Ontologie-basierte Schlagwortextraktion zur Verbesserung des Korpus für die automatische Dokumentklassifizierung des Discovery Services LIVIVO von ZB MED

Abschlussarbeit zur Erlangung des Bachelor-Grades
Bachelor of Science im Studiengang Data and Information Science
an der Fakultät für Informations- und Kommunikationswissenschaften
der Technischen Hochschule Köln

vorgelegt von: Melanie Ullrich

eingereicht bei: Prof. Dr. Konrad Förstner
Zweitgutachter/in: Prof. Dr. Klaus Lippert

Köln, 09.08.2023

Erklärung

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer oder der Verfasserin/des Verfassers selbst entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Anmerkung: In einigen Studiengängen steht die Erklärung am Ende des Textes.

Kurzfassung/Abstract

Für die einfache Literaturrecherche von Fachinformationen bietet die ZB MED eine Literaturdatenbank namens LIVIVO an. Um eine thematische Suche zu ermöglichen, befasst sich diese Bachelorarbeit mit der Themenklassifikation der in der Datenbank vorhandenen Publikationen. Das Ziel der Arbeit besteht darin, den Korpus für eine automatisierte Klassifizierung aufzubereiten, um eine relevante Klasseneinteilung zu erzielen. Ausgehend von der Annahme, dass eine Textklassifizierung durch spezifische Terme und Schlüsselwörter gezieltere und aussagekräftigere Ergebnisse liefern kann, wird eine themenspezifische Aufbereitung mithilfe von Wissensorganisationssystemen (Thesauri) eingebunden. Hierzu wird im Vorhinein eine automatisierte Spracherkennung der Publikationen implementiert. Nach der Indexierung der Schlüsselwörter in den Dokumenten werden zwei statistische Klassifikationsmodelle für die Klassifizierung angewandt. Hierzu gehört die Latent Dirichlet Allocation, sowie der Stochastic Gradient Descent Classifier. Abschließend wird die automatische Schlagwortextraktion mit einer intellektuellen Themenanalyse verglichen und die Performance der Klassifizierung mit den aufbereiteten Input-Daten auf eine Verbesserung hin analysiert.

Schlagwörter/Schlüsselwörter: Topic Modeling, Textklassifikation, Indexierung, Schlagwortextraktion, Thesaurus

For simple literature research of specialised information, the institute ZB MED offers a literature database called LIVIVO. To create a thematic search, this bachelor thesis deals with the topic classification of the publications available in the database. The aim of the thesis is to prepare the corpus for an automated classification in order to obtain a relevant classification. Based on the assumption that text classification delivers more targeted and meaningful results with an input of specific terms and keywords, a topic-specific preprocessing with the help of knowledge organisation systems (Thesauri) will be integrated. For this purpose, an automated language recognition of the publications is implemented in advance. After indexing of the keywords in the documents, two statistical classification models are applied for the classification. These are the Latent Dirichlet Allocation and the Stochastic Gradient Descent Classifier. Finally, the automatic keyword extraction is compared with an intellectual topic analysis and the performance of the classification with the processed input data is analysed for improvement.

Inhaltsverzeichnis

Tabellenverzeichnis

Formelverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

NLP	Natural Language Processing
W3C	World Wide Web Consortium
RDF	Resource Description Framework
OWL	Web Ontology Language
URI	Universal Resource Identifiers
MeSH	Medical Subject Headings
NLM	National Library of Medicine
FAO	Food and Agriculture Organization of the United Nations
SKOS	Simple Knowledge Organisation System
SKOS-XL	SKOS eXtension for Labels
SPARQL	SPARQL Protocol And RDF Query Language
LDA	Latent Dirichlet Allocation
SGDC	Stochastic Gradient Descent Classifier

1. Einleitung

In der heutigen Zeit werden die Menschen im Alltag nahezu mit Informationen überflutet. Insbesondere hat das Internet in den letzten Jahren zu einer explosionsartigen Zunahme von Informationen und Daten geführt. Dadurch wird es immer schwieriger nach den benötigten Informationen gezielt zu suchen. Da die Informationsrecherche vor allem im Bereich der Forschung von großer Bedeutung ist, wird der Bedarf an Technologien, die eine effiziente Suche und Abfrage von Informationen ermöglichen, immer größer.

