Technische Universität Wien
Diplomarbeit

WS 2006/07

Text Mining von Songtexten

Bakk.techn. Jerome Penaranda

November 2006

Betreut durch Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Rauber

Danksagung

An dieser Stelle möchte ich mich bei allen Freunden, die mich bei der Erstellung dieser Arbeit auf verschiedene Art und Weise unterstützt haben, bedanken. Besonderer Dank gebührt vor allem meinem Betreuer, Herrn Prof. Dipl.-Ing. Dr.techn. Andreas Rauber, der mir in allen Phasen meiner Arbeit mit Rat und Tat zur Seite gestanden ist. Dank gilt auch Robert Neumayer, der mir den Parallelkorpus für meine Experimente zur Verfügung gestellt hat.

Ich widme die vorliegende Arbeit meinen Eltern, die mir das Studium ermöglicht haben. Danke!

Kurzfassung

In einer Zeit der starken Verbreitung digitaler Musik versucht man mit verschiedenen Techniken die großen Mengen an Musik zu organisieren. Ein bewährtes Mittel ist die Einteilung von Musik in das entsprechende Musikgenre. In dieser Arbeit erfolgt dies durch die Analyse von Musik in Form von Songtexten. Mit Hilfe von Text Categorization Methoden wird ein Ansatz zur automatischen Klassifikation von Songtexten, welche verschiedenen Lyric-Webseiten entnommen wurden, präsentiert. Dazu werden den Songtexten verschiedene Features, welche sowohl inhaltsbasiert als auch strukturbasiert sind, extrahiert. Mit diesen Features wird ein Klassifikator trainiert, welcher dann dem jeweiligen Songtext das entsprechende Musikgenre zuordnet. Bei der Klassifikation kommen Support Vector Machines und der Naive Bayes Klassifikator zum Einsatz.

Die in dieser Arbeit durchgeführten Experimente umfassen die Evaluierung des Klassifikationsprozesses und die Kombination verschiedener Features zur Steigerung der Klassifikationsgenauigkeit. Mit Hilfe der Ergebnisse wird untersucht, wie viele Songtexte zur Definition eines Genres erforderlich sind, wie gut die Klassifikationen ausfallen und welche Featurekombinationen sich am Besten für diesen Ansatz der Songtextklassifikation eignen.

Abstract

The organization of large quantities of music is a common problem in an era, in which there is an increase in the spread of digital music. A well-tried means is the classification in appropriate music genres. In this paper we propose the use of text categorization techniques to classify music in the form of song lyrics, which are present in the internet. In addition, different features, both content-based and structure-based features, are extracted from the song lyrics. With these features a classifier is trained, which then assigns the appropriate music genre to the respective lyrics. Support Vector Machines and Naive Bayes Classifiers are primarily used in such classifications.

We present experiments comprising the evaluation of the classification process and the combination of different features to increase the classification accuracy. On the basis of these experiments, we study how many lyrics are necessary to get good results, which overall performance we can expect for classification and which feature combinations are suitable for the classification of song lyrics.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Die Verbreitung digitaler Musik wächst sehr stark. Folglich sind automatische Analyse-Techniken bzw. Algorithmen, welche die große Menge an Musik zu organisieren versuchen, eine Notwendigkeit. Die Ähnlichkeit beziehungsweise das Genre zwischen Künstlern und Musikstücken zu ermitteln ist der Kern solcher Algorithmen. Sie liefern einen skalierbaren Weg Musik zu indizieren und Vorschläge abzugeben [Logan04]. Es sind bisher viele automatische Techniken zur Bestimmung der Musik- und Künstlerähnlichkeit vorgestellt worden. Berenzweig, Logan, Ellis und Whitman haben verschiedene Verfahren basierend auf akustischen und subjektiven Informationen behandelt, um die Ähnlichkeit zwischen Musikkünstlern festzustellen [Ber03]. Diese Annäherungen basieren auf der Analyse der akustischen Informationen im Audiobereich oder der im Netz gefundenen Metadaten. Obwohl es in bestimmten Fällen zum Erfolg kam, entsprachen die Systeme nicht ganz den Erwartungen der Benutzer. Die Ähnlichkeit beziehungsweise das Genre von Musik kann jedoch auch durch die Analyse von jenen Metadaten erfolgen, welche einer reichen Quelle, dem Internet, entnommen werden können: Songtexten (Song Lyrics).

Die strukturelle und inhaltliche Analyse von Songtexten für die Ermittlung der Musikähnlichkeit ist Kernpunkt dieser Diplomarbeit. Dabei werden eine Methode zur automatischen Klassifikation von Songtexten in Musikgenres und ein Ansatz zur Entwicklung von Features, welche den von gewöhnlichen Webseiten extrahierten Songtexten entnommen werden, vorgeschlagen. Weiters werden die Resultate ausführlich beschrieben, sowie Verbesserungen und Kombinationen der Features eingebracht, mit deren Hilfe die Klassifizierungsrate des Klassifikators noch gesteigert werden kann.

Die Grundidee der in dieser Arbeit präsentierten Verfahren besteht darin, Songtexte in eine Taxonomie von n Musikgenres einzuordnen. Jedes dieser n Genres wird dadurch repräsentiert, dass m Songtexte von typischen Musikkünstlern dieses Genres dazu angegeben werden. Die Klassifikation neuer Songtexte erfolgt durch einen automatischen Klassifikator (Support Vector Machines und Naiver Bayes), der durch die vorgegebenen Songtexte die Konzepte der Genres lernt. Ein Songtext wird dabei durch verschiede-

ne Features beschrieben. Die Funktionsweise des Ansatzes wird durch folgende Schritte beschrieben:

1. Songtexte werden Musikportalen, welche im Internet vorzufinden sind, entnommen.
2. Für das in dieser Arbeit vorgestellte Klassifikationsverfahren kommt eine Menge von n Genres, von welchen ein Musikgenre von m Songtextdokumenten mit Musikkünstlern dieses Genres repräsentiert wird, zum Einsatz. Dabei wird jedem Songtext nur ein Genre zugeordnet.
3. Von den Songtexten werden charakteristische Merkmale (Features) extrahiert.
4. Anhand dieser Features wird ein Klassifikator trainiert.
5. Neue Songtexte können dann mit Hilfe des Klassifikators einer Klasse bzw. einem Genre zugeordnet werden.

Die in dieser Arbeit vorgestellten Ansätze entstammen größtenteils dem Information Retrieval (Informationswiedergewinnung), welches ein Fachgebiet ist, das sich unter anderem mit computergestütztem und inhaltsorientiertem Suchen in Dokumenten beschäftigt.

1.1 Motivation

Im folgenden Kapitel wird erläutert, warum die automatische Textklassifizierung für die Klassifizierung von Songtexten angewendet soll. Dabei wird auf die heutigen Probleme der Musikorganisation näher eingegangen und Gründe, die für die Analyse von Songtexten sprechen, genannt. Weiters wird im letzten Abschnitt die Idee hinter der Textklassifizierung in Genres näher beschrieben.

1.2 Problemstellung

Das Problem der Musikorganisation tritt in einer Zeit, wo jedem der Zugang zu mp3s online möglich ist, mit dem rasanten Zuwachs des Internets immer stärker in den Vordergrund. Je mehr Musikstücke online zur Verfügung gestellt werden, desto schwieriger wird es, den Überblick über die Musikkollektion zu behalten. Eine Lösung für dieses Problem wäre, jedes Musikstück in der Kollektion einem bestimmten Musikgenre zuzuordnen. Jedoch wird beim Download eines Musikstückes oft keine Metadaten über dessen Genre mitgegeben. Seither steigt die Notwendigkeit von Systemen, die in der Lage sind zu einem Musikstück das entsprechende Musikgenre anzugeben.

Mit dem schnellen Zuwachs des Internets wird heutzutage den Usern nicht nur der Zugang zu Musikstücken ermöglicht, sondern auch zu online Musikdaten wie z.B Songtexte, Biographien, usw. Diese Metadaten werden meistens aufgesucht, wenn User mehr über das Lied in Erfahrung bringen wollen. Songtexte können hierbei eine große Hilfe sein. Sie zählen neben dem Künstlernamen und dem Musiktitel zu den wichtigsten Metadaten eines Musikstückes, da sie nicht nur die schriftliche Form, sondern auch die zutreffendste Beschreibung eines Liedes liefern.

Mit der Anwendung von Textanalysetechniken können Songtexten wichtige Informationen entnommen werden, die Rückschlüsse auf das Musikgenre geben können. Dies wiederum könnte sehr viel zur Organisation von riesigen Musikkollektionen beitragen. Da Musikportale heutzutage ihre Sammlung an Songtexten frei im Netz zur Verfügung stellen, wäre diese Art der Musikanalyse sehr vorteilhaft gegenüber anderen. Für die Ermittlung des Musikgenres eines Liedes wäre das Herunterladen der mp3 nicht mehr von Nöten, da man nur noch die schriftliche Form des Liedes aufzusuchen bräuchte. Ein weiterer Grund, der für die Analyse von Songtexten spricht ist, dass diese in die entsprechenden Musikgenres klassifiziert werden können, sodass dem User das Navigieren in einem solchen Musikportal bzw. das Auffinden eines bestimmten Musikstückes erleichtert werden kann.

Es gibt bisher wenige Arbeiten, die sich mit der Analyse von Songtexten befassen. Die Einbeziehung von Songtexten in die Genreklassifikation von Musikstücken kann jedoch als gute Ergänzung zu anderen Ansätzen dienen. Es existieren nämlich Musikgenres, die sich leichter über ihren Text identifizieren lassen. Ein Beispiel dafür wäre das Genre

Weihnachtslied. Die Songtextanalyse kann bei der Klassifikation von Musikstücken in solchen Genres durchaus eine große Hilfe darstellen.

1.3 Textklassifizierung in Genres

Das Ziel der Textklassifizierung ist die Klassifikation von Textdokumenten in eine fix vordefinierte Anzahl von Klassen. Jedes Textdokument d kann einer, mehrerer oder keiner Klasse zugeordnet werden. Dazu wird mit Hilfe von Machine Learning Verfahren ein Klassifikator mit Trainingsdaten trainiert, auf dass dieser in der Lage ist, neue Textdokumente den entsprechenden Klassen zuzuordnen.

Die Textklassifizierung in Genres hat sich in letzter Zeit als Schlüsselverfahren für die Organisierung von Textdaten bewährt. Das Genre spielt dabei eine wichtige Rolle, da es ein wichtiges kategorisches Konzept darstellt. Es wurde oft versucht dieses für mehrere Anwendungsbereiche zu nutzen wie z.B im Dokumentenmanagement oder bei der Informationsfilterung. In verschiedenen Arbeiten wird die Textklassifizierung in Genres unter anderen dazu benutzt Nachrichten, welche in der Form von Text vorliegen, zu klassifzieren [Hay90], interessante Informationen dem Internet zu entnehmen [Lang95] oder einen User bei seiner Suche im Hypertext zu führen [Joa97]. Dabei ist der Vorgang der Genreidentifikation unumgänglich.

Es können zahlreiche Defintionen für den Begriff „Genre-Identifikation“ gefunden werden. Unter diesen fällt auch die von Santini. In seiner Arbeit definiert er das Ziel der Textgenreidentifikation hauptsächlich eine Gruppe von Dokumenten zu identifizieren, die gemeinsame Eigenschaften bezüglich Übermittlung, Zweck und Diskurs teilen [San04].

Grundsätzlich kann die Genrekategorisierung für die meisten Arten der Kommunikation angewendet werden. Dabei stellt das Genre eines der am kennzeichnendsten Eigenschaften bei der Informationssuche dar. Während sich mehrere Interpretationen bezüglich dem Begriff „Genre“ auf kulturell-eingeführte Kategorien von Text beziehen wie z.B Novellen, Briefe, Handbücher usw., basiert die automatische Genreidentifikation auf einem mengenbezogenen Ansatz, welcher extrahierbare und berechenbare Features (z.B sichtbare Eigenschaften im Text) wirksam einsetzt, um zwischen verschiedenen Klassen von Dokumenten unterscheiden zu können. Dieser Ansatz wird in dieser Arbeit ebenfalls angewendet, wobei hier mit der Hilfe von sowohl strukturellen wie auch content-basierten

Features zwischen den verschiedenen (Musik-)Genres im Songtext unterschieden werden.

Diese Arbeit ist wie folgt organisiert. Kapitel 2 befasst sich mit früheren Arbeiten, welche einen Bezug zur Songtextklassifizierung haben. Diese Arbeiten, die alle unter den Begriff „Related Work“ fallen, umfassen Arbeiten im Bereich der Songtextanalyse, Musikgenres, Featureextraktion, Musikklassifikation im Audiobereich und der automatischen Klassifizierung von Text. Es werden dabei die unterschiedlichen Ansätze und Methoden, die bei der Klassifzierung von Songtexten zum Einsatz kommen, beschrieben. Des Weiteren werden in diesem Abschnitt die Grundlagen der automatischen Textklassifizierung in einem knappen Überblick dargestellt.

Kapitel 3 befasst sich mit der Verarbeitung der Songtexte und der Extraktion der verschiedenen Features.

Im Kapitel 4 wird kurz auf die Grundlagen jener Machine Learning-Verfahren, die dann im Kapitel 5 zum Einsatz kommen, eingegangen.

Kapitel 5 befasst sich, anhand ausführlicher Experimente, eingehend mit den Fragen „Welche Klassifikationsgenauigkeit kann man erwarten?“, „Welche Genres eignen sich am besten für die Klassifikationsaufgabe?“ und „Welche Features liefern die besten Ergebnisse?“.

Im Kapitel 6 werden die wichtigsten Punkte dieser Arbeit nochmals zusammengefasst und Schlüsse gezogen, aus denen sich Perspektiven für zukünftige Verbesserungen ergeben.

2 Related Work

2.1 Einleitung

Dieses Kapitel soll einen Überblick über die verschiedenen Bereiche, die zur Songtextklassifikation in Beziehung stehen, liefern. Zu diesen Bereichen fallen Songtexte, Musikgenres, Automatische Textklassifizierung und die Musikklassifikation im Audiobereich.

Im Abschnitt 2.2 wird auf Songtexte im Allgemeinen eingegangen. Dabei wird erläutert welche Eigenschaften Songtexte hinsichtlich Struktur und Inhalt haben und inwiefern diese für die Genre-Klassifikation von Nutzen sein können.

Der Abschnitt 2.3 behandelt das Thema „Musikgenre“. Darin werden die verschiedenen Arten von Musikgenres und die Funktionen, die Musikgenres im Alltag einnehmen, beschrieben.

Die zwei darauf folgenden Abschnitte 2.4 und 2.5 behandeln die Musikklassifikation im Audio- und im Textbereich. Denn die Verfahren und Methoden, die in dieser Arbeit angewendet werden, stammen sowohl aus dem Bereich der Textklassifizierung als auch aus einem relativ jungen Forschungsbereich der Informatik: dem Music Information Retrieval (MIR). Ziel des MIR ist die „Entnahme“ von Informationen aus der Musik. Dabei versucht man Methoden zur Analyse, Modellierung, Klassifikation und Clustering von Musik zu finden. Da die Methoden und Ansätze aus dem Bereich der MIR mit denen der Textklassifizierung kombiniert werden, werden die Grundlagen aus beiden Bereichen näher beschrieben. Auf die Grundlagen der Musikklassifikation im Audiobereich wird nur kurz eingegangen, da sie nicht Schwerpunkt dieser Arbeit ist.

2.2 Songtexte

Bei der Klassifizierung von Musikstücken im Textbereich haben Songtexte viele Vorteile gegenüber anderen Formen von Metadaten. Zu allererst ist die schriftliche Form

vieler Musikstücke online verfügbar. Dadurch können Songtexte sehr einfach gesammelt werden, ganz anders als bei anderen Formen von Metadaten wie z.B. die Kritik eines Musikstückes. Weiters können Songtexte als objektiv angesehen werden. Denn es gibt nur eine „zutreffende“ Niederschrift eines Musikstückes. Dies ist ein großer Vorteil gegenüber subjektiven Formen von Metadaten wie z.B. die Expertenmeinungen eines Liedes. Und zuletzt liefern Songtexte eine bessere Beschreibung des Musikstückes als einfache Formen von Metadaten wie zum Beispiel der Titel, der Künstler oder der Songschreiber des Musikstückes.

Einige wesentlichen Merkmale eines Songtextes können in ihrer Struktur gefunden werden. Diese strukturbasierten Eigenschaften können dazu verwendet werden, die Genres einzelner Songtexte zu unterscheiden. Mahedero, Martinez und Cano beschreiben in ihrer Arbeit folgende Struturteile, in welche Songtexte getrennt werden können [Mahed05]:

- Intro: beinhaltet zum größten Teil ein Vers, bestehend aus 3 oder 4 Sätzen, um dem Zuhörer ein Kontext über das Lied zu geben.
- Vers: entspricht ungefähr einer poetischen Strophe. Der Text in einem Vers tendiert zu wenigen Wiederholungen, als im Refrain.
- Refrain: auch Chorus genannt. Wenn zwei oder mehrere Abschnitte eines Songtextes einen fast identischen Text haben, sind diese Abschnitte meist Instanzen des Refrains. Ein Vers in diesem Abschnitt wiederholt sich mindestens zweimal mit sehr geringem Unterschied. Das Thema des Liedes kommt in diesem Teil am meisten zum Ausdruck. In der Musik ist dies der Teil, an den sich die Zuhörer am meisten erinnern.

Zu den strukturbasierten Eigenschaften/Features gehören ebenfalls Text-Statistic Features, welche syntaktische Eigenschaften wie z.B. Satzpunkte oder Beistriche beinhalten. Zusätzlich kann unter anderen auch der durchschnittliche Wortschatz eines Liedes analysiert werden. In Hip-Hop Songs wiederholen sich die Wörter meistens sehr oft, wohingegen in einer Ballade der Wortschatz groß und die Wortwiederholungen klein ausfallen können.

Die Struktur eines Songtextes kann einiges über das Genre eines Liedes sagen, jedoch kann diese Analyse um die inhaltliche Analyse von Songtexten noch erweitert werden.

In Rapliedern wird sehr oft von Schimpfwörtern Gebrauch gemacht, wohingegen in Liebesliedern diese sehr selten vorzufinden sind. Dieses signifikante Merkmal kann dazu verwendet werden, das Genre Hip-Hop/Rap in Songtexten zu charakterisieren. Solche Eigenschaften können ebenfalls in anderen Genres gefunden werden: Soul und R&B Lieder handeln meist von der Liebe, von einer Beziehung zu einer Frau oder von Sex. In Country Songs können zum Beispiel Wörter wie Texas, Cowboy, Mississippi und dergleichen vorgefunden werden. Anhand dieser Beispiele kann man gut erkennen, dass die inhaltliche Analyse durchaus sehr vorteilhaft sein kann.

Es existieren frühere Arbeiten, welche die Semantik und die Struktur von Songtexten im Music Information Retrieval (MIR) analysieren und sich diese zu Nutze machen. Scott und Matwin nutzen in ihrer Arbeit mehr als 400 Volkslieder für ihre Textklassifikationsexperimente [Scott98]. Dabei erweitern sie die Bag-of-Words Methode, bei der jedes Wort in einem Dokument als Attribut definiert wird, indem sie Wordnet Hyperonyme integrieren. Dadurch konnte die Klassifikationsgenauigkeit um einiges verbessert werden. Jedoch haben sie ihr Verfahren nicht mit denen der Audioanalyse verglichen. Baumann und Klüter verwenden ontologie-basierte Dokumentenretrievalverfahren, um Songtexte zu charakterisieren [Bau02]. Auf diese Weise konnten Ähnlichkeitsmaße von Liedern basierend auf Songtext berechnet werden. Logan, Kositsky und Moreno haben in ihrer Arbeit ca. 16000 Songtexte gesammelt, um die Ähnlichkeit von Musikkünstlern bestimmen zu können [Logan04]. Für die Analyse des semantischen Inhalts wurde die Probabilistic Latent Semantic Analysis angewendet. Die PLSA ist dabei ein statistisches Verfahren aus dem Fachgebiet der automatischen Sprachverarbeitung zur Analyse von Textkorpora. Es sich heraus, dass Songtexte dazu genutzt werden können, um natürliche Genrecluster aufzufinden. Brochu und de Freitas entwickelten ein Framework, um verschiedene Eigenschaften von Daten zu modellieren [Bro02]. Diese nutzten sie, um musikalische Auswertungen zu analysieren und Textannotationen zu assoziieren, welche auch unter anderem 100 Songtexte beinhalteten. Obwohl die Ergebnisse sehr vielversprechend waren, ist die Größe ihrer Studie jedoch zu klein, sodass keine klaren Folgerungen aus diesen gezogen werden können.

Es gibt viele Gründe, die für den Einsatz von Songtexten bei der Musikklassifikation sprechen, da Songtexte eben einen wichtigen Teil der Semantik eines Liedes darstellen. Dies soll in dieser Arbeit mit Hilfe von verschiedenen Experimenten und Analysen verdeutlicht werden. Die Verwendung struktureller und inhaltlicher Informationen eines Songtextes kann nämlich viel zur Genreanalyse eines Liedes beitragen. Dies wiederum

kann sich positiv auf die Entwicklung von Music Information Retrieval (MIR) Systemen auswirken.

2.3 Musik Genres

Als Musikstil oder Musikrichtung bezeichnet man eine Klasse einer bestimmten Art von Musik. Im Folgenden werden mögliche Gliederungskriterien für Stilrichtungen der Musik vorgestellt:

- Historisch: nach der Zeit/Epoche der Entwicklung des Musikstils (alte Musik, klassische Musik, moderne Musik)
- Geographisch: woher der Stil kommt oder wo er praktiziert wird
- Funktional: wozu die Musik verwendet wird
- Quantitativ (Anzahl der Beteiligten) : Instrumental- und Vokalmusik, Einzelmusiker, Band, Big Band etc.
- Qualitativ (Wertend): wertvolle und weniger wertvolle Musik (E-Musik / U-Musik /F-Musik)

E-Musik ist eine Abkürzung für die so genannte „ernste“ („kulturell wertvolle“) Musik. U-Musik steht für „Unterhaltungsmusik“ und fasst populäre und kommerzielle Musikrichtungen (populäre Musik) zusammen. Beispiele dafür sind Popmusik, Rockmusik, Schlager usw. F-Musik ist eine Abkürzung für „funktionale“ oder „funktionelle“ Musik. Wie der Name schon sagt, handelt es sich hierbei um Musik, die an einem bestimmten Zweck gebunden ist, wie etwa Filmmusik, Musik in Kaufhäusern oder Aufzügen. An diesen Unterteilungen ist leicht zu erkennen, dass sie ein Problem für den Songtext-Klassifikator darstellen würden, da die Eigenschaften nicht objektivierbar sind. Objektivität ist ein sehr wichtiger Punkt, wenn es um die Unterteilung von Musik geht. Die Gliederung nach nichtmusikalischen Kriterien ist daher nicht für eine systematische Klassifikation geeignet.

Aus diesem Grund herrscht im Alltag eine andere Form der Unterteilung von Musik vor: die Unterteilung in Musikgenres. Diese Art der Unterteilung (in Genres wie z.B. Pop, Rock, ...) ist die am meist verwendete Form der Musikbeschreibung. Musikstücke, die von ihrer Art her ähnlich sind, werden hierbei einem Genre zugeordnet, andersartige Musikstücke anderen Genres.

Im Music Information Retrieval (MIR) sind Musikgenres sehr wichtig, da sie sehr häufig dazu verwendet werden, die zunehmende Menge an Musik, die im Netz in digitaler Form vorzufinden ist, zu strukturieren. Da sie meist eine einfache und überschaubare Unterteilung bieten, kann diese als grobe Richtlinie zum Auffinden ähnlicher Musik verwendet werden. Weiters können mit dieser einfachen Form der Strukturierung beispielsweise die Arbeiten von Musikkünstlern miteinander verglichen werden: Die Band „Green Day“ ähnelt zum Beispiel der Band „Nirvana“. Ihre Songs gehen beide in Richtung Rock. Anhand dieses einfachen Vergleichs wird gezeigt, dass es vor allem in der Populärmusik einige bestimmte Genres gibt, die allseits bekannt sind und die folglich Fixpunkte im Diskurs über Musik darstellen. Country und Popmusik sind solche, andere sind beispielsweise R&B, Reggae oder Hip-Hop. Somit bieten Musikgenres die einfache aber durchaus mächtige Möglichkeit, Musik durch Zugehörigkeit zu bekannten Kategorien zu beschreiben.

Heutzutage wird die musikalische Vielfalt immer größer, wodurch natürlich immer mehr neue Genres entstehen, um Musikstücke besser voneinander abgrenzen zu können. Dazu werden vorhandene Genres zu neuen verdichtet bzw. in Subgenres aufgespalten wie beispielsweise das Subgenre Alternative zum Genre Rock. Diese größer werdende Vielfalt wird auch von den Plattenfirmen genutzt, um dem Hörer durch die Zuordnung zu einem „neuen“ Genre die Neuheit ihrer Musik zu verdeutlichen. Dabei werden aber oft nur alte Muster neu aufgelegt und dadurch die Klassifikation in Genres erschwert. Eine Lösung zu diesem Problem wäre die Erstellung einer Genre-Taxonomie (eine Methodik zur Einordnung), die die Ordnung von Genres zu Subgenres enthält. So eine Genre-Hierarchie kommt in dieser Arbeit ebenfalls zum Einsatz und wird in einem späteren Kapitel näher behandelt. Sie wird oftmals dazu eingesetzt einen Ansatz zur Organisation von Musik zu bekommen. In Musikgeschäften werden beispielsweise die Cds meist nach Musikgenres angeordnet. Dabei verwenden sie diese Art von Genre-Hierarchien, um die Kunden bei der Erforschung großer Musikbestände zu unterstützen. Es wird typischerweise eine Taxonomie in vier Ebenen benutzt [Pachet00]:

1. eine Ebene 1 mit „globalen“ Musikkategorien (Pop, Jazz, Rock, etc.)

