Multi-Label Klassifikation in Bibliothekskatalogen

An Beispieldaten der ULB Darmstadt (15 ECTS)

Studienarbeit von Michael Gleser aus Frankfurt am Main

Tag der Einreichung:

1. Gutachten: Prof. Dr. Fürnkranz

2. Gutachten: Dr. Loza Mencia

TECHNISCHE UNIVERSITÄT DARMSTADT

FB 20 - Informatik

KE - Knowledge Engineering

Multi-Label Klassifikation in Bibliothekskatalogen

An Beispieldaten der ULB Darmstadt (15 ECTS)

Vorgelegte Studienarbeit von Michael Gleser aus Frankfurt am Main

1. Gutachten: Prof. Dr. Fürnkranz

2. Gutachten: Dr. Loza Mencia

Tag der Einreichung:

Erklärung zur Studienarbeit

Hiermit versichere ich, die vorliegende Studienarbeit ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 14. Dezember 2016

(Michael Gleser)

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

AACR	Anglo-American Cataloguing Rules
DIN	Deutsches Institut für Normung
GBV	Gemeinsamer Bibliotheksverbund
GND	Gemeinsame Normdatei
HeBIS	Hessischer BibliotheksInformationsSystem
ISO	International Standardization Organization
MARC	Machine Readable Cataloging
OPAC	Online Public Access Catalogue
PICA	Project of Integrated Catalogue Automation
RAK	Regeln für die alphabetische Katalogisierung
RDA	Resource Description and Access
RSWK	Regeln Schlagwort Katalog
SWV	Süd-westdeutscher Bibliotheksverbund
ULB	Universitäts- und Landesbibliothek

1 Einführung

Auch vor Bibliotheken macht die immer stärkere Vernetzung nicht halt. Zum einen sind Bibliotheken durch Bibliotheksverbünde untereinander immer stärker vernetzt, was auch einen erhöhten Datenaustausch nach sich zieht. Zum anderen werden Bibliotheken verstärkt durch ihre digitalen Angebote wie den Online Public Access Catalogue (OPAC) genutzt. Durch die digitale Vernetzung können Bibliotheken ihren Nutzern zum einen einen erweiterten Werkkatalog durch Zugriff auf die Kataloge der anderen Bibliotheken ihres Verbundes anbieten. Zum anderen ermöglichen digitale Angebote wie der OPAC erweiterte Recherchemöglichkeiten. Besonders die analog recht aufwändige Suche nach artverwandten Werken ist durch digitale Angebote erleichtert möglich. Um dies zu bewerkstelligen ist eine angemessene Verschlagwortung innerhalb des Bibliothekskatalogs notwendig. Eine Verschlagwortung kann hierbei durch manuelle Eingabe, abhängig von Thesauren, Schlagwortkatalogen oder völlig nach Ermessen des Eingebenden erfolgen. Durch diese manuelle Verschlagwortung kann es zu relativ präzisen Ergebnissen kommen, sie ist jedoch mit hohem Aufwand verbunden.
Für die vorliegende Arbeit wurden reale Daten eines Bibliothekskataloges seitens der ULB Darmstadt bereitgestellt. Dies eröffnet eine Reihe interessanter Untersuchungsmöglichkeiten, insbesondere da die Daten im realen Einsatz verwendet werden. Die ULB Darmstadt würde gerne die derzeitige Verschlagwortung seines digitalen Bibliothekskataloges verbessern. Diese recht offene Aufgabenstellung bildet die Basis der vorliegenden Arbeit.

1.1 Problemstellung und Zielsetzung

Das Problem der Verschlagwortung in Bibliothekskatalogen und ein Plädoyer zur Verbesserung dieser im Bezug zum RSWK-Katalog wurde von Umstätter bereits im Jahr 1991 verfasst [Ums91]. Die Zentralbibliothek Zürich hat hierzu einen digitalen Assistenten entwickelt, welcher unter anderem auf Methoden des maschinellen Lernens zurückgreift um Vorschläge für das Setzen von Schlagworten zu geben [MS14]. Die Idee, Methoden des maschinellen Lernens für die Verbesserung der Verschlagwortung zu verwenden soll auch die vorliegende Arbeit verfolgen. Grundlage der Arbeit sind bereitgestellte Daten der ULB Darmstadt. Durch die Zugehörigkeit der ULB Darmstadt zum Hessischen Bibliotheksinformationssystem (HeBIS) wird das PICA-Format zur Speicherung und zum Austausch verwendet.
Prinzipiell lässt sich die Verschlagwortung innerhalb eines Bibliothekskataloges auf zwei Arten verbessern. Zum einen können externe Quellen zur Anreicherung des Bibliothekskataloges genutzt werden. Durch Zugriff auf Informationen des Internets kann hierbei eine potentiell sehr breite Abdeckung er-

reicht werden. Eine zweiter methodischer Ansatz ist die vorliegenden Daten der ULB Darmstadt zu nutzen um mit diesen weitere Instanzen innerhalb des eigenen bibliographischen Kataloges mit Schlagworten zu versehen. Externe Daten können während dieses Prozesse potentiell auch eingebunden werden, jedoch mit Blick auf Erweiterung des Feature Raums. Methoden des maschinellen Lernens können hierbei die Basis bilden um das vorliegende Problem zu lösen. Da einem Werk mehrere Schlagworte zugewiesen werden können wird hierbei auf Methoden des Multi-Label-Learnings zurückgegriffen.

Die vorliegende Arbeit hat insgesamt drei aufeinander aufbauende Zielsetzungen. In einem ersten Schritt soll eine umfassende technische und inhaltlichen Analyse der bereitgestellten Datenbank auf Eignung für das Multi-Label-Learnings durchgeführt werden. Die hierbei gewonnenen Erkenntnisse sollen in einem zweiten Schritt beispielhaft mit Methoden des Multi-Label-Learnings analysiert werden. In einem dritten Schritt soll untersucht werden, inwiefern sich die Ergebnisse des zweiten Schritts durch Optimierung verbessern lassen. Durch die große Datenmenge wird angenommen, dass hierbei insbesondere Aspekte interessant sind, welche durch Skalierung eines Teilausschnitts der Daten auf die Gesamtmenge gewonnen werden können.

1.2 Relevanz der Arbeit

Die Ergebnisse der vorliegenden Arbeit sind aus praktischer Sicht insbesondere für die ULB Darmstadt interessant. Die ULB Darmstadt kann durch die Ergebnisse einschätzen, inwiefern durch Methoden des Multi-Label-Learnings auf den bereitgestellten Daten die Verschlagwortung verbessert werden kann. Aufgrund des praktischen Charakters der Arbeit ist eine Aussage zur wissenschaftlichen Relevanz schwierig möglich. Die durchgeführte Optimierung durch Erhöhung der Instanzanzahl liefert jedoch einige interessante Aspekte, wie sich die Ergebnisse von Multi-Label-Learning durch Vergrößerung des Instanzraums bei gleichbleibenden Feature- und Labelraum verbessern lassen.

Neben dem inhaltlichen Aspekt der Arbeit wurde, insbesondere durch den hohen Rechenbedarf bei Erhöhung der Instanzanzahl, eine Anpassung des verwendeten Pocahontas Frameworks vorgenommen. Die benötigte Rechenzeit bei zusätzlichen Instanzen skaliert nahezu linear, weshalb bei großer Anzahl an Instanzen die Rechenzeit sich stark vergrößert. Da das Pocahontas Framework keine Anpassung für Multi-Threading kennt, wurde das Framework für die Algorithmen BinaryRelevance und LibLinear hinsichtlich einer Multi-Core Verwendung optimiert. Die benötigte Rechenzeit zur Durchführung der Experimente konnte durch bessere Auslastung der zur Verfügung stehenden Ressourcen stark gesenkt werden.

1.3 Struktur der Arbeit

Die Arbeit ist in insgesamt vier Teile gegliedert. Der Grundlagenteil soll ein Verständnis für die Anforderungen und Herausforderungen der Katalogisierungen in Bibliothekskatalogen schaffen. Neben der Darstellung des verwendeten bibliographischen Datenformats und der verwendeten Frameworks zur Durchführung der Arbeit wird auf Aspekte des Multi-Label-Learnings eingegangen, insbesondere auf Bewertungsmethoden zur Güte.

Nach Darstellung der Grundlagen wird die bereitgestellte Datenbank analysiert. Hierbei werden unter anderem geeignete Features und Labels für die Verwendung im Multi-Label-Learning identifiziert, quantifiziert und analysiert. Die so gewonnenen Daten werden anschließend für die Verwendung in den bereitgestellten Frameworks technisch vorbereitet und in die Multi-Label-Learning Umgebung eingebunden.

Die nun für das Multi-Label-Learning bereitstehenden Daten werden im vierten Schritt durch ein zweistufiges Experiment analysiert. In einem ersten Schritt wird eine generelle Eignung der auf eine repräsentative Menge eingeschränkten Daten für das Multi-Label-Learning geprüft. In einem zweiten Schritt wird durch die Erhöhung der Anzahl an Instanzen eine Optimierung der Ergebnisse angestrebt. Die Arbeit schließt mit der Darstellung der Ergebnisse und Empfehlungen hinsichtlich des Einsatzes von Multi-Label-Learning zur Verbesserung der Verschlagwortung im Bibliothekskatalog der ULB Darmstadt.

2 Grundlagen

Die dargestellten Grundlagen teilen sich in zwei Bereiche. Zum einen werden die inhaltlichen Grundlagen zum Verständnis der bereitgestellten Daten dargestellt. Hierbei wird unter anderem auf Aspekte der Katalogisierung in Bibliotheken und das in der ULB Darmstadt verwendete PICA-Format eingegangen. Für ein besseres technisches Verständnis werden Grundlagen zum Multi-Label-Learning sowie zur Gütebestimmung in Multi-Label-Learning Problemen dargestellt. Die verwendeten Frameworks Mulan und Pocahontas werden hinsichtlich ihres Funktionsumfangs erläutert.

2.1 Katalogisierungen in Bibliotheken

Bibliotheken bieten Zugang zu einer Großzahl an Werken gebündelt an einem Ort. Für den Zugriff auf diese Menge an Werken ist es essentiell eine Möglichkeit zu schaffen diese auffindbar zu machen. Ein Weg dies zu bewerkstelligen sind sogenannte Bibliothekskataloge. Hierbei handelt es sich um Verzeichnisse in denen Meta-Daten zu einzelnen Werken einer Bibliothek gespeichert sind. Bibliothekskataloge eröffnen einen strukturierten Zugriff auf diese Informationen. Durch technische Neuerungen werden die einst in analoger Form gespeicherten Bibliothekskataloge, bzw. durch physische Zettel oder Karteien, durch elektronische Systeme ersetzt. Geschah dies zunächst durch Bereitstellung des Kataloges auf digitalen Datenträgern (wie beispielsweise auf CD) und dem Zugriff über einen lokalen Rechner, wurde durch die weltweite Vernetzung der Onlinezugriff auf den Bibliothekskatalog immer populärer. Diese Art des Zugriffs wird als Online Public Access Catalogue (OPAC). Durch ihn können Nutzer die Ressourcen des Bibliothekskatalogs online einsehen und in ihnen recherchiert werden. Der Onlinezugriff bietet eine Reihe von Vorteilen. Zum einen ist eine physische Recherche nicht mehr notwendig und auch eine Beschränkung hinsichtlich der Verfügbarkeit analoger Ressourcen fällt weg. Zum anderen ermöglichen digitale Bibliothekskataloge den Datenaustausch innerhalb von Bibliotheksverbünden. Dadurch ist es zum einen möglich den eigenen Katalog qualitativ zu verbessern und zum anderen den Nutzern der eigenen Bibliothek auch Werke aus den Verbundkatalogen (bspw. per Fernleihe) anzubieten[Laz15]. Um diesen Austausch auch technisch zu bewerkstelligen müssen sich die Bibliotheken innerhalb eines Verbundes abstimmen. Notwendig ist unter anderem die Standardisierung des Datenaustausches. Dieser muss zum einen technisch standardisiert werden um eine einfache Anbindung der Bibliotheken untereinander zu gewährleisten. Zum anderen müssen die Daten innerhalb des Bibliothekskataloges jedoch auch inhaltlich standardisiert werden. Die inhaltliche Standardisierung ermöglicht eine identische Interpretation der Daten. Aufgrund der Vielzahl an Bibliotheksverbünden in Deutschland und weltweit haben sich ei-

ne Reihe von bibliographischen Datenformaten herausgebildet. Die Regeln zur Katalogisierung wurden in umfassenden Regelwerken niedergeschrieben und sind für den jeweiligen Verbund verbindlich. Die theoretischen Grundlagen der Katalogisierung haben sich mit der Digitalisierung nicht wesentlich geändert. Es existieren prinzipiell zwei verschiedene Informationsbereiche bei der Erschließung von Werken. Die Formalerschließung versucht ein Werk anhand formeller d.h. faktenbasierter Kriterien zu erschließen. Diese Art der Erschließung ist unabhängig von der Person die die Erschließung durchführt. Die Sacherschließung hingegen versucht den Inhalt eines Werkes strukturiert und komprimiert darzustellen. Hierbei ist eine Abhängigkeit der Ergebnisse vom jeweiligen Sachbearbeiter nicht zu leugnen, dem wird jedoch versucht durch Festlegung formaler Kriterien Einhalt zu gebieten. Alle Aspekte sollen folgend detailliert dargestellt werden.

2.1.1 Katalogisierung

Im Rahmen der Katalogisierung werden Werke in einen Bibliothekskatalog aufgenommen. Hierbei muss ein Bibliothekar das Werk sowohl im Rahmen einer Formalerschließung als auch im Rahmen einer Sacherschließung behandeln. Die Formalerschließung behandelt die Erschließung von auf Fakten und objektiven Kriterien basierten Erfassung von Meta-Daten eines Werkes. Im Gegensatz dazu beschäftigt sich die Sacherschließung mit der inhaltlichen Erfassung eines Werkes. Der Inhalt soll hierbei komprimiert wiedergegeben werden. Hierdurch soll sowohl die Auffindbarkeit durch schnellere inhaltliche Bewertung als auch durch Suche artverwandter Werke ermöglicht werden. Für beide Bereiche der Werkerschließung existieren Richtlinien, welche basierend auf den für die ULB Darmstadt maßgebenden Richtlinien des HeBIS dargestellt und erläutert werden sollen.

Formalerschließung

Unter der Formalerschließung[1] versteht man die Erfassung eines Werkes hinsichtlich seiner Meta-Daten. Die erfassten Daten sind unabhängig vom Erfasser und sollen die Fakten eines Werkes erfassen. Hierunter fallen unter anderem Titel, Autor, Veröffentlichungsdatum und Ort[2].
Im Laufe der Zeit haben sich in Deutschland und weltweit eine Reihe von Richtlinien zur formalen Erfassung von Werken herausgebildet. Diese Regeln unter anderem Art und Umfang der zu erfassenden Informationen und die Form in welcher diese letztendlich erfasst werden. In Deutschland haben sich geographisch getrennt eine Reihe von Regelwerken herausgebildet, jedoch ohne regionenübergreifende Relevanz, was vermutlich mit dem föderalen Aufbau der deutschen Wissenschaftslandschaft zusam-

[1] engl. *descriptive cataloguing*

[2] Vgl. `http://www.dnb.de/DE/Erwerbung/Formalerschliessung/formalerschliessung.html`

menhängt. Beispielhaft wären bei diesen Regelwerken die Berliner Anweisungen (BA), die Münchner Katalogordnung (MKO) und die Preußischen Instruktionen (PI) zu nennen. Eine übergreifende Regelung hinsichtlich der formalen Werkerschließung findet sich in Deutschland in der von der Deutschen Nationalbibliothek (DN) herausgegebenen Regeln für die alphabetische Katalogisierung. Im englischsprachigen Raum, aufgrund der Dominanz der Library of Congress (LC), existiert ebenso ein solcher übergreifender Katalog, die Anglo-American Cataloguing Rules (AACR).
Zur Verbesserung auch des internationalen Austausches von Bibliotheksdaten wurde unter Federführung der LC unter mit Hilfe der DN ein Katalog entwickelt, welcher mittelfristig die existierenden Kataloge ablösen soll. Hierbei handelt es sich um den Resource Description and Access (RDA). Der RDA ist der Versuch Katalogisierungen zu vereinheitlichen und wird unter anderem vom HeBIS umgesetzt. Da sich beim HeBIS die Umsetzung noch in einer Übergangsphase befindet, befinden sich teilweise aktualisierte Deskriptoren in der jeweiligen Datenbank. Daten, welche unter einem anderen Katalog erfasst wurden, werden mit der Bezeichnung Altdaten gekennzeichnet.

Sacherschließung

Sacherschließung[3] beschäftigt sich mit der inhaltlichen Erfassung eines Werkes. Hierbei wird versucht der Inhalt des Werkes komprimiert wiedergegeben zu werden, was unter anderem eine auf dem Inhalt basierte Suche ermöglicht. Beispielsweise werden im Rahmen der Sacherschließung Zusammenfassungen erstellt oder Schlagworte vergeben, welche den Inhalt charakterisieren sollen. Die Qualität der Sacherschließung ist stark vom jeweiligen Erfasser abhängig. Dieser vergibt aufgrund seiner persönlichen Einschätzung die jeweiligen Meta-Daten, weshalb potentiell bei mehreren verschiedenen Erfassern unterschiedliche Ergebnisse und Qualitäten dieser erwarten werden können.

Um die Qualität der erfassten Informationen, insbesondere hinsichtlich der Übertragbarkeit zwischen Bibliothekskatalogen zu gewährleisten haben sich ähnlich der Formalerschließung Richtlinien für die Sacherschließung herausgebildet. Die im deutschen Raum scharfe Trennung zwischen Formal- und Sacherschließung ist in dieser Form im englischsprachigen Raum nicht gegeben. Hier werden sowohl Formal- als auch Sacherschließung gemeinsam geregelt, was sich unter anderem in der RDA zeigt. Da diese auch in Deutschland übernommen wird, verschwimmt auch hier die Grenze zwischen Formal- und Sacherschließung. Bis zur Einführung des RDA in Deutschland wurden jedoch Richtlinien verwendet, welche die Sacherschließung explizit regeln. Da Sacherschließung unter anderem die Schlagworterfassung beinhaltet hat sich ein Regelkatalog herausgebildet, welcher eine Gleichartigkeit

[3] engl. *subject cataloguing*

der erfassten Schlagwörter sicherstellen soll. Die Regeln Schlagwort Katalog (RSWK) bestimmen wie Werke mit Schlagworten versehen werden. Der RSWK[4] wird von der Deutschen Nationalbibliothek (DN) herausgegeben und besitzt für eine Großzahl an Bibliotheken in Deutschland Gültigkeit, so unter anderem auch für die Bibliotheken des HeBIS. Der RSWK regelt Form und Inhalt der Schlagworte. Er kennt unter anderem Regeln für verschiedene thematische Gruppierungen von Schlagworten, welche unter anderem Personen-, geographische, ethnographische sowie Sach- und Zeitschlagwörter umfassen. Um die Vergabe von Schlagworten in ihrer Gesamtmenge begrenzt und vereinheitlicht zu halten werden basierend auf den Regeln der Schlagwortkataloge sogenannte Normdateien gepflegt. Dies sind Dateien die normierte Schlagworte enthalten und somit einer Zersplitterung von Schlagworten gleichen Inhalts auf verschiedene Begriffe entgegen wirken sollen. Gegliedert sind die Normdateien nach verschiedene Themen- und Sachgebieten. Für den deutschsprachigen Raum ist die Gemeinsame Normdatei (GND) maßgeblich. Um diese abseits des RSWK verwenden zu können wurde sie für die Verwendung im Rahmen des RDA angepasst. Die GND besitzt auch für den HeBIS maßgebliche Gültigkeit.

2.1.2 Bibliographische Datenformate

Durch den Zusammenschluss von Bibliotheken in Bibliotheksverbünden zur besseren Kooperation und Austausch von bibliothekarischen Daten wurde die Definition von bibliographischen Datenaustauschformaten notwendig. An diese Formate werden eine Reihe von Kriterien gelegt. Zum einen sollen die Formate es ermöglichen Werke inhaltlich umfassend erschließen und dokumentieren zu können. Zum anderen soll der technische Austausch zwischen verschiedenen Bibliotheken des gleichen Verbunds ermöglicht werden. Insofern müssen die Datenformate manuelle Bearbeitung (bzw. Bearbeitung mit Hilfe vermittelnder Software) ermöglichen, also menschlich interpretierbar sein, zum anderen müssen sie einen technischen Charakter haben um den Datenaustausch effizient bewerkstelligen zu können. Durch die Digitalisierung in Bibliotheken weltweit ist es aus technischer Notwendigkeit zur Entwicklung einer Reihe von bibliographischen Datenformaten gekommen.
Federführend für die Entwicklung von Datenformaten im amerikanischen Raum ist die Library of Congress (LC). Die LC erarbeitete in den frühen 70er Jahren innerhalb einer Arbeitsgruppe das Machine Readable Cataloging (MARC) Datenformat. Bis zum heutigen Zeitpunkt ist es in den USA das dominierende Datenformat. Eine Zusammenfassung der Entwicklung des MARC Datenformats findet sich bei Avram [AC75], eine aktuelle Referenz des Formats kann auf der Website der LC abgerufen werden[5]. Im europäischen Raum wurde die Entwicklung von bibliographischen Datenformate weniger zentralisiert.

[4] Vgl. http://files.dnb.de/pdf/rswk_gesamtausgabe.pdf
[5] Vgl. https://www.loc.gov/marc/

Hier haben sich im Laufe der Zeit eine Reihe von unterschiedlichen Datenformaten herausgebildet, welche von einzelnen Bibliotheksverbünden verwendet wurden. Zu nennen sind beispielhaft die Formate Unimarc, MAB, PICA, ZDB und allegro. Durch technische Neuerungen sind diese Formate jedoch teilweise nicht mehr in Benutzung. Eine Übersicht der verschiedenen Formate inklusive einer Konkordanz von Aufbau und Inhalt findet sich bei Eversberg [Eve94], der auch die Geschichte der Entwicklung der Formate ausführlich darstellt. Da es sich bei Datenformaten im Bibliotheksbereich prinzipiell um Austauschformate handelt finden internationale Standards in diesem Bereich Anwendung. Zu nennen sind hierbei der internationale *ISO 2709:2008* [Iso] und das deutsche Gegenstück *DIN 1506:1987-03* [Din] welche den Aufbau von Formaten zum Datenaustausch regeln.
Das im amerikanischen Bereich verwendete MARC Format ist mit seiner mehr als 40 jähriger Verwendung ein Veteran unter den bibliographischen Datenformaten. Trotz ständiger technischer Anpassung des Formats haben sich die Anforderungen an bibliographische Datenformate gewandelt. Zur Neuentwicklung eines Datenformats zum bibliographischen Datenaustauschs hat die LC erneut eine Arbeitsgruppe ins Leben gerufen. Der Name des neuen Austauschformats wird hierbei mit BIBFRAME angegeben. Unter anderem soll die Neuentwicklung die Verknüpfung von Daten ermöglichen und somit den Anforderungen einer vernetzten Bibliothekslandschaft gerecht werden. Eine ausführliche Beschreibung von Anforderungen und Entwicklungen hinsichtlich BIBFRAME findet sich bei Kroeger [Kro13].

2.2 Das PICA Datenformat

Das **Project of Integrated Catalogue Automation** (PICA) Datenformat ist ein im europäischen Raum verwendetes Format zur Speicherung und zum Austausch von Daten zwischen bibliographischen Katalogen. Zu seinen Nutzern in Deutschland zählen unter anderem das Hessische Bibliotheksinformationssystem (HeBIS)[6] mit der Zentrale Frankfurt, der Gemeinsamen Bibliotheksverbund (GBV)[7] mit Sitz in Göttingen und der Südwestdeutsche Bibliotheksverband (SWB)[8] mit Sitz in Konstanz. Aufgrund seiner Mitgliedschaft im HeBIS ist das PICA Datenformat auch für die ULB Darmstadt maßgebend. Neben der ULB Darmstadt gehören dem HeBIS deutschlandweit 521 Bibliotheken an, wobei die Hauptbibliotheken hauptsächlich in Hessen und Rheinland-Pfalz liegen. Die deutschlandweite Verteilung ergibt sich über die Zweigstellen der Hauptbibliotheken welche ebenfalls Mitglied im HeBIS Verbund sind[9]. Das PICA Datenformat ist somit für einen nicht unerheblichen Teil der deutschen Bibliothekslandschaft relevant.

[6] Vgl. http://www.hebis.de/

[7] Vgl. http://www.gbv.de/

[8] Vgl. http://www.bsz-bw.de/swbverbundsystem/

[9] Vgl. http://www.hebis.de/de/1ueber_uns/verbund/bibliotheken/bib-liste.php?cat=HeBIS-Mitgliederbibliotheken&cond=Ort

Das PICA-Format muss zumindest innerhalb eines Bibliotheksverbundes standardisiert sein, wobei eventuelle lokalen Anpassungen vorgenommen werden können. Deshalb wird versucht ausschließlich auf die Dokumentation des HeBIS Bezug zu nehmen und nur falls diese nicht aussagekräftig ist auf die Dokumentationen des GBV bzw. des SWB zurückzugreifen.
Das PICA-Format ist ein zeilenbasiertes Format zur Datenhaltung und zum Datenaustausch in Bibliothekskatalogen. Es liegt derzeit in der dritten Version vor[10]. Das PICA Format besteht prinzipiell aus zwei Formaten, dem PICA3 und dem PICA+ Format, welche problemlos ineinander konvertiert werden können. Die Trennung erfolgt aufgrund des unterschiedlichen Aufgabenbereichs. Das PICA3 Format fokussiert auf menschliche Bearbeiter und wurde deshalb mit Fokus auf einfache Interpretierbarkeit entwickelt. Erreicht wird dies durch einen thematisch geordneten Katalog an Bezeichnern und speziellen Trennzeichen um Informationen innerhalb der jeweiligen Datenfelder ordnen zu können. Im Gegensatz zum PICA3 Format zielt das PICA+ Format auf effiziente technische Handhabung. Insbesondere die Speicherung und Übertragung von Datensätze zwischen Bibliothekskatalogen steht hierbei im Fokus. Durch die Konvertierbarkeit der Formate ineinander müssen diese sich inhaltlich entsprechen. Eine Gegenüberstellung der beiden Formate findet sich beispielhaft in Grafik 2.1. Diese schematische Darstellung verdeutlicht die unterschiedliche Verwendung von Trennzeichen zur Organisation von Subfeldern sowie unterschiedliche Bezeichner zum Zeilenbeginn, welche die Art der Information kennzeichnen.

Pica3-Eingabe- und Anzeigeform (mit vorgeschriebener Interpunktion):

```
4000 Hauptsachtitel // Körp. Ergänzung : Zusatz / Verfasserangabe
```

Pica+ interne Speicherungsform (Teilfeldkennungen statt Interpunktion):

```
021A ƒaHauptsachtitelƒeKörp. ErgänzungƒdZusatzƒhVerfasserangabe
```

Abbildung 2.1: Gegenüberstellung der Zeilenrepräsentation PICA3 - PICA+

Quelle: Eversberg(1994) p.41

Relevant für die vorliegende Arbeit ist das Datenformat des zugrundeliegenden Datensatzes. Dieser ist im PICA+ Format gespeichert. Aufgrund der Konvertierbarkeit der Datenformate PICA+ und PICA3 ineinander soll folgend nur das PICA+ Format ausführlicher behandelt werden. Die jeweilige Darstellung des PICA3 Formats ergibt sich durch Anpassung von Trennzeichen im Datenbereich und dem Bezeichner zum Zeilenbeginn analog. Zur Diskussion der einzelnen Bestandteile eines PICA+ Datensatzes findet sich in Grafik 2.2 eine farbliche Hervorhebung der einzelnen Elemente. Im PICA+ Format werden die einzelnen Werke in Datenblöcken gespeichert. Jeder Datenblock wird hierbei durch einen dreistelligen Code (hier in Gelb hervorgehoben) eingeleitet, welcher die Ebene der Information anzeigt. Ausprägungen der

[10] Vgl. http://www.wissenschaftsrat.de/download/archiv/10463-11.pdfS.20

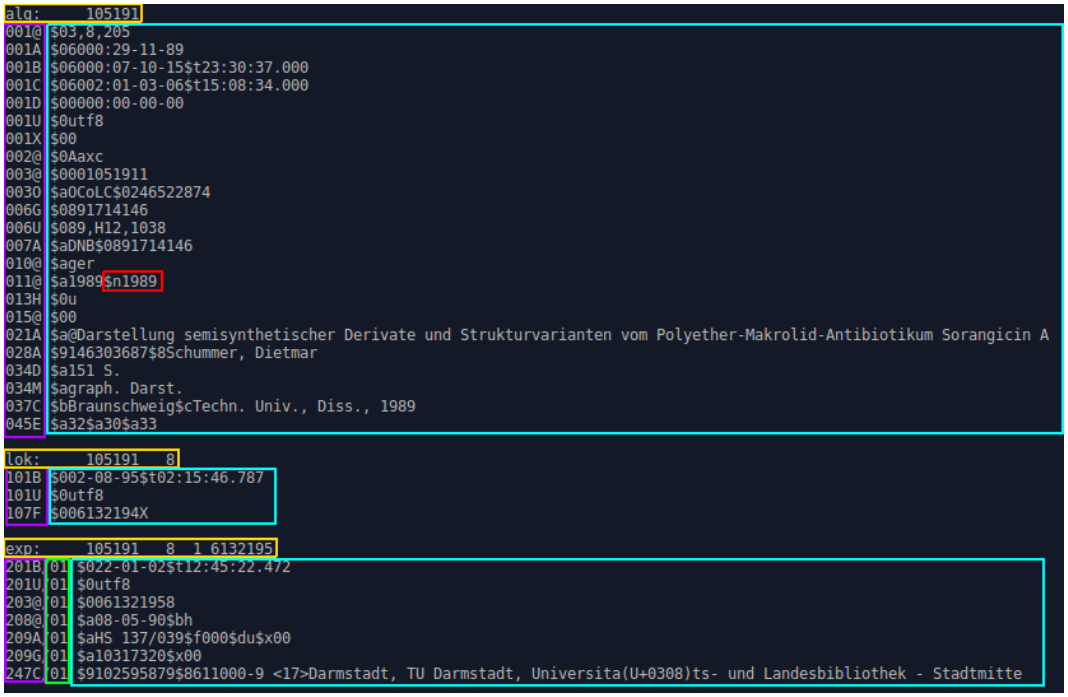


Abbildung 2.2: Farbliche Hervorhebung der Elemente eines PICA+ Datensatzes

Blockcodes sind **alg** für allgemeine Informationen zum vorliegenden Werk, **exp** für exemplarspezifische Informationen und **lok** für Informationen welche durch die jeweilige Bibliothek vergeben und somit lokal unterschiedlich sein können. Auf den Blockcode folgt nach einem Doppelpunkt und einem Leerzeichen eine jedem Werk eindeutig zuordnenbare Identifikationsnummer (ID). Diese ID ermöglicht die Zuordnung von einzelnen Blöcken zu einem Werk, im vorliegenden Datensatz sind diese Blöcke jedoch aufeinander folgend gespeichert, weshalb eine Zuordnung durch eine lineare Abarbeitung möglich ist. Die Blöcke **lok** und **exp** besitzen eine zusätzliche Identifikationsnummer, welche den Datensatz gesondert kennzeichnet.