1.1. Problemstellung und Motivation

Um die Literatur- und Informationsrecherche nach Fachinformationen zu erleichtern, werden von verschiedenen Institutionen Literaturdatenbanken zur Verfügung gestellt. Eine der größten Suchmaschinen stellt die ZB MED – Informationszentrum Lebenswissenschaften für Interessierte bereit. Die Datenbank namens LIVIVO umfasst Literatur, Informationen und Forschungsdaten für die Lebenswissenschaften. Um Forschenden die Suche nach den richtigen Informationen zu vereinfachen, werden den Publikationen verschiedene Kategorien eingeteilt. Hierzu gehören die Fachgebiete Medizin und Gesundheitswesen, Ernährung, Umweltwissenschaften, sowie die Landwirtschaft.[1] Die Kategorienzuordnung erfolgt innerhalb dieser Datenbank derzeit durch eine Software eines Drittanbieters namens Averbis GmbH. Das Unternehmen bietet Text Mining- und Machine Learning-Lösungen für Kunden aus den Bereichen HealthCare und LifeScience an. Deren Ziel ist es, unstrukturierte Texte in auswertbare Informationen zur Analyse umzuwandeln.[2]

Damit die ZB MED Einfluss auf die Klassifizierung der Publikationen nehmen kann, wird derzeit eine eigene Software entwickelt, die die Literatur in die unterschiedlichen Klassen einteilen soll. Dies bringt mehrere Vorteile mit sich. Da der Klassifizierungsalgorithmus von Averbis nicht einsehbar ist, stellt die Themenzuordnung für die ZB MED eine sogenannte Blackbox dar. Dies bedeutet, dass die Institution kein Wissen darüber hat, wie die Zuordnung der Kategorien zustande kommt. Es ist von großem Vorteil, wenn die Institution diese Aufgabe selbst übernimmt, da sie dann die Kontrolle darüber hat, wie

[1] Vgl. ZB MED - Informationszentrum Lebenswissenschaften, „LIVIVO-Suchportal," zuletzt geprüft am 25.07.2023, https://www.zbmed.de/recherchieren/livivo.

[2] Vgl. Averbis GmbH, „Informationen zum Unternehmen Averbis GmbH," zuletzt geprüft am 02.08.2023, https://averbis.com/de/unternehmen/#1592585215750-120d0d24-b750.

die Klassifikation durchgeführt wird. Ein weiterer Punkt ist die stetige Weiterentwicklung des eigenen Prozesses, um fortschrittlich zu bleiben und ‚State-of-the-Art'-Methoden implementieren zu können. So kann auf Veränderungen schneller reagiert werden und es gibt keine Abhängigkeit zu einem Dienstleister. Zusammengefasst soll also eine schnellere, flexiblere und kostengünstigere Lösung zur Klassifizierung entstehen.

1.2. Zielsetzung und Aufbau der Arbeit

Das Ziel der Arbeit besteht darin, den Korpus, der als Input für die Klassifizierung verwendet wird, für eine relevante Klasseneinteilung aufzubereiten. Im Hinblick auf die Zielsetzung lassen sich folgende Forschungsfragen definieren:

F1. Inwiefern ist eine automatische Schlagwortextraktion mit einer intellektuellen Themenanalyse vergleichbar?

F2. Inwiefern kann eine gezielte, themenspezifische Informationsextraktion zu einer gesteigerten Performance der Klassifizierung führen?

Um diese Fragestellungen beantworten zu können, wird die Arbeit wie folgt strukturiert. Sie stützt sich auf eine Bachelorarbeit, die sich mit dem Datensatz bereits befasst hat und wird aufbauend und unter Berücksichtigung dieser auf die definierten Forschungsfragen eingehen.[3] Zur Einordnung wird zuerst auf die theoretischen Grundlagen und allgemeinen Begriffe eingegangen. In einem zweiten Schritt wird die praktische Umsetzung beschrieben. Hier erfolgt eine themenspezifische Aufbereitung mithilfe von Wissensorganisationssystemen, die zur Themeneinordnung behilflich sein sollen. Auf diesen Korpus aufbauend werden zwei statistische Klassifikationsmodelle angewandt. Um die Vergleichbarkeit mit dem Vorprojekt zu gewährleisten, werden die gleichen Modelle verwendet und die Ergebnisse untereinander verglichen. Die Modelle sollen in der Lage sein, die Publikationen anhand der Informationsextraktion in ihre übergeordneten Thematiken einzuordnen. Es soll herausgefunden werden, ob mit diesem Korpus eine bessere Performance der Klassifikationsmodelle hergestellt werden kann. Daher werden die Ergebnisse des Vorprojekts in dieser Arbeit als Benchmark betrachtet. Die Ergebnisse der Modelle können in weiteren Projekten der ZB MED zur Weiterentwicklung der Textklassifizierung genutzt werden.