2. eine Ebene 2 mit Unterkategorien (z.B. Punk Rock in Rock)
3. in der 3. Ebene werden die Musikkünstler nach ihrem Namen alphabetisch sortiert
4. in der 4. Ebene liegen die Alben der verschiedenen Musikkünstler

Burred und Lerch haben diese Genre-Hierarchie in der Form übernommen, dass in deren Verfahren hierarchisch klassifiziert wird [Bur03]. Hierbei werden in einer bestimmten Reihenfolge genrespezifische Merkmale (Features) eines Musikstückes mit denen der ground truth (die als Wahrheit angenommene Zuordnung) verglichen und dann einer Klasse zugeordnet. Man unterscheidet in der obersten Stufe im Hierarchiebaum zwischen Speech, Music und Background und unterteilt diese in den jeweiligen Ästen immer weiter. Dadurch befinden sich Musikstücke, welche in einer bestimmten Stufe nicht weiter klassifizierbar sind, schon im richtigen Teilbaum und werden nicht als unklassifizierbar abgelehnt. Dabei werden Fehler, die auf einer niedrigen Stufe auftreten nicht so hoch bewertet, wie Fehler auf einer höheren Stufe. Ein West-Coast Rap, welches zum Beispiel als East-Coast Rap klassifiziert wird, kann als leichter Fehler angesehen werden, da das Musikstück unter dem Genre Rap/Hip-Hop fällt. Der Fehler würde schwerer ausfallen, wenn das Musikstück in den Klassik-Ast klassifiziert worden wäre.

Genre-Taxonomien werden heutzutage nicht nur in Musikgeschäften, sondern auch in den an Popularität gewinnenden Online-Music-Stores eingesetzt. Neben dem Interesse des durchschnittlichen Anwenders, seine Musikkollektion einfach und am besten automatisch ordnen zu lassen, besteht somit auch ein reales wirtschaftliches Interesse große Musikdatenbanken gut zu organisieren. Aus diesem Grund sind von Musikgenres in einer Zeit, wo allem der Zugriff auf digitale Musik möglich ist, gar nicht mehr wegzudenken.

2.4 Musik Genre Klassifikation im Audiobereich

Bei der Musikklassifikation im Audiobereich werden Verfahren angewendet, die Musikstücke auf Basis ihrer Klangfarbe miteinander vergleichen, um bestimmte Ähnlichkeitsmaße zu erhalten. Diese Verfahren haben sehr oft Anwendung im Bereich des MIR gefunden. Die Merkmale (Features), die bei der Klassifikation im Audiobereich aus Audiodaten gewonnen werden, können in verschiedene Kategorien eingeteilt werden. Die

erste Kategorie sind die Timbral Texture Features, welche sich an der Klangfarbe des Stückes ausrichten. Werte wie Tonumfang, Brillianz oder Schärfe des Klanges werden hier in rechenbare Merkmale gefasst, anhand derer die Klangfarben unterschieden werden können. Um aussagekräftige Merkmale zu erhalten, werden deren Ausprägungen nicht über das Signal in voller Länge berechnet. Kleine sich überlappende Ausschnitte des Signals gehen in die Rechnung ein. Die Größe dieser sogenannten Analysefenster sollte so gewählt werden, dass der Frequenz-Charakter in den Fenstern relativ stabil ist. Die Merkmale mehrerer Analysefenster werden in einem Texture-Fenster mit Mittelwert (M), Varianz (S), Mittelwert der Ableitung (DM) und Standardabweichung der Ableitung (DS) zusammengefasst. Anhand dieser wird klassifiziert. Cook und Tzanetakis haben beispielsweise in einer ihrer Arbeiten eine Analysefenstergröße von 23 ms (512 samples) und eine Texture-Fenstergröße von 1 s (43 Analysefenster) verwendet [Cook02].

Die zweite Kategorie sind die Rhythmic Content Features. Um diese zu extrahieren, werden die Daten mit Hilfe der Wavelet-Transformation bearbeitet und die rhythmischen Regelmäßigkeiten gesucht, die in einem beat-histogram dargestellt werden. Daraus kann dann die Struktur und die Stärke des Rhythmus erkannt werden. Eine weitere Kategorie sind die sogenannten Pitch Content Features. Hierbei werden Merkmale bezüglich der Tonhöhe errechnet, etwa die Anzahl der Wechsel während eines kurzen Ausschnitts des Songs.

Es sind viele Verfahren präsentiert worden, mit denen die oben angeführten Merkmale berechnet werden. Aucouturier und Pachet haben jedoch festgestellt, dass die meisten dieser Ansätze zu derselben Mustererkennungsarchitektur gehören mit unterschiedlichen Variationen und Parametern [Auc04]. Die Verarbeitungsschritte, welche untereinander sehr ähnlich sind, sind folgende:

1. Das Audiosignal wird in überlappende Segmente unterteilt (üblicherweise in Segmente von 20-50 ms mit einer 50 prozentigen Überlappung).

2. Für jedes Frame wird ein Feature-Vektor berechnet, welcher üblicherweise aus Mel-Frequency Cepstrum Coefficients (MFCC) besteht. Die Anzahl der MFCCs spielt dabei eine große Rolle und in jedem der verschiedenen Ansätze wurde eine unterschiedliche Anzahl verwendet: Baumann präsentiert in seiner Arbeit ein P2P Framework, mit dem er verschiedene Musikfeatures von Mp3 Dateien speichern und berechnen kann. Für die Berechnung dieser Musikfeatures nutzt er 13 MFCCs [Bau03]. Berenzweig und Logan untersuchen in ihrer Arbeit die Ähnlichkeit

zwischen Musikkünstlern. Dazu bewerten sie akustische Features, welche mit 20 MFCCS berechnet wurden [Ber03]. Das System, das von Foote präsentiert wird und akustische Ähnlichkeiten zwischen Audiodateien berechnet, nutzt 12 MFCCs [Foote97]. Kulesh, Sethi und Petrushin hingegen verwenden 14 MFCCs in ihrer Arbeit, welche einen Ansatz zur Indexierung und Wiederauffinden von Musik darstellt [Kulesh03].

3. Beim dritten Schritt wird ein statistisches Modell für die Verteilung der MFCCs berechnet. Die Arbeiten von Baumann [Bau03] und Berenzweig [Ber03] nutzen dazu den K-Means Ansatz mit 16 K-Means Zentren. Kulesh, Sethi und Petrushin verwenden in ihrer Arbeit Gaussian Mixture Models (GMM) mit 32 GMM Zentren [Kulesh03]. In beiden Arbeiten werden dabei die Anzahl der K-Means und GMM Zentren als Parameter mehrfach diskutiert.

4. Zuletzt werden die Modelle miteinander verglichen.

2.5 Automatische Klassifizierung von Text

In diesem Kapitel werden die Grundlagen des automatischen Klassifizierens von Text in einem knappen Überblick dargestellt. Der erste Abschnitt dieses Kapitels beschreibt die Idee und die Ziele der Textklassifizierung. Die Verfahren, mit denen ein Text einer Klasse zugeordnet wird, werden im zweiten und dritten Abschnitt näher erläutert. Dabei stützen sich die Ausführungen insbesondere auf den umfangreichen Übersichtsartikel von Sebastiani [Seb02].

2.5.1 Begriffsumfang

Textklassifizierung (im Englischen „text categorization“) wird definiert als die Tätigkeit Texte einer natürlichen Sprache mit einer von einer vordefinierten Menge an Kategorien kennzuzeichnen. Früher wurden derartige Klassifizierungsaufgaben von IR-Systemen (IR = Informationsretrieval) gelöst. Dabei wurde ein neu zu klassifizierendes Dokument, das durch einen Vektor von Termgewichten repräsentiert wurde, als Anfrage an das

IR-System gerichtet. Das IR-System führte dann einen Vergleich mit den Klassenvektoren durch, welche in der Datenbank gespeichert waren. Das Ergebnis war eine nach Ähnlichkeit ranggeordnete Liste der Klassen.

Ein automatisches Klassifizierungssystem, das auf maschinelle Lernverfahren basiert, besteht aus 2 Komponenten:

- Komponente zum Wissenserwerb: wird in der Trainingsphase eingesetzt, in welcher auf der Grundlage einer Menge bereits klassifizierter Trainingsdokumente die Konzepte der Klassen gelernt und Klassenprofile erstellt werden.
- Komponente zum Klassifizieren („Klassifikator“): wird in der Klassifizierungsphase eingesetzt, in welcher noch nicht klassifizierte Dokumente hinsichtlich ihrer Charakteristika analysiert und durch einen Vergleich mit den Klassenprofilen den passenden Klassen zugeordnet werden.

Das Ziel des automatischen Klassifizierens ist die Annäherung einer Zielfunktion, welche beschreibt, wie die Dokumente klassifiziert werden sollten. Dies geschieht mittels einer weiteren Funktion, die üblicherweise als Klassifikator („classifier“) bezeichnet wird, und zwar auf eine Art, dass diese beiden Modelle so gut wie möglich übereinstimmen.

Die Einfachklassifizierung („single-label categorization“) bezeichnet den Fall, in dem jedem Dokument genau eine Klasse zugewiesen werden soll. Kann jedoch die Klassenzahl von 0 bis Maximum variieren („overlapping categories“), so spricht man von „Multilabel categorization“.

2.5.2 Dokumentenindexierung und Merkmalsreduktion

Indexierung der Dokumente

Damit ein Klassifikator die Texte interpretieren kann, wird ein Indexierungsverfahren benötigt, bei dem der Text in eine kompakte Repräsentation transformiert wird. Dies geschieht meistens durch die Bildung eines Vektors von Termgewichten, der ausdrücken soll, zu welchem Ausmaß jeder im Dokument auftretender Term (auch „Attribut“ oder „feature“ genannt) zur Bedeutung des betreffenden Dokuments beiträgt. Dabei gibt es verschiedene Möglichkeiten für die Definition eines Attributs sowie für die Berechnung der Termgewichte.

Eine Möglichkeit ist jeden Term im Text als Attribut zu definieren. Dieser Ansatz wird „Bag-of-Words“ genannt. In dieser Arbeit werden diese „Bag-of-Words“ Features verwendet und für Experimente mit anderen kombiniert. Die zum Einsatz kommenden Features werden im Abschnitt 3.4 näher behandelt.

Die Gewichte der Features werden meist normiert, sodass diese einen Wert zwischen 0 und 1 haben. Das am häufigsten verwendete Verfahren, welches zur Darstellung der Dokumente als Vektoren aus gewichteten Termen führt, ist die bekannte TFIDF-Funktion. Im Abschnitt 3.2 wird auf die TFIDF-Funktion, welche in dieser Arbeit für die Termgewichtung verwendet wird, näher eingegangen und dessen Formel näher erläutert.

Vor der Indexierung wird der Text meist normalisiert, sodass alle unerwünschten Zeichen bzw. Terme wegfallen. Dabei kommen auch linguistische Verfahren wie z.B die Eliminierung von Stoppwörtern (sehr häufig und/oder sehr seltene Wörter) und Stemming zum Einsatz. Als Stemming bezeichnet man im Information-Retrieval ein Verfahren, mit dem verschiedene morphologische Varianten eines Wortes auf ihren gemeinsamen Wortstamm zurückgeführt werden (Beispielsweise „Suchen“ auf „Such“ und „schreibe“ auf „schreib“). Verschiedene Varianten eines Wortes können nämlich durch Komposition, Dekomposition, Flexion oder durch Hinzufügen von Affixen (Prefix, Suffix, Infix und Zirkumfix) entstehen. Stemming normalisiert somit semantisch ähnliche Worte durch Entfernung der Endsilben.

Dimensionsreduktion

Die große Menge an Termen führt meistens zu einer sehr hohen Dimensionalität des Featureraums. Dies kann bei der Erstellung eines Klassifikators zu einem großen Problem werden. Aus diesem Grund versucht man die Größe des Vektorraums auf ein „reduced term set“ zu beschränken. Eine solche Reduktion ist auch deshalb vorteilhaft, da sie dem „Overfitting“ entgegenwirkt. Unter „Overfitting“ wird die Gefahr verstanden, dass der Klassifikator eher die Spezifika der Trainingsdokumente lernt als die für die jeweiligen Klassen eigentlich konstitutiven Merkmale. Ein durch „Overfitting“ gekennzeichneter Klassifikator würde die Trainingsdokumente besonders gut klassifizieren können, neue Dokumente hingegen nicht.

Bei der Dimensionsreduktion besteht natürlich auch die Gefahr potentiell nützliche Information zu eliminieren. Aus diesem Grund wurden verschiedene methodische Alter-

nativen entwickelt, bei deren Anwendung die Reduktion klassenspezifisch oder global erfolgen kann. Die beiden zu unterscheidenden Techniken sind die Attributextraktion („term extraction“) und die Attributauswahl („feature selection“, „term space reduction“).

Die Attributextraktion geht davon aus, dass die in den Dokumenten auftretenden Terme aufgrund von Homonymie, Polysemie und Synonymie keine optimalen Dimensionen zur Repräsentierung der Dokumente darstellen und daher „künstliche“ Terme generiert werden sollten, die von diesen Problemen nicht betroffen sind.

Bei der Attributauswahl versucht man aus der Menge der Attribute jene herauszufiltern, die bei Verwendung als Indexterme die höchste Effizienz, d.h. die höchsten Werte hinsichtlich ihrer Bedeutung für die Klassifizierungsaufgabe erzielen. In dieser Arbeit kommt als „Feature selection“-Technik der χ^2-Test, welcher im nächsten Abschnitt behandelt wird, zum Einsatz.

Der χ^2-Test

Ein besonderes Merkmal bei Text Categorization Problemen ist die hohe Dimension der Featurevektoren. In einer aus verschiedenen Dokumenten bestehenden Kollektion können eine Menge unterschiedlicher Terme auftreten. Für die meisten Lernalgorithmen ist diese große Anzahl an Dimensionen viel zu hoch. Weiters sind unter den vielen Termen welche, die nur ein einziges Mal auftreten, was auf Tippfehler oder auf Irrelevanz zurückzuführen ist. Aus diesem Grund ist es meist wünschenswert die Dimension des Featurevektors zu reduzieren, ohne dabei die Klassifikationsgenauigkeit opfern zu müssen.

Für die weitere Reduzierung der Features kommen „Feature selection“-Techniken zum Einsatz. „Feature selection“, auch „subset selection“ oder „variable selection“ genannt, ist ein Prozess, bei dem eine Untermenge von Features für ein Lernalgorithmus gewählt wird. Die ausgewählten Features werden dabei als die am nützlichsten zur Repräsentation eines Dokuments erachtet. Diese werden durch eine „Evaluation Function“ bestimmt, wobei jene Features gewählt werden, welche den höchsten Wert dieser Funktion aufweisen.

Die Evaluationsfunktion, welche bei der Auswahl der Features in dieser Arbeit zum Einsatz kommt, ist der so genannte Chi-Quadrat-Test (χ^2-Test). Der χ^2-Test ist ein statistisches Verfahren, welches eingesetzt wird, um zu messen, wie unabhängig konkrete Beobachtungen von einer vermuteten Hypothese sind. Im Falle der Text Categorization

nimmt die Null-Hypothese H_0 an, dass das Auftreten eines bestimmten Terms von der Kategorie, in der dieser Term auftritt, unabhängig ist. Die Alternativhypothese H_1 lautet somit, dass das Auftreten eines Terms von dessen Kategorie abhängig ist. Die Kategorien sind in dieser Arbeit die Musikgenres, nach welchen die Songtextdokumente klassifiziert werden.

Die Prüfgröße χ^2 ist bei ausreichend großen erwarteten Häufigkeiten annähernd χ^2-verteilt. Wenn die Prüfgröße klein ist, wird vermutet, dass die Hypothese H_0 wahr ist. Daraus resultiert, dass bei einem hohen Prüfgrößenwert die Hypothese H_0 abgelehnt wird. Durch die Chi-Quadrat-Tests werden nun alle Terme ausgewählt, welche einen hohen Chi-Quadrat-Wert aufweisen, da für diese mit einer hohen Wahrscheinlichkeit die Unabhängigkeit zwischen Term und Kategorie nicht gegeben ist. Diese Terme sind jene, die die Kategorie am besten beschreiben und für die Abgrenzung von anderen Kategorien am besten geeignet sind.

Debole und Sebastiani haben die Formel zur Berechnung des Chi-Quadrat-Wertes formuliert [Deb03]. Dabei werden die Wahrscheinlichkeiten auf einem Ereignisraum aus Dokumenten interpretiert und mit Maximum-Likelihood geschätzt. Die Notation $P(t, c)$ bezeichnet die Wahrscheinlichkeit, dass für ein beliebiges Dokument x, Term t in x vorkommt und x zur Kategorie c gehört.

$$\chi^2_{tc} = \frac{[P(t,c)P(\bar{t},\bar{c}) - P(t,\bar{c})P(\bar{t},c)]^2}{P(t)P(\bar{t})P(c)P(\bar{c})} \tag{2.1}$$

Dunning behauptet, dass der normalisierte Wert des χ^2-Tests sehr vorteilhaft sein kann [Dunn93]. Dadurch, dass der Wert im Intervall zwischen 0 und 1 liegt, können Terme einer Kategorie miteinander verglichen werden. Allerdings können diese Werte nicht mehr miteinander verglichen werden, wenn ein Term eine geringe Häufigkeit aufweist. Aus diesem Grund ist die χ^2-Statistik für niedrigfrequente Terme nicht zuverlässig.

Trainings-, Test- und Validierungsdokumente

Die Menge der Trainingsdokumente („training set“), die einem lernenden Klassifikator zugrunde gelegt wird, muss bereits klassifizierte Dokumente, für jede Klasse enthalten.

Ein zu einer bestimmten Klasse zählendes Dokument gilt als positives Beispiel, ein nicht zu dieser Klasse zählendes als negatives Beispiel.

Wenn der Wunsch besteht, die Güte der durch das automatische Verfahren erzielten Ergebnisse zu evaluieren, wird vor der Erstellung des Klassifikators die Trainingsmenge in zwei (nicht notwendigerweise gleiche große) Teile geteilt:

- Die Trainingsdokumente: auf der Basis dieser Dokumente wird der Klassifikator gebildet. Sofern es in der Trainingsphase notwendig ist, verschiedene Versionen oder Parameter des Klassifikator zu testen, wird zuvor den Trainingsdokumenten eine Menge von Validierungsdokumenten entnommen, anhand derer diese Tests durchgeführt werden.

- Die Testdokumente: diese werden für den Test auf Güte herangezogen, indem jedes Dokument automatisch klassifiziert und das Resultat mit der manuellen Klassifizierung verglichen wird.

Meist ist die Trainingsmenge deutlich größer als die Testmenge. Die letztere darf keinesfalls bei der Erstellung des Klassifikators mitwirken, da dies den anschließenden Test verfälschen würde. In dieser Arbeit wird als Evaluierungsmethode auf die 10-Fold Cross Validation zurückgegriffen. Hierbei wird die zur Verfügung stehende Dokumentenmenge in 10 Teilmengen aufgeteilt. Es werden 10 Testdurchläufe gestartet, bei denen die jeweils 10-te Teilmenge als Testmenge und die verbleibenden 9 Teilmengen als Trainingsmengen verwendet werden. Die Gesamtfehlerquote errechnet sich als Durchschnitt aus den Einzelfehlerquoten der 10 Einzeldurchläufe.

3 Merkmalsextraktion

Bei der Merkmalsextraktion geht es darum den Text eines Dokuments in eine kompakte Repräsentation zu transformieren. Dies geschieht durch Standard-Indexierungsverfahren des Bereiches „Text Categorization“. Das Ergebnis ist ein Vektor von Termgewichten, welcher ausdrückt zu welchem Ausmaß jedes Attribut/Feature zur Bedeutung des Dokuments beiträgt.

Sebastiani beschreibt die Text Categorization als ein Vorgang des automatischen Erstellens von automatischen Text-Klassifikatoren, welche natürlichsprachige Texte einer Domäne D mit thematischen Kategorien aus einer vordefinierten Menge $C = \{c_1, c_2, ...\}$ durch Methoden des Machine Learning kennzeichnet [Seb02]. In diesem Fall handelt es sich bei den thematischen Kategorien um Musikgenres. Die Konstruktion eines Klassifikators erfolgt in 2 Phasen:

1. Dokumentenindexierung: hier wird der Text in eine kompakte Repräsentation transformiert. Dabei wird eine Attributextraktion und eventuell auch eine Attributauswahl (term selection) durchgeführt. Die Attributauswahl ist eine Form der Dimensionsreduktion und besteht darin, eine Auswahl aus allen vorkommenden Termen zu wählen, die bei Verwendung als Indexterme die höchste Effizienz für die Klassifikationsaufgabe erzielen.

   Für jedes Feature wird dann eine Gewichtung, meistens zwischen 0 und 1, berechnet, welche aussagt wie sehr das Feature dazu beiträgt, dass das Dokument anders ist als andere.

2. Erstellung des Klassifikators: hier wird der Klassifikator trainiert, indem dieser aus den Repräsentationen der Trainingsdokumente lernt.

In diesem Kapitel wird im Abschnitt 3.1 auf die einzelnen Vorverarbeitungsschritte, in denen die Daten für die Experimente aufbereitet werden, eingegangen.

Im Abschnitt 3.2 wird die TFIDF-Funktion, mit der die verschiedenen Terme gewichtet werden, näher erläutert.

Abschnitt 3.3 beschreibt in einem kurzen Überblick das Tool „TeSeT“, welches in dieser Arbeit zur Indexierung der Songtextdokumente zum Einsatz kommt.

Im letzten Abschnitt 3.4 werden die verschiedenen Features, die sowohl durch Attributauswahl als auch durch Attributextraktion gewonnen werden, behandelt.

3.1 Vorverarbeitung

Die Daten, die dem in dieser Arbeit präsentierten Ansatz zur automatischen Klassifikation von Songtexten zugrunde liegen, wurden dem Internet entnommen. Aus diesem Grund werden in jedem HTML-Songtext die HTML Tags entfernt und in eine einfache Textdatei verarbeitet. Dabei bleiben Zeichen wie z.B. Punkte, Beistriche, Anführungszeichen usw. erhalten.

Ein weiterer Vorverarbeitungsschritt ist die Entfernung der sogenannten Stoppwörter in jedem Songtextdokument. Stoppwörter nennt man im Information-Retrieval Wörter, die bei einer Volltextindexierung nicht beachtet werden, da sie sehr häufig auftreten und gewöhnlich keine Relevanz für die Erfassung des Dokumentinhalts besitzen. Als Experiment wurden bei mehreren Klassifikationsversuchen die Stoppwörter beibehalten, um Unterschiede in der Klassifikationsrate mit und ohne Stoppwörter festzustellen. Im Kapitel 5 wird jedoch noch darauf eingegangen.

Allgemein übliche Stoppwörter in deutschsprachigen Dokumenten sind bestimmte Artikel (der, die, das), unbestimmte Artikel (einer, eine, ein), Konjunktionen (z.B. „und“, „oder“, „doch“, ...) und häufig gebrauchte Präpositionen (z. B. „an“, „in“, „a“, ...), sowie die Negation „nicht“. Im Englischen sind unter anderem „a“, „of“, „the“, „it“, „you“ und „and“ Stoppwörter.

In manchen Experimenten wurde Stemming durchgeführt. Da Stemming oft als Vorverarbeitungsschritt angewendet wird, hat dies nicht gleich zu bedeuten, dass es dadurch automatisch zu besseren Klassifikationsergebnissen führt. Riloff behauptet, dass es durchaus mal vorkommt, dass mit einem Stemmer die Klassifikationsgenauigkeit darunter leiden kann, da nämlich auch wichtige Informationen verloren gehen können [Ril95]. Dieses Verfahren wird nicht als typischer Preprocessing-Ansatz angesehen. Die meisten Textklassifikationssysteme nutzen den TFIDF und Stopword-Removal Ansatz, um ih-

re Dokumente vorzuverarbeiten. Für das Stemming wird das Tool „TeSeT“, welches im Kapitel 3.3 behandelt wird, verwendet.

Im nächsten Schritt werden irrelevante Terme und Zahlen entfernt. Unter den irrelevanten Termen fallen jene, welche in weniger als 5 Dokumenten auftreten und aus weniger als 3 Buchstaben bestehen. Eine vollständige Liste der Stoppwörter, welche mit dem Tool „TeSeT“ entfernt werden, befindet sich im Anhang A.1.