Die eigentlichen Informationen eines jeden Datensatzes sind innerhalb der Datenblöcke zeilenbasiert gespeichert. Jede Zeile wird hierbei durch einen Bezeichner eingeleitet. Dieser Bezeichner (hierbei violett eingerahmt) repräsentieren verschiedene Informationskategorien des PICA Formats. Dies kann unter anderem der Titel des Werkes, der Autor, Veröffentlichungsjahr oder andere dem Werk zuordnenbare Informationen sein. Der Bezeichner ist hierbei ein Element, welches bei Konversion zwischen PICA3 und PICA+ angepasst wird. Eine Zuordnung der verschiedenen PICA+ Bezeichner zu den logischen Bereichen des PICA3 Formats findet sich im Anhang in Tabelle 7.1. Wird eine Information innerhalb eines Bezeichners in mehreren Zeilen gespeichert (falls es beispielsweise eine Reihe von Autoren gibt), so kann dies durch eine sogenannte Occurence angezeigt werden. Die Occurence folgt auf den Bezeichner, eingeleitet durch einen Slash (’/’) und von zwei Ziffern gefolgt (hier grün hervorgehoben).

Die Informationen einer jeden Zeile folgen getrennt durch ein Leerzeichen nach dem Bezeichner bzw. der Occurence (hier türkis hervorgehoben). Informationen müssen hierbei nicht zwingend atomar abgelegt werden, d.h. innerhalb einer Zeile können durchaus mehrere Informationen stehen. Dadurch ist die Einführung von Trennzeichen notwendig um diese Teilinformationen voneinander trennen zu können. Die jeweiligen Teilinformationen sind in der Grafik rosa hervorgehoben. Um die Teilinformationen identifizierbar und einzeln lesbar zu machen ist es notwendig unterschiedliche Trennzeichen je nach Art der Teilinformation zu verwenden. Bei PICA+ bestehen diese Trennzeichen aus einem Dollarzeichen ($) gefolgt von einem Buchstaben oder einer Ziffer. Die Informationen nach den Trennzeichen folgen einer bestimmten Logik, welche sich jedoch bei PICA+ nicht direkt erschließt. Generell werden Schlagwörter unter dem Trennzeichen $8 gespeichert, Textinformationen unter dem Trennzeichen $a. Die Verwendung der Trennzeichen beim PICA3 Format ist vielfältiger um dem jeweiligen Benutzer die Bearbeitung der Datensätze intuitiv zu ermöglichen.

Das PICA Format des HeBIS kennt insgesamt 503 verschiedene Bezeichner, welche ebenso viele Informationskategorien abdecken[11]. Diese Bezeichner teilen sich auf die verschieden codierten Blöcke des PICA+ Formats auf. Wie im Anhang in Tabelle 7.1 sind die Bezeichner des PICA+ im Bereich 0XXX dem Code **alg**, im Bereich 1XXX dem Code **exp** und im Bereich 2XXX dem Code **lok** zugeordnet. Aufgrund von Umstrukturierungen wegen der Umstellungen und Aktualisierung im Rahmen der RKA haben eine Reihe von Bezeichnern ihre Information geändert und wurden neu strukturiert. Im HeBIS werden diese Daten als Altdaten bezeichnet und werden zukünftig aktualisiert werden.

2.3 Multi-Label Learning

Multi-Label-Learning beschäftigt sich mit dem Problem, einer Instanz mehrere Label zuzuweisen [GV15]. Durch die Möglichkeit der Zuweisung mehrerer Label an eine Instanz ergeben sich eine Reihe von Besonderheiten bei Multi-Label-Problemen. Unter anderem können Label in Beziehung zueinander stehen, wobei besonders Hierachien zwischen Labels sowie deren inhaltliche Beziehung zueinander interessant sind. Insbesondere bei Zuweisung mehrerer Labels kann sich das Problem der inhaltlichen Kongruenz ergeben[Lua12]. Hierbei sind Label trotz unterschiedlicher Bezeichnung inhaltlich miteinander verbunden und eine klare inhaltliche Trennung der Label meist nur schwer möglich. Zum besseren Verständnis der Multi-Label-Klassifikation, insbesondere im Verhältnis zu klassischen Lernproblemen sollen folgend methodische Ansätze des Multi-Label-Learnings sowie Gütebestimmungen erläutert werden.

[11] Vgl. http://www.hebis.de/de/1publikationen/arbeitsmaterialien/hebis-handbuch/kategorien/kategorien-rda_index_druck.php?cat=&order=PicaPlus

2.3.1 Methodische Ansätze

Durch das Multi-Label-Learning ergeben sich eine Reihe von Herausforderungen, welche über jene bei Klassifizierungsproblemen hinausgehen. Zum einen können einer Instanz mehrere Label zugeordnet werden. Hierbei können zwischen keinem und unendlich vielen Labels eine korrekte Zuweisung im Lernproblem sein. Da eine Zuweisung mehrerer Werte mit den klassischen Klassifizierungsalgorithmen nicht ohne weiteres möglich ist, müssen Multi-Label-Learning Probleme in der Regel modifiziert werden. Hierbei gibt es prinzipiell zwei verschiedene Ansätze wie Tsoumakas[TK07] darstellt. Zum einen kann das Multi-Label-Problem transformiert werden, sodass es mit herkömmlichen Algorithmen lösbar ist, zum anderen können die verwendeten Algorithmen angepasst werden.

Ein populärer Ansatz zur Transformation von Multi-Label-Learning Problemen ist die Zerlegung in binäre Lernprobleme. Hierbei wird je Label ein Lernproblem gebildet, wobei Instanzen die das Label enthalten positiv und Instanzen die das Label nicht enthalten negativ klassifiziert sind. Die einzelnen Teilprobleme lassen sich anschließend mit Klassifizierungsalgorithmen lösen. Die Herausforderung bei diesem Ansatz liegt in der anschließenden Zusammenstellung der Ergebnisse. Dieser Ansatz wird auch als Label-Powerset bezeichnet und in einer weiteren Arbeit von Tsoumakas ausführlich dargestellt[TKV10]. Insgesamt müssen bei diesem Ansatz bei n-verschiedenen Labels insgesamt n-verschiedene Lerner gebildet werden. Somit skaliert die benötigte Rechenzeit mit der Anzahl an Labels. Erweiterungen des Ansatzes zur Handhabung der gesteigerten Komplexität durch hohe Anzahlen an Labels kann beispielsweise durch Pruning begegnet werden. Hierbei wird durch die Anwendung von Filtern die Menge an Labels reduziert, beispielsweise in dem nur Instanzen betrachtet werden, welche mehr als eine gesetzte Schranke an Label enthalten.
Neben der Transformation des Lernproblems kann auch durch die Anpassung oder Neuentwicklung von Algorithmen Multi-Label-Learning Probleme gelöst werden. In der Arbeit von Tsoumakas[TK07] sind eine Reihe von Anpassungen populärer Algorithmen für das Multi-Label-Learning aufgelistet. Unter anderem existieren Anpassungen für die Algorithmen AdaBoost [SS00], C4.5 [CK01], k-N̈earest Neighbor(̈kNN) [ZZ05] und für Support Vector Machines (SVM) [EW01] [GS04]. Da es insbesondere durch eine hohe Anzahl an Labeln beim Multi-Label-Learning zu Engpässen hinsichtlich Ressourcen und Rechenzeit kommen kann haben Tsoumakas et al. einen hierarchischen Klassifizierer namens Hierarchy Of Multilabel classifiERs (HOMER) entwickelt[TKV08].

2.3.2 Gütebestimmung

Insbesondere bei der Transformation von Multi-Label-Problemen in binäre Teilprobleme lassen sich Gütemaße von binären Klassifizierungsproblemen nutzen um die Güte des Multi-Label-Lerners zu bewerten. Um diese Gütemaße entsprechend verwenden zu können ist es notwendig diese an die Gegebenheiten des Multi-Label-Learnings anzupassen. Neben der Verwendung dieser Gütemaße existieren eine Reihe weiterer Gütemaße die speziell für das Multi-Label-Learning entwickelt wurden. Diese setzen auf den Ergebnissen binärer Klassifizierer auf, beschäftigen sich jedoch zusätzlich mit dem Ranking der einzelnen Label untereinander und je Instanz.
Binäre Klassifizierungsprobleme können während der Klassifikation vier verschiedenene Grundwerte annehmen. Hierbei sind korrekt klassifizierte Instanzen entweder True Positive, also als positiv bewertete positive Beispiele oder True Negative, also als negativ bewertete negative Beispiele. Eine Fehlklassifizierung kann in den Fehler 1.Art bei als negativ klassifizierten positiven Instanzen und den Fehler 2.Art bei als positiv klassifizierten negativen Instanzen eingeteilt werden. Tabelle 2.1 schafft einen Überblick über die verschiedenen Maße. Die auf diese Art ermittelten Werte lassen sich zu Gütemaßen zusammenfassen. Tabelle 2.2 verschafft einen Überblick über populäre Gütemaße bei binären Klassifizierungsproblemen. Gütemaße des Multi-Label-Learning basieren auf diesen Gütemaßen weshalb es essentiell ist sich diese ins Gedächtnis zu rufen. Möchte man nun die vorgestellten Gütemaße für Klassifizierungsprobleme auch

	Vorhersage Positiv (PP)	Vorhersage Negativ (PN)
Wahrer Wert Positiv (P)	True Positive (TP)	False Negative (FN)
Wahrer Wert Negativ (N)	False Positive (FP)	True Negative (TN)

Tabelle 2.1: Basis Gütewerte binäre Klassifizierung

im Multi-Label-Learning einsetzen, so müssen die einzelnen Werte der binären Teilprobleme zusammengefasst werden. Durch Tsoumakas [TKV10] werden drei Methoden aufgezeigt, welche die vorliegenden Werte je Label kombinieren und daraus ein Gütemaß für das Multi-Label-Learning bilden. Die dargestellten Ansätze sind hierbei instanzbasiert (example-based) oder labelbasiert (label-based). Instanzbasierte (example-based) Gütemaße beziehen sich in ihrer Betrachtung auf die Label einer jeden Instanz. Hierbei wird betrachtet inwiefern die Label einer Instanz korrekt zugewiesen wurden und bei wie vielen Label einer Instanz dies der Fall ist. Besonders wenn es sich um einen Datensatz mit einer hohen Anzahl an Label handelt ist die Betrachtung der instanzbasierten Gütemaße interessant. Zur Ermittlung der instanzbasierten Gütemaße werden zwei Mengen gebildet. Die erste der Mengen enthält hierbei die ursprüngliche Menge an Instanzen und ihrer Labels und die zweite Menge die Ergebnisse der Klassifi-

Name	Berechnungsmethode	Beschreibung
Precision	$= \frac{\sum TP}{\sum TP + \sum FP}$	Misst den Anteil an korrekt positiv klassifizierten Instanzen an allen Klassifizierungen. Eine hohe Precision besagt, dass der Lerner sehr genau arbeitet, also nur solche Instanzen positiv labelt die es mit hoher Wahrscheinlichkeit auch sind.
Recall	$= \frac{\sum TP}{\sum TP + \sum FN}$	Misst wie viele positive Instanzen in der Klassifizierung erfasst und als positiv klassifiziert wurden. Ein hoher Recall bedeutet, dass eine hohe Anzahl an positiven Instanzen erfasst und als positiv klassifiziert wurden.
Specificity	$= \frac{TP}{TP + FN}$	Misst wie viele negativ Instanzen auch als negativ und nicht fälschlicherweise als positiv klassifiziert wurden. Eine hohe Specificity bedeutet eine geringe Anzahl an falsch klassifizierten Negativbeispielen.
F-Measure	$= \frac{2 * Precision * Recall}{Precision + Recall}$	Ist das harmonische Mittel von Precision und Recall zusammen. Der F-Measure versucht die Aussagekraft der Güte eines Klassifizierers zu verbessern indem Precision und Recall kombiniert werden. Dadurch können die Nachteile beider Maße bedingt ausgeglichen werden.
Accuracy	$= \frac{\sum TP + \sum TN}{\sum TP + \sum FN + \sum FP + \sum TN}$	Misst wie viele Instanzen an allen vorhandenen Instanzen korrekt klassifiziert wurden. Hierbei werden sowohl korrekt positiv als auch korrekt negative klassifizierte Instanzen gezählt.

Tabelle 2.2: Gütemaße zur Bestimmung in Klassifizierungsproblemen

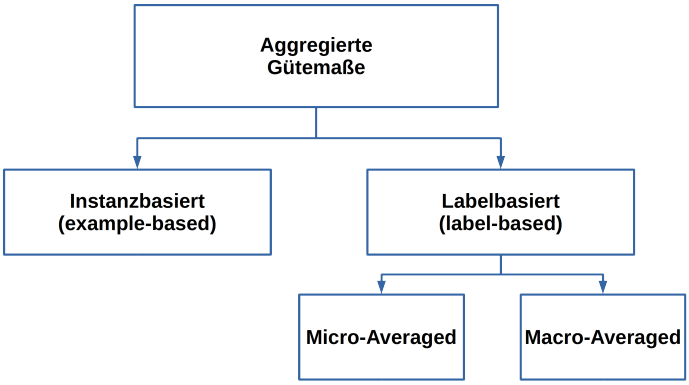


Abbildung 2.3: Übersicht aggregierte Gütemaße

kation. Durch Vergleich dieser beiden Mengen miteinander ist es nun, insbesondere durch Bildung von Schnittmengen möglich, die instanzbasierten Gütemaße zu berechnen. Eine Übersicht über Formeln zur Berechnung der Gütemaße findet sich in Tabelle 2.3. Hierbei ist Y_i die ursprüngliche Instanzmenge und Z_i die gelabelte Instanzmenge.

Im Gegensatz zu den instanzbasierten Gütemaßen, rücken labelbasierte Gütemaß die einzelnen Label in das Zentrum der Betrachtung. Durch die Zerlegung in binäre Teilprobleme können Gütemaße der binären Klassifizierung je Label ermittelt werden. Die Herausforderung ist hierbei, wie die Ergebnisse der einzelnen Label zusammengefasst werden können um sinnvolle Aussagen zuzulassen. Es existieren zwei Ansätze um dieses Problem zu lösen, das Micro- und das Macro-Averaging Verfahren. Beide Verfahren greifen auf die Gütemaße von Klassifizierungsproblemen zurück, haben jedoch unterschiedliche Methoden zur Mittlung der Werte.
Das Micro-Averaging betrachtet die absoluten Werte an korrekt oder inkorrekt zugewiesenen Labels. Bevor die Gütemaße berechnet werden, werden die einzelnen Werte je Label aufsummiert und aus diesen Summen anschließend die Gütemaße berechnet. Das Macro-Averaging hingegen berechnet einzeln für jedes Label das jeweilige Gütemaß und mittelt anschließend diese Gütemaße durch das arithmetische Mittel. Abbildung 2.4 zeigt abstrahiert die zur Berechnung der Maße herangezogenen Formeln.

Name	Berechnung	Anmerkung
Hamming-Loss	$\frac{1}{m}\sum_{i=1}^{m}\frac{\lvert Y_i \triangle Z_i\rvert}{M}$	Symmetrische Differenz beider Mengen (XOR)
ClassificationAccuracy	$\frac{1}{m}\sum_{i=1}^{m} I(Z_i = Y_i)$	Anteil an exakt gelabelten Instanzen
$Precision_{exmpl}$	$\frac{1}{m}\sum_{i=1}^{m}\frac{\lvert Y_i \cap Z_i\rvert}{\lvert Z_i\rvert}$	Gemittelter Wert aller Instanzen bei korrekt zugewiesenen Label an allen zugewiesenen Labeln der jeweiligen Instanz
$Recall_{exmpl}$	$\frac{1}{m}\sum_{i=1}^{m}\frac{\lvert Y_i \cap Z_i\rvert}{\lvert Y_i\rvert}$	Gemittelter Wert aller Instanzen bei korrekt zugewiesenen Label an allen möglichen korrekten Labeln der jeweiligen Instanz
$F1_{exmpl}$	$\frac{1}{m}\sum_{i=1}^{m} 2\frac{\lvert Y_i \cap Z_i\rvert}{\lvert Z_i\rvert + \lvert Y_i\rvert}$	Harmonisches Mittel aus Precision und Recall
$Accuracy_{exmpl}$	$\frac{1}{m}\sum_{i=1}^{m}\frac{\lvert Y_i \cap Z_i\rvert}{\lvert Y_i \cup Z_i\rvert}$	Gemittelter Wert aller Instanzen bei korrekt zugewiesenen Label an allen möglichen Labeln der jeweiligen Instanz

Tabelle 2.3: Formeln instanzbasierte Gütemaße

Quelle: [TKV10]

$$B_{\text{macro}} = \frac{1}{q}\sum_{\lambda=1}^{q} B\left(tp_\lambda, fp_\lambda, tn_\lambda, fn_\lambda\right)$$

$$B_{\text{micro}} = B\left(\sum_{\lambda=1}^{q} tp_\lambda, \sum_{\lambda=1}^{q} fp_\lambda, \sum_{\lambda=1}^{q} tn_\lambda, \sum_{\lambda=1}^{q} fn_\lambda\right)$$

Abbildung 2.4: Formeln labelbasierte aggregierte Gütemaße

Quelle: [TKV10]

Ein Vergleich der Ergebnisse zwischen dem Micro- und dem Macro-Averaging Verfahren ist hierbei lohnend. Große Unterschiede zwischen beiden Werten kann beispielsweise auf unterschiedliche Performance bei selten bzw. häufigen Label hindeuten. Werden beispielsweise seltene Label sehr gut klassifiziert, häufige hingegen weniger gut, so ergibt sich ein hoher Macro-Average Wert und ein geringer Micro-Average Wert.

Für das Multi-Label-Learning existieren noch eine Reihe weiterer Gütemaße, die sich unter anderem mit dem Ranking von Label untereinander beschäftigen. Insbesondere ist hierbei interessant ob das normalerweise wahrscheinlichste Label einer Instanz zugewiesen wird und wie das Verhältnis der Label untereinander ist. Tabelle 2.4 gibt hierbei einen Überblick über die jeweiligen Maße.

2.4 Verwendete Frameworks

Um die verwendeten Algorithmen nicht selbst programmieren zu müssen wird auf Frameworks des Multi-Label-Learnings zurückgegriffen. Hierbei wird die Grundfunktionalität durch das auf Weka aufsetzende Mulan bereitgestellt. Die im Rahmen eines Programmierpraktikums an der TU Darmstadt entstandene Erweiterung Pocahontas zur Nutzung von Mulan auf großen Datenmengen wird aufgrund der Größe der bereitgestellten Daten verwendet. Folgend werden die Funktionen und Funktionsweisen beider Frameworks vorgestellt.

2.4.1 Mulan

MULAN[12] ist eine in Java geschrieben Bibliothek zur Verwendung bei Multi-Label-Learning Problemen[Tso+11]. Es basiert auf der weit verbreiteten Machine-Learning Suite Weka [WFH11][13] und verfolgt wie diese einen Open Source Ansatz. Mulan bietet verschiedene Algorithmen zur Nutzung auf Multi-Label-Daten. Unter anderem werden Ansätze zur Problemtransformation angeboten, sodass die Algorithmen der Weka Suite zur Problemlösung verwendet werden können. Angepasste Algorithmen die direkt auf Multi-Label-Daten verwendet werden können sind im Mulan Framework ebenfalls implementiert. Unter anderem handelt es sich um Decision Tree Lerner, Nearest Neighbor Lerner, Neuronale Netze und Support Vector Machines.
Die Verwendung des Mulan Frameworks ist nur durch Einbindung der bereitgestellten Java Bibliothek möglich. Eine Verwendung als eigenständiges Programm ist bisher nicht möglich. Um Mulan

[12] Vgl. http://mulan.sourceforge.net/
[13] Vgl. http://www.cs.waikato.ac.nz/ml/weka/

Name	Berechnung	Anmerkung
$1-Error$	$\frac{1}{m}\sum_{i=1}^{m}\delta(argminr_i(\lambda))$ $\delta(\lambda)=1$ falls $\lambda \notin Y_i; 0$ andernfalls	1-error misst, wie oft das top-gerankte Label sich nicht unter den relevanten Labeln der Instanz befindet
Coverage	$\frac{1}{m}\sum_{i=1}^{m}maxr_i(\lambda)-1$	Coverage misst, wie weit man in der Liste der zugewiesenen Label herabsteigen muss um alle relevanten Label einer Instanz zu erfassen
Ranking Loss	$\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\lvert Y_i\rvert\lvert\overline{Y_i}\rvert}\lvert\{(\lambda_a,\lambda_b): r_i(\lambda_b),(\lambda_a,\lambda_b)\in Y_i x\overline{Y_i}\rvert$	Ranking loss drückt die Anzahl an Vorkommnissen aus, bei denen irrelevante Label höher als relevante Label gerankt wurden
Average Precision	$\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\lvert Y_i\rvert}\sum_{\lambda\in Y_i}\frac{\lvert\{\lambda'\in Y_i: r_i(\lambda'\leq r_i(\lambda)\}\rvert}{r_i(\lambda)}$	Average precision misst den durchschnittlichen Anteil an Label, welche über einem bestimmten Label $\lambda \in Y_i$ gerankt sind, welche sich wirklich in Y_i befinden

Tabelle 2.4: Formeln rankingbasierter Gütemaße

Quelle: [TKV10]

innerhalb des Programmcodes zu verwenden ist es notwendig, die Eingabedaten in einem passenden Format bereitzustellen. Da Mulan auf Weka aufsetzt, wird auch in Mulan das ARFF-Format für Eingabedaten verwendet. Jedoch eignet sich das ARFF-Format bei Multi-Label Daten nur bedingt, da es keine explizite Kennzeichnung von Labels ermöglicht. Aus diesem Grund wird in Mulan die ARFF-Datei durch eine XML-Datei ergänzt, in welcher die Attribute der ARFF-Datei gekennzeichnet sind, welche als Label fungieren. Dadurch ist auch die Kennzeichnung von Hierarchien unter Labels, beispielsweise Unter- und Überordnungen möglich. Zur Veranschaulichung findet sich in Abbildung 2.5 eine Gegenüberstellung der beider Dateien anhand eines Beispieldatensatzes. Zu beachten ist, dass die Organisation der ARFF-Datei essentiell bei der Verwendung innerhalb Mulan ist. Die Attribute müssen geordnet aufgelistet werden, wobei zunächst die Feature- und anschließend die Label-Attribute gelistet werden müssen. Eine gemischte Organisation der Daten führt zu Fehlern bei der Verwendung von Mulan.

```
@relation musicout

@attribute Mean_Acc1298_Mean_Mem40_Centroid numeric
@attribute Mean_Acc1298_Mean_Mem40_Rolloff numeric
@attribute Mean_Acc1298_Mean_Mem40_Flux numeric
@attribute Mean_Acc1298_Mean_Mem40_MFCC_0 numeric
        [...]
@attribute amazed-suprised {1,0}
@attribute happy-pleased {1,0}
@attribute relaxing-calm {1,0}
@attribute quiet-still {1,0}
@attribute sad-lonely {1,0}
@attribute angry-aggresive {1,0}

@data
0.036299,0.064986,0.082104,-72.710462,7.92022,0.134279,2.
0.161218,0.46782,0.096983,-71.298043,1.176349,1.871744,1.
0.115987,0.336879,0.079068,-64.570939,2.339044,0.714859,1
0.086016,0.141845,0.081554,-81.141092,6.714252,-1.338896,
0.063232,0.140621,0.082097,-66.596131,5.594724,0.350716,1
0.026461,0.041099,0.08032,-75.802612,8.103934,-0.51184,1.
0.101976,0.249644,0.081015,-58.962537,3.317153,-0.499093,
0.040637,0.066504,0.083637,-77.425609,8.75572,1.021719,1.
0.102734,0.236176,0.080773,-61.893693,3.484507,-0.122183,
0.056279,0.099741,0.085464,-83.295694,7.970949,0.139845,2
0.038516,0.408709,0.094919,-76.108186,4.806829,1.437954,1
0.034295,0.063617,0.077881,-74.429498,7.270253,1.618832,1
0.06764,0.136969,0.079322,-73.747204,6.500311,-0.774962,1
0.082958,0.205946,0.076953,-59.883939,4.698083,-0.607546,
0.103782,0.262535,0.080116,-64.95944,3.00809,0.262919,1.5
```

(a) ARFF-Datei

```
<?xml version="1.0" encoding="utf-8"?>
<labels xmlns="http://mulan.sourceforge.net/labels">
<label name="amazed-suprised"></label>
<label name="happy-pleased"></label>
<label name="relaxing-calm"></label>
<label name="quiet-still"></label>
<label name="sad-lonely"></label>
<label name="angry-aggresive"></label>
</labels>
```

(b) XML-Datei

Abbildung 2.5: Korrespondierende ARFF-Datei und XML-Datei

2.4.2 Pocahontas

Eines der zentralen Probleme des MULAN Frameworks ist laut Tsoumakas sein hoher Ressourcenbedarf [Tso+11]. Hierdurch wird insbesondere die Verwendung des Mulan Frameworks beim Lernen großer

Datenmengen erschwert, da das Framework intern mit deep-copies arbeitet und daher einen hohen Bedarf an Arbeitsspeicher hat. Um diesen Nachteil des Mulan Frameworks auszugleichen wurde im Rahmen eines IT-Praktikums an der TU Darmstadt eine Erweiterung des Mulan Frameworks erstellt. Die Erweiterung hat das Ziel das Mulan Framework für die Verwendung großer Datenmengen nutzbar zu machen. Getauft wurde die Erweiterung auf den Namen Pocahontas. Eine erste Verwendung hat die Erweiterung in einem Forschungsprojekt zum Labeling von Daten der EUR-Lex Datenbank gefunden [LMF10].
Die Erweiterung Pocahontas verfolgt hierbei den Ansatz, die interne Datenhaltung des MULAN Frameworks anzupassen. Die einzelnen Algorithmen des Mulan Framewoks arbeiten unabhängig voneinander, weshalb Anpassungen für jeden zu verwendenden Algorithmus vorgenommen werden müssen. Im Pocahontas Framework wurden die Implementierungen des BinaryRelevance (welcher ein Multi-Label-Problem in binäre Teilprobleme zerlegt) und HOMER (welcher das Lernproblem in verschiedene hierarchische Partitionen zerlegt) angepasst. Da der BinaryRelevance Algorithmus für seine Klassifizierungen auf einen binären Klassifizierer zurückgreift wurde eine Implementierung der Support Vector Machine LibLinear als binärer Klassifizierer für die Verwendung großer Datenmengen angepasst.