[3] Vgl. Max Prantz, „Verbesserung der automatischen Dokument-Klassifikation für den Discovery Service LIVIVO von ZB MED," (Technische Hochschule Köln, 2023), zuletzt geprüft am 02.08.2023.

2. Theoretische Grundlagen

In folgendem Abschnitt werden die theoretischen Grundlagen erläutert, um ein Verständnis der Arbeit zu gewährleisten. Hierzu gehört ein kurzer Einblick in die übergeordnete Thematik Natural Language Processing, sowie der Themenbereich des Semantic Webs. Es folgen die Grundlagen der verwendeten Wissensorganisationssysteme und Klassifizierungsmodelle, sowie zur Evaluation dieser.

2.1. Natural Language Processing

Im Fokus des Themengebiet Natural Language Processing (NLP), aus dem Englischen - die Verarbeitung von natürlicher Sprache, steht der Versuch menschlich geschriebene oder gesprochene Sprache zu erfassen und aufzubereiten, um sie für Computerprogramme zugänglich zu machen. Dabei werden verschiedene Methoden zur Zerlegung der Sprache in ihre Satzteile für die weitere Analyse oder Informationsextraktion eingesetzt. Zu diesen Methoden gehören beispielsweise die Tokenisierung, das Erkennen von Satzstrukturen oder das Entfernen von Satzzeichen und Stoppwörtern. Diese Schritte dienen dazu die Sprache in eine Form zu bringen, sodass sie von NLP-Algorithmen weiterverarbeitet werden kann. Diese aufbereiteten Daten können dann für nachfolgende anwendungsspezifische Aufgaben genutzt werden. Die weiteren Verarbeitungsschritte können je nach Anwendungsfall variieren und basieren auf Verfahren der Linguistik und kombiniert sie mit maschinellem Lernen oder künstlicher Intelligenz. Bei den in dieser Arbeit angewandten Verfahren handelt es sich um eine Kombination aus Informationsextraktion auf Basis von Wissensorganisationssystemen und statistischer Machine Learning Algorithmen.[4]

2.1.1. Informationsextraktion

Die Informationsextraktion ist ein Teilgebiet von Natural Language Processing und ist ein fundamentaler Schritt, da sie eine Basis für die gute Performance von komplexen Anwendungsmodellen bilden kann. Dies kann darauf zurückgeführt werden, dass die Grenzen der künstlichen Intelligenz und des maschinellen Lernens oft nicht in der Schlussfolgerung oder im Lernen selbst liegen, sondern vielmehr in der unvollständigen Informationsgrundlage. So bietet die Informationsextraktion eine Lösung, mit der

[4] Vgl. Eisenstein, *Introduction to natural language processing,* Adaptive computation and machine learning (Cambridge, Massachusetts, London: The MIT Press, 2018).

natürliche Sprache automatisiert zu strukturiertem Wissen umgewandelt werden kann. Ziel ist es, die relevanten Informationen in einem Text zu identifizieren und in eine für die weitere Verarbeitung geeignete Repräsentation zu transformieren. Hier konzentrieren sich Informationsextraktionssysteme auf die Erkennung von Schlüsselbeziehungen. Sie erhalten als Input eine Sammlung an Dokumenten, die in natürlicher Sprache verfasst wurden und geben mithilfe von definierten Kriterien die relevanten Informationen in strukturierter Form zurück. Diese Kriterien können computergestützt beispielsweise auf statistischer oder sprachwissenschaftlicher Ebene erstellt, aber auch durch manuelle Annotation von Personen entwickelt werden. Diese Merkmale werden in Entitäten, Relationen und Ereignissen definiert. Somit ist die Informationsextraktion mit einer semantischen Rollenbeschriftung vergleichbar, die durch deren Beziehungszuweisungen zu einer für Computer erschließbaren Wissensbasis führt.[5]

In dieser Arbeit wird ein Verfahren der Indexierung angewandt, welches mittels vordefinierten Wissensorganisationssystemen die wesentlichen Schlagworte aus einem natürlichsprachigem Text extrahiert. Auf dieses Verfahren wird im weiteren Verlauf der Arbeit näher eingegangen.