3.2 Termgewichtung

Im „Bag-of-Words“ Ansatz wird jeder Term als Feature definiert. Für die Berechnung der Gewichtung wird die bekannte TFIDF-Funktion (Formel 3.1) angewendet. Die Termfrequenz $tf_{i,j}$ in Formel 3.1 gibt an wie oft der Term i innerhalb eines Dokuments j auftritt. Die TFIDF-Formel setzt sich aus dieser Termfrequenz und der inversen Dokumentenfrequenz zusammen.

$$tfidf = tf_{i,j} * idf \tag{3.1}$$

Die inverse Dokumentenfrequenz idf in Formel 3.2 stellt das Maß für die allgemeine Wichtigkeit eines Terms dar mit N als die Gesamtanzahl an Dokumenten und df_i als die Anzahl der Dokumente, die den Term i beinhalten.

$$idf = \log \frac{N}{df_i} \tag{3.2}$$

Es gibt mehrere Formeln für die Berechnung des TFIDF Wertes. Man erhält jedoch immer eine hohe Gewichtung der TFIDF, wenn ein Term innerhalb eines Dokuments eine hohe Termfrequenz und eine niedrige Dokumentenfrequenz in der ganzen Dokumentensammlung aufweist. Die TFIDF Funktion filtert somit die allgemeinen Terme heraus.

Anders gesagt bestimmt das TFIDF Verfahren die Relevanz von Worten mehrerer Textdokumente und Kategorien und berechnet die Gewichtung von Termen bezüglich ihrer Einzigartigkeit in einem Textdokument. Die verschiedenen Formeln unterscheiden sich meistens nur in der genauen Gewichtung der zwei Komponenten (TF = Term frequency, IDF = Inverse Document Frequency) und in der Art, wie z.B. die Dokumentenlänge berücksichtigt wird.

3.3 TeSeT

Für die Indexierung der Songtextdokumente wird auf ein frei verfügbares Tool „TeSeT“ (Term Selection Tool) zurückgegriffen, welches sich die Libraries von Apache Lucene zu Nutze macht, um Werte wie z.B. Dokumentenfrequenz, Termfrequenz und Termlänge für jeden einzelnen Term berechnen zu können. Apache Lucene ist eine Open-Source-Java-Bibliothek zum Erzeugen und Durchsuchen von Indizes in Texten. Mit Hilfe dieser plattformunabhängigen Bibliothek lassen sich in kurzer Zeit Volltextsuchen für beliebige Inhalte erzeugen. Weiters verfügt Lucene über eine reichhaltige Auswahl zusätzlicher Funktionen und Tools, welche durch die Open-Source-Community aktiv und umfangreich weiterentwickelt werden.

Mit Hilfe des TeSeT-Tools können nun die Vektoren, welche eine kompakte Repräsentation der einzelnen Songtextdokumente darstellen, mit den entsprechenden Termgewichten gebildet werden. Die Termgewichte werden mit der TFIDF-Funktion, welche im oberen Abschnitt behandelt wurde, berechnet. Weiters werden mit dem TeSeT-Tool Preprocessing Methoden wie Stemming und das Entfernen von Stoppwörtern durchgeführt. In Abbildung 3.1 ist ein Screenshot vom Tool abgebildet.

3.3.1 Indexierung der Dokumente

Die Indexierung der Songtextdokumente erfolgt mit dem TeSeT-Tool durch die Erstellung einer Individual Vector File und einer Template Vector File. Im Individual Vector File wird für jedes Dokument im Korpus der TFIDF Wert jedes Terms im Dokument festgehalten. Dazu wird der Wert, wie oft ein bestimmter Term im jeweiligen Dokument vorkommt, ermittelt und mittels TFIDF Funktion berechnet. Das Template Vector File

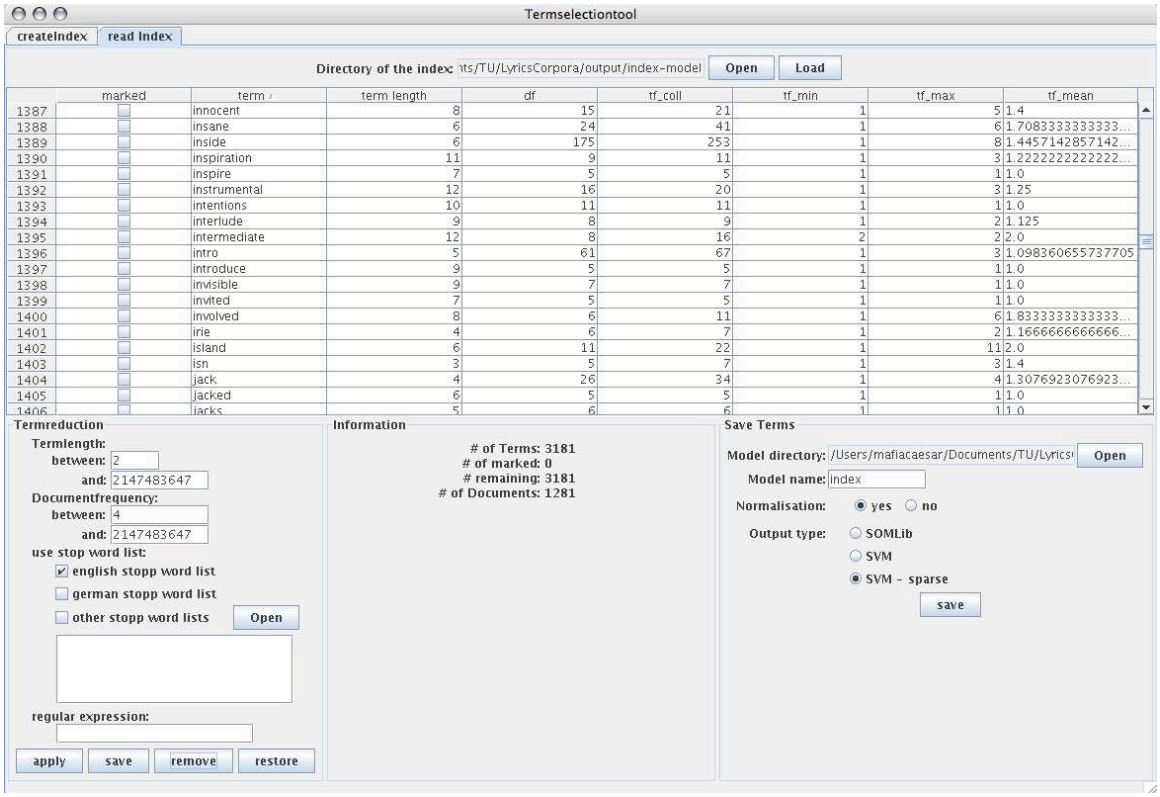


Abbildung 3.1: Benutzeroberfläche von TeSeT

hingegen listet für alle Attribute in der ganzen Dokumentensammlung auf, wie oft diese in allen Dokumenten auftreten.

Folgendes Beispiel zeigt ein Individual Vector File. In $XDIM ist die Anzahl der Dokumente eingetragen. In $VEC_DIM ist die Anzahl der Terme in der ganzen Dokumentensammlung gespeichert. Falls nun ein Term im jeweiligen Dokument vorkommt, wird der TFIDF-Wert für diesen Term in die entsprechenden Zeile des Dokumentes eingetragen. Sollte der Term im Dokument nicht vorkommen, so wird der Wert 0 an die entsprechenden Stelle geschrieben. Die horizontale Reihenfolge der Terme im Individual Vector File entspricht dabei der vertikalen Reihenfolge der Terme im Template Vector File.

```
$Type vec_tfxidf
$XDIM 4
$YDIM 1
$VEC_DIM 8
0 0 3.038 0 0 0 0.647 6.116 songtexte/text_1.txt
2.187 1.052 0 0 0.987 0.867 5.176 0.611 songtexte/text_2.txt
4.375 0 0 0 0 1.735 0.647 0 songtexte/text_3.txt
0 1.052 3.038 8.102 2.961 8.679 2.588 3.669 songtexte/text_4.txt
```

Im unteren Beispiel wird die entsprechende Template Vector Datei dargestellt. In $XDIM ist die Anzahl der Attribute, welche für jeden Term im ganzen Korpus berechnet werden, festgehalten. Diese Attribute sind unter anderen die Dokumentfrequenz, Termlänge und Termfrequenz des jeweiligen Terms. $YDIM entspricht dabei der Anzahl der eingelesenen Dokumente. Der Wert in $VEC_DIM gibt die Dimension des Vektors wieder.

```
$Type template
$XDIM 7
$YDIM 4
$VEC_DIM 8
0 addict 71 157 1 11 2.211
1 craziest 74 144 1 9 1.945
2 different 77 215 1 11 2.792
3 you 77 155 1 12 2.012
4 and 79 145 1 9 1.835
5 bedroom 111 273 1 14 2.459
6 sleeping 111 273 1 14 2.459
7 lovesongs 115 266 1 14 2.313
```

Mit dem TeSeT Tool hat man auch die Möglichkeit die Werte auf das Intervall [0,...,1] zu normalisieren. Dabei muss man lediglich nur die Option „Normalisation“ auswählen (siehe Abbildung 3.1).

3.4 Text Classification Features

Im folgenden Abschnitt werden die verschiedenen Features, welche in dieser Arbeit zum Einsatz kommen, näher beschrieben. Sie resultieren aus den beiden Techniken der Attributauswahl und der Attributextraktion, welche im Kapitel 2.5 behandelt wurden.

3.4.1 Bag of Words

Es gibt verschiedene Möglichkeiten für die Definition eines Features. Am häufigsten wird der Text in einem Dokument in einzelne Wörter zerlegt (auch „Tokenisierung“ genannt), von welchen jeder als Attribut definiert wird. In diesem Fall spricht man von einem „bag-of-words“ Ansatz (ein Beutel Wörter). Bei diesem Ansatz wird der Text mit den einzelnen Termen in einen Feature-Vektor transformiert, bei welchem jedes Element die Anwesenheit bzw. Abwesenheit eines Wortes im Dokument angibt. Die Dimension des daraus resultierenden Feature-Vektors ist dadurch sehr groß. Durch entsprechende Feature-Selection-Methoden kann die Dimension des Feature-Vektors erheblich reduziert werden, ohne dabei die Klassifikationsgenauigkeit zu verlieren.

3.4.2 Text-Statistic Features

Unter den Text-Statistic Features versteht man syntaktische Features wie z.B. Satzpunkte, Beistriche und Anführungszeichen. Die Anzahl jedes dieser Features steht in Beziehung zu der Anzahl der Wörter im Dokument. In der Tabelle 3.1 werden die Text-Statistic Features aufgelistet.

3.4.3 Part of Speech Features

Wörter einer Sprache lassen sich nach verschiedenen Gesichtspunkten in Wortarten (engl. part of speech) oder auch Wortklassen einteilen. Nach seiner Funktion innerhalb eines Satzes lässt sich ein Wort einer syntaktischen Kategorie zuordnen: wie z.B. Substantive, Verbe, Adjektive, usw. Für die POS Featureerkennung wird in dieser Arbeit auf das für

Nr	Feature Type	Feature Set
1.	Text-Statistics Features	# of question marks
2.		# of digits
3.		# of dots
4.		# of semicolons
5.		# of colons
6.		# of commas
7.		# of exclamation marks

Tabelle 3.1: Text-Statistics Features

Forschungszwecke frei verfügbare System LingPipe[1] zurückgegriffen. Die Wörter im Dokument werden hierbei mit dem Lingpipe POS-Tagger markiert, wodurch jedes einzelne Part-of-Speech Feature dem Text entnommen werden kann. Bei der Extrahierung dieser Features wird angenommen, dass die POS-Features den Stil eines Dokumentes zu einem solchen Ausmaß wiedergeben würden, dass das Dokument in verschiedene Genreklassen eingeordnet werden kann. Die Repräsentation eines Dokuments wird als Vektor mit 9 POS Features dargestellt, wobei jeder einzelne Wert im Vektor für ein POS-Feature steht. Dieser Wert steht in Relation zu der Anzahl aller Wörter im Dokument. Die vollständige Liste der POS-Features, zusammen mit den entsprechenden LingPipe-Labels, kann man der Tabelle 3.2 entnehmen.

3.4.4 Rhyme Features

Mit den Rhyme Features wird versucht verschiedene Rhyme-Muster im Songtext zu finden. Bei einem Songtext mit dem Musikgenre Hip-Hop/Rap ist zum Beispiel die Wahrscheinlichkeit größer, dass sich die letzten Wörter aufeinander folgender Zeilen reimen, als bei dem Songtext einer Ballade mit dem Musikgenre R&B. Diese Idee führte dazu verschiedene Rhyme-Features für die Unterscheidung von Songtexten nach Musikgenres zu bestimmen.

Der Reim ist ein sprachliches Phänomen, das auf dem Gleichklang von Silben beruht. Der Gleichklang von Silben zweier Wörter ist jedoch nicht nur auf die Gleichheit der letzten 2 oder 3 Zeichen zurückzuführen. Im Englischen haben zum Beispiel die beiden Wörter „tie“ (auf Deutsch: Band) und „fly“ (auf Deutsch: Fliege) nicht dieselben Endsilben. Dennoch reimen sich die Wörter in ihrer Aussprache. Wie man nun dem oben

[1] http://www.alias-i.com/lingpipe/

Nr	Feature Type	POS-Labels	Feature Set
1.	POS Features	NN, NNP, NNS	# of nouns
2.		VVB, VVD, VVG, VVI, VVN, VVNJ, VVGJ,VVGN,VVZ	# of verbs
3.		PNR	# of rel_pronouns
4.		II	# of prepositions
5.		RR, RRR, RRT	# of adverbs
6.		A, AN, THE	# of articles
7.		PN, PND, PNG	# of pronouns
8.		VM, VBB, VBD, VBG, VBI, VBN, VBZ, VDB, VDD, VDG, VDI, VDN, VDZ, VHB, VHD, VHG, VHI, VHZ	# of modals
9.		JJ, JJR, JJT	# of adjectives

Tabelle 3.2: Part of Speech Features

angeführten Beispiel entnehmen kann, ist für die Analyse von reimenden Wörtern der Vergleich der Endsilben zweier Wörter ungenügend. Aus diesem Grund wird die Analyse um den Vergleich von Phonemen erweitert.

In der menschlichen Sprache ist ein Phonem eine Menge von Sprachtönen oder Zeichenelementen, welche untereinander kognitiv equivalent sind. Es ist eine Einheit, welche zwischen verschiedenen Wörtern oder Morphemen (ist die kleinste bedeutungstragende Einheit einer Sprache auf der Inhalts- und Formebene) unterscheidet. Ändert man zum Beispiel die Einheit eines Wortes von einem Phonem zu einem anderen, resultiert dies entweder zu einem anderen oder zu einem sinnlosen Wort. Würde man jedoch das Element mit einem anderen desselben Phonems ändern, so würde man dasselbe Wort mit einer anderen Buchstabierung erhalten.

Für die Analyse der Songtexte nach verschiedenen Phonemen wird ein Java-Modul des Projektes ASP (Analysing Sound Pattern)[2] verwendet. Das Ziel dieses Projektes ist ein frei zugängliches Tool für Literaturstudenten verfügbar zu machen, um ihnen bei der Analyse von Klangmustern in Text zu unterstützen.

In dieser Arbeit erfolgt die Analyse auf Phonemen nur bei Songtexten, die in der englischen Sprache vorliegen, da die Regeln für die Bildung der Phoneme in dem oben

[2]http://www2.eng.cam.ac.uk/ tpl/asp/

angeführten Projekt nur für die englische Sprache existieren. Für die Bildung von Phonemen in verschiedenen Sprachen sind Unmengen von Regeln erforderlich. Aus diesem Grund wird in dieser Arbeit die Phonemanalyse nur auf die Sprache Englisch eingeschränkt. Im Falle eines in einer anderen Sprache vorliegenden Songtextes wird dieser auf Gleichheit der Endung jedes Wortes am Ende jeder Zeile untersucht. Die Regeln zur Extraktion der einzelnen Wortendungen sind folgende (wird nur an die letzten Wörter jeder Zeile angewendet):

1. Sollte das letzte Zeichen eines Wortes ein Konsonant sein, so wird soweit nach links gegangen, bis man auf einen Vokal trifft. Der/die Konsonant/en bilden dann mit dem Vokal die Wortendung. Handelt es sich bei dem Zeichen vor dem Vokal wieder um einen Vokal, so wird dieses Zeichen ebenfalls zur Wortendung hinzugefügt. z.B. Konzentrat-ion, Gab-el, Löff-el, Beleucht-ung, Kast-en, Räub-er, usw.

2. Sollte das letzte Zeichen eines Wortes ein Vokal sein, so geht man soweit nach links, bis man auf einen Konsonant trifft. Hier wird jedoch der Konsonant nicht zur Wortendung hinzugefügt. z.B. Beicht-e, Leucht-e, Spiegel-ei usw.

Die oben angeführten Regeln haben bei der Bestimmung der Wortendungen 2 wichtige Vorteile:

- Die Einschränkung auf eine einzelne Sprache ist nicht gegeben. Die Bestimmung der Wortendungen können sowohl in der deutschen, spanischen, türkischen und einer anderen Sprache angewendet werden.

- Die Anzahl der letzten Zeichen, mit denen die Wortendung gebildet wird, ist nicht fix festgelegt. Es können sowohl die Wortendungen von Wörtern wie z.B. Missbrauch wie auch die von Wörtern wie z.B Käs-e bestimmt werden.

Nachdem die Wortendungen bzw. Phoneme der letzten Worte in jeder Songzeile bestimmt wurden, werden diese auf Gleichheit miteinander verglichen, um verschiedene Rhyme-Muster im Songtext festzustellen. Die Rhyme-Muster bzw. Features nach welchen der Songtext analysiert wird sind folgende: AB, AA, ABAB, ABBA, AABB. Bei den ersten beiden Mustern werden alle zwei aufeinander folgenden Songzeilen auf Gleichheit (AA) bzw. Ungleichheit (AB) der beiden Wortendungen am Ende jeder Zeile untersucht.

Bei den anderen Features werden die jeweiligen Muster bei allen vier aufeinander folgenden Songzeilen festgestellt. Je nachdem welche Muster im Songtext vorgefunden werden, werden die Werte im entsprechenden Rhyme-Feature gespeichert. Die jeweiligen Werte der Rhyme-Features stehen dabei in Beziehung zu der Anzahl der Zeilen im Songtext. Im Folgenden wird die Extraktion der Rhyme Features in dieser Arbeit anhand von zwei konkreten Beispielen erläutert.

Beispiel 1	Beispiel 2
Girl our love is dying	Changing up your living
Why did you stop trying	For a loving transition
I never been a quitter	Girl it's a mission trying
But I do deserve better	Screaming at each other has become our tradition

Im ersten Beispiel sind die Wörter am Ende jeder Zeile „dying“, „trying“, „quitter“ und „better“. Somit ergeben sich folgende Werte für die Rhyme Features: AB: 1, AA: 2, ABAB: 0, ABBA: 0, AABB: 1. Im zweiten Beispiel reimt sich jede zweite Zeile, wodurch man folgende Werte für die Rhyme Features erhält: AB: 3, AA: 0, ABAB: 1, ABBA: 0, AABB: 0. Je nachdem ob ein englischer Songtext vorliegt, werden entweder die Phoneme oder die Endungen, welche nach den oben genannten Regeln bestimmt werden, am Ende jeder Zeile miteinander verglichen. Für die Identifikation der Sprache in einem Songtext, wird ein zusätzliches Feature benötigt, auf welches im nächsten Abschnitt näher eingegangen wird.

In der Tabelle 3.3 sind die Features, welche mit dem Rhyme-Feature-Modul extrahiert werden, aufgelistet. Wie man es aus der Liste entnehmen kann sind, abgesehen von den reimenden Muster-Features, Features wie z.B. Anzahl der unique words (numberofwords), Gesamtanzahl der Wörter im Text (words), durchschnittlicher Wortschatz (wordpool) und die Anzahl der Buchstaben (chars) ebenfalls inkludiert. Dabei steht das Feature „words“ für die Gesamtanzahl der Wörter, wohingegen das Feature „numberofwords“ die Anzahl der unterschiedlichen Wörter im Dokument wiedergibt. Würde man nun in einem Dokument Wörter wie z.B. „und“ zweimal, „Ich“ einmal und „Du“ ebenfalls einmal vorfinden, so würde man den Wert 4 für das Feature „words“ und den Wert 3 für das Feature „numberofwords“ erhalten. Der durchschnittliche Wortschatz (feature „word-

pool“) ergibt sich dann durch den Quotienten der beiden Werte (numberofwords/words).

Nr	Feature Type	Feature Set
1.	Rhyme Features	# of AA
2.		# of AB
3.		# of AABB
4.		# of ABAB
5.		# of ABBA
6.		# of words
7.		# of numberofwords
8.		# of wordpool
9.		# of chars

Tabelle 3.3: Rhyme Features

Evaluierung der Rhyme Detection

In diesem Kapitel soll das Rhyme Detection Modul, welches in dieser Arbeit für die Bestimmung der reimenden Wörter entwickelt wurde, auf eine Untermenge des Sing365-Korpus (Abschnitt 5.1) getestet werden. Dazu werden aus jedem Genre Lyrics entnommen, auf welchen dann die Rhyme Detection zuerst manuell und dann vom Modul durchgeführt wird. Zu den Lyrics werden ebenfalls Gedichte, welche dem Internet entnommen wurden, hinzugefügt mit der Absicht das Rhyme Detection Modul auch auf eine andere Art von Text testen zu können. Eine vollständige Liste der Lyrics bzw. der Gedichte, welche für diese Evaluierung verwendet werden, befindet sich in der Tabelle 3.4.

Wie es im vorherigen Kapitel bereits erwähnt wurde, werden die Rhyme Features, im Falle eines in der englischen Sprache vorliegenden Songtextes, über die Phoneme oder andernfalls über die mit den Regeln bestimmten Wortendungen ermittelt. Bei dem Test soll jedoch auch die einfache Variante überprüft werden, mit der man die Wortendungen aus den beiden letzten Zeichen erhält. Im Modul wurden somit alle drei eben genannten Verfahren implementiert, um deren Genauigkeit bezüglich der Rhyme Dectection miteinander zu verlgeichen. In den Tabellen 3.5, 3.6 und 3.7 sind die Testergebnisse der manuellen und der vom Modul durchgeführten Rhyme Detection bei allen drei Verfahren

Nr	Title
1.	3 Doors Down – Sarah Yelling
2.	Alanis Morissette – Perfect
3.	Babyface – Bedtime
4.	Babyface – Care for me
5.	Babyface – Gone too soon
6.	Bruce Springsteen – 10th Avenue Freeze out
7.	Confident Poem
8.	Janet Jackson – Again
9.	Janet Jackson – Alright
10.	Jimmy Cliff – Sitting in Limbo
11.	Jimmy Cliff – Vietnam
12.	Johnny Cash – Get Rhythm
13.	LL Cool J – Fast Peg
14.	Love Poem
15.	Madonna – Shine a light
16.	One Hope Present Poem
17.	Shania Twain – All fired up
18.	Shania Twain – Her Story
19.	Snoop Dogg – Balls of Steel
20.	Snoop Dogg – Fuck with Dre Day

Tabelle 3.4: Liste der Rhyme-Lyrics

ersichtlich. Für jedes Rhyme Feature wird der Wert, welcher die Anzahl des jeweils gefundenen Rhymemusters darstellt, eingetragen. Die Zeilen stellen die jeweiligen Songtexte bzw. Gedichte dar.

Am Schlechtesten fallen die Ergebnisse der Rhyme Detection mit der Variante, die Wortendung aus den letzten 2 Zeichen zu bilden, aus (Tabelle 3.5). In nur 6 von 20 Texten werden alle Rhyme Features vom Modul erfasst. Die Wortendungen auf diese Weise zu bilden macht im Allgemeinen wenig Sinn, da man mit diesem Verfahren zum Beispiel bei Worten mit zwei gleichen Zeichen am Schluss keine sinnvolle Wortendung erhält.

In der Tabelle 3.6 der Rhyme Detection, welche mit der 2. Variante durchgeführt wurde, ist zu erkennen, dass die Ergebnisse besser ausfallen als die bei der 1. Variante. Die Werte, die mit dem Modul ermittelt wurden, stimmen in 8 von 20 Texten mit denen der manuellen Rhyme Detection überein. Die Unterschiede, die in den restlichen 12 Texten vorzufinden sind, sind dabei nicht so groß wie beim ersten Ansatz. Der Tabelle ist zu entnehmen, dass in manchen Fällen ein paar AA-Muster nicht erfasst werden. Daran ist

gut zu erkennen, dass man besonders in der englischen Sprache an der Phonembildung zur Erkennung reimender Wörter nicht vorbeikommt. Dies kann mit einem einfachen Beispiel begründet werden: Wörter wie „you“ and „two“ reimen sich zwar, werden durch den Vergleich einfacher Wortendungen jedoch nicht erkannt.

Wie man anhand der Tabelle 3.7 erkennen kann, stimmen beim Phonem-Verfahren die Werte der manuellen mit denen der vom Modul durchgeführten Rhyme Detection in 14 von 20 Texten überein. Bei den übrigen 6 Texten sind die Unterschiede nur sehr gering. Dies ist ein ziemlich gutes Ergebnis und deutet darauf hin, dass sich Phoneme zur Erkennung reimender Wörter ziemlich gut eignen.