Binary Relevance

BinaryRelevance (BR) transformiert ein Multi-Label-Probelm in mehrere binäre Teilprobleme. Die einzelnen Teilprobleme werden anschließend separat gelöst und die Ergebnisse letztendlich durch den BinaryRelevance wieder kombiniert. Die Zerlegung des Ursprungsproblems erfolgt hierbei anhand der verschiedenen Label. Für jedes einzelne Label wird ein separates Datenset erstellt was beispielhaft in Abbilung 2.6 dargestellt ist. Eine Instanz in der zerlegten Datenmenge wird hierbei positiv gelabelt, falls das jeweilige Label in der Instanz vorkommt und negativ, falls dies nicht der Fall ist. Eine ausführliche Beschreibung des BinaryRelevance findet sich bei Tsoumakas [TKV10].

Die Anpassungen der Pocahontas Erweiterungen an der Implementierung des BinaryRelevance zielen auf die interne Datenhaltung. Eine neu implementierte Klasse namens LargeScaleInstances sorgt dafür, dass Features und Labels getrennt verwaltet werden. Grundlage für diese Implementierung ist die Annahme, dass sich die Features während des Lernvorgangs nicht ändern. Dadurch wird die Verwendung des BinaryRelevance bei großen Datenmengen möglich durch die Vermeidung von deep-copies je Durchlauf. Auch die Möglichkeit zur Speicherung der berechneten Modelle auf permanente Speicher wurde implementiert. Durch die neue Datenstruktur kommt es zu erhöhtem Rechenbedarf durch Re-Organisation der verwendeten Datenstrukturen.

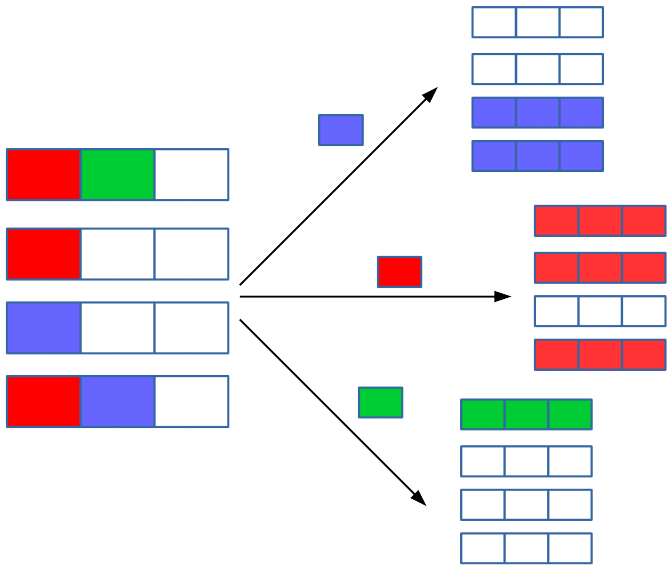


Abbildung 2.6: Beispielhafte Darstellung der Problemtransformation des BinaryRelevance Algorithmus

LibLinear

LibLinear[14] ist eine Implementierung einer Support Vector Machine (SVM) der Machine Learning Group der National Taiwan University. Hauptaspekt des LibLinear ist die Verwendung bei großen Datenmengen bei linearem Kernel[Fan+08]. Sie unterstützt unter anderem die Klassifizierer L2-regularized logistic regression, L2-loss und L1-loss linear. LibLinear kann über verschiedene Interfaces in unterschiedlichen Programmiersprachen verwendet werden. Das Interface für die Java Version greift auf LibLinear der Version 1.95 zurück[15]. Die aktuelle Implementierung des LibLinear ist Version 2.1 welche Optimierungen und Verbesserungen der Quellcode Abstraktion beinhaltet. Das Pocahontas Framework verwendet das Java Interface des LibLinear und somit die Version 1.95, welche im Rahmen des Pocahontas Framework angepasst wurde.

Die im Rahmen des Pocahontas Frameworks vorgenommenen Änderungen an LibLinear beziehen sich auf die Code Qualität sowie auf die internen Mechanismen. Neben einer Bereinigung des Codes wurden Anpassungen vorgenommen um keine unnötigen deep-copies während der Verwendung durchführen zu müssen. Hierzu wurde die interne Datenhaltung des LibLinear durch Erweiterung der internen Datenhaltungsobjekte angepasst. Die neue Datenstruktur des angepassten LibLinear wird an das ursprüngliche Framework über einen Wrapper angebunden.

[14] Vgl. `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`
[15] Vgl. `http://liblinear.bwaldvogel.de/`

3 Datenanalyse

Für die vorliegende Arbeit wurden seitens der ULB Darmstadt eine Textdatei bereitgestellt, welche unkomprimiert einen Umfang von ca. 2,9 GByte besitzt und aus über 90 Mio. Zeilen besteht. Die Datei stellt einen Abzug des internen bibliographischen Katalogs dar. In der Datei sind die einzelnen Werken, welche über die Website der ULB Darmstadt aufgefunden werden können gespeichert. Die zeilenbasierte Form der Daten eignet sich nur bedingt für Analysen. Um sinnvolle Analysen durchführen zu können, müssen diese daher zunächst in eine passende Datenrepräsentation überführt werden. Als Form wurde hierbei eine SQLite Datenbank gewählt, da diese ohne zusätzlichen Verwaltungsaufwand verwendet werden kann. Mit Hilfe dieser Datenbank lassen sich anschließend Analysen hinsichtlich möglicher Feature und Label durchführen. Die so identifizierten Feature-Label Kombinationen werden anschließend einer umfangreichen statistischen Analyse unterzogen um die Basis für das nachfolgende Multi-Label-Problem zu bilden.

3.1 Überführung in Datenbank Repräsentation

Eine Überführung der zeilenbasierten Repräsentation des Datensatzes in eine Datenbank Repräsentation erleichtert die Analyse der Daten immens. Nicht nur müssen keine umfangreichen Durchläufe der doch recht umfangreichen Datei gemacht werden, zum anderen lässt eine Datenbank Repräsentation durch die Kompatibilität mit SQL standardisierte Abfragen zu. Als Datenbankformat bietet sich SQLite[1] an. Es eignet sich besonders für die vorliegende Aufgabe, da es serverlos arbeitet und SQL unterstützt. Eine einfache Einbindung in bestehenden Code ist durch Anbindungen an alle gängigen Programmiersprachen möglich. Im vorliegenden Fall wird die SQLiteDB über einen JDBC-Connector[2] eingebunden. Die Verwendung von SQLite hat noch einen entscheidenden weiteren Vorteil. Die Datenbank lässt sich mit Hilfe von Drittprogrammen wie beispielsweise SQLiteman auslesen und auf ihr SQL Befehle über eine GUI ausführen[3]

Die Überführung der zeilenbasierten Datei in die SQLite Datenbank wird über ein in Java geschriebenes Skript bewerkstelligt. Folgende Auflistung macht die einzelnen Schritte deutlich:

[1] Vgl. https://www.sqlite.org/

[2] Vgl. https://bitbucket.org/xerial/sqlite-jdbc

[3] Vgl. https://sourceforge.net/projects/sqliteman/

1. Analyse der zeilenbasierten Daten und Extraktion vorkommender Bezeichner, Occurences und Subfelder

2. Aufbau der SQLite Datenbank basierend auf den zuvor ermittelten Bestandteilen. Je PICA+ Bezeichner wird eine Tabelle angelegt. Die Spalten der jeweiligen Tabelle enthalten neben der ID und den Rohdaten einer jeden Zeile für jedes Unterfeld eine Spalte

3. Überführung der Daten in die neu erstellte SQLite Datenbank

Die SQLite Datenbank enthält somit für jeden PICA+ Bezeichner eine Tabelle, was zu insgesamt 259 Tabellen führt. Zur Erleichterung der Referenzen wird zusätzlich eine Tabelle mit allen vorkommenden IDs gehalten. Jeder Eintrag in der Datenbank bildet zusätzlich die Rohdaten einer jeden Zeile ab. Hierdurch verdoppelt sich der benötigte Speicherbedarf im Vergleich zur Ursprungsdatei auf ca. 6,5GB. Eine Beispielrepräsentation einer Tabelle der SQLite Datenbank findet sich in Abbildung 3.1. Die einzelnen Spalten der Tabelle sind hierbei die ID (welche jedes Werk der Datenbank eindeutig identifiziert), die Rohdaten in der Spalte VAL und Occurences in der Spalte ADDON. Der dargestellte beispielhafte Bezeichner **001A** besitzt ein Unterfeld, welches durch $0 eingeleitet wird. Anhand der farblichen Hervorhebungen in den Rohdaten lässt sich sehen, wie die jeweilige Repräsentation zu Stande kommt.

	ID	VAL	ADDON	x0
1	564	$0017:28-11-88		0017:28-11-88
2	1103	$06000:10-02-88		6000:10-02-88
3	3509	$00017:23-02-89		0017:23-02-89
4	4871	$00004:23-02-89		0004:23-02-89
5	7425	$00017:17-03-89		0017:17-03-89
6	10090	$00030:23-02-89		0030:23-02-89
7	10759	$00017:28-11-88		0017:28-11-88

Abbildung 3.1: Beispiel eines Datensatzes in der SQLite Datenbank am Bezeichner 001A

Die besonderen Bezeichnungen innerhalb der Datenbank für Tabellen und Felder kommen aufgrund von Beschränkungen des SQL Standards zustande. Für Tabellen- und Feldnamen sind zum einen eine Reihe an Sonderzeichen nicht erlaubt (weshalb eine Bezeichnung der Unterfelder mit $ nicht möglich ist) und zum anderen keine Ziffern als erstes Zeichen erlaubt. Aus diesem Grund werden Tabellen und Felder mit einem vorangehenden **x** gekennzeichnet. Innerhalb von PICA+ existieren auch Bezeichner, welche case-sensitive sind. Diese werden innerhalb des jeweiligen Feldes durch das Suffix **u** gekennzeichnet.

Die Datenbank enthält insgesamt 260 Tabellen, welche aus den 259 Tabellen mit PICA+ Bezeichner und der Tabelle für die Vorhaltung aller IDs besteht. Der offiziellen Dokumentation des PICA+ Formates für das HeBIS umfasst 503 Kategorien [4]. Es wird somit knapp die Hälfte aller möglichen Bezeichner verwendet.

3.2 Auswahl der Features und Labels

Es befinden sich insgesamt 2.083.426 eindeutige IDs in der Datenbank. Da jede ID ein Werk in der ULB Darmstadt repräsentiert, existieren somit die gleiche Menge an Werken. Diese Zahl kann als Maßstab herangezogen werden um vielfältige Analysen auf den Daten durchzuführen. Die Eignung verschiedener Felder des PICA+ Formats als Feature bzw. Label muss anhand statistischer und inhaltlicher Merkmale überprüft werden. Als Label eignen sich hierbei jene Felder, welche in der Dokumentation des PICA+ Formats als Schlagworte ausgewiesen sind. Als Features eignen sich potentiell alle Felder mit Text, wobei hier anhand inhaltlicher Kriterien eine Auswahl getroffen werden muss. Um die Entscheidung für oder gegen die Verwendung eines Feldes transparent zu machen, werden umfangreiche Statistiken zu den jeweiligen Feldern dargestellt.

3.2.1 Analyse und Auswahl der Labels

Die Auswahl von Feldern, welche als Label dienen sollen, erfolgt zweistufig. In einem ersten Schritt müssen zunächst Kandidaten für die Nutzung als Label identifiziert werden. Diese Kandidaten werden anschließend auf ihre Eignung für das vorliegende Problem geprüft und tiefer gehend analysiert. Zur Identifikation geeigneter Label wird auf die Dokumentation des HeBIS zurückgegriffen. Die Übersicht des PICA Formats auf der Dokumentationswebsite[5] beinhaltet alle in Frage kommenden Bezeichner. Um die Suche nach geeigneten Labels sinnvoll einzuschränken wird eine Suche nach dem Begriff "Schlagwort" durchgeführt. Die so identifizierten 21 möglichen Schlagwort Felder finden sich mit einer kurzen Beschreibung des jeweiligen Inhalts in Tabelle 3.1. Da nicht alle dieser Felder in den realen Daten der ULB Darmstadt vorkommen, findet sich zusätzlich eine Analyse zu deren Vorkommen. Durch die Umstellung innerhalb der ULB Darmstadt auf den RDA Standard ist es zu Anpassungen gekommen. Innerhalb der Dokumentation des HeBIS werden diese als Altdaten bezeichnet und so auch in der Tabelle dargestellt. Durch die Umstellung auf RDA haben auch einige Bezeichner ihren Inhalt geändert, sodass

[4] Vgl. http://www.hebis.de/de/1publikationen/arbeitsmaterialien/hebis-handbuch/kategorien/kategorien-rda_index_druck.php?cat=&order=PicaPlus

[5] Vgl. http://www.hebis.de/de/1publikationen/arbeitsmaterialien/hebis-handbuch/kategorien/kategorien-rda_index_druck.php?cat=&order=PicaPlus

dieser nicht mehr Schlagworte enthält. Dies ist bei den Bezeichnern *039C*, *039D*, *039E* und *039F* der Fall, welche deshalb nicht weiter betrachtet werden.

Bezeichner	Kennzeichnung	Verwendung ULB	Beschreibung
035G	Altdaten	nein	Schlagwörter der Sammelschwerpunkte
041A/0X-1X	-	ja	RSWK-Schlagwort
041A	Altdaten	ja	Sachbegriff - bevorzugte Benennung
041B	Altdaten	nein	1. Unterschlagwort
041D	Altdaten	nein	Alternativform zum Hauptschlagwort
041E/01-05	Altdaten	nein	1.-5. Unterschlagwort in Alternativform
041K	Altdaten	ja	2. Unterschlagwort
041L	Altdaten	nein	3. Unterschlagwort
041M	Altdaten	nein	4. Unterschlagwort
041N	Altdaten	nein	5. Unterschlagwort
044F	Altdaten	ja	DB-Schlagwörter bis 1986
044G	Altdaten	ja	Schlagwort zu einer Notation
044K	-	ja	Einzelschlagwort
044L	Altdaten	ja	Schlagwörter der Hessischen Bibliographie
044M	Altdaten	ja	SWD-Schlagwörter der BDSL
044O	Altdaten	ja	Fremddaten-Schlagwörter
044P	Altdaten	ja	Schlagwörter (von Autoren selbst vergeben)
044Q	Altdaten	ja	Schlagwörter für Schulprogramme
045R	-	ja	Zeitschlagwort
144Z/XY	Altdaten	ja	Lokale Schlagwörter für die ganze ILN
244Z/XY	Altdaten	ja	Lokale Schlagwörter (exemplarbezogen)

Tabelle 3.1: Bezeichner mit Schlagwortbezug

Von den identifizierten 21 Bezeichner finden sich nur 15 Bezeichner im vorliegenden Datensatz der ULB wieder. Ein besonderer Fall ist das Feld 041A, welches ohne eine Occurence kein Schlagwort darstellt, jedoch mit Occurence als RSWK-Schlagwort firmiert. Folgend wird, wenn vom Bezeichner 041A die Rede ist, das RSWK-Schlagwort[6] gemeint.

Innerhalb des PICA Formats kann eine Zeile mehrere Teilinformationen enthalten. Bei Schlagworten ist

[6] RSWK ist das Regelwerk Schlagwortkatalog. Das Regelwerk der Deutschen Nationalbibliothek findet sich unter `http://files.dnb.de/pdf/rswk_gesamtausgabe.pdf`

Bezeichner	Unterfelder	Hauptunterfeld
041A(w/o Occ.)	$8, $x, $9, $z, $a, $b, $c, $d, $f, $g, $h, $i, $l, $m, $n, $p, $r, $s, $P	$8
041A/0X-1X	$8, $x, $9, $z, $a, $b, $c, $d, $f, $g, $h, $i, $l, $m, $n, $p, $r, $s, $P	$8
041K	$0, $a	$a
044F	$a, $b, $c	$a
044G	$0, $a	$a
044K	$8, $x, $9, $z, $a, $b, $c, $d, $f, $g, $l, $m, $n, $p, $0, $s, $P	$8
044L	$8, $9, $x, $l, $I, $n, $a, $b, $c, $d, $g, $P	$8
044M	$p, $t	$p
044O	$a, $h, $2	$a
044P	$a, $e	$a
044Q	$8, $9, $k, $s, $S, $g	$s
045R	$a, $c	$a
144Z/XY	$a	$a
244Z/XY	$8, $9, $l, $n, $x, $a, $b, $c, $P, $g	$a

Tabelle 3.2: Unterfelder der Bezeichner für Schlagworte

dies besonders dahingehend interessant, da eventuelle Hierarchien innerhalb eines Feldes abgebildet werden. Um die Komplexität der Untersuchung einzuschränken wird nur das Hauptfeld des jeweiligen Schlagwortes zur Bewertung herangezogen. In Tabelle 3.2 findet sich ein Überblick, welche der Unterfelder eines jeden Bezeichners als Hauptunterfeld und somit als Schlagwort identifiziert wurden. Die Identifikation des Hauptunterfelds wurde hierbei manuell anhand der Vorkommnisse an schlagwortartigen Daten durchgeführt. Denkbar wäre, die zusätzlichen Felder eines jeden Datensatzes als zusätzliche Schlagworte hinzuzufügen. Um die Ergebnisse einfacher interpretierbar zu machen wird auf diese zusätzliche Komplexität jedoch verzichtet.

Die identifizierten Kandidaten zur Verwendung als Label müssen für eine inhaltliche Eignung einer statistischen Analyse unterzogen werden. Hierbei wird unter anderem ermittelt, bei wie vielen Instanzen der Datenbank ein Eintrag des jeweiligen Label vorhanden ist. Bezug wird auf das Hauptfeld des jeweiligen Schlagworts genommen. Zur statistischen Analyse werden Maße wie Median, arithmetisches Mittel, sowie minimale und maximale Label Anzahl pro Instanz ermittelt. Die Ergebnisse der statistischen Ana-

lyse findet sich in Tabelle 3.3.
Wie die Tabelle 3.3 zeigt kommen für eine nähere Betrachtung die beiden Label Kandidaten 041A/XX und 044K in Betracht. Sie verfügen über ein hohes Vorkommen innerhalb der Daten und durch ihr hohes arithmetischen Mittels kombiniert mit einem Median größer 1 kann davon ausgegangen werden, dass diese zu interessanten Ergebnissen hinsichtlich Multi-Label-Problemen führen. Das Feld Spezifizierung des Hauptschlagworts ist eine Auflistung jener Felder, welche zusätzlich zum Hauptschlagwort gesetzt sind. Dies können Informationen unterschiedlichster Art sein, darunter auch Werte, welche keine schlagwortartigen Daten darstellen. Eine exakte Analyse der einzelnen Felder ist aufgrund der mangelhaften Dokumentation hinsichtlich PICA+ Subfelder des HeBIS nicht möglich.

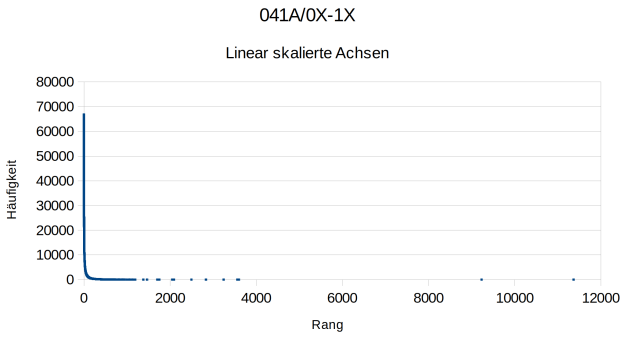
Die identifizierten Label Kandidaten *044K* und *041A/XX* sollen folgend tiefer gehend untersucht werden. Interessant sind die beiden Label Kandidaten insbesondere inhaltlich. Bei den Label des *041A/XX* handelt es sich um Schlagworte die einer Taxonomie folgen. Gesetzt wurden diese Schlagworte nach den Regeln Schlagwort Katalog (RSWK). Bei den Labeln des Bezeichners *044K* ist keine klare Taxonomie ersichtlich. Sie werden als Einzelschlagworte bezeichnet und werden vermutlich abhängig von der Einschätzung des jeweiligen Bibliothekars vergeben.
Neben der statistischen Analyse ist auch eine inhaltliche Analyse der identifizierten Label Quellen interessant. Eine Auflistung der am häufigsten vergebenen Schlagworte anhand der TOP-10 beider Label findet sich in Tabelle 3.4. Interessant ist, dass sich beide Bezeichner inhaltlich hinsichtlich ihrer TOP-10 Schlagworte ähneln. Was herausstricht ist, dass beim Bezeichner 041A öfters die Veröffentlichungsform (Zeitschrift und Online-Publikation) gesetzt wird, der Bezeichner 044K einen Fokus auf spezielle Themengebiete wie Juden und Judentum besitzt. Vermutlich ist dies auf die Taxonomie bzw. auf den jeweiligen Interessenschwerpunkt des Bibliothekars zurückzuführen. Die Pareto Distribution (auch bekannt als Zipf's Law) beschäftigt sich mit der Verteilung von Daten bei Betrachtung großer Datenmengen [New05]. Bei der Betrachtung von Texten natürlicher Sprache stellt sich häufig eine Pareto Distribution bei der Verteilung von Worten ein, da in natürlicher Sprache meist wenige Worte sehr häufig und viele Worte sehr selten verwendet werden. Auch bei der Verwendung von Label kann es zu einer Pareto Verteilung kommen. Sichtbar wird die Pareto Verteilung indem die lineare Darstellung von Häufigkeit und Rang an beiden Achsen logarithmisch gestaucht werden. Ergibt sich hierbei eine annähernd lineare Darstellung so handelt es sich mit hoher Wahrscheinlichkeit um eine Pareto Distribution. Abbildung 3.2 zeigt die Verteilung der Label abhängig von deren Rang hinsichtlich Häufigkeit. Wie zu sehen ergibt sich bei logarithmischer Stauchung eine annähernd lineare Darstellung, weshalb davon ausgegangen werden kann, dass beide Label einer Pareto Distribution folgen.

Bezeichner	Anzahl Einträge insgesamt in DB	Unterschiedliche IDs	Unterschiedliche Schlagwörter	Zus. Inform. Hauptschlag-wort	Median pro ID	arithm. Mittel pro ID	min. Label pro ID	max. Label pro ID
041A/XX	724.880	288.306	66.842	26.331	2	2,51	1	23
041K	45.150	15.782	12.275	0	3	2,86	1	21
044F	110.605	54.736	26.867	0	2	2,02	1	13
044G	613	605	56	0	1	1,01	1	2
044K	902.265	342.445	83.161	46.585	2	2,63	1	43
044L	53.042	19.600	7.431	2.922	2	2,71	1	48
044M	182	150	177	68	1	1,21	1	4
044O	86.060	41.924	39.598	86.060	2	2,05	1	12
044P	2.759	744	2.326	30	4	3,71	1	11
044Q	124	112	24	0	1	1,11	1	2
045R	106.771	101.340	21.316	2	1	1,05	1	8
144Z/XY	98.523	36.086	32.862	0	2	2,73	1	21
244Z/XY	38.274	20.943	9.822	0	1	1,83	1	16

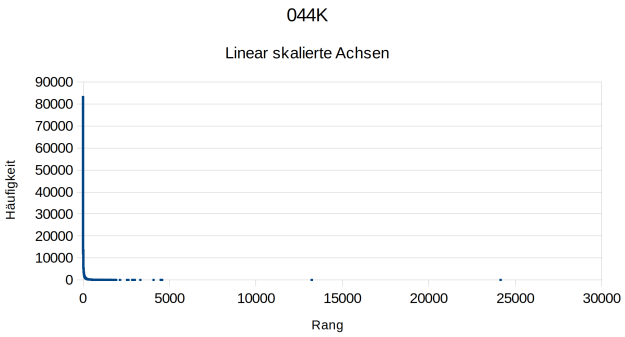
Tabelle 3.3: Statistische Analyse der Hauptfelder von Schlagwort-Bezeichnern

	041A/0X-1X		044K	
Rang	Label	Häufigkeit	Label	Häufigkeit
1	Zeitschrift	11366	Deutschland	24149
2	Deutschland	9232	USA	4577
3	Literatur	3596	Deutsch	4517
4	Architektur	3242	Architektur	4501
5	Online-Publikation	2834	Literatur	4080
6	Philosophie	2496	Juden	3315
7	Kultur	2093	Judentum	2988
8	Rezeption	2049	Philosophie	2930
9	Recht	1753	Großbritannien	2871
10	Kunst	1703	Europa	2852

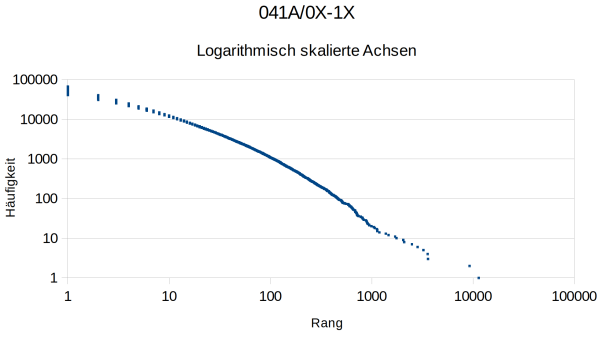
Tabelle 3.4: TOP-10 Label der Bezeichner 041A/0X-1X und 044K



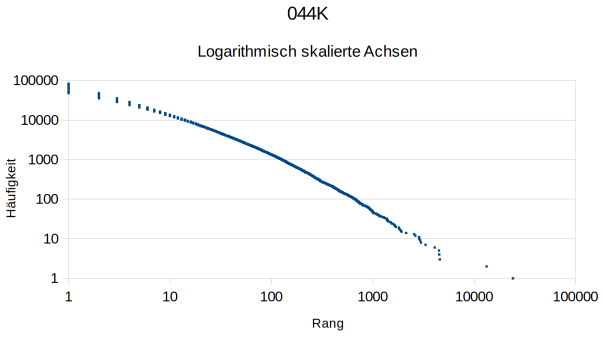
(a) 041A (RSWK-Schlagwort)



(b) 044K (Einzelschlagwort)



(c) 041A (RSWK-Schlagwort)



(d) 044K (Einzelschlagwort)

Abbildung 3.2: Doppelt-Logarithmisch skalierte Graphik von Häufigkeit und Rang

3.2.2 Analyse und Auswahl der Features

Die Ermittlung geeigneter Features ist im Gegensatz zur Ermittlung geeigneter Label breiter gestreut. Theoretisch eignen sich alle Felder, welche Text enthalten als Feature Kandidat. Auch können die Features der einzelnen Felder kombiniert werden um so eine größere Menge an Features je Instanz zu schaffen. Wichtig im vorliegenden Fall ist, dass die Auswahl an Features eine große Überschneidung mit den ausgewählten Labeln besitzen um eine möglichst breite Basis für das Multi-Label-Problem zu schaffen. Um die Auswahl an Features objektiv zu gestalten, wird sie auf PICA3 Bezeichner reduziert werden, welche potentiell Text enthalten. Die PICA3 Bereiche **4XXX** (Titelbeschreibung incl. Fußnoten), **5XXX** (Sacherschließung) und **6XXX** (lokale Sacherschließung) bieten sich hierfür an. Die korrespondierenden PICA+ Bezeichner können im Anhang in Tabelle 7.1 entnommen werden.
Den drei genannten Bereichen des PICA3 Formats entsprechen nach der Dokumentation des HeBIS bis zu 130 PICA+ Bezeichner. Auswahlkriterium bei Durchsicht der einzelnen Bezeichner ist, dass diese Text enthalten, welcher sich potentiell zur Verwendung im Multi-Label-Learning als Feature eignet. Aus der Dokumentation lässt sich solch eine Eignung nur schwer ableiten. Aus diesem Grund müssen alle Einträge der Datenbank manuell gesichtet und bewertet werden. Die Durchsicht der Felder der Datenbank hat zwei Gruppen von potentiellen Features hervorgebracht. Zum einen sind dies jene Felder, die die oben genannten Kriterien an Features erfüllen, d.h. subjektiv Text in ausreichender Menge beinhalten. Die relevanten Bezeichner finden sich in Tabelle 3.5. Jene Bezeichner, welche subjektiv nicht genügend Text beinhalten, können potentiell ergänzend zu den identifizierten Feldern verwendet werden.

Für die identifizierten Feature Kandidaten können eine Reihe statistischer Erhebungen durchgeführt werden. Interessant ist hierbei zum einen bei wie vielen Instanzen die Features vorkommen. Eine hohe Anzahl an Vorkommnissen ermöglicht bei einer praktischen Anwendung die Verschlagwortung vieler Instanzen. Zusätzlich ist interessant, bei wie vielen Instanzen es Überschneidungen hinsichtlich der identifizierten Label aus den Bezeichnern 041A und 044K gibt.

Interessant als Feature Kandidaten sind die Bezeichner 021A und 047I. Beide Bezeichner sind hierbei auf ihre Art interessant für eine Untersuchung. Während es sich bei 021A um den Haupttitel eines Werkes handelt, also meist aus wenigen Worten besteht, handelt es sich bei 047I um eine inhaltliche Zusammenfassung des Werkes mit viel Text. Bei so gut wie jedem Werk ist der Haupttitel gesetzt, jedoch nur bei einem Bruchteil der Werke die inhaltliche Zusammenfassung.