2.2. Semantic Web

Um die Informationsextraktion in diesem Kontext einordnen zu können, wird zunächst auf das Thema Semantic Web eingegangen.

Das Semantic Web kann als eine Erweiterung des World Wide Webs verstanden werden und beschreibt ein Konzept, welches die Maschinenlesbarkeit der Inhalte von Webdokumenten sicherstellen soll. Die grundlegende Idee ist die Vorstellung, dass Daten im Internet nicht nur für Menschen, sondern auch für Computer verständlich sind. Die Daten von Webdokumenten werden mit zusätzlichen Metadaten versehen, sodass eine Darstellung entsteht, die die Bedeutung dieser Dokumente durch Semantik und logische Verbindungen von Begriffen herstellt. Eine solche Darstellung wird meist als Ontologie bezeichnet und ist eine fundamentale Technologie des Semantic Webs. Die Verknüpfungen der Ontologien basieren auf Links, die einheitliche Standards und Identifikatoren verwenden. Die Links erleichtern eine anwendungsübergreifende

[5] Vgl. Eisenstein, *Introduction to natural language processing*.

Integration, sowie die Austauschbarkeit der Daten und werden auch als *Universal Resource Identifiers* (URI) bezeichnet.[6] Dies wird im nächsten Abschnitt näher erläutert.

2.2.1. Ontologien und kontrollierte Vokabulare

Eine Ontologie und auch kontrollierte Vokabulare werden im Kontext der Wissensrepräsentation als eine Beschreibung von Konzepten und Beziehungen in einem Anwendungsbereich definiert. Sie sollen Wissen und Gegebenheiten repräsentieren und deren Relationen darstellen.[7] Das *World Wide Web Consortium* (W3C) weist darauf hin, dass es keine eindeutige Abgrenzung zwischen einem Vokabular und einer Ontologie gibt. Ontologien können komplexer und in ihren Regeln strikter sein als ein kontrolliertes Vokabular, welches in einem Thesaurus oder einer Normdatei festgehalten wird.[8] Im weiteren Teil werden die Begriffe „kontrolliertes Vokabular“ oder „Thesaurus“ verwendet, da nur diese Form von Wissensorganisationssystem in der Arbeit Anwendung findet. Da ein solches Wissensorganisationssystem je nach Anwendungsfall von Menschen oder auch von Maschinen verstanden werden soll, werden verschiedene Technologien eingesetzt. Zur Kodierung der Darstellungen werden Technologien wie die *Resource Description Framework* Spezifikation (RDF) oder die *Web Ontology Language* (OWL) eingesetzt. Da diese Arbeit ausschließlich das RDF-Schema verwendet, wird sich auf diese Technologie beschränkt. RDF ist eine Repräsentationssprache für URIs und basiert auf sogenannten Tripel. URIs identifizieren Ressourcen und bieten die Grundlage für die eindeutige Identifizierung der Objekte. Ein RDF-Triple ist somit eine Entität, die aus Subjekt, Prädikat, Objekt besteht und in einem Triple-Store gespeichert wird (vgl. Abb. 1). Somit entsteht eine semantische Zuordnung der Begriffe untereinander und es wird den Maschinen ermöglicht, diese Daten direkt zu verarbeiten.[9]

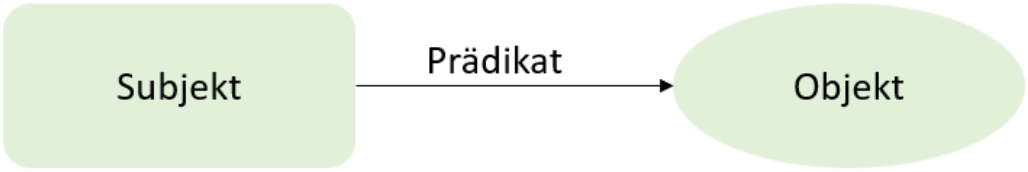

Abbildung 1 - Aufbau eines RDF-Triples

[6] Vgl. Steffen Staab, „The Semantic Web Revisited,” 2006, zuletzt geprüft am 25.07.2023, https://web.archive.org/web/20130320130521/http:/eprints.soton.ac.uk/262614/1/Semantic_Web_Revisted.pdf. 2006.