	Manuell					Detected				
	AA	AB	AABB	ABAB	ABBA	AA	AB	AABB	ABAB	ABBA
1.	1	56	0	3	0	1	56	0	3	0
2.	0	46	0	0	0	0	46	0	0	0
3.	4	50	0	1	0	6	48	0	1	0
4.	2	41	0	0	0	2	41	0	0	0
5.	4	29	1	0	0	4	29	1	0	0
6.	4	49	0	0	0	6	47	0	0	0
7.	0	8	0	1	0	0	8	0	1	0
8.	2	36	1	0	1	5	33	1	0	1
9.	10	20	4	2	2	5	25	1	0	0
10.	6	28	0	2	0	3	31	0	2	0
11.	10	14	4	1	1	8	16	2	1	1
12.	9	22	4	0	1	11	20	6	0	0
13.	10	33	6	0	0	7	36	2	0	0
14.	0	17	0	0	0	0	17	0	0	0
15.	12	15	7	1	1	8	19	4	1	0
16.	6	6	5	0	0	6	6	3	1	1
17.	3	22	0	2	0	2	23	0	2	0
18.	7	26	0	1	0	7	26	0	1	0
19.	13	53	5	0	0	10	56	4	0	0
20.	18	69	7	3	1	14	75	4	3	1

Tabelle 3.5: Rhyme Detection über die letzten beiden Zeichen eines Wortes

	Manuell					Detected				
	AA	AB	AABB	ABAB	ABBA	AA	AB	AABB	ABAB	ABBA
1.	1	56	0	3	0	1	56	0	3	0
2.	0	46	0	0	0	0	46	0	0	0
3.	4	50	0	1	0	4	50	0	1	0
4.	2	41	0	0	0	5	38	1	1	1
5.	4	29	1	0	0	4	29	1	0	0
6.	4	49	0	0	0	8	45	1	0	1
7.	0	8	0	1	0	0	8	0	1	0
8.	2	36	1	0	1	4	34	1	0	1
9.	10	20	4	2	2	6	24	2	0	0
10.	6	28	0	2	0	3	31	0	2	0
11.	10	14	4	1	1	8	16	2	1	1
12.	9	22	4	0	1	8	23	3	0	0
13.	10	33	6	0	0	6	37	1	0	0
14.	0	17	0	0	0	0	17	0	0	0
15.	12	15	7	1	1	12	15	7	1	1
16.	6	6	5	0	0	6	6	3	1	1
17.	3	22	0	2	0	4	21	1	3	1
18.	7	26	0	1	0	10	23	2	1	1
19.	13	53	5	0	0	9	57	2	0	0
20.	18	69	7	3	1	18	69	7	3	1

Tabelle 3.6: Heuristische Rhyme Detection

	Manuell					Detected				
	AA	AB	AABB	ABAB	ABBA	AA	AB	AABB	ABAB	ABBA
1.	1	56	0	3	0	1	56	0	3	0
2.	0	46	0	0	0	0	46	0	0	0
3.	4	50	0	1	0	3	51	0	1	0
4.	2	41	0	0	0	2	41	0	0	0
5.	4	29	1	0	0	4	29	1	0	0
6.	4	49	0	0	0	5	50	0	0	0
7.	0	8	0	1	0	0	8	0	1	0
8.	2	36	1	0	1	2	36	1	0	1
9.	10	20	4	2	2	10	20	4	2	2
10.	6	28	0	2	0	5	29	0	2	0
11.	10	14	4	1	1	10	14	4	1	1
12.	9	22	4	0	1	8	23	3	0	1
13.	10	33	6	0	0	9	34	4	0	0
14.	0	17	0	0	0	2	15	0	0	0
15.	12	15	7	1	1	12	15	7	1	1
16.	6	6	5	0	0	5	7	3	0	0
17.	3	22	0	2	0	3	22	0	2	0
18.	7	26	0	1	0	8	25	0	0	0
19.	13	53	5	0	0	13	53	5	0	0
20.	18	69	7	3	1	19	70	8	3	2

Tabelle 3.7: Rhyme Detection über Phoneme

3.4.5 Language Feature

Im Language Feature wird die jeweilige Sprache des Songtextes gespeichert. Obwohl der größte Teil der Songlyrics in den verwendeten Korpora (Abschnitt 5.1) in englischer Sprache sind, wird dieses Feature in dieser Arbeit in die Featuremenge aufgenommen, da Songtexte durchaus in einer anderen Sprache vorliegen können. Es ist nicht selten, dass Songtexte mit dem Genre Klassik in Italienisch oder Französisch vorzufinden sind. Die Sprache des Textes wird hierbei mit dem Perl Script „TextCat“ identifiziert. Dieses Script implementiert den Text Categorization-Algorithmus von Canvar [Can94], welcher einen N-gram-basierten Ansatz zur Kategorisierung von Text darstellt. Ein N-Gram kann als Teil eines längeren Strings betrachtet werden. Zum Beispiel kann das Wort DATA als Tri-Grams _DAT, DATA, ATA_, oder Quad-Grams _DA, DAT, ATA, TA_ dargestellt werden, wobei das Unterzeichen ein führendes oder zurückhängendes Leerzeichen sein kann.

Dieser N-gram-basierter Algorithmus, welcher auch tolerant gegenüber Schreibfehlern ist, eignet sich sehr gut für „language classification“. In einem Experiment erzielt Canvar eine 99,8 prozentige Klassifikationsrate am Usenet Newsgroup - Korpus [Can94]. Die natürlichen Sprachen, welche in dieser Arbeit mit TextCat identifiziert werden, sind folgende:

- Englisch
- Finnisch
- Französisch
- Deutsch
- Griechisch
- Indonesisch
- Italienisch
- Norwegisch
- Portugiesisch
- Spanisch

- Schwedisch

4 Klassifikation

Es existieren viele Arten von Klassifikatoren wie z.B Entscheidungsbäume, Regressionsmethoden, probabilistische Klassifikatoren und künstliche neuronale Netzwerke. In dieser Arbeit wird jedoch der Erstellung der verschiedenen Features mehr Aufmerksamkeit geschenkt. Aus diesem Grund werden die zwei typischen Textklassifikatoren Support Vektor Maschinen, welche sich bei der Textklassifizierung schon sehr oft bewährt haben, und der Naive Bayes für die Songtextklassifikation angewendet. Dieses Kapitel soll einen Überblick über die theoretischen Grundlagen der beiden Klassifikatoren geben.

4.1 Support Vektor Maschinen

Support Vector Machines (SVM) sind ein neues und vielversprechendes Machine Learning-Verfahren, welches von Vapnik und dessen Gruppe im AT&T Bell Labor entwickelt wurde ([Burges96], [Vap95]). Joachims führte erstmals in seiner Arbeit diese Methode in die „Text Categorization“ ein [Joa98]. Darin untersucht er, welche Eigenschaften Support Vektor Maschinen bei Lernen eines Text Klassifikators haben. Weiters erläutert er die Gründe warum sich Support Vektor Machines für die Kategorisierung von Textdokumenten sehr gut eignen. Dazu führt er in dieser Arbeit die wesentlichen Eigenschaften von Text an:

- Hochdimensionaler Featureraum: Beim Lernen von Text-Klassifikatoren, haben diese es oft mit sehr vielen Features zu tun. Da SVMs robust gegenüber „Overfitting“ sind, sind diese durchaus in der Lage mit einem derart großen Featureraum umzugehen. Der Ausdruck „Overfitting“ steht für das Phänomen, dass Klassifikatoren nur jene Daten, mit denen sie trainiert wurden, gut und neue Daten schlecht klassifizieren.

- Wenige irrelevante Features: Eine Möglichkeit die hohe Dimensionalität des Textes zu vermeiden, ist die Annahme, dass der Großteil der Features bedeutungslos für die Semantik des Textes ist. Diese irrelevanten Features werden bei der Feature

Selection, welche im Kapitel 2.5 erläutert wurde, bestimmt. Ein Experiment in Joachims Arbeit zeigt jedoch, dass die am unwichtigsten angesehenen Features wichtige Informationen tragen können, wodurch sie in einer gewissen Weise als relevant angesehen werden können.

- Die meisten Text Categorization Probleme sind linear trennbar: Viele der bekannten Textkorpora beinhalten Kategorien, die linear trennbar sind. Und die Idee der Support Vector Machines ist eben derartige lineare Seperatoren zu finden.

Der Einsatz von SVMs ist sehr vielversprechend, da ihre Stärken in den Bereichen der wichtigsten Aspekte von TC-Aufgaben liegen. Die wesentliche Funktionsweise der SVM ist die unter all den möglichen Trennebenen ((n-1)-dimensionale Hyperebenen im n-dimensionalen Raum) eine Hyperebene zu bestimmen, welche die Trainingsbeispiele so voneinander trennt, dass der kleinste Abstand zur Hyperebene, dem sogenannten „margin“, für die Beispiele beider Klassen maximiert wird (largest margin). Dabei wird jene Ebene gewählt, welche eine Äquidistanz zu den beiden parallelen Ebenen, welche die Beispiele trennen, aufweist. Die Hyperebene wird nach dem Training, d.h. der Berechnung der Hyperebene zwischen den Trainingsbeispielen, als Entscheidungsfunktion benutzt. Diese „optimale“ Entscheidungsebene wird von jenen Trainingsbeispielen bestimmt, die am nächsten zu den Beispielen aus der anderen Klasse liegen. Diese Trainingsbeispiele werden Support Vectors genannt.

Es ist oft der Fall, dass das Klassifikationsproblem nicht linear ist. Die SVM lösen dieses Problem, indem sie die Daten durch eine Kernel-Transformation in einen Raum höherer Dimension abbilden, in welchem das Klassifikationsproblem linear trennbar ist. In der folgenden Abbildung wird die Funktionsweise der SVM für 2 Klassen im 2-dimensionalen Raum illustriert.

In der Abbildung 4.1 kämen verschiedene Trennungslinien als Entscheidungsebene in Frage. Die SVM-Methode wählt jedoch die mittlere Linie (dickere Linie), welche den größten Abstand zwischen zwei Beispielelementen aufweist. Die kleinen Kreise und Kreuze repräsentieren hierbei positive und negative Trainingsbeispiele. Die kleinen Kästchen markieren die Support Vectors.

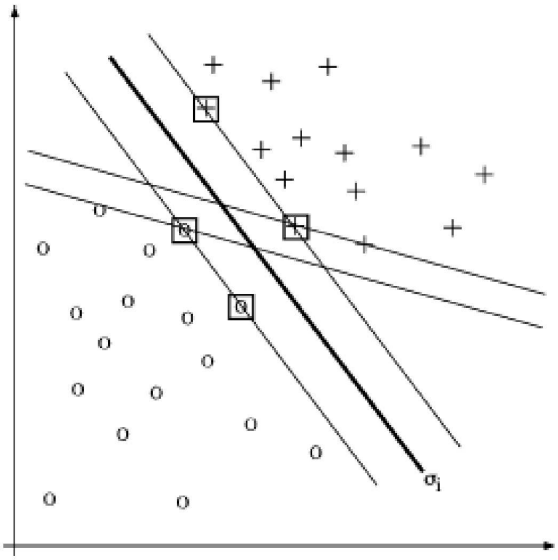


Abbildung 4.1: SVM im 2 dimensionalen Raum [Seb02]

4.2 Naive Bayes

Der Naive Bayes Klassifikator ist ein einfacher probabilistischer Klassifikator, welcher aus dem berühmten Bayestheorem (Formel 4.1) hergeleitet wurde [Lewis98]. Hierbei wird angenommen, dass alle möglichen Ereignisse (in unserem Fall sind es die Dokumente) in eine von ec Klassen, $C = (c_1, \ldots, c_k, \ldots, c_{ec})$, fallen. C ist eine beliebige Variable, dessen Werte die ec Klassen beinhalten, wohingegen X ebenfalls eine beliebige Variable darstellt mit den Vektoren als Wert, wobei ein Vektor für ein Dokument steht und die verschiedenen Featurewerte $x = (x_1, ..., x_j, ..., x_d)$ beinhaltet. $P(C = c_k|X = x)$ ist dabei die bedingte Wahrscheinlichkeit, dass ein Dokument zur Klasse c_k gehört, mit der Vorraussetzung, dass das Dokument den Featurevektor x hat.

$$P(C = c_k|X = x) = P(C = c_k) \times \frac{P(X = x|C = c_k)}{P(x)} \tag{4.1}$$

Die Formel des Bayes Theorems kann nun durch die folgende Formel (Formel 4.2) vereinfacht werden, wissend dass c_k und x Werte der Zufallsvariablen C und X sind.

$$P(c_k|x) = P(c_k) \times \frac{P(x|c_k)}{P(x)} \tag{4.2}$$

Der Ausdruck $P(x)$ steht hierbei für die Wahrscheinlichkeit, dass ein Dokument den Vektor x als Repräsentation hat. $P(c_k)$ gibt die Wahrscheinlichkeit an, dass ein Dokument der Klasse c_k angehört. Da man nun die Wahrscheinlichkeit $P(c_k|x)$ normalerweise nicht kennt, muss diese aus den Daten geschätzt werden. Dazu schätzt man die Wahrscheinlichkeit $P(x|c_k)$, $P(c_k)$ und $P(x)$ und kombiniert diese, um einen Schätzwert für $P(c_k|x)$ zu bekommen. Da die Anzahl der Kombinationen für die Werte von $x = (x_1, \ldots, x_j, \ldots, x_d)$ astronomisch hoch ist, kann dies für die Schätzung der Wahrscheinlichkeit $P(x|c_k)$ zu Problemen führen. Aus diesem Grund wird beim Naive Bayes „naiv“ angenommen, dass zwei beliebige Attribute eines Dokumentenvektors statistisch voneinander unabhängig sind. Diese Annahme wird durch die folgende Gleichung verdeutlicht:

$$P(x|c_k) = \prod_{j=1}^{d} P(x_j|c_k) \tag{4.3}$$

Mit der Annahme, dass die Attribute voneinander unabhängig sind, können die Parameter jedes Attributs getrennt „gelernt“ werden, wodurch das Lernen um ein Vielfaches vereinfacht wird. Dem Dokument wird dann die Klasse mit der höchsten Wahrscheinlichkeit $P(c_k|x)$ aus der Formel 4.2 zugewiesen.

Es gibt verschiedene Variationen des Naive-Bayes Ansatzes. Sie erzielen bei praktischen Anwendungen häufig gute Ergebnisse, obwohl diese statistische Unabhängigkeit in der Realität selten zutrifft. Lewis gibt einen guten Überblick über die verschiedenen Richtungen, die die Forschung über Naive Bayes eingeschlagen hat [Lewis98].

5 Experimente

5.1 Data Sets

Die Daten, die den in dieser Arbeit durchgeführten Experimente zugrunde liegen, sind Songtextdokumente, die einem Musikgenre zugeordnet werden können. Anhand von diesen Dokumenten werden typische Merkmale extrahiert, die zur Beschreibung dieses Genres verwendet werden.

In dieser Arbeit werden eigens erstellte Textkorpora benutzt, da für die Klassifizierung von Songtexten derzeit noch keine geeigneten Korpora existieren. Ein Textkorpus (od. einfach Korpus genannt) ist eine große und strukturierte Sammlung von Textdokumenten, welche meist für linguistische Analysen genutzt werden. Die Songtextkorpora, die für die Klassifikationsexperimente verwendet werden, wurden mit Hilfe von Lyric-Webseiten, welche in Songtextportalen vorzufinden sind, erstellt. Dieser großen gesammelten Menge an Webseiten wurden die Songtexte entnommen und in entsprechende Songtextdokumente weiterverarbeitet. Die manuelle Erstellung jedes einzelnen Dokumentes wäre hierbei eine sehr aufwendige Aufgabe gewesen. Aus diesem Grund kamen so genannte WebCrawler zum Einsatz, mit welchen die Lyric-Webseiten dem Internet entnommen werden konnten. Ein WebCrawler ist ein Computerprogramm, das automatisch das World Wide Web durchsucht und Webseiten analysiert. Wie beim Internetsurfen gelangt ein Webcrawler über Hyperlinks von einer Webseite zu weiteren URLs. Dabei werden alle aufgefundenen Adressen gespeichert und der Reihe nach besucht.

Die Korpora wurden daraufhin so strukturiert, dass alle Songtexte, welche einem Musikgenre angehören, in ein (Genre-)Verzeichnis gespeichert wurden. Das Musikgenre jedes einzelnen Songtextes wurde über das jeweilige Musikportal ermittelt. Auf diese Weise wurden zwei Songtextkorpora erstellt, welche in den folgenden Abschnitten näher beschrieben werden.

5.1.1 Sing365-Korpus

Die Songtexte aus diesem Korpus wurden der Webseite www.sing365.com[1], welche über eine sehr große Sammlung an Songlyrics verfügt, entnommen. Wie es in der Einführung schon erwähnt wurde, geschah dies mit Hilfe eines Webcrawlers. Das Ergebnis war eine Sammlung von über 8000 HTML-Webseiten. Diese wurden daraufhin mittels Preprocessing in Textdokumente weiterverarbeitet.

Da die gesammelten HTML-Dokumente keine Metadaten über das Genre des jeweiligen Songtextes enthalten, wurden aus diesem Grund die Songtexte, entsprechend deren Zuordnung in www.allmusic.com[2], in die jeweiligen Musikgenres eingeteilt. Die gewählten Musikgenres sind: Hip-Hop/Rap, Country, R&B, Pop, Rock und Reggae. Bei der Wahl der Musikgenres wird bewusst auf globale Genres, welche als umfassende Kategorie angesehen werden, geachtet. Musikstile, welche feiner definierte Strömungen innerhalb eines solchen Genres darstellen, finden hierbei keine Verwendung. Dies kann mit dem folgenden Argument begründet werden, dass sich Musikstile meistens nur durch ihre musikalischen Eigenschaften unterscheiden, jedoch nicht durch ihren Text. Ein typisches Beispiel dafür wären Musikstile wie Punk Rock und Hard Rock. Sie mögen sich durch Rhythmus und Musiktempo unterscheiden, jedoch können beide dem Genre Rock zugeordnet werden. Dies ist ein Punkt, welcher unbedingt berücksichtigt werden muss, da bei der Songtextklassifizierung auf keine Eigenschaften im Audiobereich zurückgegriffen werden kann.

In der Tabelle 5.1 wird die vollständige Liste der Musikgenres mit der Anzahl der jeweiligen Dokumente im Korpus dargestellt. Es ist ersichtlich, dass es sich hierbei um einen kleinen Korpus handelt. Da es derzeit keine öffentlich zugänglichen Songtext-Korpora gibt, musste für diese Arbeit anfangs ein für die ersten Experimente geeigneter Korpus erstellt werden. Und da die Erstellung von Korpora allgemein sehr zeitaufwendig ist, wird der Sing365-Korpus für die ersten kleinen Experimente als ausreichend erachtet.

5.1.2 Parallelkorpus

Dieser Korpus besteht aus Songtexen, welche den Musikdateien einer großen Musikkollektion entsprechend im Internet aufgesucht wurden. Das Ergebnis ist eine Songtext-

[1] http://www.sing365.com
[2] http://www.allmusic.com

Genre	Anzahl
Hip-Hop/Rap	236
Rock	217
Pop	204
Reggae	192
R&B	229
Country	203
Total	1281

Tabelle 5.1: Sing365-Korpus

Sammlung, bei dem es zu jedem Songtext auch die entsprechende Musikdatei gibt. Die Musikkollektion, die hierbei zum Einsatz kam, weist über 7000 Musikdateien auf und ist privat von Robert Neumayer zusammengestellt worden. Für die Erstellung des Songtext-Korpus wurden bei der Suche der jeweiligen Songtexte drei verschiedene Musikportale verwendet. Bei den drei Musikportalen handelt es sich um die Webseiten lyrc.com.ar[3], sing365.com[4] und oldielyrics.com[5]. Diese wurden der Reihe nach mittels eines Skriptes abgefragt. Dabei wurde eine Suchanfrage, bestehend aus dem Namen des Musikkünstlers und des Musikstückes, an das Musikportal geschickt. Wenn die Antwort von dem jeweiligen Musikportal gültig war, wurde der Songtext der entsprechenden Musikdatei zugewiesen. Sobald die Antwort jedoch ungültig war, wurde die Anfrage an das nächste Musikportal weitergeleitet. Im Falle, dass keine Antwort von allen drei Musikportalen zurückkam, wurde der Songtext über Google aufgefunden. Nachdem die Songtexte dem Internet entnommen wurden, wurden diese daraufhin mittels Preprocessing in reine Textdokumente weiterverarbeitet. Hinsichtlich Zuordnung der einzelnen Genres bekam jedes Songtextdokument das Genre der entsprechenden Musikdatei zugewiesen.

Bei diesem großen Korpus kommen 37 verschiedene Musikgenres zum Einsatz, wobei hier nicht zwischen globalen Genres und Musikstilen unterschieden wird. So wird bei dieser Genreeinteilung darauf Bedacht gelegt durchaus auch ähnliche Genres miteinzubeziehen, um zu sehen, wie sich das Verfahren in diesen Fällen verhält. Die Zuordnung in globale Genres, von welchen die Vorteile bereits im vorherigen Abschnitt erläutert wurden, wird für zusätzliche Experimente dennoch durchgeführt, um Unterschiede in der Klassifikationsrate beim Einsatz von Genres und Musikstile festzustellen. Die Ergebnisse dieser Versuche werden im unteren Abschnitt behandelt (Abschnitt 5.2).

[3]http://www.lyrc.com.ar
[4]http://www.sing365.com
[5]http://www.oldielyrics.com

In der Tabelle 5.2 wird die vollständige Liste der Musikgenres mit der Anzahl der jeweiligen Dokumente im Korpus dargestellt. Es ist deutlich ersichtlich, dass die Dokumentenanzahl der Musikgenres nicht so ausgeglichen ist, wie beim Sing365-Korpus. Dies ist beabsichtigt, da auf diese Weise realistische Bedingungen für das Erlernen von verschiedenen Genrekonzepten geschaffen werden können. Denn in einer privaten Musiksammlung verfügt nicht jedes Musikgenre über diesselbe Anzahl an Musikdateien.

Das Genre Classic war anfangs in der Genreliste nicht vertreten. Um ein breiteres Spektrum an Musik zu erhalten, wurde die Musikkollektion in einer späteren Phase mit klassischen Musikdateien, welche verschiedenen klassischen Konzert-Cds entnommen wurden, ergänzt. Die entsprechenden Songtexte wurden ebenfalls in den Songtext-Korpus aufgenommen.

5.2 Evaluierung

Zur Evaluierung wird auf die Genreeinteilung der jeweiligen Songtextkorpora zurückgegriffen. Beim Sing365-Korpus handelt es sich bei der Genreeinteilung der Songtexte um eine Taxonomie von 6 bekannten Genres: Pop, Rock, Reggae, Country, R&B und Hip-Hop/Rap. Dabei ist es wichtig Songtexte von Künstlern zu wählen, die eindeutig einem bestimmten Genre zugeordnet werden können. Dem Genre Pop sind beispielsweise bekannte Künstler wie Britney Spears oder Michael Jackson zugeordnet. Das Genre Hip-Hop/Rap wird z.B. durch Künstler wie Eminem, 2pac oder 50 Cent repräsentiert. Die Zuordnung der Künstler in das jeweilige Musikgenre erfolgte nach dem Musikportal allmusic.com[6]. Beim Parallelkorpus hingegen sind 37 Musikgenres vertreten, wobei hier nicht zwischen globalen Genres und Musikstilen unterschieden wurde und die Genrezuordnung manuell erfolgte. Die vollständige Liste der Künstler mit dem jeweiligen Genre befindet sich im Anhang A.2.

Auf Grundlage der Genrezuordnungen, welche bei beiden Korpora für die durchgeführten Experimente als ground truth verwendet werden, ist es nun möglich einen Klassifikator zu trainieren und dessen Klassifikationen zu bewerten. Dazu werden einige der Songtextdokumente desselben Musikgenres herangezogen, um diesen zu definieren und andere um zu überprüfen, ob der Klassifikator tatsächlich jenes Musikgenre voraussagt,

[6]http://www.allmusic.com

Genre	Anzahl
Acid Punk	19
Alternative	478
Ambient	15
Avantgarde	110
Blues	23
BritPop	56
Christian Rock	38
Classic	513
Country	134
Dance	13
Dance Hall	10
Electronic	143
Emo	254
Experimental	10
Folk	46
Garage	41
Goth Metal	46
Grunge	120
Hard Rock	24
Hardcore	184
Hip-Hop	613
Indie	334
Industrial	22
Metal	572
New Metal	98
Pop	860
Post Punk	26
Punk Rock	1390
R&B	254
Reggae	49
Rock	715
Ska	37
Slow Rock	501
Soundtrack	27
Speech	53
Trip-Hop	52
World	4
Total	7884

Tabelle 5.2: Parallelkorpus

welchem das Songtextdokument zugeordnet ist. Dabei ist es entscheidend, dass Trainingsset und Testset disjunkt sind, d.h. dass keines der Songtextdokumente, dessen Zugehörigkeit geprüft wird, dazu verwendet wird, das jeweilige Musikgenre zu definieren, da Evaluationen auf dem Trainingsset im Allgemeinen zu optimistisch sind und daher einen falschen Eindruck der Qualität des Verfahrens vermitteln. Aus diesem Grund wird als Evaluierungsmethode auf die 10-Fold Cross Validation zurückgegriffen, bei der ersichtlich wird wie groß der Anteil an falsch klassifizierter Dokumente ist.

Die Ergebnisse, welche im folgenden Abschnitt behandelt werden, ergeben sich aus Experimenten mit folgendem Ablauf:

1. Für alle Songtextdokumente jedes Musikgenres werden die verschiedenen Features berechnet. Die BOW-Features werden mittels TFIDF-Funktion gewichtet.