PICA3	PICA+	Beschreibung	Haupt-unterfeld	Einträge insge-samt	Über-schneidung 044K IDs	Über-schneidung 041A/XX IDs
4000	021A	Haupttitel, Titelzusätze, Paralleltitel, Verant-wortlichkeitsangabe	$a	2.036.194	338.988	373.733
4005	021C	Titel von Unterreihen fortlaufender Ressourcen	$a	14.069	13	2.807
4010	021M	Haupttitel, Titelzusätze, Paralleltitel und Ver-antwortlichkeitsangabe von Teilen bei Zusam-menstellungen ohne übergeordneten Titel	$a	5.779	1.733	862
4011	021N	Gemeinsame Titelzusätze und Verantwortlich-keitsangaben bei Zusammenstellungen ohne übergeordneten Titel	$a	1.706	479	201
4207	047I	Inhaltliche Zusammenfassung	$a	76.184	2.894	2.532

Tabelle 3.5: Kandidaten-Bezeichner als Haupt-Features

Die zweite Gruppe an Feature Lieferanten, welche sich nicht als primärer Lieferant für Features eignen, sei es durch die geringe Anzahl an Features oder die Art der Informationen, sind in Tabelle 3.6 aufgelistet. Hierbei ist eine Trennung hinsichtlich der PICA3 Kategorien vorgenommen worden. Denkbar ist die Verwendung dieser Feature Quellen ergänzend zu den bereits identifizierten Feature Quellen um den Feature Raum und die Anzahl an Features zu erhöhen und somit die vorhandenen Daten bestmöglich zu Nutzen.

PICA3-Bereich	PICA+ Bezeichner
YYYY	021C, 021M, 021N
0XXX	022A, 028A, 044F
3XXX	022S, 025@, 028B, 028C, 028D, 028E, 028F, 029A, 029E, 029F, 029G, 055A, 055B
4XXX	021B, 021M, 036A, 036B, 036C, 036D, 036E, 036F, 036G, 036L, 036M, 037B, 037D, 039B, 039C, 039D, 039E, 039I, 039S, 039T, 039U, 039X, 039Z, 046A, 046C, 046D, 047B, 047C, 060B, 060C
5XXX	041K, 044A, 044G, 044L, 044M, 044N, 044O, 044P, 044Q, 045R, 045W, 045X

Tabelle 3.6: Kandidaten-Bezeichner als Ergänzungs-Features

3.3 Analyse der identifizierten Feature-Label Kombinationen

Es konnten insgesamt 2 Label Quellen und 2 Feature Quellen identifiziert werden. Durch Kombination der verschiedenen Quellen lassen sich interessante Experimente konstruieren. In Tabelle 3.7 ist eine statistische Übersicht für die verschiedenen Feature-Label Kombinationen dargestellt.

Nahezu jede Instanz der Datenbank besitzt einen Haupttitel (021A). Hierdurch existieren auch große Überschneidungen mit den Label Quellen. Die Schnittmenge der inhaltlichen Zusammenfassung (047I) mit den beiden Label Quellen ist weit geringer. Auch bei der Quote der Verschlagwortung unterscheiden sich die beiden Feature Quellen. Beträgt sie bei der inhaltlichen Zusammenfassung ca. 4% so liegt sie bei gesetztem Haupttitel bei knapp 12%. Es besteht somit für beide Gruppen an Features das Potential die Verschlagwortung durch automatisierte Methoden zu verbessern. Die Schnittmenge an gelabelten Instanzen, welche sowohl Haupttitel als auch inhaltliche Zusammenfassung besitzen ist zwar stark verringert, erlaubt jedoch vergleichende repräsentative Experimente anhand verschiedener Feature-Label

Feature IDs	Label IDs	Schlagwort Einträge insgesamt	Unterschiedliche IDs (Werke)	Unterschiedliche Schlagworte	Median	arithm. Mittel	min. Label pro ID	max. Label pro ID
021A	041A/XX	724.525	288.085	66.834	1	2,51	1	23
021A	044K	896.771	339.586	83.090	1	2,64	1	11
047I	041A/XX	6.916	2.340	2.220	1	2,96	1	43
047I	044K	9914	2.894	4.244	1	3,43	1	30
$021A \cap 047I$	041A/XX	6.916	2.340	2220	1	2,96	1	11
$021A \cap 047I$	044K	9.906	2.890	4242	1	3,43	1	30

Tabelle 3.7: Statistiken Cross-Over Features/Labels

Kombinationen.

Aufbauend auf der so ermittelten repräsentativen Menge wird ein zweistufiges Experiment angestrebt. In einem ersten Schritt soll die repräsentative Menge an Instanzen, also jene Instanzen mit Haupttitel (021A) und inhaltlicher Zusammenfassung (047I) verwendet werden. Es soll untersucht werden, inwiefern sich die Güte der Lerner mit steigender Feature Anzahl entwickelt. Die drei verschiedenen zu bildenden Feature Mengen sind hierbei Hauptitel (021A), inhaltliche Zusammenfassung (047I) und die Kombination aus beiden (021A+047I). Hierdurch kann abgeschätzt werden, welche Ergebnisse bei Verwendung der verschiedenen Feature erwartet werden kann.
In einem zweiten Schritt soll, durch die große Verbreitung des Haupttitels, eine Optimierung vorgenommen werden. Die im ersten Schritt verwendete Menge an repräsentativen Instanzen kann durch eine Erhöhung der Anzahl an verwendeten Instanzen angereichert werden. Hierdurch kann abgeschätzt werden, wie sich die Güte der Klassifizierer durch die Erhöhung an Instanzen verhält. Ziel beider Experimente ist eine Einschätzung hinsichtlich der Eignung der verschiedenen Feature-Label Kombinationen für den praktischen Anwendungsfall.

4 Datentransformation

Die ausgewählten Feature-Label Kombinationen müssen für das Multi-Label-Learning in ein geeignetes Format gebracht werden. Diese Transformation spielt sich zum einen auf der inhaltlichen Ebene durch Veränderung der Features und Labels und auf der technische Ebene ab. Insbesondere die inhaltliche Transformation bietet eine Reihe von Möglichkeiten die Güte der verwendeten Daten zu verbessern. Es können Features entfernt, abgeändert oder hinzugefügt werden. Die Anzahl an Label kann ebenfalls, beispielsweise durch die Anwendung von Filtern, verändert werden.

4.1 Inhaltliche Transformation

Die ausgewählten Feature Quellen Haupttitel (021A) und inhaltliche Zusammenfassung (047I) liegen in natürlicher Sprache vor. Aus diesem Grund bieten sich Techniken zur Reduktion des Feature Raums an. Bekannte Techniken sind unter anderem das Stop-Word Filtering und das Stemming. Beim Stop-Word Filtering werden funktionale Worte wie Präpositionen, Artikel und Konjunktionen durch Abgleich mit Wortlisten entfernt. Beim Stemming werden Worte eines Textes auf seinen Wortstamm reduziert und somit ähnliche Worte (welche beispielsweise durch Konjugation entstanden sind) zusammengefasst. Während die Stop-Word Filterung meist problemlos angewandt werden kann ist die Anwendung des Stemming umstritten[Seb02].

Nach Durchsicht der Features ist es möglich, dass diese sowohl in deutscher als auch in englischer Sprachen verfasst sind und dies teilweise im gleichen Unterfeld. Die Trennung zwischen den einzelnen Sprachen ist hierbei uneinheitlich, sodass es schwierig ist diese automatisch zu trennen. Aus diesem Grund und da die Qualität des Stemming nicht eingeschätzt werden kann, wird auf Stemming verzichtet. Die Stop-Word Filterung hingegen findet mit Stop-Word Listen in Deutsch[1] und Englisch[2] statt. Durch die wenigen Überschneidungen der deutschen und englischen Liste führt die Stop-Word Filterung nicht zu einer starken Verfälschung der Ergebnisse. Um die Features zu vereinheitlichen werden diese normalisiert. Hierbei wird eine generelle Kleinschreibung der Features durch Transformation erzwungen, um die auch im Englischen übliche Großschreibung am Satzbeginn nicht zur Erhöhung der Feature Anzahl führen zu lassen. Bei Features mit einer Läge von weniger als 3 Chars wird angenommen, dass diese nur eine geringe Aussagekraft besitzen. Daher werden diese ebenso wie Sonderzeichen aus dem Feature

[1] Deutsche Liste: `http://www.ranks.nl/stopwords/german`

[2] Englische Liste: `http://www.ranks.nl/stopwords`

Raum entfernt. Obwohl die Verwendung des Haupttitels als Feature Quelle den Nachteil eines stark begrenzten Umfangs an Features pro Eintrag besitzt, soll in der vorliegenden Arbeit auf eine Erweiterung des Feature Raums verzichtet werden.

4.2 Überführung in geeignetes Datenformat

Für die Verwendung des Mulan Frameworks und seiner Erweiterung Pocahontas werden sowohl eine ARFF-Datei mit Definition aller Attribute und Instanzen als auch eine XML-Datei mit allen vorkommenden Labels benötigt. Wichtig für ARFF-Datei ist hierbei, dass zuerst die Feature Attribute aufgelistet werden müssen bevor die Label Attribute folgen. Eine inverse oder gemischte Sortierung der verschiedenen Attribute führt zu Laufzeitfehlern bei Verwendung von Mulan und Pocahontas. Für eine bessere Unterscheidbarkeit von Features und Labels (und um zu ermöglichen, dass identische Features und Labels existieren können) werden innerhalb der ARFF- und XML-Datei die Label durch das Präfix "LABEL_" gekennzeichnet.

Die Instanzen der ARFF-Datei werden durch eine Sparse-Repräsentation angegeben. Hierbei werden jene Attribute, welche in einer Instanz nicht vorkommen, nicht durch Nullen gekennzeichnet aufgeführt. Nachteil ist, dass die einzelnen Attribute nun durch einen Index gekennzeichnet werden müssen. Insbesondere bei einer hohen Anzahl an Features und Labels hat die Sparse Repräsentation durch seinen geringeren Speicherbedarf große Vorteile. Sowohl Features als auch Labels werden binär codiert, d.h. sie können entweder den Wert 1 annehmen, falls sie bei der Instanz auftreten oder 0 falls dies nicht der Fall ist. Durch die Sparse Repräsentation werden die 0 Werte jedoch nicht explizit dargestellt. Auf eine Anpassung beispielsweise durch die Term Frequency - Inverse Document Frequency (TF-IDF) wird in der vorliegenden Arbeit verzichtet.

4.3 Technische Einbindung und Multi-Threading Anpassung

Die Experimente werden durch die Einbindung des Mulan Frameworks und der Pocahontas Erweiterung durchgeführt. Ein minimaler Beispielcode zur Einbindung der jeweiligen Algorithmen findet sich in Grafik 4.1. Die Verwendung vorgefertigter Frameworks hat den Vorteil immenser Zeit- und Aufwandsersparnis. Durch sie muss ein Großteil an Funktionalität nicht selbst programmiert werden.

Nachteil der Verwendung externer Frameworks ist jedoch, dass diese eventuell nicht exakt den eigenen Anforderungen entsprechen oder dass unvorhergesehene Fehler bei der Verwendung auftreten, welche nicht dokumentiert sind. Während mehrerer Testläufe mit dem Pocahontas Framework wurde festgestellt, dass der Bedarf an Arbeitsspeicher weit über den prognostizierten Bedarf liegt, der laut Pocahontas

```java
public static void main(String[] args) {

        String arffFilename = "emotions.arff";
        String xmlFilename = "emotions.xml";

        MultiLabelInstances dataset = new MultiLabelInstances(arffFilename, xmlFilename);

        LargeScaleBinaryRelevance learner1 = new LargeScaleBinaryRelevance(new LSLibLINEAR());
        LargeScaleHOMER learner2 = new LargeScaleHOMER();

        Evaluator eval = new Evaluator();
        MultipleEvaluation results;

        int numFolds = 10;
        results = eval.crossValidate(learner1, dataset, numFolds);
        System.out.println(results);

        int numFolds = 10;
        results = eval.crossValidate(learner2, dataset, numFolds);
        System.out.println(results);
}
```

Abbildung 4.1: Beispielcode zur Einbindung von HOMER bzw. BinaryRelevance+LL

Dokumentation eigentlich gebraucht werden sollte. Besonders bei Einsatz der Cross-Validation kann es zu Out-of-Memory Fehlern kommen. Ein weiterer auftretender Fehler ist bei vollem Arbeitsspeicher, dass der Garbage Collector der Java VM zu viel Zeit zum Aufräumen verwaister Objekte benötigt und dadurch ebenfalls das Programm zu einem Abbruch kommt. Der Overhead der Cross-Validation an Arbeitsspeicher wurde in der vorliegenden Arbeit nicht tiefer gehend analysiert. Als Workaround wird der Garbage Collector nach jedem Fold der Cross-Validation aufgerufen um zumindest den Programmabbruch durch zu lange Garbage Collection zu verhindern.

Ein weiterer Nachteil des Einsatzes von externen Frameworks ist, dass diese nur bedingt hinsichtlich der Rechenauslastung optimiert sind. Das Pocahontas Framework beispielsweise wurde nicht hinsichtlich Multi-Threading optimiert. Dadurch kommt es zu sehr langen Laufzeiten, da die meist vorhandenen Multi-Core Ressourcen nicht optimal ausgelastet werden.
Um diesem Problem zu begegnen wurde der Code des BinaryRelevance Algorithmus innerhalb der Pocahontas Erweiterung für Multi-Threading optimiert. Insbesondere die geringe Komplexität des BinaryRelevance Algorithmus und dessen lineare Skalierung eignen sich hierbei dafür [Rea+11]. Innerhalb der Implementierung des BinaryRelevance in der Pocahontas Erweiterung wurden an zwei Stellen Änderungen vorgenommen, dem Trainings- und dem Klassifizierungsvorgang. Das Training der Instanzen kann parallelisiert werden, da für jedes Label ein Klassifizierer gebildet wird. Die Bildung der Klassifizierer ist hierbei unabhängig voneinander. Bei der Implementierung fiel auf, dass die Ergebnisse des auf Multi-Threading optimierten BinaryRelevance von denen ohne Optimierung abweichen. Grund hierfür war der globale Zugriff auf einen Zufallszahlengenerator in der Implementierung des verwendeten LibLinear. Hierduch kam es durch jeden neu erstellten Thread zu einer Zurücksetzung des Zufallszahlengenerators. Durch die nun abweichenden Zufallszahlen stimmten in Folge auch die Ergebnisse nicht mehr überein. Für den konkret vorliegenden Fall wurde die Implementierung des LibLinear angepasst und dem Zufallszahlengenerator die globale Verfügbarkeit entzogen. Es ist jedoch nicht auszuschließen,

dass innerhalb des Code des LibLinear weitere Zugriffe auf die globale Variable erfolgen. Eine Überarbeitung der LibLinear Bibliothek hinsichtlich Multi-Threading Eignung oder die Verwendung der Multi-Core Variante des LibLinear (welche jedoch noch nicht für Java verfügbar ist)[3] wird daher für zukünftige Projekte empfohlen.
Neben dem Training lässt sich auch die Klassifizierung hinsichtlich Multi-Threading optimieren. Die Klassifizierung erfolgt für jede Instanz einzeln. Innerhalb der Klassifizierung werden die einzelnen Klassifizierer linear abgearbeitet und schreiben ihre Ergebnisse in ein zentrales Array. Da die einzelnen Klassifizierungsschritte unabhängig voneinander ablaufen, lässt sich eine Parallelisierung an zwei Stellen vornehmen. Zum einen auf der Ebene der Instanzen, welche in der jeweiligen Testmenge vorkommen und klassifiziert werden. Zum anderen auf der Ebene der einzelnen Gütemaße, welche im Rahmen der Klassifizierung erstellt werden. In der vorliegenden Arbeit wurde sich für letzteren Fall entschieden.

Durch die Implementierungen des Multi-Threading kann auf Multi-Core Systemen eine immense Zeitersparnis erreicht werden. Je nach Anzahl verwendeter Kerne kann die Zeit pro Durchlauf, bei beispielsweise 8 Kernen, auf ca. 1/5 im Vergleich zu einem Single-Core System reduziert werden. Da die Rechenressourcen bei Multi-Core Rechnern geschätzt nur zu ca. 80% ausgelastet sind besteht hierbei jedoch noch einiges an Optimierungspotential.

[3] Vgl. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/multicore-liblinear/

5 Durchführung und Ergebnisanalyse

Ziel der folgenden Analyse ist zu ermitteln, inwiefern sich die Daten innerhalb der bibliographischen Datenbank der ULB Darmstadt eignen um die interne Verschlagwortung zu verbessern. Hierzu wurden in den vorangehenden Schritten verschiedene Feature-Label Kombinationen identifiziert, welche für ein Multi-Label-Problem geeignet sind. Für die Durchführung und Analyse der folgenden Experimente müssen aufgrund der Daten Annahmen getroffen werden. Zum einen kann davon ausgegangen werden, dass die Instanzen, bei welchen keine Schlagworte gesetzt sind, diese gewollt nicht gesetzt sind. Dies kommt höchstwahrscheinlich daher, dass der jeweilige Bibliothekar diese Instanzen noch nicht bearbeitet hat. Diese Annahme bildet auch die Grundlage der vorliegenden Arbeit. Eine zweite Annahme ist, dass bei jenen Instanzen, bei welchen Label gesetzt sind diese nicht zwingend vollständig sind. Folgende Auflistung fasst beide Annahmen noch einmal zusammen.

- A_1: Nicht gelabelte Instanzen in der Datenbank sind nicht gewollt ungelabelt. Wenn eine Instanz keine Label besitzt bedeutet dies nicht, dass keine der Label zutreffen. Vielmehr wird davon ausgegangen, dass die entsprechenden Label noch nicht manuell gesetzt wurden.
- A_2: Hat eine Instanz Label so wird davon ausgegangen, dass die Liste an gesetzten Label nicht abschließend ist. D.h. selbst wenn in einer Instanz Label gesetzt wurden sind diese nicht zwingend vollständig.

Beide Annahmen spielen insbesondere bei der Bewertung der Güte der Klassifizierer eine Rolle. Es kann beispielsweise nicht zwingend davon ausgegangen werden, dass nicht zugewiesene Label (True Negative) auch wirklich nicht zwingend zutreffen. Folgt man diesem Gedanken, so macht eine Optimierung hinsichtlich der Accuracy wenig Sinn, da diese sowohl **True Positive** als auch **True Negative** erfasst. Insbesondere wenn ein möglichst breites Spektrum an Schlagworten angestrebt wird, sind durch den Klassifizierer inkorrekt gesetzte Schlagwort nicht zwingend falsch. Auch sonst wären diese nur bedingt schädlich, da diese sich beispielsweise durch den Nutzer ignorieren oder korrigieren ließen. Sinnvoll erscheint daher eine Bewertung hinsichtlich Precision und Recall. Um einen Ausgleich beider Maße zu finden lassen sich diese durch das F-Measure als harmonisches Mittel beider Werte abbilden.

Durch die verschiedenen Feature-Label Kombinationen ergeben sich eine Reihe interessanter inhaltlicher Aspekte. Es können Label basierend auf einer Taxonomie (Bezeichner 041A - RSWK-Schlagwort)

oder basierend auf der Einschätzung des Bibliothekars (Bezeichner 044K - Einzelschlagwort) mit verschieden umfangreichen Featureräumen verglichen werden.
Die Experimente laufen zweistufig ab. In einem ersten Schritt soll anhand einer repräsentativen Menge ermittelt werden, wie gut sich die jeweiligen Kombinationen an Features und Labels für das Multi-Label-Learning eignen. Im Rahmen dieses Schrittes werden verschiedenen Feature Mengen gebildet und auf den beiden verschiedenen Labeln gelernt. Um die Güte der SVM besser einschätzen zu können werden verschiedene Werte für den C-Parameter ebenso abgetragen. Der zweite Abschnitt des Experiments soll ermitteln, inwiefern sich eine Erhöhung der Anzahl an Instanzen auf die Güte der Klassifizierer auswirkt. Hierbei wird nur Bezug auf die Features des Haupttitel genommen und Instanzen in steigender Menge dem Trainingsprozess zugeführt.

5.1 1. Experiment - Vergleich Feature-Label Kombinationen auf repräsentativen Instanzen

Im ersten Experiment werden verschiedene Feature-Label Kombinationen auf ihre Eignung zur Verbesserung der Verschlagwortung im bibliographischen Katalog der ULB Darmstadt untersucht. Hierbei wird, um die einzelnen Feature-Label Kombinationen miteinander vergleichen zu können eine repräsentative Menge an Instanzen gebildet. Abbildung 5.1 stellt schematisch dar, wie diese Menge aus allen Instanzen gebildet wird. Die Untersuchungen werden für jede der beiden Label Quellen getrennt durchgeführt, da ein direkter Vergleich aufgrund der geringen Überschneidungen und der unterschiedlichen inhaltlichen Ausrichtung wenig sinnvoll erscheint. Tabelle 5.1 listet die einzelnen zu untersuchenden Feature-Label Kombinationen auf. Zu beachten ist, dass innerhalb der Experimente die Menge an Features systematisch ausgedehnt wird. Hierdurch soll untersucht werden, wie sich eine Erhöhung der Feature Anzahl auf die Güte der Klassifizierer auswirkt.
Da die verwendeten Algorithmen hinsichtlich Multi-Threading optimiert wurden und die Implementierung einen erhöhten Bedarf an Arbeitsspeicher benötigen ist in der Tabelle zusätzlich die Untergrenze des Bedarfs an Arbeitsspeicher abhängig von den verwendeten Features und Labels angegeben. Die Berechnung erfolgt anhand folgender Formel, welche aus der Dokumentation des Pocahontas Framework abgeleitet wurde:

$$f_{RAM}(X_{Features}, Y_{Labels}) = X_{Features} * Y_{Labels} * 8\ Byte$$

Ziel des ersten Experiments ist es zu ermitteln, inwiefern sich die unterschiedlichen Feature-Label Kombinationen für das Labeling der bibliographischen Daten eignen. Durch Variation des C-Parameters während des Experiments soll ein objektiver Vergleich sichergestellt werden. Das Verhalten abhängig vom C-Parameter der SVM soll genutzt werden um diese Einflussquelle auf die Güte der Ergebnisse

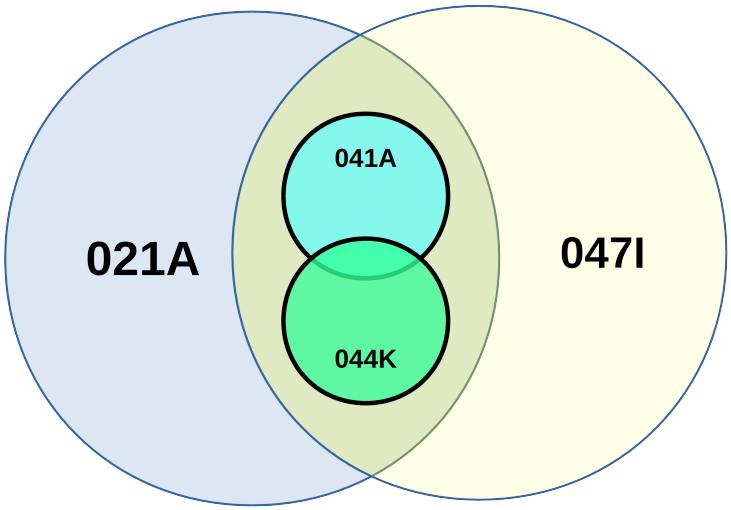


Abbildung 5.1: Repräsentative Mengen zur Durchführung des ersten Experiments

Features	Anzahl Features	Labels	Anzahl Labels	Progn. Untergrenze RAM Bedarf
021A	3.904	041A	2.200	0,064 Gb
047I	24.653	041A	2.200	0,407 Gb
$021A \cup 047I$	25.607	041A	2.200	0,423 Gb
021A	8.028	044K	4.239	0,253 Gb
047I	43.273	044K	4.239	1,366 Gb
$021A \cup 047I$	45.821	044K	4.239	1,447 Gb

Tabelle 5.1: Zu vergleichende Feature-Label Kombinationen auf repräsentativer Menge

abzudecken.

5.1.1 Vorgehensweise

Um die einzelnen Feature-Label Kombinationen miteinander vergleichen zu können wird jede Kombination durch den BinaryRelevance Algorithmus in Verbindung mit der SVM des LibLinear gelernt. Die SVM lernt das primale Problem mit dem SVM Typ "L2-regularized logistic regression (primal)". Die einzelnen Kombinationen werden mit einer Cross-Validation mit 10 Folds durchgeführt. Der C-Parameter wird mit insgesamt 7 verschiedenen Werten variiert. Diese folgen der Formel:

$$C = 10^i, i \in \{-6, -4, -2, 0, 2, 4, 6\}$$

Hierdurch werden zum einen sehr kleine, als auch sehr große Werte des C-Parameters abgedeckt. Sollte sich während der Experimente zeigen, dass die gewählten C-Parameter zu eng gefasst sind, können ergänzende Durchläufe mit größerem beziehungsweise kleinerem C-Parametern erfolgen. Die Darstellung des Algorithmus 1 verdeutlicht die Vorgehensweise.

Algorithm 1 Darstellung des ersten Experiments in Pseudocode

procedure FIRSTEXPERIMENT(T) ▷ T: Repräsentative Menge der verschiedenen Feature-Label Kombinationen

  for Feature-Label Kombinationen T **do**

    for $c = 10^{-6}, 10^{-4}, ..., 10^6$ **do**

      for $i = 1, ..., 10$ **do**

        $Train_i, Test_i \leftarrow splitCV(T, i)$

        $SVM_i^c \leftarrow trainSVM(c, Train_i)$

        $Results_i^c \leftarrow testModel(SVM_c^i, Test_i)$

      end for

      $AvgResult_T^c \leftarrow avg(Results_{i=1...10})$

    end for

  end for

end procedure

5.1.2 Ergebnisse

Die Darstellung der Ergebnisse wird getrennt nach Label Quellen durchgeführt. Hierbei wird insbesondere dargestellt, wie sich die einzelnen label-basierten Gütemaße bei Änderung des C-Parameters und

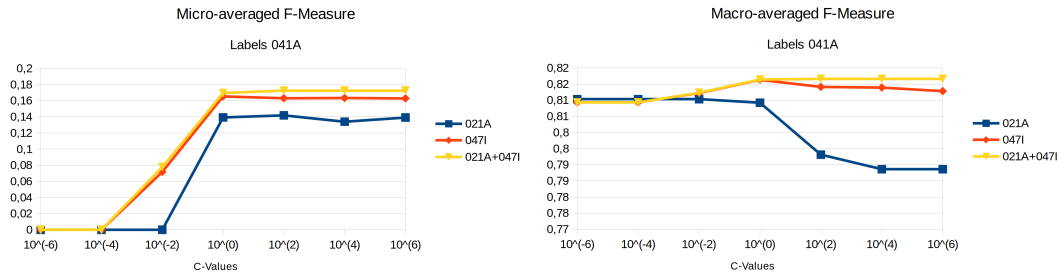
der Variation der Feature Menge verhalten. Hieraus wird erwartet, dass Rückschlüsse zum einen über das Potenzial der Erweiterung des Feature Raums, als auch durch über den Bereich des mutmaßlich optimalen C-Parameters gezogen werden können. Zur Analyse wurden die label-basierten Gütemaße herangezogen, da sich die instanz-basierten Gütemaße jenen der Micro-Averaged Gütemaße stark ähneln. Die Untersuchung bezieht sich hierbei auf F-Measure und deren Zusammensetzung aus Recall und Precision. Eine Untersuchung der Specificity Werte erscheint aufgrund der Struktur der Label innerhalb der Instanzen wenig sinnvoll. Durch die hohe Anzahl unterschiedlicher Label und der geringen Anzahl an Label pro Instanz kommt es hierbei zu Werten der Specificity nahe dem maximalen Wert 1, unabhängig von der erfassenden Methode. Zur Vollständigkeit finden sich die Graphiken der Specificity im Anhang.

041A - RSWK-Schlagwort

F-Measure

Die Ergebnisse für das F-Measure bei Label Quelle 021A sind in Abbildung 5.2 dargestellt. Die Performance in absoluten Zahlen (Micro-Averaged) bewegen sich im Bereich 0,14-0,18 und jene in relativen Ergebnissen (Macro-Averaged) im Bereich 0,78-0,82. Ein Grund hierfür ist vermutlich die Verteilung der Label. Es existieren sehr viele sehr seltene Label, welche scheinbar relativ gut zugewiesen werden. Allgemein lässt sich feststellen, dass eine erhöhte Anzahl an Features zu einer besseren Performance führt. Der optimale C-Parameter für das vorliegenden Experiment liegt vermutlich im Bereich um $C = 1$. Die Performance der Features aus dem Haupttitel nimmt mit steigendem C-Parameter bei der Macro-Averaged Methode ab. Dies ist bedeutsam, falls die Performance seltener Label optimiert werden soll.

