$	optimale relative Textüberdeckung der Sprache L durch die höchstfrequenten k Wörter der Sprache L
$u_d(L,k)$	gemessene relative Textüberdeckung durch die höchstfrequenten k Wörter der Sprache L im Dokument d

Tabellenverzeichnis

Abbildungsverzeichnis

Eidesstattliche Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann.

Leipzig, den 6. Februar 2011