2. Verschiedene Kombinationen der Features werden zu einem Feature-Set kombiniert.

3. Der χ^2-Wert jedes Features wird berechnet und der Größe nach in einer Liste angeordnet.

4. Die Termauswahl erfolgt durch das Aufnehmen der ersten n Features in den Featurevektor.

5. Trainieren des Klassifikators (SVM oder Naiver Bayes).

6. Bewertung der Genauigkeit des Klassifkators erfolgt mittels 10-Fold Cross Validation.

Alle beschriebenen Experimente werden mit den BOW, Rhyme, Language und POS Features durchgeführt. Auf Experimente mit Text-Statistic Features wird verzichtet, da syntaktische Eigenschaften wie z.B Punkte und Beistriche nicht in allen Songtexten vorhanden sind, wodurch die Ergebnisse der Experimente verzerrt werden können. Das Language Feature hingegen wird als fixer Bestandteil in allen Featurekombinationen aufgenommen. Anhand dieser Kombinationen soll überprüft werden, welche Feature-Sets sich am Besten für die Klassifikation von Songtextdokumenten eignen.

Mit Hilfe des χ^2-Tests werden die verschiedenen Features (BOW, Rhyme, POS, Language) nach deren χ^2-Werten absteigend in einer Liste angeordnet. So stehen Features

mit einem hohen Chi-Quadrat-Wert ganz oben in der Liste. Im Musikgenre Hip-Hop/Rap haben z.B Schimpfwörter wie n***a, sh*t, f*ck, usw. einen sehr hohen Chi-Quadrat-Wert, da sie signifikant für dieses Genre sind. In keinem anderen Genre können derartige Terme zahlreich vorgefunden werden.

Weiters soll überprüft werden, wie sehr die Anzahl der Features die Qualität beeinflusst. Deshalb werden von der χ^2-Liste die ersten n Features in den Featurevektor aufgenommen, mit welchem dann der Klassifikator in der Lage ist, jedes einzelne Songtextdokument im Korpus zu interpretieren. Die Experimente werden in drei Konfigurationen ausgeführt, bei welchen der Cut-off-Index n variiert wird: $n = 600$ (C600), $n = 800$ (C800) und $n = 1000$ (C1000).

5.2.1 Evaluierung der Experimente mit dem Sing365-Korpus

Auffallend an den Ergebnissen der folgenden Experimente ist, dass diese sehr gut ausgefallen sind. Als Höchstwert für die Klassifikationsgenauigkeit kann ein Wert von 91,710 Prozent verzeichnet werden. Dies kann man der Tabelle 5.3 (Experiment 5) entnehmen. In dieser Tabelle sind die Klassifikationsergebnisse der Experimente in allen drei Konfigurationen des Cutoff-Indexes ersichtlich. Die Konfigurationen der einzelnen Experimente, in welchen Stemming, Stopword Removal und verschiedene Featurekombinationen zum Einsatz kommen, werden in der Tabelle 5.4 aufgelistet.

Sing365-Korpus	**C600**		**C800**		**C1000**	
	SVM	**NBayes**	**SVM**	**NBayes**	**SVM**	**NBayes**
Exp. 1	91.334	88.076	**91.456**	89.071	89.773	84.933
Exp. 2	88.212	85.480	**88.729**	85.316	85.402	80.249
Exp. 3	89.929	87.465	**90.007**	87.665	87.431	85.245
Exp. 4	91.178	87.744	**91.432**	87.900	90.163	85.870
Exp. 5	91.556	87.744	**91.710**	88.134	89.805	85.636

Tabelle 5.3: Klassifikationsergebnisse des Sing365-Korpus. Die Konfiguration jedes einzelnen Experiments kann der Tabelle 5.4 entnommen werden. Die Werte sind in Prozent angegeben und wurden mit den Cut-off-Indizes von 600 (C600), 800 (C800) und 1000 (C1000) erzielt. Das beste Ergebnis jedes Experiments wird fett angezeigt.

	Konfiguration
Exp. 1	BOW, w/o SW, no stemming
Exp. 2	BOW, w/o SW, stemming
Exp. 3	BOW+POS, w/o SW, no stemming
Exp. 4	BOW+Rhyme, w/o SW, no stemming
Exp. 5	BOW+POS+Rhyme, w/o SW, no stemming

Tabelle 5.4: Die Konfigurationen der Experimente, die in Tabelle 5.3 durchgeführt wurden. Versuche mit/ohne Stopwörter werden mit „w/ SW“ bzw. mit „w/o SW“ bezeichnet. Versuche, bei denen das Verfahren Stemming durchgeführt wurde, werden mit „stemming“ bezeichnet oder andernfalls mit „no stemming“.

In der Tabelle 5.3 verdeutlichen die Experimente 1 und 2, dass Versuche, in denen „Stemming“ durchgeführt wurde, schlechtere Ergebnisse liefern, als jene, bei denen dieses Verfahren nicht angewendet wird. Da ersichtlich ist, dass Stemming in allen 3 Konfigurationen des Cutoff-Indexes zu schlechteren Ergebnissen führt, kann angenommen werden, dass dieses Verfahren sich allgemein nicht für die durchgeführten Experimente eignet. Es scheint nämlich so zu sein, dass die verschiedenen Varianten eines Wortes wichtige Informationen für den Klassifikator darstellen. So können zum Beispiel die Wörter „doing“ in einem Country-Text und „doin“ mit derselben Bedeutung in einem Hip-Hop-Text vorgefunden werden. In Hip-Hop-Texten sind derartige Abkürzungen sehr charakteristisch, da diese sehr oft beobachtet werden. Andere Beispiele für solche sind: „cruisin“ (cruising), „em“ (them) und „ya“ (you). Wendet man nun den Porter-Algorithmus [Port98] an den oben angeführten Worten „doing“ und „doin“ an, so würde man das auf dem gemeinsamen Wortstamm gebildete Wort „do“ für beide Worte erhalten. Bei den Worten „cruising“ und „cruisin“, würde man für beide Varianten das Wort „cruis“ erhalten. So ist deutlich zu erkennen, dass in beiden Fällen durch das Stemming wichtige Eigenschaften eines Textes verloren gehen, wodurch die Klassifikation von diesem erschwert wird. Die Möglichkeit des auf Stemming zurückführenden Verlustes wichtiger Informationen wurde bereits im Abschnitt 3.1 angesprochen.

Das beste Ergebnis in jedem Experiment wird in der Tabelle 5.3 fett angezeigt. Bei näherer Betrachtung fällt auf, dass die Klassifikationsrate bei allen Kombinationen mit dem Cutoff-Index 800 am höchsten ist. Dies lässt vermuten, dass die Features bei C800 die verschiedenen Songtextdokumente bereits so gut beschreiben, dass das Hinzufügen oder Entfernen von weiteren Attributen die Qualität nur beeinträchtigt. Allerdings muss fest-

gehalten werden, dass diesbezüglich sämtliche Formen der Interpretation spekulativer Natur sind, da statistisch gesehen zwischen den Ergebnissen kein signifikanter Unterschied besteht.

Aus den Werten kann man ebenfalls deutlich schließen, dass Support Vektor Maschinen die in dieser Arbeit durchgeführten Klassifikationsaufgaben besser meistern als der Naive Bayes-Klassifikator. Dies kann man besonders in der Tabelle 5.3 beim 4. Experiment mit dem Cutoff-Index 1000 gut erkennen, in welchem ein Unterschied von 5 Prozent zwischen SVM und dem Naive Bayes zu verzeichnen ist. Da die Stärken der Support Vektor Maschinen in den Bereichen der wichtigsten Aspekte von Textklassifizierungsaufgaben liegen und man die Songtextklassifikation der Textklassifizierung zuordnen kann, ist dieses Ergebnis nicht sehr verwunderlich.

An den Werten in der Tabelle 5.5 ist ebenfalls gut zu erkennen, dass Stemming nur zu einer unnötigen Reduktion von Termen führt, bei dem wertvolle Informationen verloren gehen. An derselben Tabelle ist ersichtlich, dass bei beiden Klassifikatoren das Weglassen der Stoppwörter zu besseren Ergebnissen führt. Dies verdeutlichen auch die Tabellen 5.6 und 5.7, in denen die Ergebnisse der Versuche mit verschiedenen Featuremengen angeführt werden. In vielen Fällen sind die Unterschiede ohne Stoppwörter deutlich bemerkbar wie z.B in Tabelle 5.6, worin die BOW Features mit den POS Features und den Rhyme Features kombiniert werden. Hier zeigt sich ein Unterschied von 3 Prozent. Die angeführten Werte wurden in allen drei Tabellen mit dem Cutoff-Index 800 erzielt, da mit diesem die Klassifikationsgenauigkeit am Besten ist.

Sing365-Korpus	**BOW (w/ SW, no stemming)**	**BOW (w/o SW, no stemming)**	**BOW (w/ SW, stemming)**	**BOW (w/o SW, stemming)**
SVM	90.632	91.456	86.416	88.729
N.Bayes	86.182	89.071	85.245	85.316

Tabelle 5.5: Klassifikationsergebnisse des Sing365-Korpus bei Benutzung des Bag-of-Word-Ansatzes (BOW). Die Werte sind in Prozent angegeben und wurden mit einem Cut-off-Index von 800 (C800) erzielt. Andere Bezeichnungen wie in Tabelle 5.4.

Bezüglich der verschiedenen Featurekombinationen kann festgestellt werden, dass der Ansatz, in denen BOW und POS Features kombiniert werden, die schlechtesten Resultate erzielen, der BOW-Ansatz am zweitschlechtesten und der BOW+Rhyme-Ansatz am

Sing365-Korpus	**BOW + POS (w/ SW)**	**BOW + POS (w/o SW)**	**BOW + Rhyme (w/ SW)**	**BOW + Rhyme (w/o SW)**
SVM	87.275	90.007	88.524	91.432
N.Bayes	86.807	87.665	86.651	87.900

Tabelle 5.6: Klassifikationsergebnisse des Sing365-Korpus bei Benutzung des Bag-of-Word- (BOW), Part-of-Speech- (POS) und des Rhyme-Detection Ansatzes. Die Werte sind in Prozent angegeben und wurden mit einem Cut-off-Index von 800 (C800) erzielt. Andere Bezeichnungen wie in Tabelle 5.4.

Sing365-Korpus	**BOW + POS + Rhyme (w/ SW)**	**BOW + POS + Rhyme (w/o SW)**
SVM	89.149	91.710
N.Bayes	86.573	88.134

Tabelle 5.7: Klassifikationsergebnisse des Sing365-Korpus bei Benutzung des Bag-of-Word- (BOW), Part-of-Speech- (POS) und des Rhyme-Detection Ansatzes. Die Werte sind in Prozent angegeben und wurden mit einem Cut-off-Index von 800 (C800) erzielt. Andere Bezeichnungen wie in Tabelle 5.4.

zweitbesten abschneiden. Bei diesem Vergleich werden nur jene Werte, die in Experimenten mit Stopword-Removal erzielt wurden, in Betracht gezogen, da diese in allen Fällen die besseren Werte aufweisen. Die besten Ergebnisse werden mit der Kombination aller 3 Featuremengen erzielt, wobei jedoch zu erwähnen ist, dass sich dies nur für die SVM-Klassifikationen abzeichnen. Für Klassifikationen, die mit dem Naive Bayes durchgeführt wurden, sind die Resultate, die mit den BOW-Features alleine erzielt wurden, die besten. Bei diesem Klassifikator schneidet der BOW+POS+Rhyme-Ansatz als Zweiter ab. Anhand dieser Werte kann man somit keine allgemein gültige Aussage treffen, welche Features besser geeignet sind. Man kann jedoch festhalten, dass die Tendenz dahin geht, dass die Rhyme-Features für die Songtextklassifikation besser geeignet sind als die POS-Features. Dies kann man der Tabelle 5.3 entnehmen. Wenn man die Ergebnisse in Experiment 3 mit jenen in Experiment 4 vergleicht, so ist klar ersichtlich, dass in allen Fällen mit den Rhyme Features bessere Resultate erzielt werden als mit den POS Features. So ist die Annahme, dass reimende Wörter mehr über die Struktur und den Aufbau von Songtexten aussagen als die POS Eigenschaften eines Textes, durchaus nicht abwegig. Mit den Rhyme-Features können nämlich verschiedene Muster ermittelt werden, wohingegen dies bei den POS-Features nicht der Fall ist.

Die Stärken und Schwächen des Verfahrens in seiner besten Konfiguration (BOW + POS + Rhyme, C800) werden mit Hilfe der Confusion Matrix in der Tabelle 5.8 verdeutlicht. Als äußerst positiv muss festgehalten werden, dass die meisten Musikgenres mit einer sehr hohen Genauigkeit klassifiziert werden. Vier der 6 Genres haben eine Klassifikationsgenauigkeit von über 90 Prozent zu verzeichnen. Diese sind Country (96,5 Prozent), Hip-Hop (91,9 Prozent), R&B (92,5 Prozent) und Rock (95,8 Prozent). Bemerkenswert ist, dass das Musikgenre Country am Besten klassifiziert wird und dass die Ergebnisse des Genres Rock nur knapp hinter diesem fällt. Dies zeigt, dass die Konzepte dieser beiden Genres sehr gut von den anderen unterscheidbar sind. Die anderen beiden Genres, welche eine hohe Genauigkeit aufweisen, sind Hip-Hop und R&B.

Die Genres, bei denen die Rate unter 90 Prozent liegen, sind Pop (84,3 Prozent) und Reggae (81,7 Prozent). Bei Reggae zeigt sich, dass ein paar Verwechslungen mit Pop, Rock und Country auftreten. Die Anzahl falsch klassifizierter Dokumente ist in diesem Fall jedoch viel zu klein, als dass nun nachvollziehbare Rückschlüsse gemacht werden können. Die Ergebnisse beim Genre Pop fallen ebenfalls nicht so gut aus. Dies kann man jedoch durchaus auf das Konzept des Genres zurückführen, da als Popmusik im Allgemeinen Musik angesehen wird, welche in erster Linie als populär gilt. So ist es leicht nachzuvollziehen, dass es hierbei leicht zu einer Durchmischung mit anderen Genres kommen kann.

In Tabelle 5.9 befindet sich eine Liste von Features, welche typischerweise durch das in dieser Arbeit beschriebene Verfahren entsteht. Von allen Songtextdokumenten werden die n durch den χ^2-Test am höchsten eingestuften Features genommen und als Featuremenge für die kompakte Repräsentation eines Dokumentes verwendet. In diesem Fall ist $n = 800$ und die Featuremenge wurde mit dem BOW-,POS- und dem Rhyme-Detection-Ansatz ermittelt. In der Tabelle werden die 10 besten Terme dargestellt, wobei hier besonders auffällig ist, dass vier der insgesamt neun Rhyme-Features es unter die besten 10 geschafft haben. Daran ist zu erkennen, dass die Länge und Struktur der Songtexte in diesem Korpus bei den einzelnen Musikgenres sehr spezifisch sind. Weiters ist besonders gut ersichtlich, dass in der Liste 3 Schimpfwörter vorzufinden sind. Dies ist nachvollziehbar, da derartige Wörter meistens nur in Songtexten mit dem Genre Rap/Hip-Hop vorkommen. Da zeigt sich deutlich, dass der χ^2-Test seinen Zweck erfüllt, da Terme, welche für ein Genre als signifikant angesehen werden, hoch eingestuft werden. Auffällig in der Liste sind ebenfalls die Namen der Künstler. Da gerade Eigennamen sehr spezifisch sind und daher gute Diskriminatoren darstellen ist auch dies nicht sehr verwunderlich.

Ein anderer vielleicht nicht auf den ersten Blick nachvollziehbares Feature ist „nouns“, welcher den POS-Features angehört.

	Country	Hip-Hop	Pop	R&B	Reggae	Rock
Country	196	1	1	0	2	3
Hip-Hop	1	217	5	6	3	4
Pop	0	4	172	13	8	7
R&B	3	3	7	212	3	1
Reggae	8	1	11	4	157	11
Rock	2	1	2	2	2	208

Tabelle 5.8: Confusion Matrix der Klassifikationsergebnisse des Sing365-Korpus mit der Konfiguration: Sing365, SVM, C800, BOW+POS+Rhyme Features. Gesamtscore beträgt 91,710 Prozent. Die Werte stehen für die Anzahl der Songtextdokumente.

Nr	Terme
1.	words (*)
2.	AB (*)
3.	chars (*)
4.	numberofwords (*)
5.	nouns (-)
6.	shit (+)
7.	nigga (+)
8.	marley (+)
9.	parton (+)
10.	fuck (+)

Tabelle 5.9: Die 10 Features mit den höchsten χ^2-Werten. Das Ergebnis wurde mit der Konfiguration: Sing365-Korpus, BOW (+) + POS (-) + Rhyme (*)-Features, C800 erzielt.

5.2.2 Auswertungen der Experimente mit dem Parallelkorpus

Beim Parallelkorpus zeichnet sich ein komplett anderes Bild ab, da hier nun völlig andere Bedingungen gegeben sind. Denn sowohl die Anzahl der Songtextdokumente, als auch die der eingesetzten Musikgenres sind um ein Vielfaches größer als die vom Sing365-Korpus. Dies wirkt sich stark auf die Klassifikationsrate der beiden Klassifikatoren aus. Die maximal erreichte Klassifikationsgenauigkeit ist 65,119 Prozent (Tabelle 5.11: Experiment 4). Die Ergebnisse, welche den beiden folgenden Tabellen (Tabelle 5.10 und Tabelle 5.11) zu entnehmen sind, verzeichnen Werte um die 55 Prozent mit SVM und um die 33 Prozent mit dem Naive Bayes. Es ist ersichtlich, dass hier hinsichtlich der Klassifikationsgenauigkeit ein markanter Unterschied zum Sing365-Korpus besteht.

Wie es bereits beim Sing365-Korpus der Fall war, fielen auch hier die Ergebnisse ohne dem Stemmingverfahren besser aus. Dies wird in der Tabelle 5.10 illustriert. Sowohl beim SVM-Klassifikator, als auch beim Naive Bayes sind höhere Werte vorzufinden. An derselben Tabelle kann man auch gut erkennen, dass bei beiden Klassifikatoren das Weglassen der Stoppwörter ebenfalls zu besseren Ergebnissen führt.

Das Stemming wird im Gegensatz zum Stopword-Removal Ansatz nicht als typischer Preprocessing-Ansatz angesehen. Aus diesem Grund werden die Ergebnisse dieses Verfahrens in allen drei Konfigurationen nochmals analysiert und in der Tabelle 5.11 (Experiment 1 und 2) dargestellt. Darin kann man gut erkennen, dass Experiment 1, in welchem die Ergebnisse ohne Stemmingverfahren verzeichnet sind, in allen Fällen höhere Werte aufweisen als Experiment 2. So bestätigt sich die Annahme, dass man ohne

Stemmingverfahren zu besseren Ergebnissen kommt. Da dies ebenfalls im vorherigen Abschnitt beim Sing365-Korpus festgestellt wurde, kann mit ziemlicher Sicherheit behauptet werden, dass sich das Stemmingverfahren als Vorverarbeitungsschritt für die Experimente nicht eignet. Denn für die in dieser Arbeit durchgeführten Klassifikationsaufgaben scheinen die Wortendungen der einzelnen Terme sehr wichtige Informationen zu beinhalten, mit deren Hilfe die einzelnen Musikgenres spezifiziert werden können.

Parallelkorpus	**BOW (w/ SW, no stemming)**	**BOW (w/o SW, no stemming)**	**BOW (w/ SW, stemming)**	**BOW (w/o SW, stemming)**
SVM	54.071	55.796	53.082	54.667
N.Bayes	31.988	34.081	31.303	32.927

Tabelle 5.10: Klassifikationsergebnisse des Parallelkorpus bei Benutzung des Bag-of-Word-Ansatzes (BOW). Versuche mit/ohne Stopwörter werden mit „w/ SW“ bzw. mit „w/o SW“ bezeichnet. Versuche, bei denen das Verfahren Stemming durchgeführt wurde, werden mit „stemming“ bezeichnet oder andernfalls mit „no stemming“. Die Werte sind in Prozent angegeben und wurden mit einem Cut-off-Index von 800 (C800) erzielt.

Parallelkorpus	**C600**		**C800**		**C1000**	
	SVM	**NBayes**	**SVM**	**NBayes**	**SVM**	**NBayes**
Exp. 1	53.094	33.523	55.796	34.081	**56.113**	32.090
Exp. 2	51.369	31.988	54.667	32.927	**55.124**	31.380
Exp. 3	58.358	28.602	62.265	29.261	**63.952**	30.720
Exp. 4	59.310	28.881	62.874	29.502	**65.119**	30.758
Exp. 5	59.005	29.211	62.899	29.794	**65.030**	31.088

Tabelle 5.11: Klassifikationsergebnisse des Parallelkorpus. Die Konfiguration jedes einzelnen Experiments kann der Tabelle 5.12 entnommen werden. Die Werte sind in Prozent angegeben und wurden mit den Cut-off-Indizes von 600 (C600), 800 (C800) und 1000 (C1000) erzielt. Das beste Ergebnis jedes Experiments wird fett angezeigt.

Ein weiterer Unterschied, welcher beim Parallelkorpus zu beobachten war, ist, dass hier nun die Klassifikationsgenauigkeit mit größerer Anzahl an Features steigt. Beim Sing365-Korpus wurden bessere Ergebnisse mit dem Cut-off-Index C800 erzielt. Dies führte zu der Annahme, dass mit dem Index C800 die optimale Anzahl an Attributen

	Konfiguration
Exp. 1	BOW, w/o SW, no stemming
Exp. 2	BOW, w/o SW, stemming
Exp. 3	BOW+POS, w/o SW, no stemming
Exp. 4	BOW+Rhyme, w/o SW, no stemming
Exp. 5	BOW+POS+Rhyme, w/o SW, no stemming

Tabelle 5.12: Die Konfigurationen der Experimente, die in Tabelle 5.11 durchgeführt wurden. Bezeichnungen wie in Tabelle 5.10.

gefunden wurde, um die verschiedenen Genres zu beschreiben und dass das Hinzufügen und Entfernen weiterer Features nur zur einer Verschlechterung der Ergebnisse führt. Wie man jedoch der Tabelle 5.11 entnehmen kann, tritt dieser Fall hier nicht mehr auf. In der Tabelle können mit größerem Cut-off-Index bessere Ergebnisse festgestellt werden. In Experiment 4 und 5 kann in der Klassifikationsgenauigkeit ein Unterschied von 6 Prozent zwischen C600 und C1000 verzeichnet werden. Somit scheint es so zu sein, dass die ersten 600 und 800 Features nicht ausreichend genug sind, um die 37 Musikgenres in diesem Korpus zu beschreiben. Dies ist leicht nachvollziehbar, da der Parallelkorpus aufgrund seiner Größe von über 7000 Songtextdokumenten dazu eine entsprechend größere Anzahl an Features zu benötigen scheint. Aus diesem Grund wurde Experiment 4, mit dessen Konfiguration das beste Klassifikationsergebnis erzielt wurde, mit größeren Cut-off-Indizes von 1500 und 2000 wiederholt. Die Ergebnisse dieses Experiments kann der Tabelle 5.13 entnommen werden. Es ist deutlich zu erkennen, dass die Klassifikationsgenauigkeit mit einem Cut-off-Index von 2000 noch gesteigert werden kann. Dies zeichnet sich jedoch nur für die SVM-Klassifikationen ab. Beim Naive Bayes scheint die Tendenz dahin zu gehen, dass die Genauigkeit mit größerer Featureanzahl sinkt.

Parallel-korpus	**C1000**		**C1500**		**C2000**	
	SVM	**NBayes**	**SVM**	**NBayes**	**SVM**	**NBayes**
Exp. 4	65.119	30.758	65.969	28.399	**66.704**	27.524

Tabelle 5.13: Klassifikationsergebnisse des Parallelkorpus. Die Konfiguration des Experiments kann der Tabelle 5.12 entnommen werden. Die Werte sind in Prozent angegeben und wurden mit den Cut-off-Indizes von 1000 (C1000), 1500 (C1500) und 2000 (C2000) erzielt. Das beste Ergebnis wird fett angezeigt.

Ebenfalls interessant ist, dass die Ergebnisse des BOW-Ansatzes am Schlechtesten ausgefallen sind (Tabelle 5.11: Experiment 1). Die besten Ergebnisse wurden mit dem BOW+Rhyme-Ansatz erzielt (Tabelle 5.11: Experiment 4), wobei hier eine Verbesserung von ca. 9 Prozent zu beobachten ist (mit dem Cut-off-Index C1000). Eine derart große Steigerung der Klassifikationsrate durch Einsatz von Rhyme-Features ist bemerkenswert und legt nun die Vermutung nahe, dass die verschiedenen Muster der reimenden Wörter viel zur Beschreibung der verschiedenen Genrekonzepte beitragen. Weiters wurde auch bei diesem Korpus festgestellt, dass mit den Rhyme Features bessere Ergebnisse erzielt wurden als mit den POS Features. Dies verdeutlichen in der Tabelle 5.11 die Experimente 3 und 4.

In der Tabelle 5.15 werden die Werte von Precision und Recall, welche typische Kennwerte für die Qualität von Klassifikatoren darstellen, miteinander verglichen. Unter Recall versteht man hierbei den Anteil der korrekt klassifizierten Beispiele an allen klassifizierten Beispielen einer Kategorie (die Vollständigkeit eines Ergebnisses). Der Kennwert Precision gibt den Anteil der korrekten Vorhersagen unter allen zu einer Kategorie als zugehörig vorausgesagten Beispielen wieder (die Genauigkeit eines Ergebnisses). Die beiden Maße sind in gewisser Weise gegenläufig, da im Allgemeinen mit steigendem Recall die Precision und umgekehrt mit steigender Precision der Recall sinkt. Im Übersichtsartikel von Sebastiani [Seb02] und im vom Mitchell verfassten Buch [Mitch97] befindet sich eine genaue Beschreibung dieser beiden Werte.