Die großen Unterschiede zwischen Micro- und Macro-Averaged F-Measure lassen auf eine schein-



(a) Micro-averaged F-Measure **(b)** Macro-averaged F-Measure

Abbildung 5.2: Vergleich verschiedener F-Measure Werte - Label 041A

bar hohe Performance von seltenen Label und auf eine durchwachsene Performance häufiger Label schließen. Insbesondere das Macro-Averaged F-Measure nimmt hohe Werte an, wenn eine große Anzahl

an Label eine gute Performance erzielt. Die hohe Performance des Macro-Averaged F-Measure wird in der Funktionsweise des Frameworks vermutet. Kommt in der Trainingsmenge ein Label nicht vor, so setzt Mulan den F-Measure für dieses Label auf 1[1]. Um diesen Zusammenhang zu verdeutlichen ist in Abbildung 5.3 die Performance der einzelnen Label abhängig von ihrem Häufigkeitsrang abgetragen. Als C-Parameter wurde hierbei $C = 1$ gewählt, da dieser Parameter eine gute Performance hinsichtlich des F-Measure besitzt. Es zeigt sich deutlich, dass die Kurve mit sinkender Häufigkeit höhere Macro-Averaged F-Measure Werte annimmt. Da sich die Werte ungefähr invers entwickeln (bspw. führt das einmalige Vorkommen zu einem Macro-Averaged F-Measure von 0,9[2]). Es kann somit eher das Gegenteil der Performance seltener Label angenommen werden, d.h. dass diese weit schlechter klassifiziert werden als häufige Label da die beiden Graphen aus Häufigkeit und F-Measure sich gegenläufig bewegen.

Für die Gütebestimmung ist somit realistischerweise nur der Wert der Micro-Averaged Gütemaße

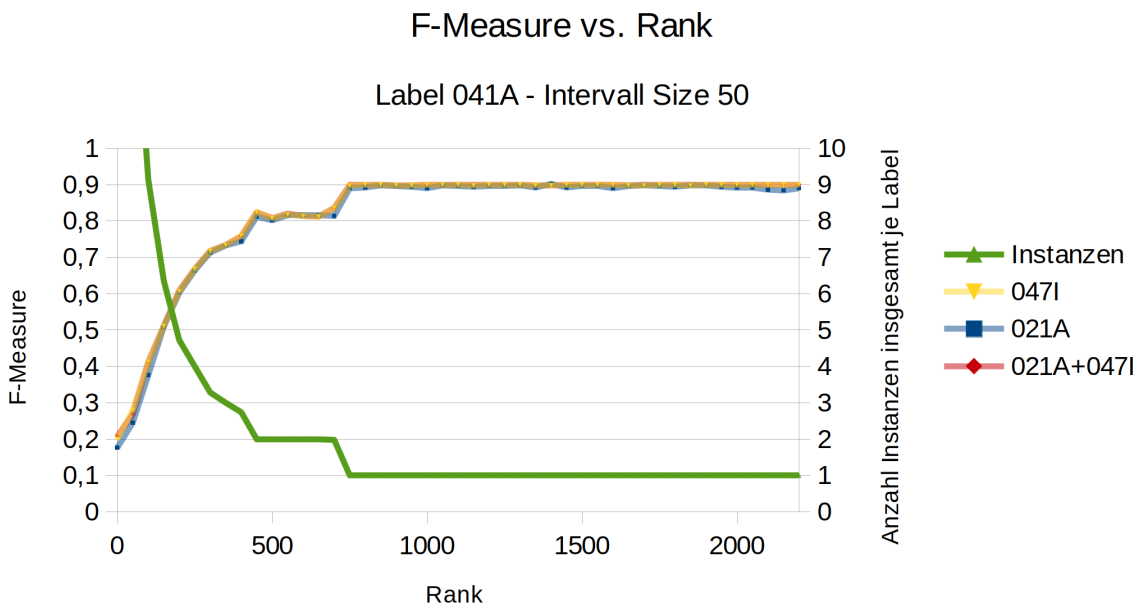


Abbildung 5.3: Rang eines Labels und dessen F-Measure Wert - Label 041A

sinnvoll heranzuziehen, da die Werte der Macro-Averaged Gütemaße einen starken Bias bei geringem Vorkommen von Labels haben. Die Analyse des Micro-Averaged F-Measure zeigt, dass eine Erhöhung der Feature Anzahl für eine bessere Performance sorgt. Der optimale C-Parameter wird um den Wert 1

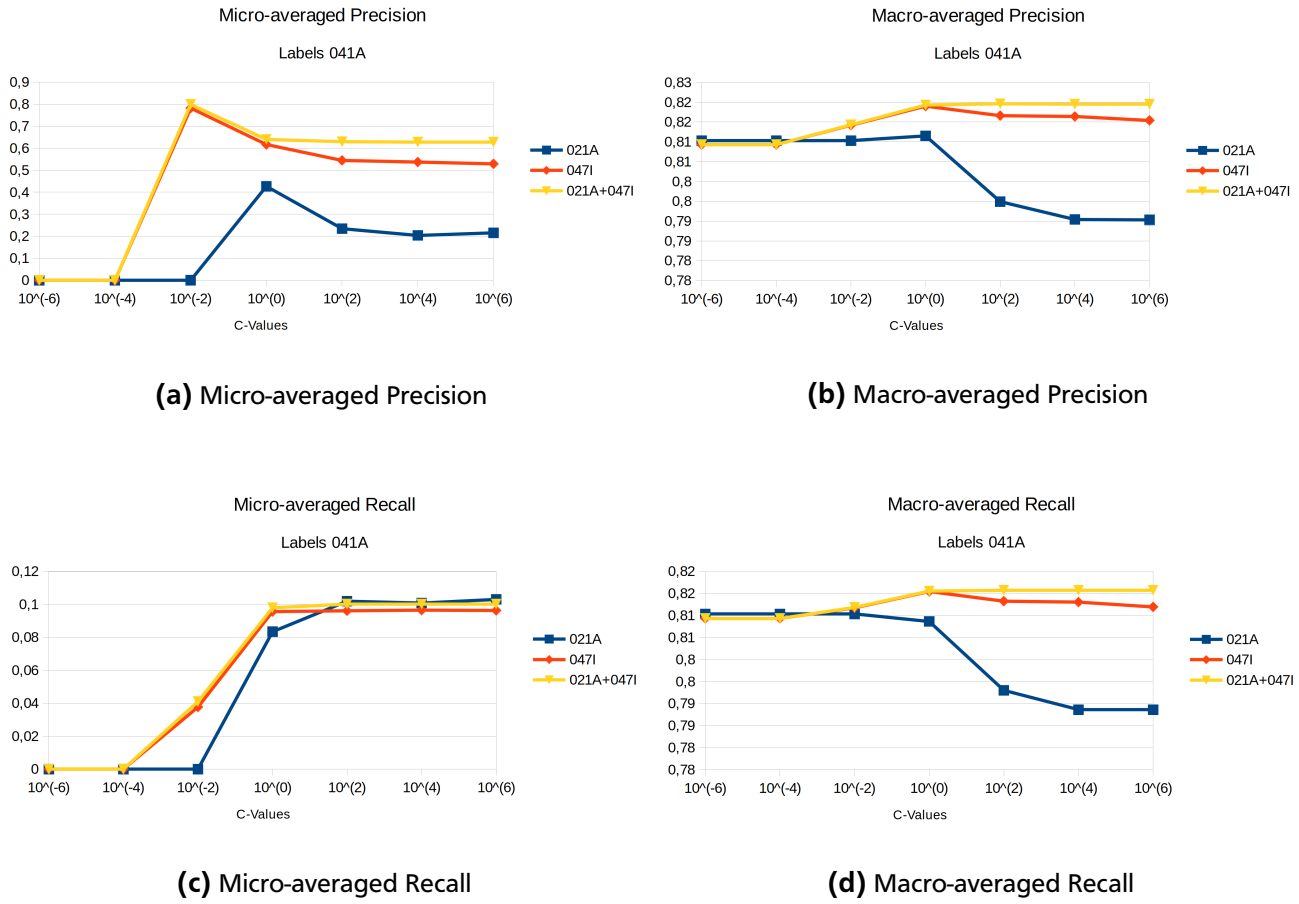
[1] Das Handling von 0/0 bei der Gütewertberechnung wird somit als 1 interpretiert

[2] Dies kann bei einem 10-fold Cross-Validation berechnet werden durch 1-maliges Vorkommen in der Testmenge (und inkorrekter Klassifizierung) und 9-maliges nicht-Vorkommen in der Testmenge. Hierdurch ergibt sich aufgrund der Berechnungsmethode des Mulan Frameworks ein Wert von $0,9 = \frac{9*1+1*0}{10}$

vermutet. Folgend soll die Zusammensetzung des F-Measure aus den einzelnen Werten für Precision und Recall ermittelt werden.

Precision und Recall

Da sich der Wert des F-Measure als harmonisches Mittel aus den Werten für Precision und Recall zusammensetzt ist interessant inwiefern diese den F-Measure beeinflussen. Abbildung 5.4 stellt sowohl Precision als auch Recall für die Label Quelle 041A dar, zur besseren Vergleichbarkeit untereinander in einer Graphik.



(a) Micro-averaged Precision

(b) Macro-averaged Precision

(c) Micro-averaged Recall

(d) Macro-averaged Recall

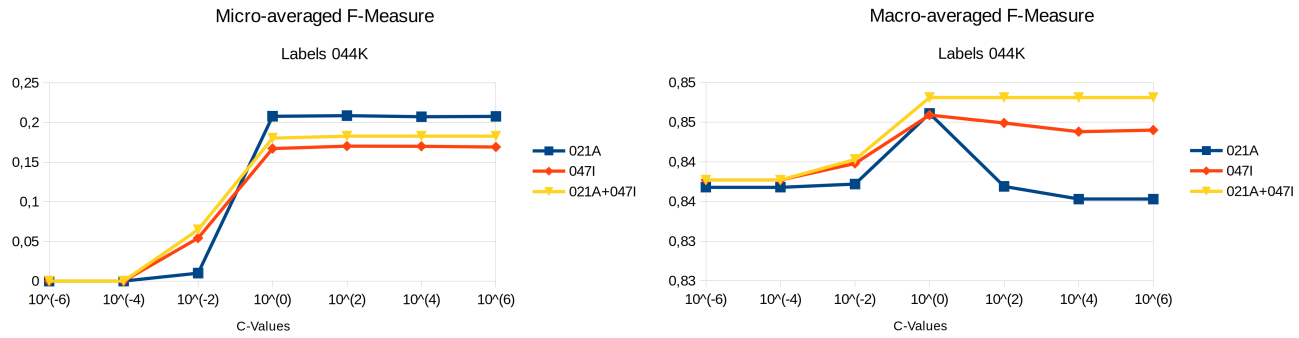
Abbildung 5.4: Vergleich Precision und Recall - Label 041A

Es zeigt sich, dass der F-Measure insbesondere von einer hohen Precision profitiert. Diese bewegt sich beim Micro-Averaged Verfahren insbesondere bei größerem Feature Raum im Bereich 0,5-0,65. Der Beitrag des Recall zum F-Measure schwankt hingegen um den Bereich 0,1. Dies bedeutet, dass relativ wenige Label erfasst werden, diese jedoch mit einer relativ hohen Präzision. Der charakteristische Unterschied zwischen Micro-Averaged und Macro-Averaged Gütemaßen aufgrund der Berechnungsmethode des Mulan Frameworks zeigt sich auch hier. Interessant hinsichtlich des C-Parameters ist, dass dieser unterschiedliche Optima für Precision und Recall zu besitzen scheint und somit Optimierungspotential je nach angestrebten Zweck ermöglicht.

044K - Einzelschlagwort

Bei der Verwendung der Einzelschlagworte (044K) zeigt sich eine Performance des F-Measure bei höheren C-Parametern im Bereich 0,16-0,21. Die Performance des F-Measure Gütemaße ist in Abbildung 5.5 dargestellt. Im Gegensatz zu den RSWK-Schlagworten (041A) zeigt sich bei den Einzelschlagworten eine Verschlechterung der Micro-Averaged F-Measure Werte bei Einbindung von Features der inhaltlichen Zusammenfassung. Der optimale C-Parameter liegt vermutlich im Bereich größer 1.

F-Measure



(a) Micro-averaged F-Measure **(b)** Macro-averaged F-Measure

Abbildung 5.5: Vergleich verschiedener F-Measure Werte - Label 044K

Der charakteristische Unterschied zwischen Micro-Averaged und Macro-Averaged F-Measure zeigt sich auch bei den Einzelschlagworten (044K). Auch hier kann dies auf die interne Arbeitsweise des Mulan Frameworks zurückgeführt werden, welches Label die nicht in der Testmenge vorkommen mit 1 bewertet. Zur Verdeutlichung stellt auch hier Abbildung 5.6 die Performance der einzelnen Label abhängig vom Rang (bzw. Häufigkeit) der Label dar. Die scheinbar gute Performance seltener Label ist somit wiederum

auf die Berechnungsmethode des Mulan Frameworks zurückzuführen. Es kann auch hier eher das Gegenteil an Performance angenommen werden, d.h. dass seltene Label eher schlecht klassifiziert werden da sich die Kurven aus Häufigkeit und F-Measure gegenläufig bewegen.

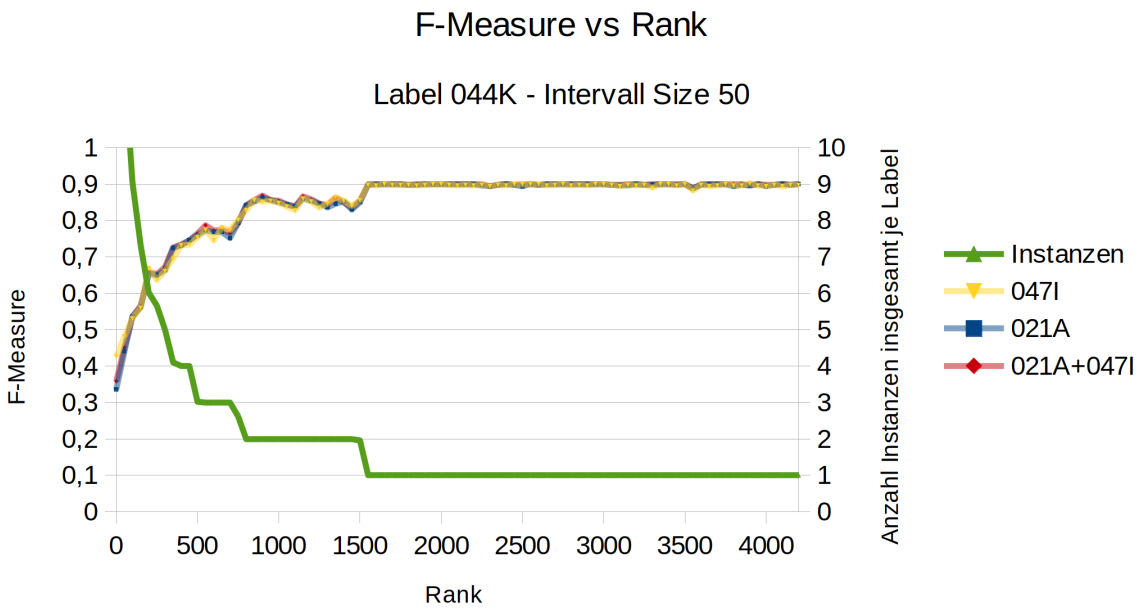
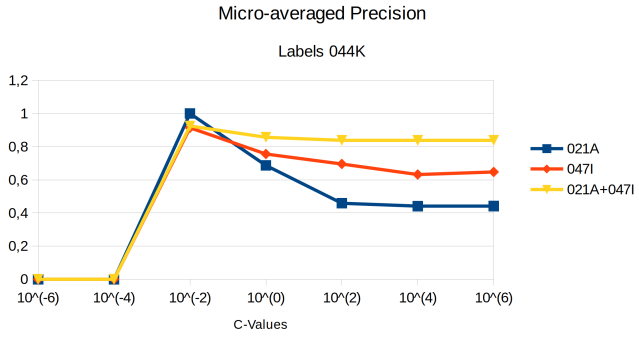


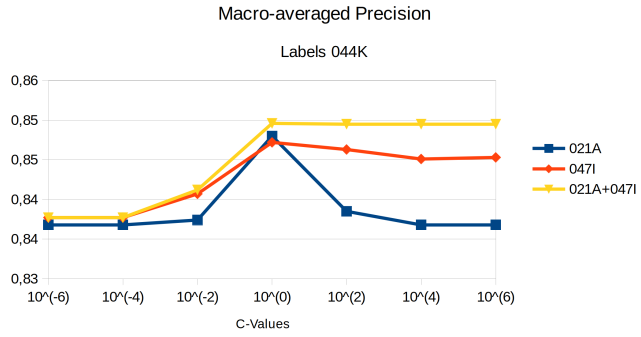
Abbildung 5.6: Rang eines Labels und dessen F-Measure Wert - Label 044K

Precision und Recall

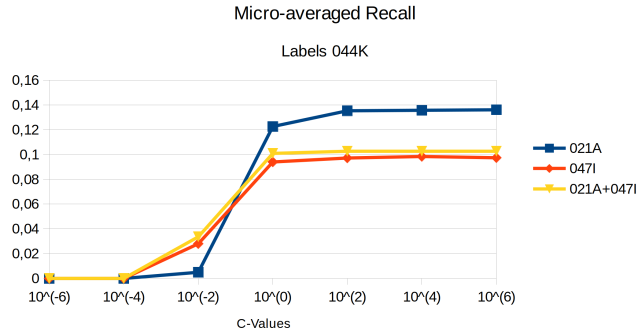
Die Zusammensetzung des F-Measure aus Precision und Recall soll auch hier analysiert werden. Der charakteristische Unterschied zwischen Micro-Averaged und Macro-Averaged Gütemaßen zeigt sich auch hier. Interessant bei der Betrachtung der Micro-Averaged Gütemaße ist, dass sich der F-Measure aus einer hohen Precision und einem geringen Recall zusammensetzt. Eine Erhöhung der Anzahl an Features sorgt für eine starke Verbesserung der Precision, scheint jedoch eine gegenteilige Auswirkung auf den Recall zu haben. Hierdurch lässt sich auch der Unterschied hinsichtlich des F-Measure bei unterschiedlichem Feature Raum erklären.



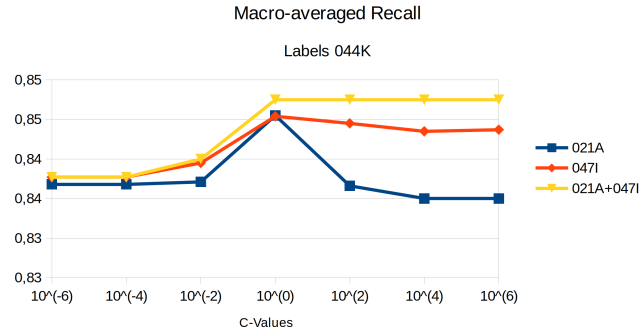
(a) Micro-averaged Precision



(b) Macro-averaged Precision



(c) Micro-averaged Recall



(d) Macro-averaged Recall

Abbildung 5.7: Vergleich verschiedener Precision Werte - Label 044K

Hinsichtlich des C-Parameters scheint es unterschiedliche Optima für Precision und Recall zu geben. Insbesondere der Recall scheint sich erst mit einem C-Parameter größer 1 sinnvollen Werten zu nähern, während bei der Precision ein Wert kleiner 1 zu höheren Werten führt.

5.1.3 Analyse

Die Ergebnisse der Experimente, insbesondere hinsichtlich der Performance der Macro-Averaged Gütemaße sind mit Vorsicht zu genießen. Die scheinbar hohe Performance seltener Label konnte durch Analyse der internen Berechnungsmethoden des Mulan Frameworks in das Gegenteil verkehrt werden. Seltene Label werden also tendenziell sehr schlecht klassifiziert. Die relevante Klasse im Mulan Framework *mulan.evaluation.measure.InformationRetrievalMeasures*, bzw. der Mechanismus zur Berechnung der Ergebnisse der Cross-Validation müsste für aussagekräftige Ergebnisse zu einzelnen Label angepasst werden.

Zur Analyse der Güte des Klassifizierers ist es somit sinnvoller nur die Werte der Micro-Averaged Gütemaße heranzuziehen, da diese absolute Werte an korrekt bzw. inkorrekt zugewiesenen Label betrachten. Betrachtet diese absoluten Werte, so zeigt sich ein recht niedriger Wert des Micro-Averaged F-Measure für die RSWK-Schlagworte (041A) im Bereich 0,14-0,18, wobei die Klassifizierung besser funktioniert, je größer der Feature-Raum ist. Bei Einzelschlagworten (044K) bewegen sich die Gütemaße des Micro-Averaged F-Measure im Bereich 0,16 - 0,21. Eine Vergrößerung des Feature-Raums führt hierbei interessanterweise zu einer Verschlechterung der Ergebnisse des Micro-Averaged F-Measure. Der optimale C-Parameter liegt bei beiden Experimenten höchstwahrscheinlich im Bereich um 1 und kann je nach Anforderung an die einzelnen Gütemaße explizit optimiert werden.
Bei beiden Schlagworten zeigt sich, dass sich der F-Measure aus einer relativ hohen Precision und einem relativ niedrigen Recall zusammensetzt. Dies ist für den praktischen Anwendungsfall interessant, da eine hohe Precision bedeutet, dass wenige Falschklassifizierungen erfolgen und somit die Ergebnisse zwar nicht umfangreich dafür jedoch präzise sind. Als Hilfestellung für einen Bibliothekar, beispielsweise durch Empfehlungen von höchstwahrscheinlichen Label wären die betrachteten Algorithmen somit durchaus interessant.

5.2 2. Experiment - Einbindung zusätzlicher Instanzen und Optimierung

Im ersten Experiment wurden verschiedene Feature-Label Kombinationen miteinander verglichen, wobei der Feature Raum sich zwischen den einzelnen Mengen unterschieden hat. Die Auswahl der Instanzen erfolgte hierbei anhand einer repräsentativen Menge die sich für die Vergleichbarkeit der Messwerte anhand mehrerer Schnittmengen gebildet hatte. Um einschätzen zu können, wie sich die Güte der verwendeten Lerner bei Vergrößerung der Instanz Anzahl verhält, soll im folgenden zweiten Experiment die Menge an Instanzen schrittweise erhöht werden. Grundlage bilden hierbei wiederum die repräsentativen Instanzen des ersten Experiments, welche durch Instanzen des gleichen Feature- und Label-Raums angereichert werden. Da hierfür nur die Instanzen mit Haupttitel in Betracht kommen wird das zweite Experiment auf diese Instanzen beschränkt. Um den Einfluss des C-Parameters zumindest teilweise auszugleichen wird dieser innerhalb der Experimente optimiert.

5.2.1 Vorgehensweise

Die Erhöhung der Anzahl an Instanzen des zweiten Experiments erfolgt schrittweise. Hierbei werden in Abständen zu 10.000 aus einer Menge P zusätzliche Instanzen in das Experiment eingebunden. Die zusätzlichen Instanzen dienen zur Ermittlung des optimalen C-Parameters und zum finalen Erstellen des Lerners. Das zweite Experiment wird ebenfalls für beide Label Quellen durchgeführt. Die einzelnen Schritte des zweiten Experiments werden anhand einer 10-fold Cross-Validation durchgeführt. Eine Darstellung des zweiten Experiments in Pseudocode findet sich in Algorithmus 2.

Als Grundmenge an Instanzen wird die repräsentative Menge aus dem ersten Experiment verwendet. Diese wird im Rahmen der Optimierung des C-Parameters und zur finalen Konstruktion des optimierten Modells durch zusätzliche Instanzen aus der Gesamtmenge an Instanzen angereichert. Eine schematische Darstellung der Anreicherung durch zusätzliche Instanzen finde sich in Abbildung 5.8. Für die Interpretation der Ergebnisse ist es wichtig, dass die zusätzlichen Instanzen sich Feature- und Label-Raum mit den repräsentativen Instanzen teilen. Dies bedeutet, dass nur jene Instanzen hinzugezogen wurden, welche Features und Labels besitzen die auch in der repräsentativen Menge an Instanzen vorkommt. Hierdurch wird eine Untersuchung anhand der Erhöhung der Menge an Instanzen bei gleichbleibendem Feature- und Label Raum möglich.
Der C-Parameters wird anhand des Micro-Averaged F-Measure optimiert. Die Verwendung des Micro-Averaged F-Measure hat sich im ersten Experiment und durch die interne Arbeitsweise des Mulan Frameworks als geeignetes Maß herausgestellt. Insbesondere Label die nicht in der Testmenge vorkom-

Algorithm 2 Darstellung des zweiten Experiments in Pseudocode

procedure SECONDEXPERIMENT(T, P) ▷ T: Repräsentative Menge aus 1. Experiment

P: Menge zusätzlicher Instanzen

for $p = 0, 10.000, ...60.000$ aus P **do**

for $i = 1, ..., 10$ **do**

$Train_i, Test_i \leftarrow splitCV(T, i)$

$TrainTrain_i, TrainValid_i \leftarrow splitCV(Train_i, 1)$

for $c = 10^{-6}, 10^{-4}, ..., 10^{6}$ **do**

$SVM_i^c \leftarrow trainSVM(c, TrainTrain_i + p)$

$Results_i^c \leftarrow testModel(SVM_i^c, TrainValid_i)$

end for

$C^* \leftarrow bestResult(Results_c^i, Measure_m)$

$SVM_i^* \leftarrow train(c^*, Train_i + p)$

$Results_i^* \leftarrow test(SVM_i, Test_i)$

end for

$AvgResult^p \leftarrow avg(Results_{i=1...10})$

end for

end procedure

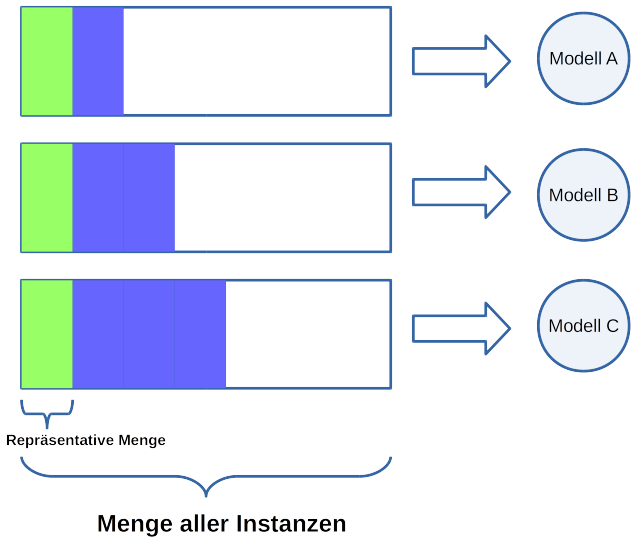


Abbildung 5.8: Darstellung Instanzmenge zweites Teilexperiment

Label-Quelle	Features	Labels	Repr. Instanzen	Max. zusätzliche Instanzen
041A	3904	2220	2335	67339
044K	8028	4239	2884	148594

Tabelle 5.2: Statistiken zusätzliche Instanzen pro Label Quelle

men werden durch das Macro-Averaged Gütemaß als besonders erfolgreich klassifiziert dargestellt. Die Verwendung des Micro-Averaged F-Measure als Optimierungsmaß erhöht die absolute Anzahl an korrekt klassifizierten Label und sollte dadurch ein gutes Maß für das vorliegende Experiment darstellen.

Durch die Hinzunahme zusätzlicher Instanzen erhöht sich die Rechenzeit annähernd linear. Dies bedeutet, dass bei Verdopplung der Anzahl an Instanzen sich auch die benötigte Rechenzeit verdoppelt. Aus diesem Grund muss eine Abstufung an zusätzlichen Instanzen gefunden werden, welche zum einen präzise genug ist, um aussagekräftige Ergebnisse zu produzieren, zum anderen zur Berechnung nicht unverhältnismäßig viel Zeit benötigt. Es wurden Schritte zu 10.000 Instanzen gewählt, die maximal mögliche Anzahl an zusätzlichen Instanzen findet sich in Tabelle 5.2. Falls sich während der Experimente herausstellen sollte, dass interessante Phänomene auftreten, so können zusätzliche Experimente mit variierender Menge zusätzlicher Instanzen durchgeführt werden.