Die Werte, die man der Tabelle entnehmen kann, sind mit dem SVM-Klassifkator und der Konfiguration: BOW+Rhyme, w/o SW, no stemming, C1000 erzielt worden. Auffallend an den Werten ist, dass die Genres Dancehall und Experimental trotz geringer Trainingsdokumente perfekt klassifiziert worden sind. Beide Genres weisen einen Precision- und Recall-Wert von 1 auf. Ein möglicher Grund dafür ist, dass das Genre Experimental nur durch eine einzige Gruppe mit dem Namen Kaizers Orchestra vertreten wird (Anhang A.2). Ihre Lieder schreiben sie hauptsächlich in norwegisch, da diese Gruppe aus Norwegen stammt. Der Hauptgrund, weshalb die Klassifikation perfekt ausgefallen ist, liegt daher vermutlich in der Sprache des Songtextes. Dies wird in der Tabelle 5.14 deutlich ersichtlich, in der die 10 am höchsten eingestuften Features basierend auf die χ^2-Werte aufgelistet sind. In der Liste ist das Language Feature, in der die Sprache des jeweiligen Songtextes gespeichert wird, an oberster Stelle. Somit kann vermutet werden, dass die Sprachen der Songtexte eines Genres in diesem Korpus sehr spezifisch waren. Weiters fällt ebenfalls auf, dass der Name der Band es unter

die besten 10 Terme/Features geschafft hat. Dies kann damit begründet werden, dass Kaizers Orchestra als einzige Band das Genre Experimental beschreibt. Da der Name dieser Band sehr spezifisch ist, wurden die Terme/Features „Kaizers“ und „Orchestra“ als sehr hoch eingestuft. Bezüglich dem Genre Dance Hall kann festgestellt werden, dass die Worte in den Songtexten sich sehr oft wiederholen. Dieses Genre wird ebenfalls von nur einem Musikkünstler definiert. Hierbei handelt es sich um Sean Paul, einem jamaikanischen Dance-Hall-Interpreten (Anhang A.2). Da dieser seine Texte in englischer Sprache schreibt und der Großteil der Songtexte im Korpus auf Englisch ist, ist das Ergebnis der Klassifikation überraschend. Das gute Ergebnis kann möglicherweise mit der Vermutung begründet werden, dass in jamaikanischen Songtexten, wie in Hip-Hop Texten, die derbe Umgangssprache „Slang“ ihre Verwendung findet. Somit könnten spezifische Terme in Songtexten dieses Genres vorgefunden werden.

Ein anderes Genre, das eine sehr hohe Genauigkeit bei einer großen Anzahl an Trainingsdokumenten aufweist, ist Classic. Dieses hat bei 513 Trainingsdokumenten ein Precisionwert von 0.977 und ein Recallwert von 0.994 aufzuweisen. Auch hier könnte die Vermutung dahin gehen, dass das Language Feature einen großen Beitrag zum guten Ergebnis leistet. Denn besonders die klassischen Songtexte, die von Musikkünstlern wie Verdi, Gustav Mahler, Beethoven und Mozart geschrieben wurden, können in mehreren Sprachen vorgefunden werden.

Bei den Genres Acid Punk, Alternative, Christian Rock, Hard Rock, Rock und anderen Musikstilen desselben Genres treten Verwechslungen in allen Richtungen auf. Der Grund dafür ist, dass diese Genres bzw. Stile einander viel zu ähnlich sind, als dass hundertprozentige Trennungen erwarten werden können. Eine mögliche Lösung dieses Problems wäre der Einsatz von globalen Musikgenres, die als umfassende Kategorie angesehen werden. Hierbei würden die verschiedenen Musikstile desselben Genres zu einer Kategorie zusammengefasst werden, wodurch das Erlernen dieser erheblich erleichtert werden kann. Um zu überprüfen, welcher Unterschied beim Einsatz von globalen Genres festgestellt werden kann, werden die Songtextdokumente dieses Korpus in globale Genres zusammengefasst, woraufhin dieselben Experimente auf diesen durchgeführt werden. Die Ergebnisse dieser Experimente werden im nächsten Abschnitt diskutiert.

Nr	Terme
1.	language (-)
2.	det (+)
3.	kaizers (+)
4.	caddies (+)
5.	orchestra (+)
6.	caturn (+)
7.	words (*)
8.	norah (+)
9.	filter (+)
10.	som (+)

Tabelle 5.14: Die 10 Features mit den höchsten χ^2-Werten. Das Ergebnis wurde mit der Konfiguration: Parallelkorpus, BOW (+) + Rhyme (*)-Features, C1000 erzielt. Das Language Feature (-), welches in allen Featurekombinationen als fixer Bestandteil aufgenommen wurde, steht an oberster Stelle.

5.2.3 Auswertungen der Experimente mit dem Parallelkorpus (Genres 1.Stufe)

Die Songtexte des Parallelkorpus sind in 37 Musikgenres eingeteilt, wobei bei diesen nicht zwischen globalen Genres und Musikstilen unterschieden wird. Dies wirkt sich sehr stark auf die Klassifikationsrate der Klassifikatoren aus, da bei der Genreauswahl darauf Bedacht gelegt wurde, auch ähnliche Genres miteinzubeziehen. Und so kommt es bei der Klassifizierung zu einer Durchmischung ähnlicher Genres.

In diesem Abschnitt soll nun die Klassifikationsrate bei Einsatz von globalen Musikgenres überprüft werden. Dazu werden die verschiedenen Musikstile im Parallelkorpus in gemeinsame Genres zusammengefasst. Die Taxanomie der Musikgenres und Musikstile wurde der Seite www.allmusic.com[7], welche beim Sing365-Korpus im Abschnitt 5.1 bereits zum Einsatz kam, entnommen, nach dessen Zuordnungen die Songtexte dann in die globalen Genres: Rock, Pop, R&B, Rap/Hip-Hop, Blues, Reggae, Electronic, Classic, Country und Avantgarde eingeteilt werden. Eine genaue Auflistung dieser Genres mit deren dazugehörigen Musikstilen befindet sich im Anhang A.3.

Die Ergebnisse, die bei den folgenden Experimenten mit globalen Musikgenres zu beobachten sind, fallen deutlich besser aus, als jene, die im vorherigen Abschnitt zu sehen waren. Die maximal erreichte Klassifikationsgenauigkeit beträgt 83,974 Prozent, wobei diese mit der Konfiguration: Parallelkorpus mit Genres 1.Stufe, SVM, C1000, BOW+POS+Rhyme Features erzielt wird (Tabelle 5.17: Experiment 5). Der BOW+Rhyme-

[7]http://www.allmusic.com

Genre	Anzahl	Precision	Recall
Acid Punk	19	0.636	0.368
Alternative	478	0.887	0.548
Ambient	15	1	0.133
Avantgarde	110	0.984	0.545
Blues	23	0.944	0.739
BritPop	56	0.828	0.421
Christian Rock	38	0.879	0.602
Classic	513	0.977	0.994
Country	134	0.646	0.381
Dance	13	1	0.615
Dance Hall	10	1	1
Electronic	143	0.787	0.259
Emo	254	0.859	0.311
Experimental	10	1	1
Folk	46	0.875	0.457
Garage	41	0.867	0.317
Goth Metal	46	0.944	0.639
Grunge	120	0.958	0.383
Hard Rock	24	0.874	0.732
Hardcore	184	0.938	0.163
Hip-Hop	613	0.839	0.723
Indie	334	0.939	0.509
Industrial	22	0.333	0.045
Metal	572	0.782	0.278
New Metal	98	0.95	0.582
Pop	860	0.849	0.54
Post Punk	26	1	0.346
Punk Rock	1390	0.615	0.872
R&B	254	0.868	0.571
Reggae	49	1	0.082
Rock	715	0.504	0.957
Ska	37	0.962	0.676
Slow Rock	501	0.855	0.437
Soundtrack	27	0.2	0.037
Speech	53	0.95	0.717
Trip-Hop	52	0.8	0.077
World	4	0	0

Tabelle 5.15: Precision & Recall Werte des Parallelkorpus bei Benutzung des Bag-of-Word- (BOW) und Rhyme-Detection Ansatzes. Die Werte sind mit dem SVM-Klassifikator und dem Cut-off-Index von 1000 (C1000) erzielt worden.

Ansatz in Experiment 4 (Tabelle 5.17) fällt mit einem Wert von 83,961 Prozent nur knapp hinter diesem Ergebnis zurück.

Wie bereits in den vorherigen Experimenten festgestellt wurde, kommt man hier ebenfalls ohne Stemmingverfahren und Stoppwörter zu den besten Resultaten. Dies wird in den Tabellen 5.16 und 5.17 (Experiment 1 und 2) verdeutlicht. Allerdings zeichnet sich dies diesmal nur für die SVM-Klassifikationen ab. Interessanterweise treten beim Naive Bayes die besten Ergebnisse bei Experimenten auf, in denen das Stemmingverfahren angewendet und Stoppwörter miteinbezogen werden. In der Tabelle 5.17 wird deutlich gezeigt, dass Experiment 2, in der das Stemmingverfahren angewendet wird, höhere Werte aufweist als Experiment 1. Dies ist überraschend, weil bisher sowohl mit dem SVM-Klassifikator als auch mit dem Naive Bayes bessere Ergebnisse ohne Stemming erzielt wurden. Da jedoch die Werte des Naive Bayes mit durchschnittlich 35 Prozent weit hinter den Werten der SVM mit ca. 80 Prozent fallen, fällt diese Beobachtung nicht besonders schwer ins Gewicht.

1. Stufe				
Parallel-korpus	**BOW (w/ SW, no stem-ming)**	**BOW (w/o SW, no stem-ming**	**BOW (w/ SW, stem-ming**	**BOW (w/o SW, stem-ming**
SVM	81.871	82.897	80.743	80.910
N.Bayes	36.666	32.961	37.628	35.487

Tabelle 5.16: Klassifikationsergebnisse des Parallelkorpus mit Genres 1.Stufe bei Benutzung des Bag-of-Word-Ansatzes (BOW). Versuche mit/ohne Stopwörter werden mit „w/ SW“ bzw. mit „w/o SW“ bezeichnet. Versuche, bei denen das Verfahren Stemming durchgeführt wurde, werden mit „stemming“ bezeichnet oder andernfalls mit „no stemming“. Die Werte sind in Prozent angegeben und wurden mit einem Cut-off-Index von 1000 (C1000) erzielt.

Beim Vergleich aller Konstellationen schneidet der Parallelkorpus mit den Genres 1. Stufe viel besser ab, als der alte. Die Ergebnisse in der Tabelle 5.17 verdeutlichen den markanten Unterschied in der Klassifikationsrate. Ansonsten zeigt sich auch in diesem Korpus das erwartete Bild, dass die Klassifikationsgenauigkeit mit steigender Anzahl an Attributen steigt. So wurde auch bei diesem Parallelkorpus das Experiment, welches das beste Ergebnis lieferte (Experiment 5), mit größeren Cut-off-Indizes wiederholt. Die Ergebnisse in der Tabelle 5.19 verdeutlichen auch hier, dass mit einem Cut-off-Index von

1. Stufe						
Parallel-korpus	**C600**		**C800**		**C1000**	
	SVM	**NBayes**	**SVM**	**NBayes**	**SVM**	**NBayes**
Exp. 1	81.166	30.512	81.500	32.756	**82.897**	32.961
Exp. 2	79.397	32.897	80.371	34.589	**80.910**	35.487
Exp. 3	81.371	30.448	81.730	33.333	**82.602**	33.589
Exp. 4	82.794	30.602	83.153	33.320	**83.961**	33.653
Exp. 5	82.500	30.192	83.128	33.179	**83.974**	33.833

Tabelle 5.17: Klassifikationsergebnisse des Parallelkorpus mit Genres 1. Stufe. Die Konfiguration jedes einzelnen Experiments kann der Tabelle 5.18 entnommen werden. Die Werte sind in Prozent angegeben und wurden mit den Cut-off-Indizes von 600 (C600), 800 (C800) und 1000 (C1000) erzielt. Das beste Ergebnis jedes Experiments wird fett angezeigt.

	Konfiguration
Exp. 1	BOW, w/o SW, no stemming
Exp. 2	BOW, w/o SW, stemming
Exp. 3	BOW+POS, w/o SW, no stemming
Exp. 4	BOW+Rhyme, w/o SW, no stemming
Exp. 5	BOW+POS+Rhyme, w/o SW, no stemming

Tabelle 5.18: Die Konfigurationen der Experimente, die in Tabelle 5.17 durchgeführt wurden. Bezeichnungen wie in Tabelle 5.16.

2000 die Klassifikationsgenauigkeit sowohl mit dem SVM-Klassifikator als auch mit dem Naive Bayes noch gesteigert werden kann.

In der Tabelle 5.20 werden Precision und Recall jedes einzelnen Musikgenres gegenüberstellt. Dabei sind die Werte mit der besten Konfiguration: BOW+POS+Rhyme, w/o SW, no stemming und C1000 erzielt worden. Es ist klar ersichtlich, dass das Genre Classic mit einem Precisionwert von 0,984 und einem Recallwert von 0,988 als Bestes abschneidet. Dass das Konzept dieses Genres sehr gut von anderen unterscheidbar ist, wurde bereits im vorherigen Abschnitt festgestellt. Andere Genres mit einer hohen Genauigkeit sind Rock (98,7 Prozent), Hip-Hop (73,2 Prozent) und Blues (73,9 Prozent). Ebenfalls auffallend sind die Ergebnisse der Musikstile Dancehall und Experimental. Im vorherigen Abschnitt wurden diese mit einer 100 prozentigen Genauigkeit klassifiziert. Da sie nun mit anderen Musikstilen des selben Genres gemischt werden, verlieren sie auch

1. Stufe						
Parallel-korpus	**C1000**		**C1500**		**C2000**	
	SVM	**NBayes**	**SVM**	**NBayes**	**SVM**	**NBayes**
Exp. 5	83.974	33.833	85.166	36.269	**86.000**	37.256

Tabelle 5.19: Klassifikationsergebnisse des Parallelkorpus mit Genres 1. Stufe. Die Konfiguration des Experiments kann der Tabelle 5.18 entnommen werden. Die Werte sind in Prozent angegeben und wurden mit den Cut-off-Indizes von 1000 (C1000), 1500 (C1500) und 2000 (C2000) erzielt. Das beste Ergebnis wird fett angezeigt.

gleichzeitig ihre Spezifität. Das Genre Reggae, welches dem Musikstil Dancehall übergeordnet wird, hat eine Genauigkeit von 16,9 Prozent vorzuweisen. Avantgarde, worin das Musikstil Experimental eingeordnet wird, verzeichnet einen Genauigkeitswert von 62,5 Prozent. Wenn man diese Werte mit den Werten der selben Genres im alten Parallelkorpus vergleicht, kann man eine Verbesserung der Ergebnisse erkennen. Avantgarde hatte im alten Korpus einen Recallwert von 0,545 und Reggae einen Wert von 0,082. Dass die Verbesserung dieser Werte auf das Hinzufügen der beiden Musikstile Dancehall und Experimental im neuen Korpus zurückzuführen ist, versteht sich von selbst.

Genre	**Anzahl**	**Precision**	**Recall**
Avantgarde	120	0.974	0.625
Blues	23	0.944	0.739
Classic	513	0.984	0.988
Country	180	0.828	0.4
Electronic	286	0.79	0.224
Hip-Hop	613	0.874	0.732
Pop	860	0.938	0.441
Reggae	59	0.667	0.169
Rock	4892	0.815	0.987
R&B	254	0.879	0.602

Tabelle 5.20: Precision & Recall des Parallelkorpus (Genres 1.Stufe)

In der Tabelle 5.21 wird anhand der Confusion Matrix, welche ebenfalls mit der oben angeführten Konfiguration gebildet wurde, die Verteilung falsch klassifizierter Songtextdokumente ersichtlich. Auffällig an den Werten der Matrix ist, dass bei allen Musikgenres Verwechslungen mit dem Genre Rock auftreten. Dies ist intuitiv nachvollziehbar, da in

die Rockmusik viele Elemente der meisten Musikrichtungen, darunter sogar auch Blues und Reggae, einfliessen. Natürlich hat die sehr große Anzahl an Trainingsdokumenten dieses Genres ebenfalls Einfluss auf dieses Ergebnis. Da von den Musikgenres im alten Parallelkorpus zahlreiche Rockstile vertreten sind, führte das Zusammenfassen dieser zu einem Ergebnis von 4892 Dokumenten im Genre Rock. Beim Genre Classic kann die hohe Genauigkeit gut abgelesen werden. Von 513 Songtextdokumenten werden nur 6 fälschlicherweise als Rock klassifziert.

Die meisten Probleme gibt es bei den Genres Country, Electronic, Pop und Reggae. Bei den Genres Reggae und Country können die schwachen Genauigkeitswerte, wie oben schon erwähnt, damit erklärt werden, dass das Genre Rock durch die große Dokumentenanzahl einen starken Einfluss auf das Ergebnis ausübt. Es liegt aber auch durchaus im Bereich des Möglichen, dass bei beiden Genres im Text Ähnlichkeiten mit dem Genre Rock existieren, wodurch es natürlich zu Verwechslungen kommen kann. Diese Vermutung kann ebenfalls beim Genre Electronic aufgestellt werden. Mit Electronic wird Musik bezeichnet, welche mit Hilfe von elektronischen Geräten erzeugt wurde. Auf diese Eigenschaft kann bei der Songtextklassifizierung jedoch nicht zurückgegriffen werden, wodurch sowohl im Text als auch bei der Klassifizierung Verwechslungen mit dem Genre Rock nicht unwahrscheinlich sind.

Das Genre Pop stellt sich mit 44,1 Prozent Genauigkeit ebenfalls als Problemfall dar. Dass bei diesem Genre es leicht zu einer Durchmischung mit anderen Genres kommen kann, wurde bereits bei der Evaluierung der Ergebnisse mit dem Sing365-Korpus festgestellt. Da mit Popmusik im Allgemeinen „populäre“ Musik gemeint ist, wird eigentlich durch Musikcharts und Radiostationen festgelegt, welche Musik in diesem Genre fallen. Somit kann es durchaus vorkommen, dass auch Lieder, welche einem anderen Genre angehören, als Popmusik bezeichnet werden.

Abschliessend werden die Ergebnisse aller drei Korpora, die in dieser Arbeit zum Einsatz kamen, gegenübergestellt. In der Abbildung 5.1 sind die Unterschiede in der Klassifikationsgenauigkeit aller drei Korpora klar ersichtlich. Die Werte wurden mit dem BOW+POS+Rhyme-Ansatz erzielt. Deutlich zeigt sich, dass mit dem Sing365-Korpus die besten Ergebnisse erreicht wurden. Da dieser Korpus jedoch im Vergleich zum Parallelkorpus sehr klein ist, hat der Vergleich seiner Ergebnisse mit denen des Parallelkorpus wenig Sinn. Wenn man jedoch die Ergebnisse des alten und neuen Parallelkorpus betrachtet, kann ein Unterschied von ca. 20 Prozent beim Cut-off-Index 600 und 15 Prozent beim Cut-off-Index 1000 festgestellt werden. Dies zeigt, was für eine große Rolle die Aus-

wahl der Musikgenres bei der Songtextklassifikation spielt. Im Anhang A.4 befinden sich die restlichen Abbildungen, in denen die Ergebnisse mit den restlichen Konfigurationen aller drei Korpora gegenübergestellt werden.

	Avantgarde	Blues	Classic	Country	Electronic	Hip-Hop	Pop	Reggae	Rock	R&B
Avantgarde	75	0	1	1	0	1	0	0	42	0
Blues	0	17	0	0	0	0	0	0	6	0
Classic	0	0	507	0	0	0	0	0	6	0
Country	0	1	1	72	1	5	2	0	98	0
Electronic	0	0	0	0	64	6	0	0	215	1
Hip-Hop	1	0	0	6	2	449	2	0	144	9
Pop	0	0	1	2	6	11	379	0	454	7
Reggae	0	0	0	0	4	1	1	10	43	0
Rock	1	0	5	6	6	27	12	3	4828	4
R&B	0	0	0	0	0	14	8	0	79	153

Tabelle 5.21: Confusion Matrix der Klassifikationsergebnisse des Parallelkorpus mit der Konfiguration: Parallelkorpus, SVM, C1000, BOW+POS+Rhyme Features. Gesamtscore beträgt 83.974 Prozent. Die Werte stehen für die Anzahl der Songtextdokumente.

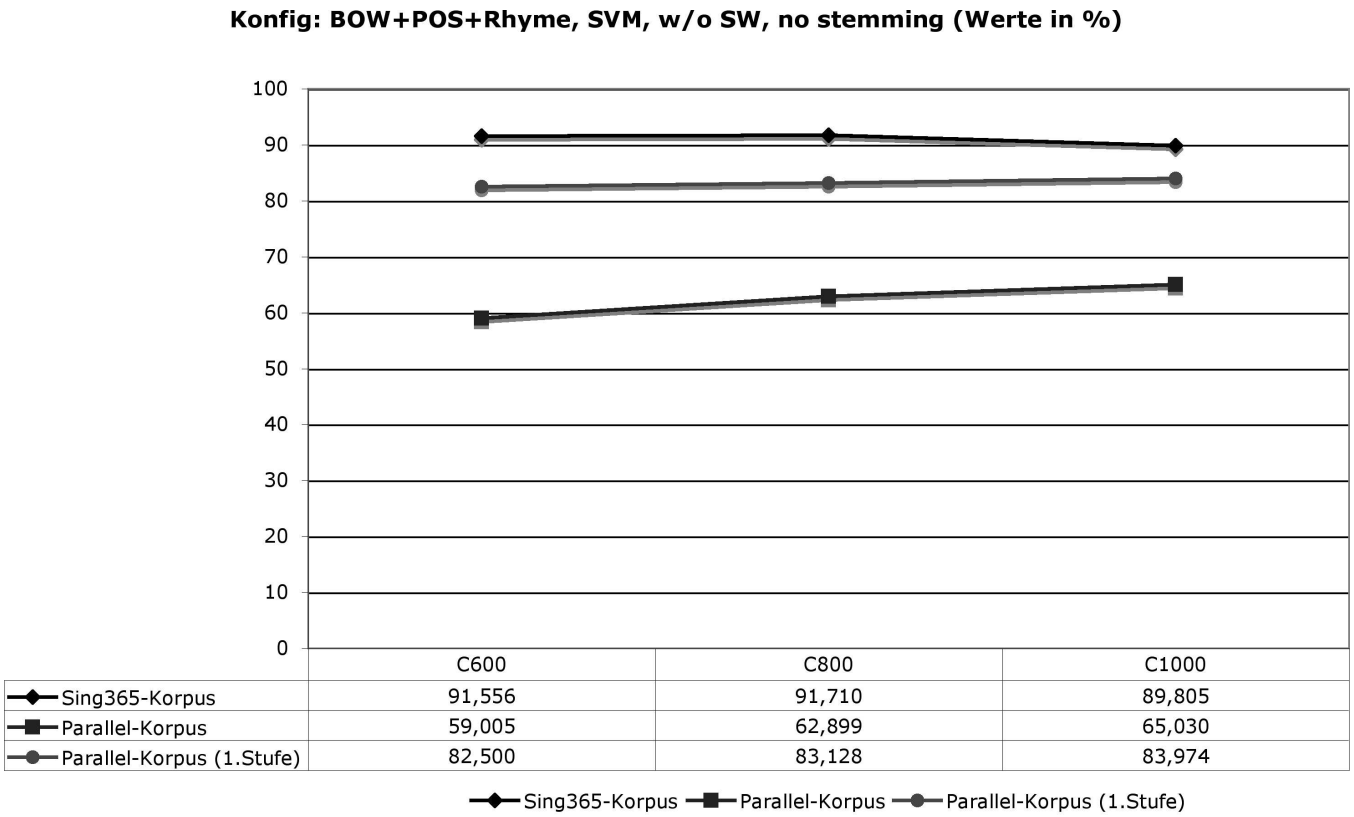


Abbildung 5.1: Klassifikationsergebnisse des Sing-365-, Parallel- und Parallelkorpus mit Genres 1.Stufe mit der Konfiguration: BOW+POS+Rhyme Features, SVM, w/o SW, no stemming. Die Werte sind in Prozent angegeben und wurden mit den Cut-off-Indizes von 600 (C600), 800 (C800) und 1000 (C1000) erzielt.

6 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Ansatz zur automatischen Klassifikation von Musik in Form von Songtexten vorgestellt. Durch mehrere Experimente mit den Kombinationen der verschiedenen Features konnten folgende Erkenntnisse gewonnen werden. Mit einer 6 Musikgenres umfassenden Taxonomie und einem aus 1281 Dokumenten bestehenden Korpus kann eine Klassifikationsgenauigkeit von 91 Prozent erzielt werden. Bei einem Korpus mit einer Größe von 7884 Songtextdokumenten und 37 Musikgenres kann eine Genauigkeit von 65 Prozent erreicht werden. Dabei spielt die Auswahl der Musikgenres eine große Rolle. Es hat sich herausgestellt, dass mit dem Einsatz von globalen Musikgenres die besten Resultate erzielt werden. Die Verwendung von Musikstilen, welche feiner definierte Strömungen innerhalb eines Musikgenres darstellen, führt bei der Klassifikation leichter zu einer Durchmischung mit anderen Genres. So wurden bei einem Korpus mit einer Größe von 7884 Songtextdokumenten und 37 Musikgenres die Musikstile in 10 globale Musikgenres zusammengefasst, woraufhin eine Klassifikationsgenauigkeit von 83 Prozent erreicht werden konnte.

Für die Songtextklassifikation kamen mehrere Features (Bag-of-Words, Part-of-Speech, Text-Statistic) zum Einsatz. Als eigener Beitrag wurde in dieser Arbeit der Einsatz von Language und Rhyme Features eingeführt. Im Language Feature wird die Sprache des Songtextes gespeichert. Mit den Rhyme Features können verschiedene Reimmuster im Text ermittelt werden. In einem Test wurde gezeigt, dass sich Phoneme zur Bestimmung dieser Reimmuster am Besten eignen. Anhand von Ergebnissen konnte ebenfalls gezeigt werden, dass beim Vergleich der beiden strukturbasierten Features POS und Rhyme mit den Rhyme Features in allen Fällen bessere Resultate erzielt werden. Für eine Überraschung sorgte das Language Feature. In den Experimenten wurde gezeigt, dass Musikgenres mit nur 10 Trainingsdokumenten perfekt klassifziert werden konnten. Zu diesem guten Ergebnis trug das Language Feature eine Menge bei. Dies wurde in der χ^2-Liste der 10 am höchsten eingestuften Features verdeutlicht, worauf das Language Feature an oberster Stelle stand.

Weiters lässt sich anhand der Ergebnisse argumentieren, dass Support Vektor Maschinen für die in dieser Arbeit durchgeführten Klassifikationsaufgaben besser geeignet sind, als der Naive Bayes Klassifikator.

In der Vorverarbeitungsphase hat sich das Entfernen von Stoppwörtern als typischer Vorverarbeitungsschritt bewährt, da dadurch die Klassifikationsgenauigkeit in den meisten Fällen gesteigert werden konnte. Überraschend war die Erkenntnis, dass das Stemmingverfahren, mit welchem man normalerweise zu besseren Ergebnissen kommt, die Werte der einzelnen Klassifikationsversuche verschlechterte. Es hat sich dabei herausgestellt, dass die Endsilben der Terme wichtige Informationen beinhalten, welche viel zur Spezifität der einzelnen Musikgenres beitragen.

Nichts desto trotz bringt der Einsatz von Songtexten, welche den Musikportalen im Internet entnommen werden, auch Beschränkungen und Nachteile mit sich. Zum Ersten können die Songtexte falsch formatiert sein. Bei der Rhyme-Detection kann es zu falschen Ergebnissen kommen, wenn die Zeilen in ein Dokument nicht mit jenen Worten enden, wie sie im Musikstück zu hören sind. Ein weiterer Nachteil ist, dass bei der Genrezuordnung des Songtextes das Aufsuchen des Musikgenres zu sehr von den zugrunde liegenden Suchmaschinen und Musikportalen abhängt. Mit der Qualität der Suchmaschinen bzw. der Musikportale steht und fällt somit auch die Präzision des in dieser Arbeit vorgestellten Verfahrens. Das falsche Musikgenre eines Songtextes ist natürlich nicht entscheidend für die gesamte Klassifikation, jedoch sollten Möglichkeiten gefunden werden dieses Problem zu beseitigen.

Ein Ansatz zur Lösung falsch formatierter Songtextdokumente wäre das Aufsuchen von Lyrics mittels „Multiple Sequence Alignment“. Knees, Schedl und Widmer wenden dieses Verfahren an, um bei der Songtextsuche im Internet das Auffinden von fehlerhaften Songtexten zu vermeiden [Knees05]. Dies wird durch den Vergleich mehrerer Lyricwebseiten, welche alle den selben Songtext beinhalten, erreicht. Dabei werden die verschiedenen Lyricwebseiten, welche über Google dem Internet entnommen und gesammelt wurden, aneinander gereiht und in diesen nach übereinstimmenden Wortsequenzen, bei denen es sich mit größter Wahrscheinlichkeit um den Songtext handelt, gesucht. Auf die Art kann zum Beispiel unter all den Songtextformatierungen jene gewählt werden, welche am meisten vorgefunden wurde.

Eine Möglichkeit die Songtextklassifikation in Zukunft zu verbessern ist die Zuhilfenahme von webbasierten Metadaten. Das Wissen über die einzelnen Musikkünstler kann dazu genutzt werden, die Musikgenres der Songtexte, welche eben von diesen Künstlern

geschrieben wurden, besser zu definieren. Weiters kann der in dieser Arbeit vorgestellte Language Feature-Ansatz erweitert werden. Durch die Identifizierung der Sprache eines Songtextes könnte für jede Songtextsprache ein eigener Klassifikator erstellt werden. Andere Perspektiven für zukünftige Verfahren beinhalten die Einbeziehung zusätzlicher Informationen von Google und die Kombination mit Audio-basierten Ansätzen.

Literaturverzeichnis

[Auc04] J. Aucouturier and F. Pachet. „Improving timbre similarity: How high is the sky?“, Journal of Negative Research Results in Speech and Audio Sciences, 1 (1), 2004.

[Bau02] S. Baumann, A. Klüter. „Super-Convenience for Non-Musicians: Querying mp3 and the Semantic Web.“ In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR'02)*, Paris, France, 2002.

[Bau03] S. Baumann. „Music similarity analysis in a p2p environment.“ In *Proceedings of the 4th European Workshop on Image Analysis for Multimedia Interactive Services*, London, UK, April 2003.

[Ber03] A. Berenzweig, B. Logan, D.P.W. Ellis, and B. Whitman. „A large-scale evaluation of acoustic and subjective music similarity measures.“ In *Proceedings of the Fourth International Conference on Music Information Retrieval (ISMIR'03)*, Baltimore, 2003.

[Bro02] E. Brochu and N. de Freitas. „Name that song!: A probabilistic approach to querying on music and text.“ *In NIPS: Neural Information Processing Systems*, 2002.

[Burges96] C.J.C. Burges. *Simplified support vector decision rules.*, 1996.

[Bur03] J.J Burred and A. Lerch. „A hierarchical approach to automatic musical genre classification.“ In *Proceedings of the 6th International Conference on Digital Audio Effects*, London, UK, September 2003.

[Can94] W.B. Canvar and J.M. Trenkle. „N-Gram-Based Text Categorization“ In *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval*, p. 161-175, Las Vegas, NV, April 1994.

[Cook02] P. Cook and G. Tzanetakis. „Musical Genre Classification of Audio Signals.“ In *IEEE Transaction on Speech and Audio Processing*, Vol.10, No. 5, July 2002.

[Deb03] F. Debole and F. Sebastiani. „Supervised term weighting for automated text categorization.“ In *Proceedings of the 18th ACM Symposium on Applied Computing*, p. 784-788, Melbourne (USA), 2003.

[Dunn93] T. E. Dunning. „Accurate methods for the statistics of surprise and coincidence.“ In *Computational Linguistics*, volume 19:1, p. 61-74, 1993.

[Foote97] J. T. Foote. „Content-based retrieval of music and audio. In Multimedia Storage and Archiving Systems II.“ In *Proceedings of SPIE*, p. 138-147, 1997.

[Hay90] P. Hayes and S. Weinstein. „Construct this: a system for content-based indexing of a database of news stories.“ In *Annual Conference on Innovative Applications of AI*, 1990.

[Joa97] T. Joachims, D. Freitag and T. Mitchell. „Webwatcher: A tour guide for the world wide web.“ In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1997.

[Joa98] T. Joachims. „Text categorization with support vector machines: learning with many relevant features.“ In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, p.137-142, Chemnitz, Germany, 1998.

[Knees05] P. Knees, M. Schedl and G. Widmer. „Multiple lyrics alignment: Automatic Retrieval of Song lyrics.“ In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, p. 564-569, London, UK, 11-15 September, 2005.

[Kulesh03] V. Kulesh, I. Sethi and P. V. „Indexing and retrieval of music via Gaussian mixture models.“ In *Proceedings of the 3rd International Workshop on Content-Based Multimedia Indexing (CBMI)*, Rennes (France), 2003.

[Lang95] K. Lang. „Newsweeder: Learning to filter netnews.“ In *International Conference on Machine Learning (ICML'95)*, 1995.

[Lewis98] D. D. Lewis. „Naive (Bayes) at forty: The independence assumption in information retrieval.“ In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, p.4-15, Chemnitz, Germany, 1998.

[Logan04] B. Logan, A. Kositsky and P. Moreno. Semantic Analysis of Song Lyrics. Technical Report HPL-2004-66, HP Laboratories Cambridge, 2004.

[Mahed05] J. P. G. Mahedero, A. Martinez and P. Cano. „Natural language processing of lyrics.“ In *Proceedings of the 13th annual ACM international conference on Multimedia*, p. 475 - 478, Singapore, November 2005.

[Mitch97] Tom M. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

[Pachet00] F. Pachet and D. Cazaly. „A taxonomy of musical genres.“ In *Proceedings of RIAO 2000 Content-Based Multimedia Information Francois PachetAccess*, Paris, France, 2000.

[Port98] M.F. Porter. „An algorithm of suffix stripping.“ *Program*, 14(3), p. 130-137, 1998.

[Ril95] E. Riloff. „Little words can make a big difference for text classification.“ *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, p. 130-136, Seattle, US, 1995.

[San04] M. Santini, „State-of-the-Art on Automatic Genre Identification“, January 2004.

[Scott98] S. Scott, S. Matwin. „Text Classification using WordNet Hypernyms.“ In *Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference*, p. 38-44, New Jersey, 1998.

[Seb02] F. Sebastiani. „Machine learning in automated text categorization.“ ACM Computing surveys, 34(1) p. 1-4, Italy, March 2002.

[Vap95] V. Vapnik. *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.

A Anhang

A.1 Liste der Stoppwörter

- **http://thomas.loc.gov/home/stopwords.html:** a about above across adj after afterwards again against albeit all almost alone along already also although always among amongst an and another any anyhow anyone anything anywhere are around as at be became because become becomes becoming been before beforehand behind being below beside besides between beyond both but by can cannot co could down during each eg either else elsewhere enough etc even ever every everyone everything everywhere except few first for former formerly from further had has have he hence her here hereafter hereby herein hereupon hers herself him himself his how however ie if in inc indeed into is it its itself last latter latterly least less ltd many may me meanwhile might more moreover most mostly much must my myself namely neither never nevertheless next no nobody none noone nor not nothing now nowhere of off often on once one only onto or other others otherwise our ours ourselves out over own per perhaps rather same seem seemed seeming seems several she should since so some somehow someone something sometime sometimes somewhere still such than that the their them themselves then thence there thereafter thereby therefor therein thereupon these they this those though through throughout thru thus to together too toward towards under until up upon us very via was we well were what whatever whatsoever when whence whenever whensoever where whereafter whereas whereat whereby wherefrom wherein whereinto whereof whereon whereto whereunto whereupon wherever wherewith whether which whichever whichsoever while whilst whither who whoever whole whom whomever whomsoever whose whosoever why will with within without would xsubj xcal xauthor xother xnote yet you your yours yourself yourselves;
- **TeSeT :** a a's able about above according accordingly across actually after afterwards again against ain't all allow allows almost alone along already also although

always am among amongst an and another any anybody anyhow anyone anything anyway anyways anywhere apart appear appreciate appropriate are aren't around as aside ask asking associated at available away awfully b be became because become becomes becoming been before beforehand behind being believe below beside besides best better between beyond both brief but by c c'mon c's came can can't cannot cant cause causes certain certainly changes clearly co com come comes concerning consequently consider considering contain containing contains corresponding could couldn't course currently d definitely described despite did didn't different do does doesn't doing don't done down downwards during e each edu eg eight either else elsewhere enough entirely especially et etc even ever every everybody everyone everything everywhere ex exactly example except f far few fifth first five followed following follows for former formerly forth four from further furthermore g get gets getting given gives go goes going gone got gotten greetings h had hadn't happens hardly has hasn't have haven't having he he's hello help hence her here here's hereafter hereby herein hereupon hers herself hi him himself his hither hopefully how howbeit however I i'd i'll i'm i've ie if ignored immediate in inasmuch inc indeed indicate indicated indicates inner insofar instead into inward is isn't it it'd it'll it's its itself j just k keep keeps kept know knows known l last lately later latter latterly least less lest let let's like liked likely little look looking looks ltd m mainly many may maybe me mean meanwhile merely might more moreover most mostly much must my myself n name namely nd near nearly necessary need needs neither never nevertheless new next nine no nobody non none noone nor normally not nothing novel now nowhere o obviously of off often oh ok okay old on once one ones only onto or other others otherwise ought our ours ourselves out outside over overall own p particular particularly per perhaps placed please plus possible presumably probably provides q que quite qv r rather rd re really reasonably regarding regardless regards relatively respectively right s said same saw say saying says second secondly see seeing seem seemed seeming seems seen self selves sensible sent serious seriously seven several shall she should shouldn't since six so some somebody somehow someone something sometime sometimes somewhat somewhere soon sorry specified specify specifying still sub such sup sure t t's take taken tell tends th than thank thanks thanx that that's the their theirs them themselves then thence there there's thereafter thereby therefore therein theres thereupon these they they'd they'll they're they've think third this thorough

thoroughly those though three through throughout thru thus to together too took toward towards tried tries truly try trying twice two u un under unfortunately unless unlikely until unto up upon us use used useful uses using usually uucp v value various very via viz vs w want wants was wasn't way we we'd we'll we're we've welcome well went were weren't what what's whatever when whence whenever where where's whereafter whereas whereby wherein whereupon wherever whether which while whither who who's whoever whole whom whose why will willing wish with within without won't wonder would would wouldn't x y yes yet you you'd you'll you're you've your yours yourself yourselves z zero

A.2 Liste der Künstler

A.2.1 Sing365-Korpus

Country: Dixie Chicks, Dolly Parton, Faith Hill, Johnny Cash, Shania Twain
Hip-Hop: 2Pac, 50 Cent, Black Eyed Peas, Eminem, LL Cool J, Snoop Dogg
Pop: Britney Spears, Christina Aguilera, Janet Jackson, Jennifer Lopez, Justin Timberlake, Madonna, Michael Jackson, Robbie Williams, Shakira
R&B: Alicia Keys, Ashanti, Babyface, Beyonce Knowles, Boyz II Men, Marvin Gaye, Mary J. Blidge, Usher
Reggae: Bob Marley, Jimmy Cliff, Sean Paul, Shaggy, UB40, Ziggy Marley
Rock: 3 Doors Down, A Perfect Circle, Alanis Morissette, Audioslave, AvrilLavigne, Beck, Billy Talent, Bright Eyes, Bruce Springsteen

A.2.2 Parallelkorpus

Acid Punk: Blood Brothers, Melt Banana, The Locust
Alternative: Audioslave, Billy Corgan, Bush, Faith No More, Hard-Fi, Keane, Maroon 5, Muse, Our Lady Peace, Tool, 3 Doors Down, Arctic Monkeys, At the Drive-In, Breaking Benjamin, Coheed and Cambria, Deftones, Finger Eleven, Foo Fighters, Incubus, Porcupine Tree, Sparta, The Goo Goo Dolls, Transplants
Ambient: Enya, Sigur Rs, The Album Leaf

Avantgarde: Mclusky, Mogwai, Mr. Bungle, The Mars Volta, The Rapture, Tomahawk, Yeah Yeah Yeahs
Blues: Norah Jones
BritPop: Oasis, Richard Ashcroft, Suede, The Verve
Christian Rock: Switchfoot
Classic: Beethoven, Franz Schubert, Gustav Mahler, Wolfgang Amadeus Mozart, Richard Strauss, Verdi
Country: Johnny Cash, Kid Rock, Leanne Rimes, Wilco
Dance: Faithless
Dance Hall: Sean Paul
Electronic: Bent, Boards of Canada, IAMX, Lamb, Massive Attack, Mnemonic, Moby, Nine Inch Nails, The Crystal Method
Emo: Autopilot Off, Boysetsfire, Brand new, Circa Survive, Dashboard Confessional, Fall Out Boy, Glassjaw, Matchbook Romance, Story of the Year, Taking Back Sunday, The Used, Thrice, Thursday
Experimental: Kaizers Orchestra
Folk: Adam Green, Dave Matthews Band, The Pogues
Garage: The Streets
Goth Metal: Rob Zombie, White Zombie
Grunge: Nirvana, Pearl Jam, Soundgarden
Hard Rock: A Perfect Circle
Hardcore: Against Me, Alexisonfire, Atreyu, Avenged Sevenfold, Bleeding Through, From Autumn to Ashes, Grade, Ill Repute, Poisen the Well, Red Tape, Shadows Fall, Sick of it all, Silverstein, The dillinger Escape Plan, The Distillers, The Suicide Machines
Hip Hop: 2Pac, 8 Ball & Mjg, 50 Cent, Absolute Beginner, Anti-Pop Consortium, Atmosphere, Beastie Boys, Beginner, Black Eyed Peas, Boyz n da Hood, Camron, Cassidy, D12, Die Fantastischen Vier, Eminem, Everlast, Fugees, Ghostface Killah, House of Pain and Everlast, Jadakiss, Jay-Z, Jedi Mind Tricks, Kanye West, Lauryn Hill, Ludacris, Lumidee, Masta Killa, Max Herre, Method Man, Missy Elliott, Mobb Deep, Nerd, Obie Trice, Royce Da 59, RZA, Slum Village, Snoop Dogg, The Beastie Boys, Ugly Duckling, X-Ecutioners, Xzibit
Indie: Ambulance LTD, And you will know us by the Trail of Dead, Architecture in Helsinki, Bright Eyes, Calexico, Cardia, Death Cab For Cutie, Donnas, Echo & The Bunnymen, Good Life, Graham Coxon, Grandaddy, Head Automatica, Hood, Pave-

ment, Snapcase, Team Sleep, The Appleseed Cast, The Beta Band, The Kooks, The Weakerthans, Tomte

Industrial: Filter

Metal: Planet Earth, Crazyfists, Apocalyptica, Blindside, Chimaira, Coal Chamber, Converge, Earshot, Fear Factory, Finche, Godsmack, Helmet, Ill Nino, Killswitch Engage, Kittie, Korn, M.O.D., Marilyn Manson, Methods of Mayhem, Most Precious Blood, Mudvayne, Murderdolls, Probot, Rammstein, Seether, Slipknot, Soulfly, Spineshank, Static-X, Stone Sour, System of A Down, Therapy, Wolfmother

New Metal: Adema, Hundred Reasons, Limp Bizkit, Papa Roach, TRUSTCompany

Pop: Anastacia, Ashlee Simpson, Atomic Kitten, Avril Lavigne, Britney Spears, Busted, Charlotte Hatherley, Christina Aquilera, Corrs, David Hasselhoff, Die Happy, Geri Halliwell, Gorillaz, Hilary Duff, James, Jeanette, Jez & Superhandz, Lene Marlin, Lindsay Lohan, Liz Phair, Madonna, Madsen, Michael Jackson, Natalie Imbruglia, Nelly Furtado, Novastar, Pink, Robbie Williams, Silbermond, Sportfreunde Stiller, Sugababes, TATU, Texas, The Beatles, The Cardigans, The Cranberries, The Flaming Lips, Tori Amos, Vanilla Ninja, Zornik, Zwan

Post Punk: Interpol, She wants Revenge

Punk Rock: 3 Feet Smaller, 28 Days, 1208, A, AFI, Alkaline Trio, All American Rejects, Andrew WK, Angels and Airwaves, Anti-Flag, Antimaniax, Authority Zero, Bad Religion, Beatsteaks, Belvedere, Billy Talent, Blink 182, Bouncing Souls, Bowling for Soup, Boyhitscar, Descendents, Dirtbags, Division of Laura Lee, Dropkick Murphys, Farin Urlaub, Flogging Molly, Frank Black, Goldfinger, Good Charlotte, Good Riddance, Green Day, Guttermouth, Heiderroosjes, Hot Water Music, Lagwagon, Less than Jake, Me First and the Gimme Gimmes, Midtown, Millencolin, Motion City, Mxpx, My Chemical Romance, New Found Glory, No use for a Name, NOFX, Pennywise, Pulley, Rancid, Rise Against, Satanic Surfers, Saves the Day, Simple Plan, Strike Anywhere, Strung Out, Subhumans, Sum 41, Ten Foot Pole, The Alkaline Trio, The Damned, The Epoxies, The Lawrence Arms, The Offspring, The Pogues, The Soviettes, The Vandals, Toy Dolls, Tsunami Bomb, Wizo, Yellowcard, Zebrahead

R&B: Alicia Keys, Ashanti, Beyonce Knowles, Christina Milian, Ciara, Destinys Child, Dru Hill, Jamelia, Kelis, Kiley Dean, Mary J Blidge, Natasha Bedingfield, Nelly Furtado, TLC, Usher, Vera

Reggae: Benjie, Damian „Jr. Gong“ Marley, Sam Ragga Bandd, UB40

Rock: A Change of Pace, Alanis Morissette, Beck, Black Rebel Motorcycle Club, Bloc

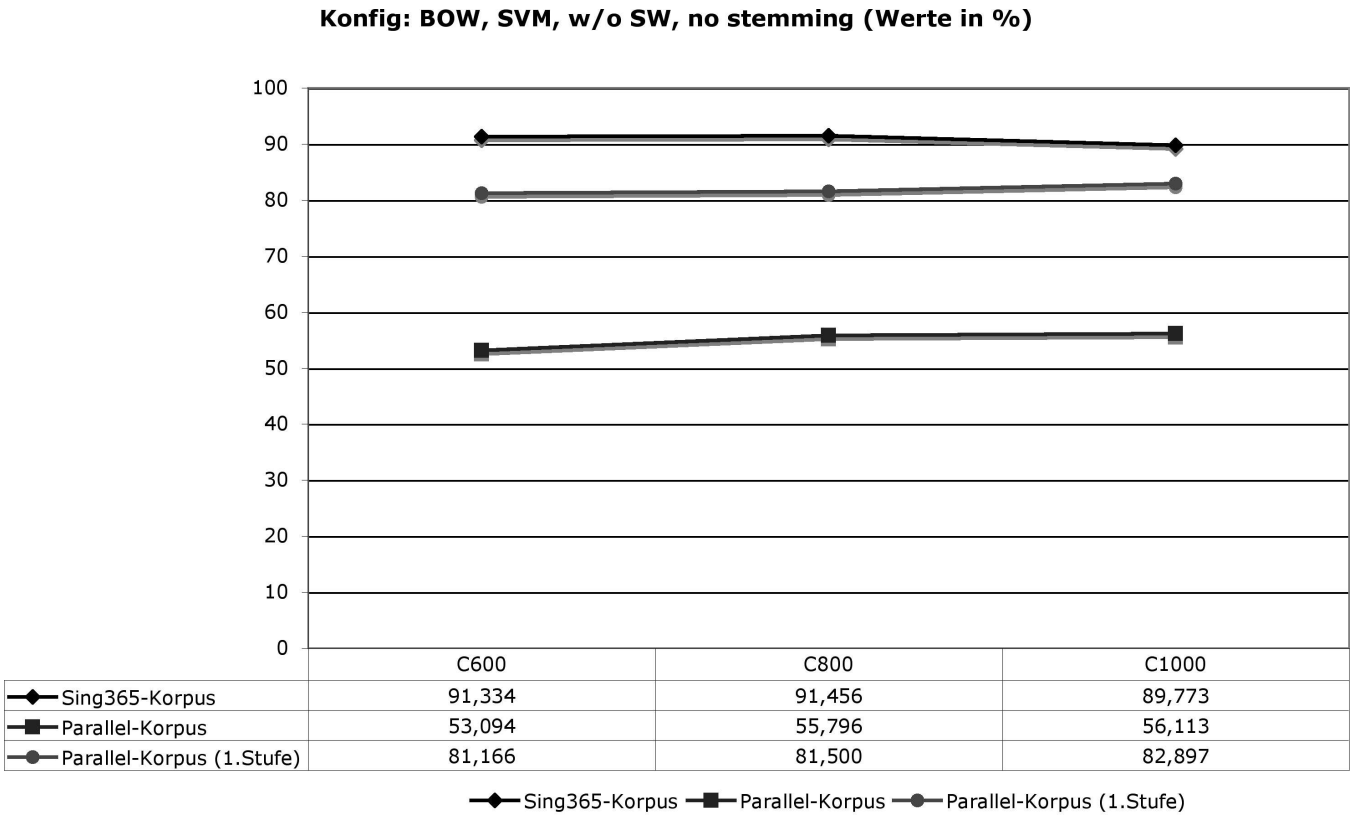
Party, Caesars, Courtney Love, Creed, Danko Jones, Embrace, Feeder, Fugazi, Hole, Hoobastank, Joe Strummer, Killers, Libertines, Mando Diao, Manic Street Preachers, Melissa auf der Maur, Modest Mouse, Mundy, Nada Surf, Phantom Planet, Placebo, Queens of the Stone Age, Ryan Adams, Secret Machines, Seven Mary Three, Taproot, The Czars, The Juliana Theory, The Strokes, The Subways, The White Stripes, Theory of a deadman, They Might be Giants, Three Days Grace, Tommy Lee
Ska: Mad Caddies
Slow Rock: Aqualung, Athlete, Badly Drawn Boy, Doves, Dredg, Elbow, Garbage, Heather Nova, James Blunt, Kasabian, Kashmir, Kate Bush, Krezip, Lambchop, Mercury Rev, Morcheeba, Radiohead, Rilo Kiley, Saybia, Snow Patrol, The Crash, The Cure, The Raveonettes, The Smashing Pumpkins, Travis, Turin Brakes
Speech: Alfred Brehm, Frank Kafka, Friedrich Schiller, Georg Trakl, Giovanni Boccacio, Hans Christian Andersen, Johann Wolfgang von Goethe, Joseph Freiherr von Eichendorf, Klabund, Leopold von Sacher, Rainer Maria Rilke, The Brothers Grimm, Theodor Fontane, Tori Amos, Wilhelm Hauff, William Shakespeare
Trip Hop: Goldfrapp, Moloko, Sneaker Pimps
World: Myslovitz

A.3 Taxanomie der Musikgenres vom Parallelkorpus

1. Rock:
    - Christian Rock
    - Alternative
        - Grunge
        - BritPop
        - Indie
        - Ska
        - Emo
    - Hard Rock

        - Heavy Metal
        - Goth Metal
        - New Metal
    - Punk Rock
        - Acid Punk
        - Hardcore
        - Post Punk
    - Slow Rock
2. Pop
3. R&B
4. Rap/Hip-Hop
5. Blues
6. Reggae
    - Dancehall
7. Electronic:
    - Dance
        - Garage
    - Ambient
    - Trip-Hop
    - Industrial
8. Classic
9. Country:
    - Folk
10. Avantgarde:
    - Experimental

A.4 Ergebnisse aller 3 Korpora



	C600	C800	C1000
Sing365-Korpus	91,334	91,456	89,773
Parallel-Korpus	53,094	55,796	56,113
Parallel-Korpus (1.Stufe)	81,166	81,500	82,897

Abbildung A.1: Klassifikationsergebnisse des Sing-365-, Parallel- und Parallelkorpus mit Genres 1.Stufe mit der Konfiguration: BOW Features, SVM, w/o SW, no stemming. Die Werte sind in Prozent angegeben und wurden mit den Cut-off-Indizes von 600 (C600), 800 (C800) und 1000 (C1000) erzielt.