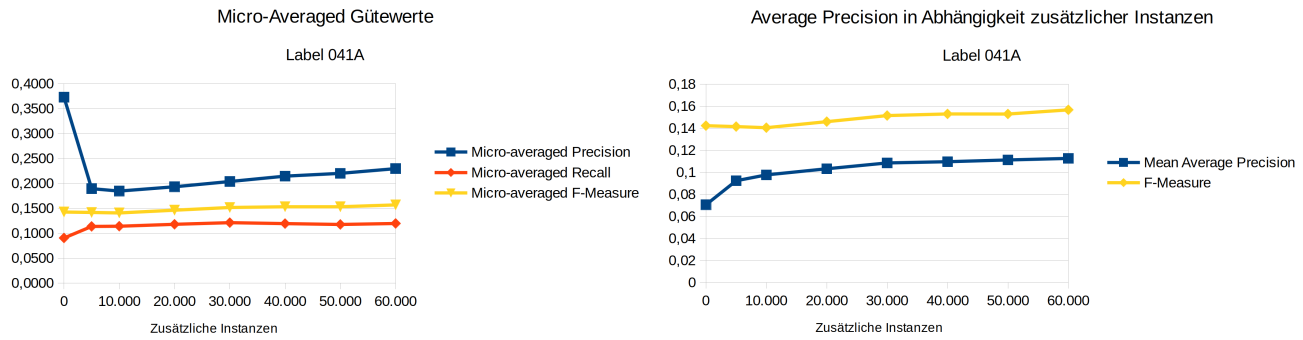
5.2.2 Ergebnisse

Die Auswertung des zweiten Experiments wird wie das erste Experiment getrennt nach den beiden Label Quellen 041A und 044K durchgeführt. Da die Ergebnisse hinsichtlich des Micro-Averaged F-Measure optimiert wurde, wird auf dieses Maß bei der Auswertung fokussiert. Besonders interessant ist hierbei auch, wie sich Micro-Averaged Precision und Recall bei Erhöhung der Anzahl an Instanzen verhalten.

041A - RSWK-Schlagwort

Die Ergebnisse für die Label Quelle 041A für das Micro-Averaged F-Measure sind in Graphik 5.9 abgetragen. Die verschiedenen Stufen an zusätzlichen Instanzen befinden sich hierbei auf der X-Achse. In der Graphik wurden zusätzlich die Maße für Recall und Precision abgetragen um einen direkten Vergleich zu ermöglichen.

Wie sich zeigt führt eine Erhöhung der Anzahl an Instanzen zu einer leichten Verbesserung der Werte des Micro-Averaged F-Measure. Interessant zu sehen ist, dass die Werte für Precision bei den ersten 10.000 zusätzlichen Instanzen abfallen während die Werte für den Recall ansteigen. Um dieses Phänomen zu-



(a) Micro-Averaged Gütemaße

(b) Mean Average Precision und F-Measure

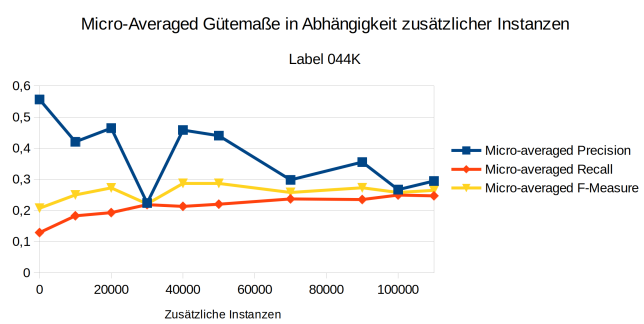
Abbildung 5.9: Micro-Averaged Gütemaße und Mean Average Precision - Label 041A

sätzlich zu präzisieren wurde noch einmal ein Durchlauf mit 5.000 zusätzlichen Instanzen gestartet. Da das Phänomen auch hier Auftritt, scheint schon eine geringe Erhöhung der Anzahl an Instanzen einen Einfluss auf die Güte zu haben. Es lässt sich feststellen, dass durch die Einbindung zusätzlicher Instanzen sich insbesondere die Precision steigern lässt und diese Auswirkungen auf den Wert des F-Measure besitzt. Ist man an möglichst wenigen fehlerhaften Klassifizierungen interessiert ist eine Erhöhung der Instanz Anzahl also zielführend.

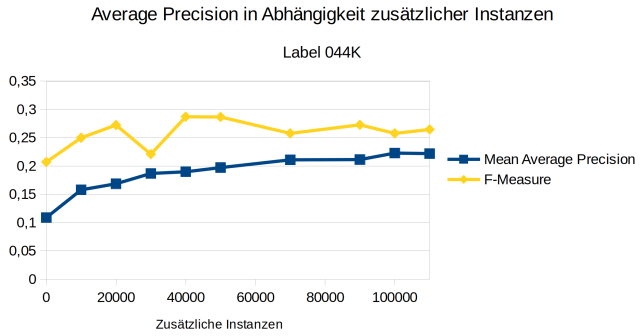
Die Gesamtgüte des Modells, dargestellt durch den Mean Average Precision, erhöht sich durch Einbindung zusätzlicher Instanzen kontinuierlich. Auch im Vergleich zum F-Measure zeigt sich eine stärkere Verbesserung bei Einbindung von mehr Instanzen. Daraus ist zu schließen, dass eine Erhöhung der Anzahl an Instanzen auch für die Güte des Gesamtmodells eine positive Auswirkung hat.

044K - Einzelschlagwort

Die Ergebnisse für die Einzelschlagworte (044K) hinsichtlich der Einbindung zusätzlicher Instanzen weisen im Vergleich zu den RSWK-Schlagworten (041A) einige Besonderheiten auf. Es zeigt sich auch bei den Einzelschlagworten, dass eine Erhöhung der Anzahl an Instanzen tendenziell zu einer Verbesserung der Micro-Averaged F-Measure führen kann. Die Gütewerte der Precision scheinen sich jedoch durch Einbindung zusätzlicher Instanzen zu verschlechtern, wohingegen sich die Werte des Recll zu verbessern scheinen. Interessant ist, dass es bei gewissen Mengen zusätzlicher Instanzen zu Einbrüchen der Precision und folglich auch des F-Measure kommt. Eine kontinuierliche Verbesserung wie bei den RSWK-Schlagworten (041A) zeigt sich hier somit nicht. Ein Grund kann hierbei in der Optimierung des C-Parameters liegen. Wird dieser auf einen für die Testmenge unvorteilhaften Wert optimiert, so können sich tendenziell starke Schwankungen der Precision ergeben, wie im ersten Experiment gezeigt wurde. Die Gesamtgüte des Modells, mit Blick auf die Mean Average Precision, verbessert sich durch



(a) Micro-Averaged Gütemaße



(b) Mean Average Precision mit F-Measure

Abbildung 5.10: Micro-Averaged Gütemaße und Mean Average Precision - Label 044K

Einbindung zusätzlicher Instanzen kontinuierlich. Bei ihr zeigen sich die Einbrüche des F-Measure nicht in einem Rückgang der Werte. Somit kann auch für die Modelle basierend auf den Einzelschlagworten (044K) eine Verbesserung der Güte der Modelle durch Einbindung zusätzlicher Instanzen angenommen werden.

5.2.3 Analyse

Die Erhöhung der Anzahl an Instanzen bei gleichbleibenden Feature Raum scheint die Qualität der gelernten Modelle sowohl für RSWK-Schlagwörter (041A) als auch für Einzelschlagwörter (044K) zu verbessern. Insbesondere die Mean Average Precision verbessert sich bei beiden Label Quellen bei Zunahme der Anzahl an Instanzen, was auf eine allgemeine Verbesserung der Güte des Lernmodells schließen lässt.

Bei den RSWK-Schlagwörtern (041A) zeigt sich ein recht kontinuierliches Bild bei Hinzufügung zusätzlicher Instanzen. Fällt zunächst die Precision scharf ab, so stabilisieren sich die Werte durch das Hinzufügen zusätzlicher Instanzen. Der Recall hingegen verbessert sich zunächst, verharrt dann jedoch auf einem Wert um 0.12. Die Werte des Micro-Averaged F-Measure verbessern sich kontinuierlich, jedoch auf niedrigem Niveau. Bei der Berechnung des F-Measure dominiert ohne zusätzliche Instanzen die Precision, durch zusätzliche Instanzen wird jedoch der Einfluss des Recall auf den F-Measure verstärkt. Die Mean Average Precision verbessert sich durch Hinzunahme weiterer Instanzen kontinuierlich um ca. 61% (0,07 [ohne zusätzliche Instanzen] vs. 0,113 [mit max. zusätzlichen Instanzen])

Die Einzelschlagworte (044K) zeigen hinsichtlich der Micro-Averaged Gütemaße kein einheitliches Bild. Insbesondere scheint der Micro-Averaged F-Measure großen Schwankungen ausgesetzt zu sein, sodass sich hier kein einheitlicher Trend abzeichnet. Bei den zusätzlichen Instanzmengen um 30.000 und 70.000 nimmt der F-Measure sogar geringere Werte an. Der Grund für das Verhalten des F-Measure

wird in der Optimierung des C-Parameters vermutet. Womöglich sorgen einzelne Instanzen bei der Optimierung des C-Parameters zu einer Verschiebung desselben und zu einer unvorteilhaften Auswahl des optimalen C-Parameters für das entscheidende Testen und Bewerten des Modells auf der repräsentativen Menge. Die Gütebewertung des Modells auf Basis der Mean Average Precision hingegen verbessert sich kontinuierlich um ca. 103% (0,109 [ohne zusätzliche Instanzen] vs. 0,222 [mit max. zusätzlichen Instanzen]), sodass auch bei den Einzelschlagworten von einer positiven Beeinflussung der Ergebnisse durch Erhöhung der Instanzanzahl ausgegangen werden kann, jedoch mit der gebotenen Vorsicht der Beeinflussung der Ergebnisse durch Ausreißer.

Die Erhöhung der Anzahl an Instanzen zur Bildung des Modells scheint für die RSWK-Schlagwörter (041A) definitiv lohnend zu sein. Hier verbessern sich die Ergebnisse mit Erhöhung der Anzahl an Instanzen kontinuierlich. Bei den Einzelschlagworten (044K) muss während der Erhöhung der Anzahl an Instanzen der C-Parameter beachtet werden. Es wird stark vermutet, dass dieser Ergebnisse negativ beeinflussen kann. Da durch das erste Experiment dieser Arbeit vermutet wird, dass der optimale C-Parameter um 1 liegt, könnte auch ein fester C-Parameter gewählt werden vor Durchführung der Experimente.

6 Fazit, Kritik und Ausblick

Ziel der vorliegenden Arbeit ist es zu ermitteln, wie sich die Verschlagwortung innerhalb der bibliographischen Datenbank der ULB Darmstadt verbessern lässt. Hierzu wurde ein Abzug der bibliographischen Datenbank der ULB Darmstadt bereitgestellt, welcher im PICA+ Format formatiert ist. In einem ersten Schritt wurden diese Daten einer Analyse unterzogen um zu ermitteln, welche Bestandteile sich für die Verwendung in einem Multi-Label-Problem eignen. Die hieraus gewonnen Erkenntnisse über Inhalt und Struktur der Daten hat zur Bildung einer repräsentativen Menge geführt. Diese Menge wurde in einem ersten Experiment verwendet um verschiedene Feature-Label Kombinationen miteinander vergleichen zu können. Aufbauend auf dem ersten Experiment wurde in einem zweiten Experiment versucht, die Güte der Lernmodelle durch Erhöhung der Anzahl an Instanzen zu verbessern. Hierbei blieb sowohl der Feature- als auch der Label-Raum identisch mit jenem der repräsentativen Menge.

Im ersten Experiment wurde ein Vergleich verschiedener Feature-Label Kombinationen durchgeführt. Die Erhöhung des Feature-Raums scheint je nach Label Quelle lohnend zu sein. Es wird daher erwartet, dass durch die Einbindung externer Quellen, bzw. die Anreicherung der Features durch externe Quellen sich die Qualität der Modelle weiter verbessern lässt. Außerdem wurde der Grund für eine scheinbar gute Performance von seltenen Labels ausgemacht. Das Mulan Framework behandelt Label, die in der Testmenge nicht vorkommen als perfekt klassifiziert, was zu Verzerrung der Gütewerte führt und bei der Auswertung beachtet werden muss.
Das zweite Experiment hatte die Erhöhung der Anzahl an Instanzen und die Untersuchung der Auswirkung auf die Güte der Modelle zum Ziel. Es hat sich ergeben, dass eine Verbesserung der Güte der Ergebnisse bei RSWK-Schlagworten durch die Erhöhung der Instanzen bei gleichbleibendem Feature Raum erreicht werden kann. Auch die Ergebnisse der Einzelschlagworte verbessern sich durch Erhöhung der Anzahl an Instanzen. Es zeigt sich hierbei jedoch kein lineares Bild der Ergebnisse hinsichtlich der absoluten Anzahl der klassifizierten Label.

6.1 Praktische Relevanz

Praktische Relevanz haben die Ergebnisse für die ULB Darmstadt. Um ein Multi-Label-Learning gewinnbringend auf der gesamten Datenbank durchzuführen müssen jedoch einige Dinge beachtet werden. Zum einen stellt sich durch die Größe der Datenbank die Frage der Skalierung. Die Durchführung der Experimente, insbesondere bei hoher Anzahl an Instanzen, hat großen Zeit- und Ressourcenaufwand

zur Folge gehabt. Eine Erstellung einer Lernmodells auf dem gesamten Datensatz wird aufgrund der Beschränkung von Ressourcen schwierig. Ein Weg diese Einschränkung zu umgehen wurde durch die Einbindung zusätzlicher Instanzen bei gleichbleibenden Feature- und Label-Raum gezeigt. Da angenommen werden kann, dass die Qualität der Ergebnisse mit der Anzahl and Qualität der Features korreliert wäre es für den praktischen Anwendungsfall interessant, inwiefern sich durch Feature Engineering, durch Bereinigung oder Einbindung externer Feature Quellen sich die Ergebnisse verbessern lassen. Ein automatisches Labeling basierend auf den untersuchten Algorithmen scheint nicht zielführend. Die Güte der Modelle, insbesondere der niedrige F-Measure Wert, lassen eine automatische Erstellung von Schlagworten nicht sinnvoll erscheinen. Da die Precision jedoch je nach verwendetem Modell sehr hoch sein kann wäre es denkbar, ein Empfehlungssystem für den Bibliothekar aufzusetzen. Hierbei würden dem Bibliothekar Schlagworte empfohlen, welche mit hoher Wahrscheinlich zutreffen. Die Entscheidung diese Schlagworte zu setzen liegt somit letztendlich jedoch immer noch einer menschlichen Kontrolle.

Neben den inhaltlichen Aspekten der vorliegenden Arbeit wurden die verwendeten Algorithmen um die Möglichkeit des Multi-Threading erweitert. Dies gebot sich aufgrund von Zeit- und Ressourcenbeschränkungen und hat zur Anpassung des Pocahontas Framework geführt. Unter anderem wurde der BinaryRelevance Algorithmus an entscheidenden Stellen des Trainings- und Testvorgangs parallelisiert. Bei Verwendung des Multi-Threading BinaryRelevance ist darauf zu achten, dass der eingesetzte binäre Klassifizierer für Multi-Threading geeignet sein muss. Insbesondere globale Variablen und Referenzen können zu Seiteneffekten führen, welche schwer aufspürbar sind. Im vorliegenden Fall kam es zu solchen Seiteneffekten durch die Implementierung des LibLinear, welche jedoch behoben werden konnte. Für zukünftige Projekte wäre entweder eine Einbindung der Multi-Core Version des LibLinear oder eine umfassende Analyse und Optimierung des Pocahontas Framework ratsam, insbesondere durch seinen Fokus auf große Datenmengen und dem damit verbundenen Zeitaufwand für Training und Testing.

6.2 Zukünftige Arbeiten

Die vorliegende Arbeit hat einen Vergleich der verschiedenen Feature-Label Kombinationen sowie eine Optimierung durchgeführt. Was in der vorliegenden Arbeit jedoch nicht geleistet wurde ist ein umfassendes Feature Engineering, bspw. durch Einbindung externer Ressourcen. Inwiefern sich die Einbindung externer Quellen auf die Güte der Ergebnisse auswirkt wäre daher noch zu untersuchen.
Ein weiterer wichtiger Punkt ist die Untersuchung der Skalierung der Ergebnisse auf die Gesamtmenge

der Daten. Innerhalb der Experimente wurde nur eine kleine repräsentative Datenmenge verwendet. Inwiefern sich diese Ergebnisse auf die gesamte Datenmenge skalieren lassen und wie dabei Ressourcen optimal genutzt werden können wäre durch eine weitere Untersuchung interessant zu erfahren. Eine Möglichkeit wäre die Einbindung von Algorithmen die explizit für große Datenmengen geeignet sind (wie beispielsweise HOMER).

7 Anhang

Gütemaß	C = 10^(-6)	C = 10^(-4)	C = 10^(-2)	C = 10^(0)	C = 10^(2)	C = 10^(4)	C = 10^(6)
Hamming Loss	0.0013±0.0000	0.0013±0.0000	0.0013±0.0000	0.0014±0.0000	0.0016±0.0000	0.0017±0.0001	0.0017±0.0000
Subset Accuracy	0.0000±0.0000	0.0000±0.0000	0.0000±0.0000	0.0351±0.0099	0.0308±0.0095	0.0214±0.0094	0.0244±0.0098
Example-Based Precision	0.0000±0.0000	0.0000±0.0000	0.0000±0.0000	0.1439±0.0181	0.1396±0.0207	0.1286±0.0204	0.1315±0.0199
Example-Based Recall	0.0000±0.0000	0.0000±0.0000	0.0000±0.0000	0.0962±0.0147	0.1187±0.0182	0.1177±0.0186	0.1196±0.0206
Example-Based F Measure	0.0000±0.0000	0.0000±0.0000	0.0000±0.0000	0.1055±0.0136	0.1152±0.0158	0.1096±0.0149	0.1116±0.0170
Example-Based Accuracy	0.0000±0.0000	0.0000±0.0000	0.0000±0.0000	0.0847±0.0117	0.0894±0.0133	0.0825±0.0124	0.0847±0.0143
Example-Based Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.9998±0.0000	0.9996±0.0000	0.9995±0.0001	0.9995±0.0000
Micro-averaged Precision	0.0000±0.0000	0.0000±0.0000	0.0000±0.0000	0.4265±0.0515	0.2343±0.0248	0.2035±0.0236	0.2153±0.0233
Micro-averaged Recall	0.0000±0.0000	0.0000±0.0000	0.0000±0.0000	0.0834±0.0095	0.1019±0.0105	0.1008±0.0109	0.1030±0.0113
Micro-averaged F-Measure	0.0000±0.0000	0.0000±0.0000	0.0000±0.0000	0.1392±0.0150	0.1418±0.0135	0.1339±0.0108	0.1391±0.0141
Micro-averaged Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.9998±0.0000	0.9996±0.0000	0.9995±0.0001	0.9995±0.0000
Macro-averaged Precision	0.8103±0.0059	0.8103±0.0059	0.8103±0.0059	0.8115±0.0070	0.7949±0.0067	0.7904±0.0066	0.7903±0.0065
Macro-averaged Recall	0.8103±0.0059	0.8103±0.0059	0.8103±0.0059	0.8086±0.0071	0.7930±0.0069	0.7886±0.0068	0.7886±0.0069
Macro-averaged F-Measure	0.8103±0.0059	0.8103±0.0059	0.8103±0.0059	0.8092±0.0070	0.7931±0.0068	0.7886±0.0068	0.7886±0.0067
Macro-averaged Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.9998±0.0000	0.9995±0.0000	0.9995±0.0001	0.9995±0.0000
Average Precision	0.0061±0.0015	0.0061±0.0015	0.0061±0.0015	0.0915±0.0114	0.1002±0.0136	0.0917±0.0135	0.0946±0.0157
Coverage	1598.3480±38.7936	1598.3480±38.7936	1598.3480±38.7936	1499.8951±36.4468	1477.4672±41.3840	1481.6239±40.0146	1477.1283±40.2508
OneError	0.9970±0.0034	0.9970±0.0034	0.9970±0.0034	0.8582±0.0133	0.8630±0.0190	0.8865±0.0270	0.8792±0.0171
IsError	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.9619±0.0104	0.9602±0.0098	0.9704±0.0093	0.9666±0.0119
ErrorSetSize	3492.7106±130.6601	3492.7106±130.6601	3492.7106±130.6601	3177.2557±120.0729	3107.8852±117.5378	3111.9217±118.9854	3104.8640±122.2922
Ranking Loss	0.5329±0.0142	0.5329±0.0142	0.5329±0.0142	0.4780±0.0123	0.4654±0.0134	0.4661±0.0131	0.4651±0.0132
Mean Average Precision	0.0272±0.0019	0.0272±0.0019	0.0272±0.0019	0.0667±0.0098	0.0761±0.0095	0.0757±0.0094	0.0760±0.0095
Geometric Mean Average Precision	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Mean Average Interpolated Precision	0.0281±0.0020	0.0281±0.0020	0.0281±0.0020	0.0684±0.0098	0.0780±0.0095	0.0776±0.0094	0.0779±0.0095
Geometric Mean Average Interpolated Precision	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Micro-averaged AUC	0.5000±0.0000	0.5000±0.0000	0.5000±0.0000	0.5416±0.0047	0.5507±0.0052	0.5501±0.0054	0.5512±0.0056
Macro-averaged AUC	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Logarithmic Loss	54.2756±1.6849	54.2756±1.6849	54.2834±1.6786	55.9085±1.6566	66.8907±1.8019	70.7133±5.2973	69.1160±1.7465

Abbildung 7.1: 1.Experiment Messwerte 021A - 041A

Gütemaß	C = 10^(-6)	C = 10^(-4)	C = 10^(-2)	C = 10^(0)	C = 10^(2)	C = 10^(4)	C = 10^(6)
Hamming Loss	0.0013±0.0000	0.0013±0.0000	0.0013±0.0001	0.0013±0.0001	0.0013±0.0001	0.0013±0.0001	0.0013±0.0001
Subset Accuracy	0.0000±0.0000	0.0000±0.0000	0.0120±0.0037	0.0312±0.0143	0.0308±0.0140	0.0308±0.0129	0.0308±0.0129
Example-Based Precision	0.0000±0.0000	0.0000±0.0000	0.0830±0.0160	0.1742±0.0238	0.1722±0.0253	0.1727±0.0242	0.1724±0.0259
Example-Based Recall	0.0000±0.0000	0.0000±0.0000	0.0392±0.0081	0.0947±0.0144	0.0952±0.0151	0.0956±0.0136	0.0953±0.0142
Example-Based F Measure	0.0000±0.0000	0.0000±0.0000	0.0501±0.0098	0.1139±0.0167	0.1138±0.0176	0.1141±0.0165	0.1138±0.0172
Example-Based Accuracy	0.0000±0.0000	0.0000±0.0000	0.0389±0.0080	0.0899±0.0152	0.0895±0.0160	0.0897±0.0149	0.0895±0.0154
Example-Based Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.9999±0.0000	0.9999±0.0000	0.9999±0.0000	0.9999±0.0000
Micro-averaged Precision	0.0000±0.0000	0.0000±0.0000	0.7818±0.0611	0.6166±0.0620	0.5442±0.0682	0.5369±0.0661	0.5287±0.0647
Micro-averaged Recall	0.0000±0.0000	0.0000±0.0000	0.0377±0.0094	0.0956±0.0170	0.0961±0.0162	0.0965±0.0153	0.0963±0.0164
Micro-averaged F-Measure	0.0000±0.0000	0.0000±0.0000	0.0717±0.0171	0.1653±0.0272	0.1630±0.0252	0.1633±0.0240	0.1628±0.0259
Micro-averaged Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.9999±0.0000	0.9999±0.0000	0.9999±0.0000	0.9999±0.0000
Macro-averaged Precision	0.8093±0.0046	0.8093±0.0046	0.8142±0.0045	0.8190±0.0056	0.8166±0.0046	0.8164±0.0052	0.8154±0.0062
Macro-averaged Recall	0.8093±0.0046	0.8093±0.0046	0.8117±0.0044	0.8154±0.0054	0.8132±0.0046	0.8130±0.0052	0.8119±0.0060
Macro-averaged F-Measure	0.8093±0.0046	0.8093±0.0046	0.8122±0.0044	0.8163±0.0055	0.8141±0.0046	0.8139±0.0052	0.8128±0.0060
Macro-averaged Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.9999±0.0000	0.9999±0.0000	0.9999±0.0000	0.9999±0.0000
Average Precision	0.0061±0.0012	0.0061±0.0012	0.0446±0.0082	0.0952±0.0156	0.0945±0.0165	0.0948±0.0150	0.0947±0.0156
Coverage	1598.4368±39.9197	1598.4368±39.9197	1562.2803±35.7062	1508.0312±39.9160	1507.7803±41.3341	1507.3308±39.8708	1507.8962±40.4209
OneError	0.9970±0.0027	0.9970±0.0027	0.9162±0.0171	0.8278±0.0265	0.8312±0.0291	0.8308±0.0271	0.8308±0.0286
IsError	1.0000±0.0000	1.0000±0.0000	0.9876±0.0045	0.9679±0.0137	0.9684±0.0134	0.9684±0.0123	0.9684±0.0123
ErrorSetSize	3493.9115±149.4349	3493.9115±149.4349	3345.6726±147.9049	3118.9363±150.4387	3116.8556±154.6078	3115.3902±149.0549	3115.9607±152.5038
Ranking Loss	0.5328±0.0125	0.5328±0.0125	0.5094±0.0102	0.4773±0.0128	0.4768±0.0135	0.4765±0.0124	0.4768±0.0127
Mean Average Precision	0.0285±0.0058	0.0285±0.0058	0.0401±0.0086	0.0628±0.0128	0.0632±0.0125	0.0632±0.0125	0.0631±0.0127
Geometric Mean Average Precision	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Mean Average Interpolated Precision	0.0294±0.0058	0.0294±0.0058	0.0415±0.0087	0.0647±0.0129	0.0651±0.0127	0.0651±0.0127	0.0650±0.0129
Geometric Mean Average Interpolated Precision	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Micro-averaged AUC	0.5000±0.0000	0.5000±0.0000	0.5188±0.0047	0.5478±0.0085	0.5480±0.0081	0.5482±0.0076	0.5481±0.0082
Macro-averaged AUC	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Logarithmic Loss	54.3017±1.9531	54.3017±1.9531	52.8296±2.0841	52.3022±2.4272	53.4672±2.3848	53.5932±2.4327	53.7191±2.5351

Abbildung 7.2: 1.Experiment Messwerte 047I - 041A

Gütemaß	C = 10^(-6)	C = 10^(-4)	C = 10^(-2)	C = 10^(0)	C = 10^(2)	C = 10^(4)	C = 10^(6)
Hamming Loss	0.0013±0.0000	0.0013±0.0000	0.0013±0.0001	0.0013±0.0001	0.0013±0.0001	0.0013±0.0001	0.0013±0.0001
Subset Accuracy	0.0000±0.0000	0.0000±0.0000	0.0137±0.0050	0.0338±0.0130	0.0333±0.0140	0.0338±0.0138	0.0338±0.0138
Example-Based Precision	0.0000±0.0000	0.0000±0.0000	0.0901±0.0160	0.1810±0.0237	0.1842±0.0233	0.1836±0.0229	0.1836±0.0229
Example-Based Recall	0.0000±0.0000	0.0000±0.0000	0.0430±0.0080	0.0981±0.0153	0.1002±0.0165	0.1005±0.0162	0.1005±0.0162
Example-Based F Measure	0.0000±0.0000	0.0000±0.0000	0.0546±0.0099	0.1178±0.0176	0.1202±0.0183	0.1203±0.0180	0.1203±0.0180
Example-Based Accuracy	0.0000±0.0000	0.0000±0.0000	0.0426±0.0080	0.0930±0.0157	0.0948±0.0166	0.0949±0.0163	0.0949±0.0163
Example-Based Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.9999±0.0000	0.9999±0.0000	0.9999±0.0000	0.9999±0.0000
Micro-averaged Precision	0.0000±0.0000	0.0000±0.0000	0.7984±0.0708	0.6395±0.0656	0.6294±0.0591	0.6276±0.0602	0.6276±0.0602
Micro-averaged Recall	0.0000±0.0000	0.0000±0.0000	0.0410±0.0095	0.0979±0.0178	0.1002±0.0173	0.1002±0.0170	0.1002±0.0170
Micro-averaged F-Measure	0.0000±0.0000	0.0000±0.0000	0.0778±0.0171	0.1695±0.0283	0.1725±0.0271	0.1724±0.0267	0.1724±0.0267
Micro-averaged Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.9999±0.0000	0.9999±0.0000	0.9999±0.0000	0.9999±0.0000
Macro-averaged Precision	0.8093±0.0046	0.8093±0.0046	0.8143±0.0044	0.8193±0.0053	0.8196±0.0057	0.8195±0.0056	0.8195±0.0056
Macro-averaged Recall	0.8093±0.0046	0.8093±0.0046	0.8118±0.0043	0.8155±0.0051	0.8157±0.0054	0.8157±0.0054	0.8157±0.0054
Macro-averaged F-Measure	0.8093±0.0046	0.8093±0.0046	0.8123±0.0044	0.8164±0.0052	0.8166±0.0055	0.8166±0.0055	0.8166±0.0055
Macro-averaged Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.9999±0.0000	0.9999±0.0000	0.9999±0.0000	0.9999±0.0000
Average Precision	0.0061±0.0012	0.0061±0.0012	0.0482±0.0081	0.0989±0.0164	0.1008±0.0172	0.1009±0.0167	0.1009±0.0167
Coverage	1598.4368±39.9197	1598.4368±39.9197	1559.3124±36.7201	1503.9214±39.7522	1502.1496±40.0221	1501.6983±39.6649	1501.6983±39.6649
OneError	0.9970±0.0027	0.9970±0.0027	0.9094±0.0171	0.8184±0.0274	0.8145±0.0266	0.8150±0.0252	0.8150±0.0252
IsError	1.0000±0.0000	1.0000±0.0000	0.9859±0.0054	0.9654±0.0127	0.9654±0.0134	0.9650±0.0132	0.9650±0.0132
ErrorSetSize	3493.9115±149.4349	3493.9115±149.4349	3332.6526±143.3145	3106.7128±137.1302	3097.3543±139.8858	3097.3568±139.5056	3097.3568±139.5056
Ranking Loss	0.5328±0.0125	0.5328±0.0125	0.5074±0.0102	0.4748±0.0122	0.4734±0.0133	0.4732±0.0130	0.4732±0.0130
Mean Average Precision	0.0285±0.0058	0.0285±0.0058	0.0408±0.0090	0.0624±0.0129	0.0638±0.0126	0.0637±0.0125	0.0637±0.0125
Geometric Mean Average Precision	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Mean Average Interpolated Precision	0.0294±0.0058	0.0294±0.0058	0.0421±0.0090	0.0643±0.0130	0.0657±0.0127	0.0656±0.0126	0.0656±0.0126
Geometric Mean Average Interpolated Precision	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Micro-averaged AUC	0.5000±0.0000	0.5000±0.0000	0.5205±0.0047	0.5489±0.0089	0.5500±0.0086	0.5500±0.0085	0.5500±0.0085
Macro-averaged AUC	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Logarithmic Loss	54.3017±1.9531	54.3017±1.9531	52.6407±2.0920	51.9715±2.3574	52.0503±2.3243	52.0817±2.3337	52.0817±2.3337

Abbildung 7.3: 1.Experiment Messwerte 021A+047I - 041A

Name	C = 10^(-6)	C = 10^(-4)	C = 10^(-2)	C = 10^(0)	C = 10^(2)	C = 10^(4)	C = 10^(6)
Hamming Loss	0.0007±0.0000	0.0007±0.0000	0.0007±0.0000	0.0007±0.0000	0.0007±0.0000	0.0007±0.0000	0.0007±0.0000
Subset Accuracy	0.0000±0.0000	0.0000±0.0000	0.0028±0.0030	0.0829±0.0155	0.0808±0.0122	0.0804±0.0121	0.0804±0.0129
Example-Based Precision	0.0000±0.0000	0.0000±0.0000	0.0128±0.0044	0.2084±0.0215	0.2079±0.0203	0.2071±0.0200	0.2074±0.0199
Example-Based Recall	0.0000±0.0000	0.0000±0.0000	0.0065±0.0031	0.1410±0.0167	0.1538±0.0171	0.1540±0.0164	0.1544±0.0167
Example-Based F Measure	0.0000±0.0000	0.0000±0.0000	0.0081±0.0035	0.1566±0.0179	0.1641±0.0169	0.1640±0.0162	0.1643±0.0164
Example-Based Accuracy	0.0000±0.0000	0.0000±0.0000	0.0065±0.0031	0.1343±0.0162	0.1382±0.0149	0.1380±0.0144	0.1383±0.0147
Example-Based Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.9999±0.0000	0.9999±0.0000	0.9999±0.0000
Micro-averaged Precision	0.0000±0.0000	0.0000±0.0000	1.0000±0.0000	0.6868±0.0425	0.4585±0.0379	0.4410±0.0346	0.4411±0.0342
Micro-averaged Recall	0.0000±0.0000	0.0000±0.0000	0.0050±0.0020	0.1226±0.0151	0.1353±0.0155	0.1357±0.0153	0.1361±0.0156
Micro-averaged F-Measure	0.0000±0.0000	0.0000±0.0000	0.0100±0.0040	0.2078±0.0236	0.2086±0.0212	0.2073±0.0206	0.2077±0.0210
Micro-averaged Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.9999±0.0000	0.9999±0.0000	0.9999±0.0000
Macro-averaged Precision	0.8368±0.0056	0.8368±0.0056	0.8374±0.0057	0.8480±0.0051	0.8385±0.0053	0.8368±0.0049	0.8368±0.0050
Macro-averaged Recall	0.8368±0.0056	0.8368±0.0056	0.8371±0.0056	0.8455±0.0049	0.8366±0.0053	0.8350±0.0050	0.8350±0.0051
Macro-averaged F-Measure	0.8368±0.0056	0.8368±0.0056	0.8372±0.0056	0.8461±0.0050	0.8369±0.0053	0.8353±0.0050	0.8353±0.0050
Macro-averaged Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.9999±0.0000	0.9999±0.0000	0.9999±0.0000
Average Precision	0.0030±0.0008	0.0030±0.0008	0.0095±0.0032	0.1387±0.0171	0.1429±0.0164	0.1427±0.0159	0.1429±0.0162
Coverage	3070.6464±48.9329	3070.6464±48.9329	3060.4236±47.5053	2760.6441±79.6595	2735.5458±74.4907	2736.2513±74.6234	2734.4958±75.9979
OneError	0.9997±0.0010	0.9997±0.0010	0.9868±0.0046	0.7909±0.0266	0.7986±0.0277	0.7986±0.0278	0.7989±0.0274
IsError	1.0000±0.0000	1.0000±0.0000	0.9972±0.0030	0.9151±0.0150	0.9147±0.0133	0.9154±0.0132	0.9151±0.0136
ErrorSetSize	6736.1151±157.9348	6736.1151±157.9348	6704.7112±160.7820	5861.4462±209.7114	5764.4347±197.0432	5760.7726±198.3698	5757.8951±201.1731
Ranking Loss	0.5240±0.0112	0.5240±0.0112	0.5205±0.0114	0.4468±0.0148	0.4394±0.0130	0.4391±0.0128	0.4389±0.0129
Mean Average Precision	0.0231±0.0029	0.0231±0.0029	0.0248±0.0027	0.1050±0.0129	0.1139±0.0119	0.1147±0.0117	0.1149±0.0119
Geometric Mean Average Precision	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Mean Average Interpolated Precision	0.0234±0.0029	0.0234±0.0029	0.0251±0.0027	0.1060±0.0128	0.1150±0.0118	0.1158±0.0117	0.1159±0.0119
Geometric Mean Average Interpolated Precision	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Micro-averaged AUC	0.5000±0.0000	0.5000±0.0000	0.5025±0.0010	0.5613±0.0075	0.5676±0.0077	0.5678±0.0076	0.5680±0.0078
Macro-averaged AUC	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Logarithmic Loss	55.9711±1.3190	55.9711±1.3190	55.6902±1.3480	52.2094±1.5626	57.3456±1.7209	58.0031±1.6005	57.9968±1.5772

Abbildung 7.4: 1.Experiment Messwerte 021A - 044K

Name	C = 10^(-6)	C = 10^(-4)	C = 10^(-2)	C = 10^(0)	C = 10^(2)	C = 10^(4)	C = 10^(6)
Hamming Loss	0.0007±0.0000	0.0007±0.0000	0.0007±0.0000	0.0007±0.0000	0.0007±0.0000	0.0007±0.0000	0.0007±0.0000
Subset Accuracy	0.0000±0.0000	0.0000±0.0000	0.0125±0.0070	0.0678±0.0084	0.0671±0.0079	0.0619±0.0080	0.0644±0.0104
Example-Based Precision	0.0000±0.0000	0.0000±0.0000	0.0621±0.0094	0.1615±0.0166	0.1650±0.0167	0.1630±0.0180	0.1623±0.0162
Example-Based Recall	0.0000±0.0000	0.0000±0.0000	0.0323±0.0088	0.1091±0.0096	0.1126±0.0107	0.1139±0.0120	0.1130±0.0116
Example-Based F Measure	0.0000±0.0000	0.0000±0.0000	0.0396±0.0092	0.1209±0.0108	0.1242±0.0118	0.1237±0.0131	0.1231±0.0119
Example-Based Accuracy	0.0000±0.0000	0.0000±0.0000	0.0320±0.0087	0.1053±0.0098	0.1076±0.0103	0.1058±0.0111	0.1060±0.0106
Example-Based Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000
Micro-averaged Precision	0.0000±0.0000	0.0000±0.0000	0.9141±0.0847	0.7559±0.0422	0.6956±0.0530	0.6318±0.0452	0.6474±0.0513
Micro-averaged Recall	0.0000±0.0000	0.0000±0.0000	0.0280±0.0065	0.0940±0.0109	0.0972±0.0128	0.0984±0.0132	0.0974±0.0129
Micro-averaged F-Measure	0.0000±0.0000	0.0000±0.0000	0.0542±0.0122	0.1671±0.0176	0.1702±0.0202	0.1700±0.0205	0.1691±0.0205
Micro-averaged Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000
Macro-averaged Precision	0.8377±0.0065	0.8377±0.0065	0.8407±0.0061	0.8472±0.0070	0.8463±0.0076	0.8451±0.0072	0.8453±0.0075
Macro-averaged Recall	0.8377±0.0065	0.8377±0.0065	0.8395±0.0060	0.8454±0.0069	0.8445±0.0075	0.8435±0.0072	0.8437±0.0075
Macro-averaged F-Measure	0.8377±0.0065	0.8377±0.0065	0.8398±0.0060	0.8459±0.0069	0.8449±0.0075	0.8438±0.0072	0.8440±0.0075
Macro-averaged Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000
Average Precision	0.0030±0.0005	0.0030±0.0005	0.0350±0.0082	0.1094±0.0092	0.1122±0.0096	0.1117±0.0110	0.1116±0.0106
Coverage	3069.2792±47.7572	3069.2792±47.7572	3016.5163±35.3807	2819.3280±53.7980	2813.7277±53.5236	2810.6453±54.7114	2812.1083±57.3570
OneError	0.9997±0.0010	0.9997±0.0010	0.9377±0.0088	0.8377±0.0173	0.8343±0.0167	0.8349±0.0193	0.8353±0.0169
IsError	1.0000±0.0000	1.0000±0.0000	0.9875±0.0070	0.9311±0.0080	0.9315±0.0075	0.9349±0.0084	0.9332±0.0107
ErrorSetSize	6729.1301±265.1340	6729.1301±265.1340	6550.0941±229.6799	6078.6381±268.7018	6056.7090±276.9731	6047.6284±275.3772	6055.4567±278.7859
Ranking Loss	0.5239±0.0104	0.5239±0.0104	0.5090±0.0091	0.4673±0.0118	0.4650±0.0115	0.4644±0.0118	0.4650±0.0117
Mean Average Precision	0.0240±0.0036	0.0240±0.0036	0.0359±0.0092	0.0822±0.0107	0.0848±0.0116	0.0854±0.0108	0.0848±0.0108
Geometric Mean Average Precision	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Mean Average Interpolated Precision	0.0243±0.0036	0.0243±0.0036	0.0364±0.0091	0.0829±0.0107	0.0855±0.0116	0.0861±0.0107	0.0856±0.0107
Geometric Mean Average Interpolated Precision	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Micro-averaged AUC	0.5000±0.0000	0.5000±0.0000	0.5140±0.0032	0.5470±0.0054	0.5486±0.0064	0.5492±0.0066	0.5487±0.0065
Macro-averaged AUC	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Logarithmic Loss	55.9250±1.5175	55.9250±1.5175	54.5036±1.3486	52.3747±1.7854	52.9037±2.0144	53.6431±2.0146	53.4583±2.1634

Abbildung 7.5: 1.Experiment Messwerte 047I - 044K

Name	C = 10^(-6)	C = 10^(-4)	C = 10^(-2)	C = 10^(0)	C = 10^(2)	C = 10^(4)	C = 10^(6)
Hamming Loss	0.0007±0.0000	0.0007±0.0000	0.0007±0.0000	0.0007±0.0000	0.0007±0.0000	0.0007±0.0000	0.0007±0.0000
Subset Accuracy	0.0000±0.0000	0.0000±0.0000	0.0176±0.0081	0.0754±0.0108	0.0768±0.0102	0.0768±0.0102	0.0768±0.0102
Example-Based Precision	0.0000±0.0000	0.0000±0.0000	0.0712±0.0096	0.1750±0.0179	0.1784±0.0153	0.1784±0.0153	0.1784±0.0153
Example-Based Recall	0.0000±0.0000	0.0000±0.0000	0.0388±0.0084	0.1164±0.0119	0.1191±0.0115	0.1191±0.0115	0.1191±0.0115
Example-Based F Measure	0.0000±0.0000	0.0000±0.0000	0.0468±0.0088	0.1297±0.0132	0.1325±0.0121	0.1325±0.0121	0.1325±0.0121
Example-Based Accuracy	0.0000±0.0000	0.0000±0.0000	0.0386±0.0083	0.1135±0.0117	0.1159±0.0109	0.1159±0.0109	0.1159±0.0109
Example-Based Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000
Micro-averaged Precision	0.0000±0.0000	0.0000±0.0000	0.9250±0.0727	0.8567±0.0297	0.8383±0.0290	0.8383±0.0290	0.8383±0.0290
Micro-averaged Recall	0.0000±0.0000	0.0000±0.0000	0.0336±0.0075	0.1009±0.0121	0.1027±0.0113	0.1027±0.0113	0.1027±0.0113
Micro-averaged F-Measure	0.0000±0.0000	0.0000±0.0000	0.0648±0.0141	0.1803±0.0193	0.1828±0.0179	0.1828±0.0179	0.1828±0.0179
Micro-averaged Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000
Macro-averaged Precision	0.8377±0.0065	0.8377±0.0065	0.8412±0.0062	0.8496±0.0066	0.8495±0.0067	0.8495±0.0067	0.8495±0.0067
Macro-averaged Recall	0.8377±0.0065	0.8377±0.0065	0.8400±0.0062	0.8475±0.0067	0.8475±0.0067	0.8475±0.0067	0.8475±0.0067
Macro-averaged F-Measure	0.8377±0.0065	0.8377±0.0065	0.8403±0.0062	0.8481±0.0066	0.8481±0.0067	0.8481±0.0067	0.8481±0.0067
Macro-averaged Specificity	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000
Average Precision	0.0030±0.0005	0.0030±0.0005	0.0416±0.0080	0.1168±0.0115	0.1193±0.0109	0.1193±0.0109	0.1193±0.0109
Coverage	3069.2792±47.7572	3069.2792±47.7572	3005.5145±35.5568	2804.7436±52.0619	2796.7744±54.7183	2796.7744±54.7183	2796.7744±54.7183
OneError	0.9997±0.0010	0.9997±0.0010	0.9287±0.0099	0.8266±0.0179	0.8228±0.0153	0.8228±0.0153	0.8228±0.0153
IsError	1.0000±0.0000	1.0000±0.0000	0.9824±0.0081	0.9239±0.0108	0.9228±0.0107	0.9228±0.0107	0.9228±0.0107
ErrorSetSize	6729.1301±265.1340	6729.1301±265.1340	6515.8740±240.0150	6037.6976±267.6221	6023.3574±266.3111	6023.3574±266.3111	6023.3574±266.3111
Ranking Loss	0.5239±0.0104	0.5239±0.0104	0.5061±0.0097	0.4634±0.0112	0.4617±0.0110	0.4617±0.0110	0.4617±0.0110
Mean Average Precision	0.0240±0.0036	0.0240±0.0036	0.0387±0.0098	0.0878±0.0094	0.0890±0.0093	0.0890±0.0093	0.0890±0.0093
Geometric Mean Average Precision	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Mean Average Interpolated Precision	0.0243±0.0036	0.0243±0.0036	0.0392±0.0097	0.0885±0.0093	0.0897±0.0093	0.0897±0.0093	0.0897±0.0093
Geometric Mean Average Interpolated Precision	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Micro-averaged AUC	0.5000±0.0000	0.5000±0.0000	0.5168±0.0038	0.5504±0.0061	0.5513±0.0057	0.5513±0.0057	0.5513±0.0057
Macro-averaged AUC	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Logarithmic Loss	55.9250±1.5175	55.9250±1.5175	54.1849±1.4603	51.2338±1.6247	51.2975±1.6066	51.2975±1.6066	51.2975±1.6066

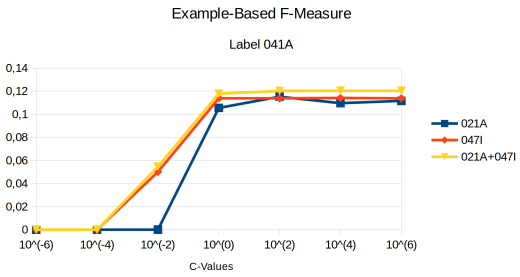
Abbildung 7.6: 1.Experiment Messwerte 021A+047I - 044K

Gütemaß	P = 0	P = 5000	P = 10000	P = 20000	P = 30000	P = 40000	P = 50000	P = 60000
Hamming Loss	0.0014±0.0001	0.0018±0.0001	0.0018±0.0001	0.0018±0.0001	0,0018±0,0001	0.0017±0.0001	0,0017±0,0001	0,0017±0,0001
Subset Accuracy	0.0347±0.0107	0.0283±0.0107	0.0300±0.0105	0.0300±0.0094	0,0291±0,0126	0.0338±0.0120	0,0334±0,0143	0,0373±0,0110
Example-Based Precision	0.1451±0.0233	0.1324±0.0139	0.1368±0.0149	0.1428±0.0125	0,1519±0,0204	0.1573±0.0193	0,1581±0,0138	0,1610±0,0189
Example-Based Recall	0.1051±0.0214	0.1350±0.0167	0.1362±0.0125	0.1395±0.0122	0,1409±0,0191	0.1402±0.0154	0,1377±0,0136	0,1399±0,0172
Example-Based F Measure	0.1100±0.0184	0.1194±0.0120	0.1212±0.0117	0.1237±0.0105	0,1282±0,0161	0.1310±0.0154	0,1303±0,0125	0,1335±0,0165
Example-Based Accuracy	0.0874±0.0152	0.0900±0.0091	0.0918±0.0104	0.0933±0.0081	0,0964±0,0134	0.0998±0.0134	0,0992±0,0113	0,1029±0,0136
Example-Based Specificity	0.9998±0.0001	0.9994±0.0001	0.9993±0.0001	0.9993±0.0001	0,9994±0,0000	0.9994±0.0000	0,9994±0,0000	0,9995±0,0001
Micro-averaged Precision	0.3730±0.0915	0.1893±0.0154	0.1844±0.0171	0.1931±0.0154	0,2036±0,0221	0.2144±0.0242	0,2200±0,0181	0,2296±0,0264
Micro-averaged Recall	0.0902±0.0151	0.1134±0.0139	0.1138±0.0124	0.1177±0.0143	0,1210±0,0155	0.1191±0.0129	0,1174±0,0119	0,1193±0,0154
Micro-averaged F-Measure	0.1423±0.0171	0.1415±0.0142	0.1405±0.0131	0.1460±0.0147	0,1515±0,0175	0.1530±0.0162	0,1529±0,0136	0,1567±0,0184
Micro-averaged Specificity	0.9998±0.0001	0.9994±0.0001	0.9993±0.0001	0.9993±0.0001	0,9994±0,0000	0.9994±0.0000	0,9994±0,0000	0,9995±0,0001
Macro-averaged Precision	0.8067±0.0117	0.7746±0.0068	0.7707±0.0066	0.7719±0.0071	0,7729±0,0089	0.7771±0.0090	0,7797±0,0084	0,7812±0,0093
Macro-averaged Recall	0.8040±0.0117	0.7730±0.0074	0.7689±0.0073	0.7700±0.0078	0,7706±0,0089	0.7742±0.0089	0,7769±0,0084	0,7781±0,0095
Macro-averaged F-Measure	0.8044±0.0118	0.7727±0.0072	0.7688±0.0071	0.7699±0.0076	0,7706±0,0088	0.7744±0.0090	0,7772±0,0085	0,7785±0,0095
Macro-averaged Specificity	0.9998±0.0001	0.9993±0.0001	0.9993±0.0001	0.9993±0.0001	0,9994±0,0000	0.9994±0.0000	0,9994±0,0000	0,9995±0,0001
Average Precision	0.0960±0.0155	0.1016±0.0106	0.1040±0.0117	0.1075±0.0094	0,1118±0,0139	0.1133±0.0115	0,1129±0,0105	0,1165±0,0131
Coverage	1490.7032±35.4871	1468.2744±33.4538	1462.2742±39.8982	1456.4940±38.9619	1464,1922±40,4118	1467.3855±39.5629	1475,7314±39,4698	1471,5688±39,2455
OneError	0.8574±0.0185	0.8796±0.0142	0.8741±0.0226	0.8642±0.0212	0,8493±0,0201	0.8497±0.0144	0,8497±0,0143	0,8446±0,0154
IsError	0.9597±0.0094	0.9627±0.0094	0.9597±0.0100	0.9585±0.0084	0,9572±0,0131	0.9550±0.0133	0,9550±0,0156	0,9520±0,0127
ErrorSetSize	3149.9959±104.2120	3076.6437±137.5322	3073.8059±145.6497	3055.5262±151.2343	3054,1440±155,4889	3061.0439±144.7869	3067,5920±145,4158	3060,7602±151,8309
Ranking Loss	0.4728±0.0116	0.4595±0.0132	0.4586±0.0151	0.4557±0.0146	0,4572±0,0146	0.4582±0.0150	0,4597±0,0147	0,4582±0,0155
Mean Average Precision	0.0705±0.0103	0.0923±0.0156	0.0976±0.0145	0.1032±0.0147	0,1085±0,0154	0.1096±0.0154	0,1112±0,0145	0,1126±0,0156
Geometric Mean Average Precision	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Mean Average Interpolated Precision	0.0722±0.0103	0.0944±0.0156	0.0995±0.0143	0.1052±0.0145	0,1107±0,0154	0.1117±0.0155	0,1132±0,0145	0,1147±0,0156
Geometric Mean Average Interpolated Precision	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Micro-averaged AUC	0.5450±0.0075	0.5564±0.0070	0.5566±0.0062	0.5585±0.0071	0,5602±0,0078	0.5593±0.0065	0,5584±0,0059	0,5594±0,0077
Macro-averaged AUC	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Logarithmic Loss	58.8448±5.4762	74.4558±2.1311	75.4899±2.7096	74.5592±2.5351	73,3841±2,3946	71.5602±2.4914	70,4969±2,2245	69,6045±3,0994

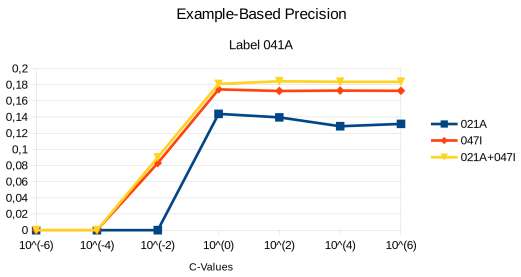
Abbildung 7.7: 2.Experiment Messwerte 021A - 041A

Gütemaß	P = 0	P = 10000	P = 20000	P = 30000	P = 40000	P = 50000	P = 70000	P = 90000	P = 100000	P = 110000
Hamming Loss	0.0007±0.0000	0,0008±0,0001	0.0007±0.0000	0,0011±0,0000	0.0008±0.0001	0,0008±0,0001	0,0010±0,0001	0,0009±0,0002	0,0010±0,0000	0,0010±0,0001
Subset Accuracy	0.0836±0.0151	0,0825±0,0164	0.0901±0.0188	0,0745±0,0158	0.0884±0.0195	0,0877±0,0146	0,0818±0,0177	0,0856±0,0115	0,0818±0,0155	0,0856±0,0181
Example-Based Precision	0.2071±0.0185	0,2766±0,0293	0.3072±0.0186	0,2463±0,0215	0.3286±0.0269	0,3294±0,0336	0,2950±0,0369	0,3178±0,0423	0,2994±0,0234	0,3078±0,0270
Example-Based Recall	0.1474±0.0148	0,2043±0,0199	0.2156±0.0177	0,2424±0,0203	0.2366±0.0135	0,2447±0,0232	0,2627±0,0184	0,2615±0,0242	0,2751±0,0225	0,2735±0,0213
Example-Based F Measure	0.1601±0.0153	0,2158±0,0170	0.2344±0.0169	0,2226±0,0177	0.2530±0.0170	0,2572±0,0142	0,2526±0,0211	0,2614±0,0194	0,2597±0,0204	0,2627±0,0181
Example-Based Accuracy	0.1365±0.0140	0,1749±0,0162	0.1908±0.0170	0,1745±0,0164	0.2036±0.0180	0,2059±0,0143	0,1986±0,0210	0,2072±0,0176	0,2033±0,0190	0,2067±0,0182
Example-Based Specificity	0.9999±0.0000	0,9998±0,0001	0.9998±0.0000	0,9995±0,0000	0.9998±0.0001	0,9998±0,0001	0,9996±0,0001	0,9996±0,0002	0,9995±0,0000	0,9995±0,0001
Micro-averaged Precision	0.5564±0.1298	0,4204±0,0890	0.4642±0.0241	0,2238±0,0107	0.4580±0.0820	0,4400±0,0961	0,2979±0,1005	0,3553±0,1239	0,2667±0,0216	0,2945±0,0756
Micro-averaged Recall	0.1288±0.0148	0,1826±0,0214	0.1929±0.0186	0,2185±0,0196	0.2129±0.0143	0,2199±0,0207	0,2369±0,0166	0,2349±0,0213	0,2493±0,0201	0,2468±0,0191
Micro-averaged F-Measure	0.2070±0.0208	0,2498±0,0256	0.2723±0.0221	0,2209±0,0145	0.2870±0.0262	0,2866±0,0245	0,2576±0,0327	0,2727±0,0331	0,2576±0,0200	0,2645±0,0239
Micro-averaged Specificity	0.9999±0.0000	0,9998±0,0001	0.9998±0.0000	0,9995±0,0000	0.9998±0.0001	0,9998±0,0001	0,9996±0,0001	0,9996±0,0002	0,9995±0,0000	0,9995±0,0001
Macro-averaged Precision	0.8420±0.0098	0,8315±0,0126	0.8394±0.0053	0,7894±0,0064	0.8375±0.0184	0,8331±0,0197	0,8014±0,0236	0,8128±0,0272	0,7943±0,0081	0,7999±0,0177
Macro-averaged Recall	0.8399±0.0094	0,8300±0,0124	0.8376±0.0052	0,7907±0,0057	0.8361±0.0173	0,8316±0,0189	0,8016±0,0224	0,8127±0,0260	0,7951±0,0082	0,8003±0,0169
Macro-averaged F-Measure	0.8403±0.0096	0,8300±0,0126	0.8377±0.0053	0,7890±0,0061	0.8359±0.0180	0,8315±0,0193	0,8004±0,0231	0,8117±0,0268	0,7935±0,0082	0,7990±0,0174
Macro-averaged Specificity	0.9999±0.0000	0,9998±0,0001	0.9998±0.0000	0,9995±0,0000	0.9998±0.0001	0,9998±0,0002	0,9996±0,0001	0,9996±0,0002	0,9995±0,0000	0,9995±0,0001
Average Precision	0.1407±0.0152	0,1818±0,0159	0.1981±0.0181	0,1853±0,0187	0.2134±0.0190	0,2158±0,0160	0,2092±0,0235	0,2187±0,0209	0,2161±0,0222	0,2182±0,0199
Coverage	2747.2330±73.8114	2655,7649±75,5613	2643.3902±78.3067	2597,6596±77,4875	2608.5217±71.1597	2592,7801±91,1839	2564,2228±63,4331	2564,5277±83,2744	2546,8577±76,7286	2541,8864±84,6746
OneError	0.7958±0.0267	0,7330±0,0324	0.6959±0.0212	0,7715±0,0282	0.6741±0.0324	0,6789±0,0323	0,7213±0,0505	0,6942±0,0498	0,7157±0,0289	0,7095±0,0328
IsError	0.9140±0.0144	0,9106±0,0144	0.9015±0.0176	0,9123±0,0139	0.8998±0.0197	0,9008±0,0161	0,9074±0,0182	0,9029±0,0149	0,9047±0,0173	0,9029±0,0181
ErrorSetSize	5812.1337±190.0736	5447,8314±203,2585	5370.9673±208.2748	5210,3490±212,4168	5253.1710±191.1679	5198,3976±215,1393	5085,7234±187,5543	5099,4715±197,9850	5006,0627±207,1657	5023,5012±197,9740
Ranking Loss	0.4430±0.0141	0,4125±0,0157	0.4061±0.0144	0,3934±0,0143	0.3967±0.0118	0,3923±0,0155	0,3822±0,0113	0,3834±0,0151	0,3767±0,0139	0,3776±0,0124
Mean Average Precision	0.1087±0.0123	0,1578±0,0170	0.1685±0.0148	0,1867±0,0146	0.1897±0.0115	0,1971±0,0159	0,2109±0,0134	0,2113±0,0177	0,2228±0,0164	0,2219±0,0173
Geometric Mean Average Precision	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Mean Average Interpolated Precision	0.1097±0.0123	0,1587±0,0170	0.1694±0.0148	0,1877±0,0146	0.1907±0.0116	0,1981±0,0160	0,2120±0,0134	0,2124±0,0178	0,2240±0,0164	0,2231±0,0173
Geometric Mean Average Interpolated Precision	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Micro-averaged AUC	0.5644±0.0074	0,5912±0,0107	0.5964±0.0093	0,6090±0,0098	0.6063±0.0071	0,6098±0,0103	0,6182±0,0083	0,6172±0,0106	0,6244±0,0100	0,6232±0,0095
Macro-averaged AUC	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN	NaN±NaN
Logarithmic Loss	55.2731±3.8373	61,7300±8,4876	57.6063±1.7409	86,0992±2,0754	59.8521±9.2101	61,9960±10,8858	77,6298±10,8712	71,5801±13,4709	80,4287±3,4147	77,2843±7,9012