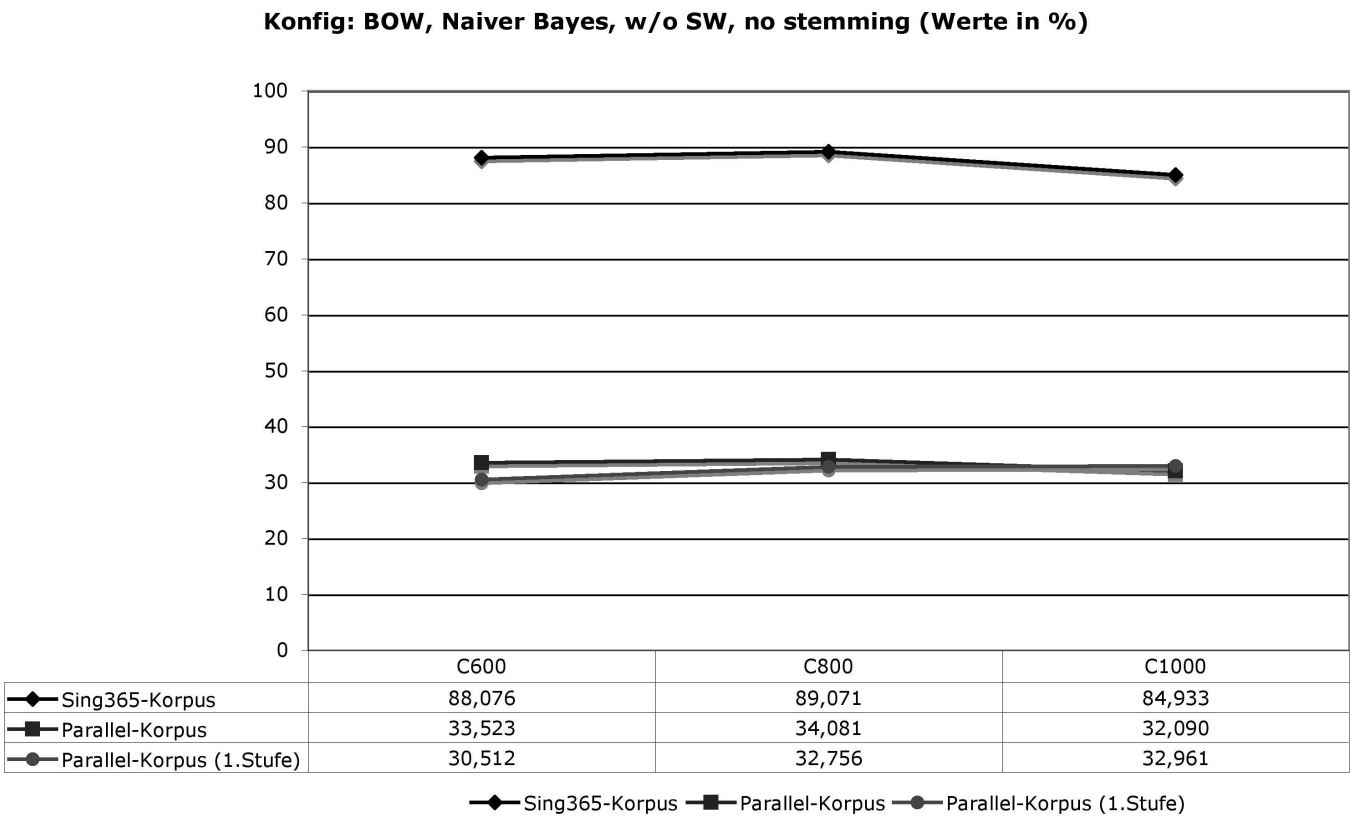


Abbildung A.2: Klassifikationsergebnisse des Sing-365-, Parallel- und Parallelkorpus mit Genres 1.Stufe mit der Konfiguration: BOW Features, Naive Bayes, w/o SW, no stemming. Die Werte sind in Prozent angegeben und wurden mit den Cut-off-Indizes von 600 (C600), 800 (C800) und 1000 (C1000) erzielt.

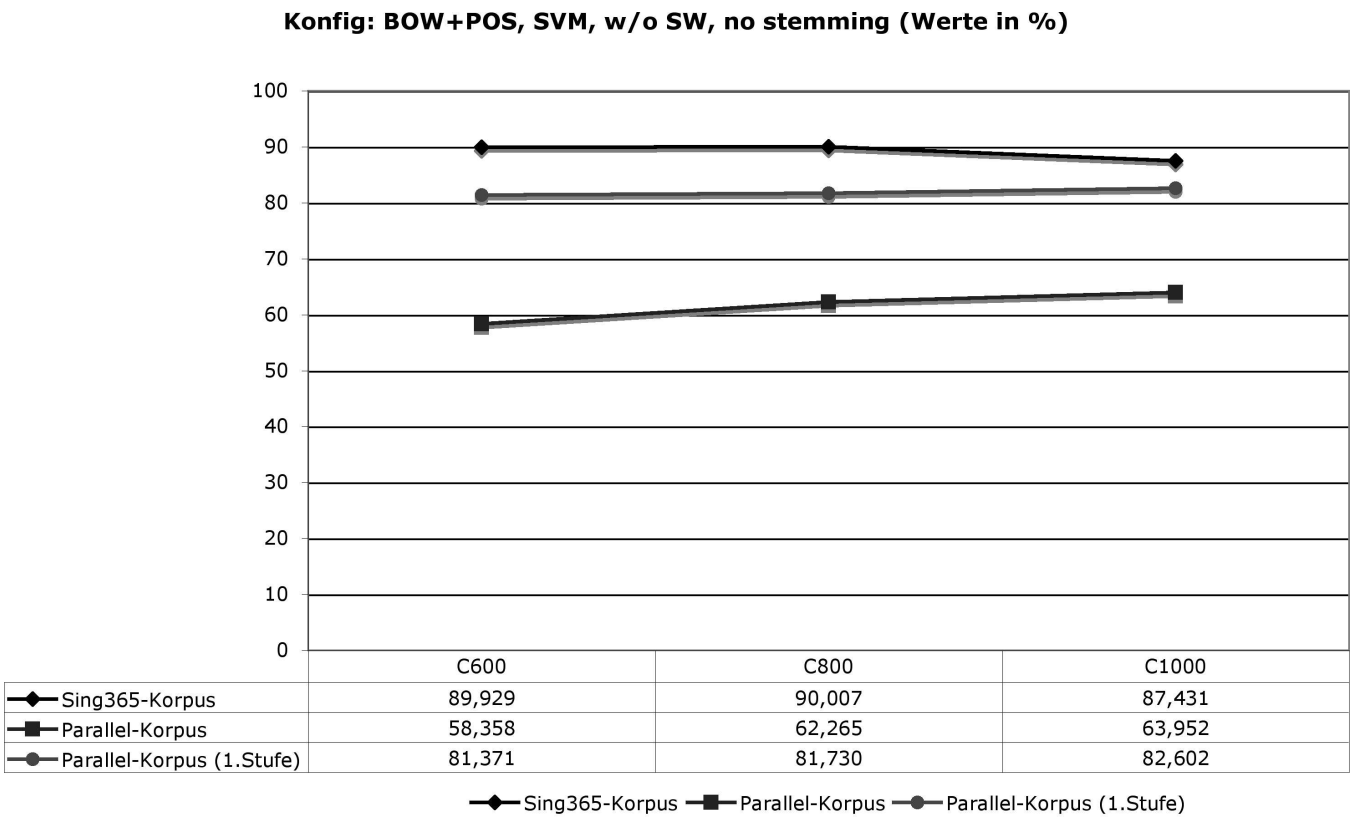


Abbildung A.3: Klassifikationsergebnisse des Sing-365-, Parallel- und Parallelkorpus mit Genres 1.Stufe mit der Konfiguration: BOW+POS Features, SVM, w/o SW, no stemming. Die Werte sind in Prozent angegeben und wurden mit den Cut-off-Indizes von 600 (C600), 800 (C800) und 1000 (C1000) erzielt.

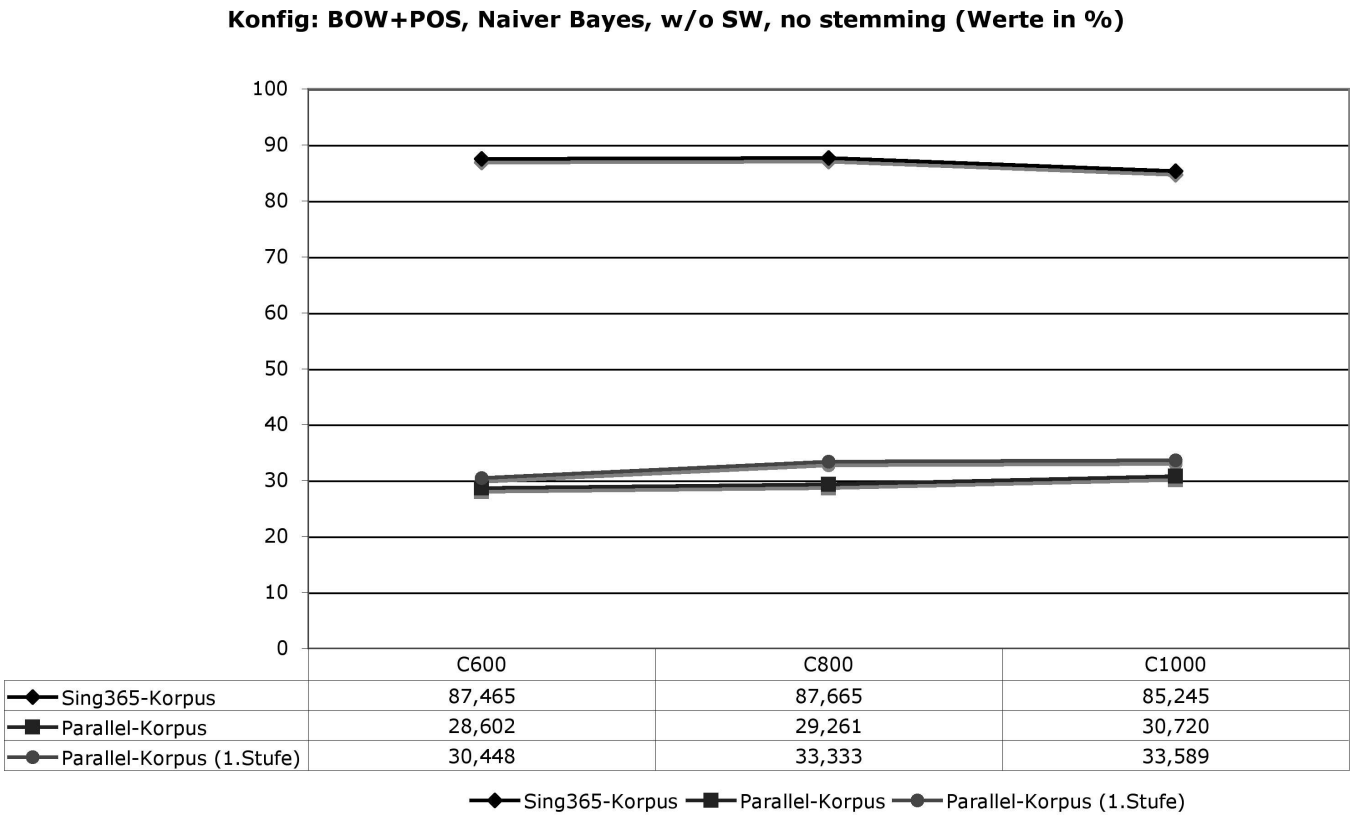


Abbildung A.4: Klassifikationsergebnisse des Sing-365-, Parallel- und Parallelkorpus mit Genres 1.Stufe mit der Konfiguration: BOW+POS Features, Naive Bayes, w/o SW, no stemming. Die Werte sind in Prozent angegeben und wurden mit den Cut-off-Indizes von 600 (C600), 800 (C800) und 1000 (C1000) erzielt.

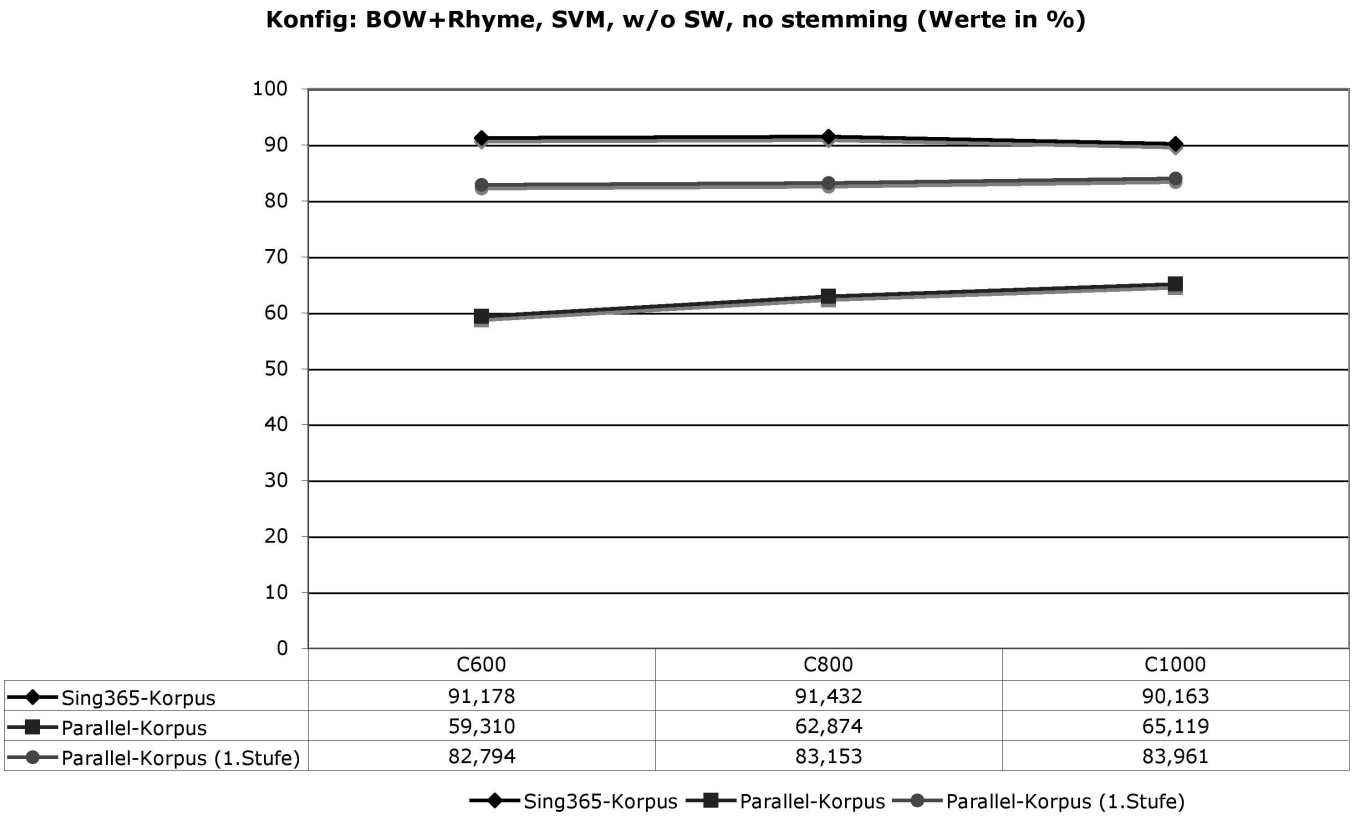


Abbildung A.5: Klassifikationsergebnisse des Sing-365-, Parallel- und Parallelkorpus mit Genres 1.Stufe mit der Konfiguration: BOW+Rhyme Features, SVM, w/o SW, no stemming. Die Werte sind in Prozent angegeben und wurden mit den Cut-off-Indizes von 600 (C600), 800 (C800) und 1000 (C1000) erzielt.

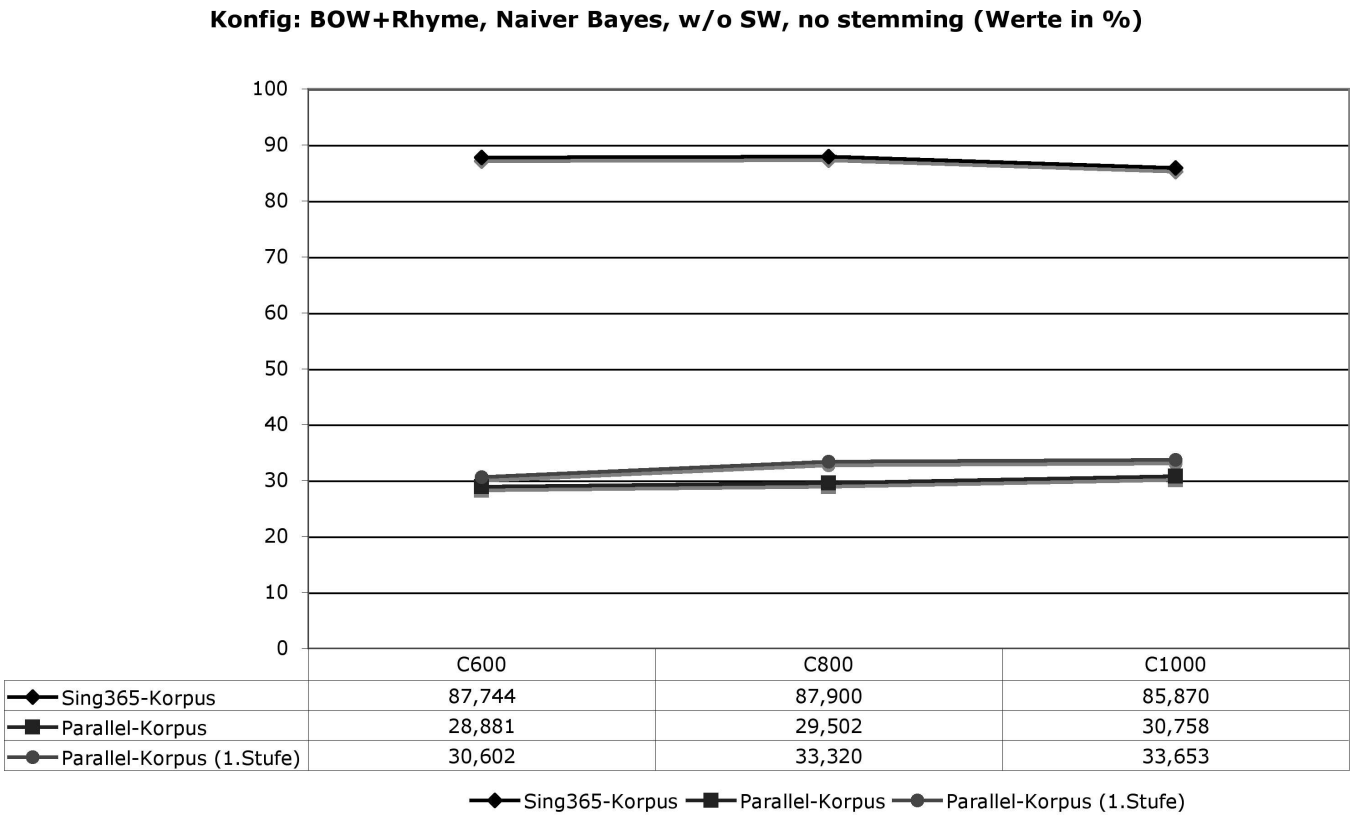


Abbildung A.6: Klassifikationsergebnisse des Sing-365-, Parallel- und Parallelkorpus mit Genres 1.Stufe mit der Konfiguration: BOW+Rhyme Features, Naive Bayes, w/o SW, no stemming. Die Werte sind in Prozent angegeben und wurden mit den Cut-off-Indizes von 600 (C600), 800 (C800) und 1000 (C1000) erzielt.

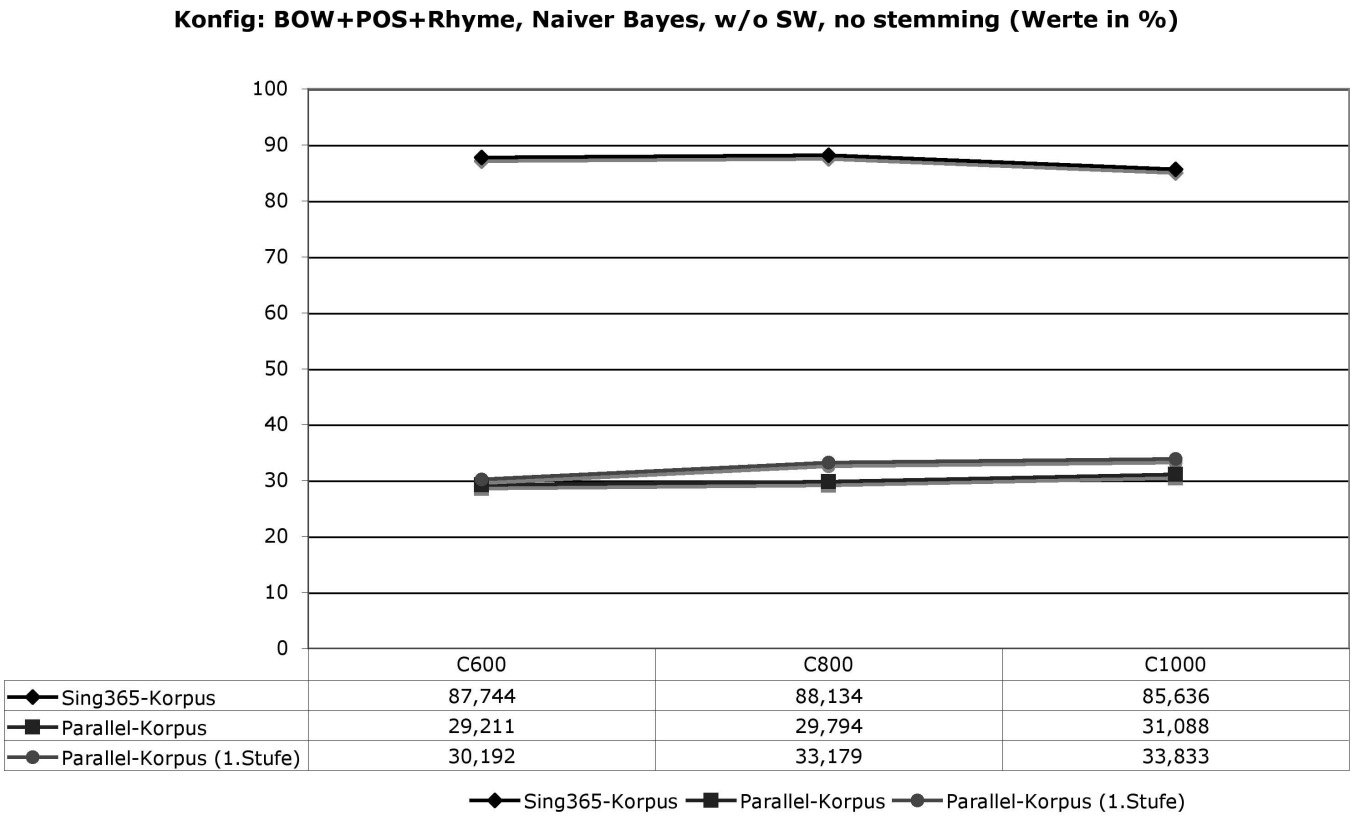


Abbildung A.7: Klassifikationsergebnisse des Sing-365-, Parallel- und Parallelkorpus mit Genres 1.Stufe mit der Konfiguration: BOW+POS+Rhyme Features, Naive Bayes, w/o SW, no stemming. Die Werte sind in Prozent angegeben und wurden mit den Cut-off-Indizes von 600 (C600), 800 (C800) und 1000 (C1000) erzielt.