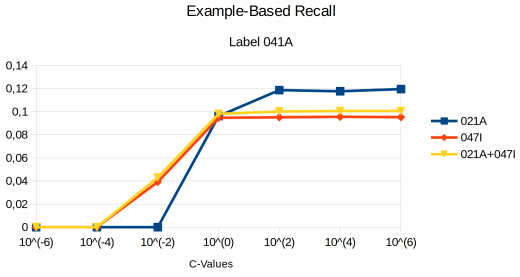
Abbildung 7.8: 2.Experiment Messwerte 021A - 044K



(a) Example-based F-Measure

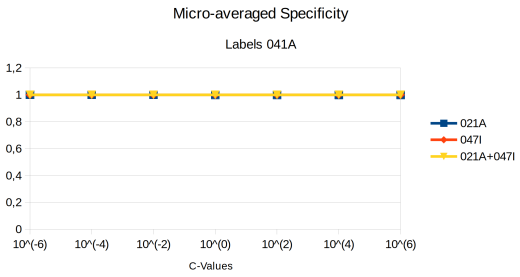


(b) Example-based Precision

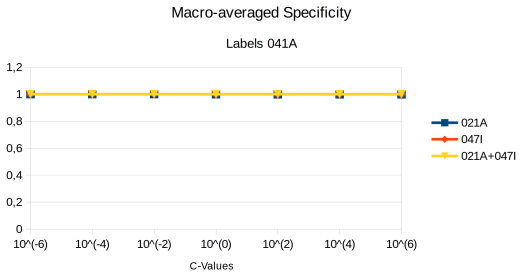


(c) Example-based Recall

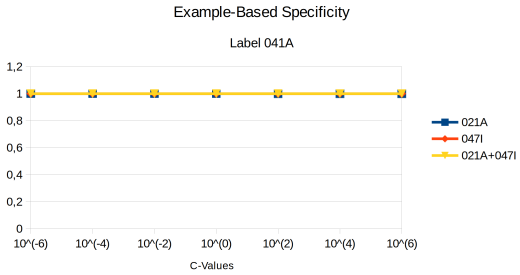
Abbildung 7.9: 1. Experiment - 041A (RSWK-Schlagwort) - Example-based Gütewerte



(a) Micro-averaged Specificity

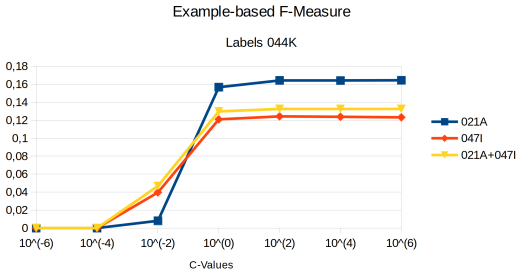


(b) Macro-averaged Specificity

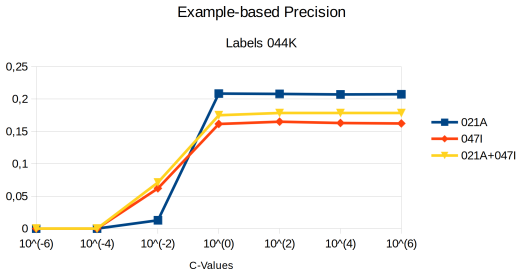


(c) Example-based Specificity

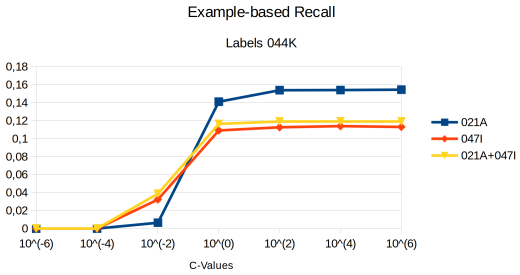
Abbildung 7.10: 1. Experiment - 041A (RSWK-Schlagwort) - Specificity



(a) Example-based F-Measure

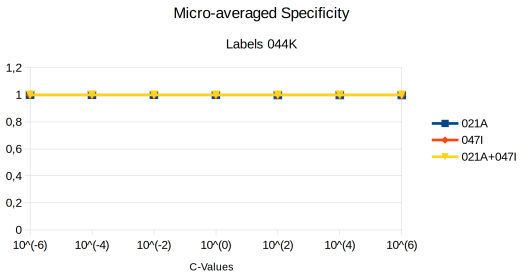


(b) Example-based Precision

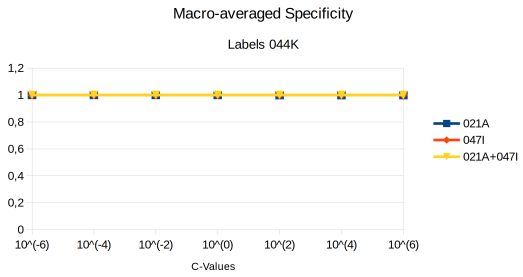


(c) Example-based Recall

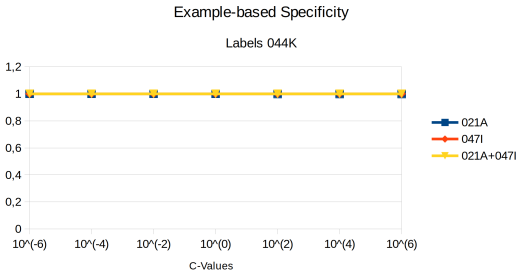
Abbildung 7.11: 1. Experiment - 044K (Einzelschlagwort) - Example-based Gütewerte



(a) Micro-averaged Specificity



(b) Macro-averaged Specificity



(c) Example-based Specificity

Abbildung 7.12: 1. Experiment - 044K (Einzelschlagwort) - Specificity

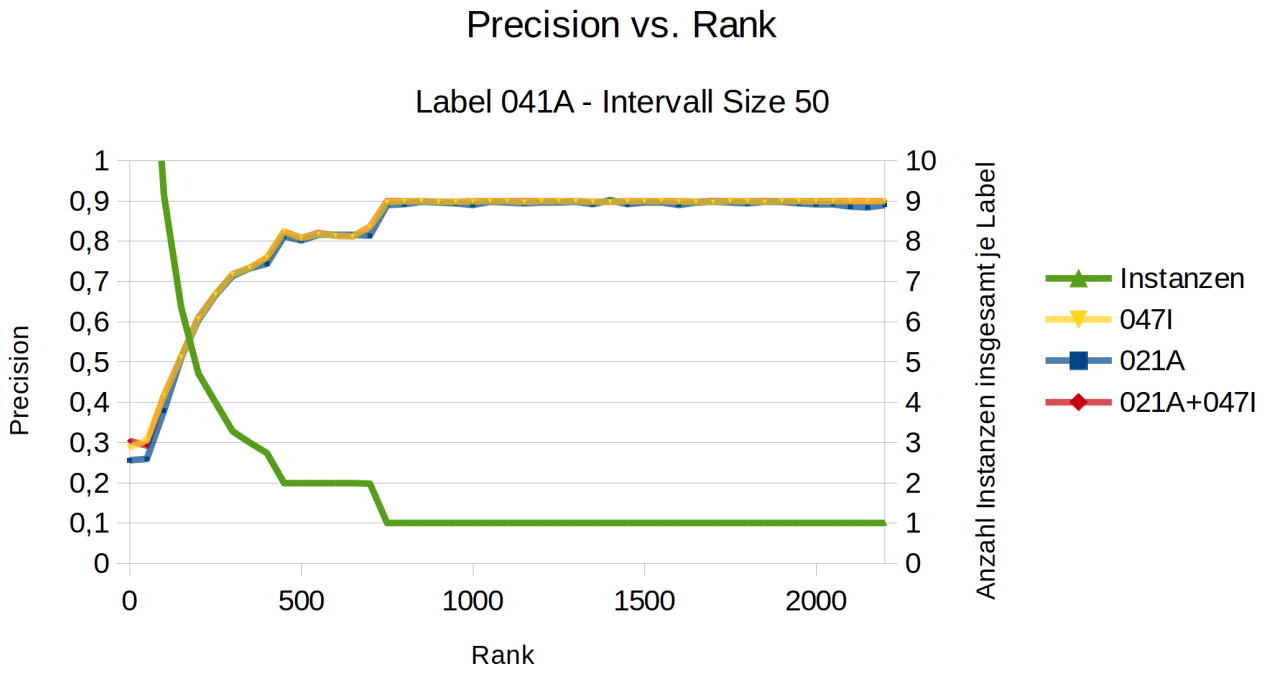


Abbildung 7.13: 1. Experiment Ranking vs. Precision - 041A (RSWK-Schlagwort)

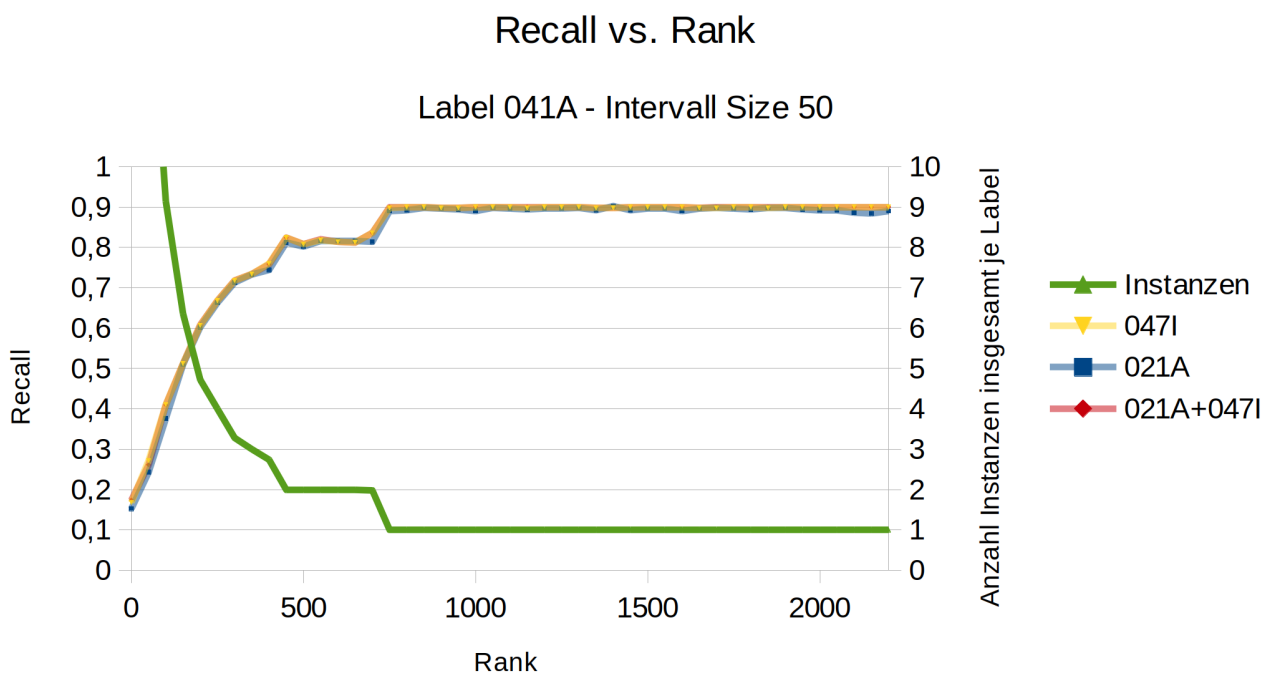


Abbildung 7.14: 1. Experiment Ranking vs. Recall - 041A (RSWK-Schlagwort)

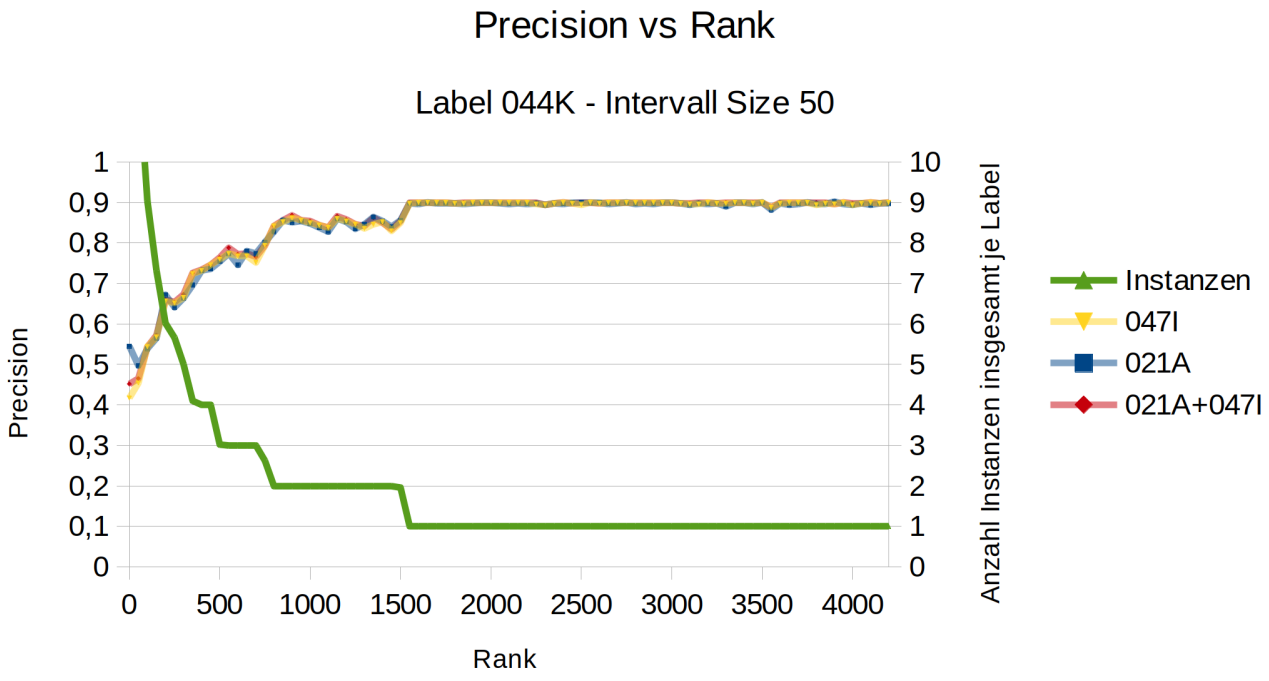


Abbildung 7.15: 1. Experiment Ranking vs. Precision - 044K (Einzelschlagwort)

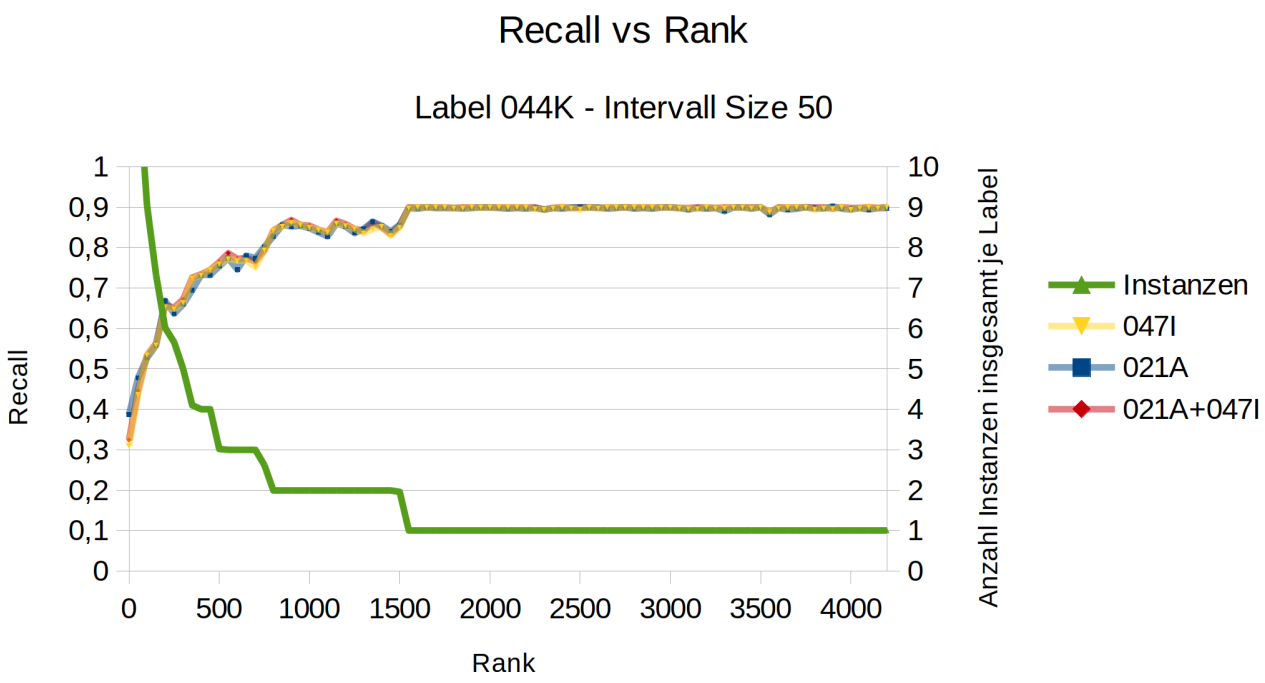


Abbildung 7.16: 1. Experiment Ranking vs. Recall - 044K (Einzelschlagwort)

7.1 PICA-Bezeichner

PICA3 Bereich	Beschreibung	PICA+ Felder
0XXX	Verarbeitungsangaben (Datum, Uhreit, interne Satznummer)	001@, 001B, 001D, 002@, 001X, 001U, 006Y, 037H, 010E, 037G, 001X, 003@, 028A, 029A, 022A, 041A, 045A, 001A, 001B, 001D, 001U, 044F, 002@, 002C, 002D, 002E, 016B, 017A, 017B, 039I, 046G, 003@, 035E, 009Q
1XXX	Codierte Angaben, incl. Erscheinungsjahr	016A, 016E, 011F, 011B, 013@/01, 013D, 013F, 012@, 015@, 010@, 010E, 038L, 019@, 018@, 011@, 013H/0X

2XXX	Identifikationsnummern aller Art	004A, 005I, 004D, 005A, 005J, 005K, 005P, 004G, 004H, 005G, 005B, 004F, 004I, 006B, 004L, 004M, 004K, 004C, 004U, 004P, 004R, 007L, 006G, 006A, 006C, 003O, 006T, 006U, 006Z, 006S, 006L, 006D, 006H, 006N, 006Y, 007C, 007F, 007H, 007M, 007E, 007B, 007G, 007D, 007A, 007I, 007P, 007S, 007T, 004E/01, 004E/02, 004E/03, 004E/04, 004E/05, 004E/06, 004E/07
30XX	Personennamen	028A, 028C, 028B/0Y, 028C/D/0X, 028E/0X, 028F/0X
31XX	Körperschaftsnamen	029A, 029F, 029A/0X, 029F/G/0X, 029E/0X
32XX	Sachtitel für Nebeneintragungen	022A, 022A/01, 025@, 026C, 027A/09, 022S/0X, 027A/00-08, 055A/0X, 055A/1X/2X, 055B/1X
4XXX	Titelbeschreibung incl. Fußnoten	021A, 021B, 021C, 021M, 021N, 032@, 032B, 032C, 031N, 031@, 035E, 037H, 033A, 033E, 033B, 033C, 033N, 033O, 034D, 034M, 034I, 034K, 033P, 033Q, 031A, 009Q, 009P, 009R, 036L/09, 036A, 036B, 036C, 036D, 047C, 037A, 037B, 047P, 037C, 037D, 047I, 020F, 046A, 046B, 046C, 046D, 046F, 046G, 046I, 046J, 046K, 046L, 046M, 046S, 046O, 046P, 046Q, 046R, 046X, 037F, 037G, 039B, 039C, 039D, 039E, 039S, 039Z, 048H/01, 039I, 039X, 039T, 039U, 060B, 060C, 047A, 047G, 047B, 047N, 047M, 047R, 047E, 036L/00-02, 036M/00-02, 036A/0Y, 036C/0Y, 036E/0X, 036F/0X, 036G/0X

5XXX	Sacherschließung	045B, 045A, 045E, 045T, 045U, 045Z, 041A/80, 041A/90, 045X, 045V, 045W, 045K, 045F, 045F/01, 045F/02, 045F/03, 045F/04, 044A, 044B, 044F, 044K, 044L, 044M, 044N, 044O, 044P, 044Q, 044G, 052A, 041A/0X-1X, 041K/XX, 045Q, 045R, 045G-045J, 045G-J/01, 045G-J/02, 045G-J/03, 045G-J/04
6XXX	Lokaldaten (lokale Sacherschließung etc.)	145B, 244Z/XY
7XXX	Exemplardaten (eigener Datensatz)	209A/XY, 231@, 209S, 204U/XY, 204P/XY, 204R/XY, 209T/XY, 209U, 209F, 203@/XY, 201C/XY, 201B/XY, 201D/XY
8XXX	weitere Exemplardaten (eigener Datensatz)	209E/XY, 205B/XY, 209C/XY, 209G/XY, 233O/XY, 233Q/XY, 233R/XY, 233P/XY, 245G/XY, 206V/XY, 206W/XY, 206x/XY

Tabelle 7.1: Bereiche und Beschreibung der Bezeichner des PICA3 Formats

Literatur

[AC75] Henriette D. Avram und Library of Congress. *MARC, its history and implications*. English. Library of Congress Washington, 1975, 49 p. ISBN: 0844401765.

[CK01] Amanda Clare und Ross D. King. „Knowledge Discovery in Multi-label Phenotype Data“. In: *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*. PKDD '01. London, UK, UK: Springer-Verlag, 2001, S. 42–53. ISBN: 3-540-42534-9. URL: http://dl.acm.org/citation.cfm?id=645805.670013.

[Din] *Format für den Austausch von bibliographischen Daten*. Standard. Berlin, DE: Deutsches Institut für Normung, 1978.

[Eve94] Bertram Eversberg. *Was sind und was sollen Bibliothekarische Datenformate*. Bd. 9. TU Braunschweig: Univ.-Bibl. der TU Braunschweig, 1994. ISBN: 3-927115-21-5.

[EW01] André Elisseeff und Jason Weston. „A Kernel Method for Multi-Labelled Classification“. In: *In Advances in Neural Information Processing Systems 14*. MIT Press, 2001, S. 681–687.

[Fan+08] Rong-En Fan u. a. „LIBLINEAR: A library for large linear classification“. In: *Journal of machine learning research* 9.Aug (2008), S. 1871–1874.

[GS04] Shantanu Godbole und Sunita Sarawagi. „Discriminative Methods for Multi-labeled Classification“. In: *Advances in Knowledge Discovery and Data Mining: 8th Pacific-Asia Conference, PAKDD 2004, Sydney, Australia, May 26-28, 2004. Proceedings*. Hrsg. von Honghua Dai, Ramakrishnan Srikant und Chengqi Zhang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, S. 22–30. ISBN: 978-3-540-24775-3. DOI: 10.1007/978-3-540-24775-3_5. URL: http://dx.doi.org/10.1007/978-3-540-24775-3_5.

[GV15] Eva Gibaja und Sebastián Ventura. „A Tutorial on Multilabel Learning“. In: *ACM Comput. Surv.* 47.3 (Apr. 2015), 52:1–52:38. ISSN: 0360-0300. DOI: 10.1145/2716262. URL: http://doi.acm.org/10.1145/2716262.

[Iso] *ISO 2709:2008 - Information and documentation – Format for information exchange*. Standard. Geneva, CH: International Organization for Standardization, 2008.

[Kro13] Angela Kroeger. „The Road to BIBFRAME: The Evolution of the Idea of Bibliographic Transition into a Post-MARC Future“. In: *Cataloging & Classification Quarterly* 51.8 (2013), S. 873–890. DOI: 10.1080/01639374.2013.823584. eprint: http://dx.doi.org/10.1080/01639374.2013.823584. URL: http://dx.doi.org/10.1080/01639374.2013.823584.

[Laz15] Fotis Lazarinis. „Cataloguing and Classification“. In: Boston: Chandos Publishing, 2015. ISBN: 978-0-08-100161-5. DOI: http://dx.doi.org/10.1016/B978-0-08-100161-5.09979-6. URL: http://www.sciencedirect.com/science/article/pii/B9780081001615099796.

[LMF10] Eneldo Loza Mencía und Johannes Fürnkranz. „Semantic Processing of Legal Texts“. In: Hrsg. von Enrico Francesconi u. a. Berlin, Heidelberg: Springer-Verlag, 2010. Kap. Efficient Multilabel Classification Algorithms for Large-scale Problems in the Legal Domain, S. 192–215. ISBN: 3-642-12836-X, 978-3-642-12836-3. URL: http://dl.acm.org/citation.cfm?id=2167945.2167959.

[Lua12] Oscar Luaces. „Binary relevance efficacy for multilabel classification“. In: *Progress in Artificial Intelligence* 1.4 (2012), S. 303–313. ISSN: 2192-6360. DOI: 10.1007/s13748-012-0030-x. URL: http://dx.doi.org/10.1007/s13748-012-0030-x.

[MS14] A. Malits und P. Schaeuble. „The digital assistant: a semi-automated system for subject cataloguing in the Zentralbibliothek Zurich.“ In: *ABI Technik* 34.3 (2014), S. 132–143.

[New05] M.E.J. Newman. „Power laws, Pareto distributions and Zipf's law“. In: *Contemporary Physics* 46.5 (2005), S. 323–351. DOI: 10.1080/00107510500052444. eprint: http://dx.doi.org/10.1080/00107510500052444. URL: http://dx.doi.org/10.1080/00107510500052444.

[Rea+11] Jesse Read u. a. „Classifier chains for multi-label classification“. In: *Machine Learning* 85.3 (2011), S. 333. ISSN: 1573-0565. DOI: 10.1007/s10994-011-5256-5. URL: http://dx.doi.org/10.1007/s10994-011-5256-5.

[Seb02] Fabrizio Sebastiani. „Machine Learning in Automated Text Categorization“. In: *ACM Comput. Surv.* 34.1 (März 2002), S. 1–47. ISSN: 0360-0300. DOI: 10.1145/505282.505283. URL: http://doi.acm.org/10.1145/505282.505283.

[SS00] Robert E. Schapire und Yoram Singer. „BoosTexter: A Boosting-based System for Text Categorization“. In: *Machine Learning* 39.2 (2000), S. 135–168. ISSN: 1573-0565. DOI: 10.1023/A:1007649029923. URL: http://dx.doi.org/10.1023/A:1007649029923.

[TK07] Grigorios Tsoumakas und Ioannis Katakis. „Multi-label classification: An overview“. In: *Int J Data Warehousing and Mining* 2007 (2007), S. 1–13.

[TKV08] Grigorios Tsoumakas, Ioannis Katakis und Ioannis P. Vlahavas. „Effective and Efficient Multilabel Classification in Domains with Large Number of Labels“. In: *ECML/PKDD 2008 Workshop on Mining Multidimensional Data*. Antwerp, Belgium, 2008. URL: http://mlkd.csd.auth.gr/publication_details.asp?publicationID=276.

[TKV10] Grigorios Tsoumakas, Ioannis Katakis und Ioannis Vlahavas. „Mining multi-label data“. In: *In Data Mining and Knowledge Discovery Handbook*. 2010, S. 667–685.

[Tso+11] Grigorios Tsoumakas u. a. „MULAN: A Java Library for Multi-Label Learning“. In: *J. Mach. Learn. Res.* 12 (Juli 2011), S. 2411–2414. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=1953048.2021078.

[Ums91] W. Umstaetter. „Isn't it time to give information technology the proper consideration for library subject cataloguing? a word to the RSWK“. In: *ABI Technik* 11.4 (1991), S. 277–288.

[WFH11] Ian H. Witten, Eibe Frank und Mark A. Hall. „Chapter 10 - Introduction to Weka“. In: *Data Mining: Practical Machine Learning Tools and Techniques (Third Edition)*. Hrsg. von Ian H. Witten, Eibe Frank und Mark A. Hall. Third Edition. The Morgan Kaufmann Series in Data Management Systems. Boston: Morgan Kaufmann, 2011, S. 403 –406. ISBN: 978-0-12-374856-0. DOI: http://dx.doi.org/10.1016/B978-0-12-374856-0.00010-9. URL: http://www.sciencedirect.com/science/article/pii/B9780123748560000109.

[ZZ05] Min-Ling Zhang und Zhi-Hua Zhou. „A k-nearest neighbor based algorithm for multi-label classification.“ In: *GrC*. Hrsg. von Xiaohua Hu u. a. IEEE, 2005, S. 718–721. ISBN: 0-7803-9017-2.