[7] Vgl. Ebenda

[8] Vgl. W3C, „Web Standards,” zuletzt geprüft am 25.07.2023, https://www.w3.org/standards/semanticweb/ontology.

[9] Vgl. Staab, „The Semantic Web Revisited”2006.

Im nachfolgenden Teil der Arbeit werden die Thesauri beschrieben, die in dieser Thematik Verwendung finden. Für einen ersten Test der Implementierung wird sich hier zunächst auf zwei Thesauri beschränkt. Diese umfassen hauptsächlich die zwei Themen, die den größten Anteil an Veröffentlichungen in der Datenbank vorweisen. Dies sind „MEDLINE“ und „AGRICOLA“.[10] Um das Thema Medizin abzudecken, wird der Thesaurus *Medical Subject Headings* (MeSH) verwendet. Dies ist ein kontrolliertes und hierarchisch organisiertes biomedizinisches Vokabular, welches von der *National Library of Medicine* (NLM) entwickelt und gepflegt wird.[11]

Der zweite verwendete Thesaurus nennt sich *AGROVOC*. Dies ist ein mehrsprachiges, kontrolliertes Vokabular mit einer Anzahl von 40.983 Termen, welche die Interessensbereiche der *Food and Agriculture Organization of the United Nations* (FAO) umfassen. Dies beinhaltet hauptsächlich das Themengebiet Landwirtschaft, aber auch verwandte Wissenschaften wie zum Beispiel Ernährung und Umwelt. Folglich ist es mit diesen beiden Vokabularen möglich, die vier thematischen Bereiche im weiten Sinne abzudecken. AGROVOC ist der meistgenutzte Thesaurus der Welt und daher sehr bekannt für Indexierung und Extraktion von Informationen.[12]
Die beiden Thesauri sind in verschiedenen Formaten im Web öffentlich einsehbar. Wie bereits im oberen Teil beschrieben wird das RDF-Format verwendet.
Ein Thesaurus besteht aus mehreren Konzepten, welche die verschiedenen Themen mit einem oder mehreren Begriffen repräsentieren. Konzepte können hierarchisch angeordnet sein, wobei übergeordnete Konzepte allgemeinere Begriffe darstellen und untergeordnete Konzepte spezifischere Aspekte oder Unterkategorien repräsentieren. Auf der technischen Ebene wird ein solches Konzept mit der Sprache *Simple Knowledge Organisation System* (SKOS) beschrieben. Sie dient zur Strukturierung der Wissenskonzepte. Der Inhalt dieses Konzeptschemas basiert auf einer Erweiterung dieser Sprache namens *SKOS eXtension for Labels* (SKOS-XL). Sie versieht die Begriffe mit Labeln, um diese sprachspezifisch, hierarchisch und mehrsprachig definieren zu können. Mit diesen Sprachen wird es ermöglicht die einzelnen Konzepte über ihre Verknüpfungen abzurufen. Um auf die Begriffe zugreifen zu können, wird eine RDF-Abfragen Sprache namens *SPARQL Protocol And RDF Query Language* (SPARQL) verwendet.

[10] Vgl. Prantz, „Verbesserung der automatischen Dokument-Klassifikation für den Discovery Service LIVIVO von ZB MED.”

[11] Vgl. „Medical Subject Headings - Home Page,” zuletzt geprüft am 25.07.2023, https://www.nlm.nih.gov/mesh/meshhome.html.

[12] Vgl. Imma Subirats-Coll et al., „AGROVOC: The linked data concept hub for food and agriculture,” *Computers and Electronics in Agriculture* 196 (2022), https://doi.org/10.1016/j.compag.2020.105965.

SPARQL ermöglicht es, Abfragen über die RDF-Tripel auszuführen und die Begriffe und deren Beziehungen abzurufen.[13]

Zur Einordnung des Aufbaus eines Konzeptes und seiner Tripel wurde nachfolgend eine graphische Darstellung entwickelt:

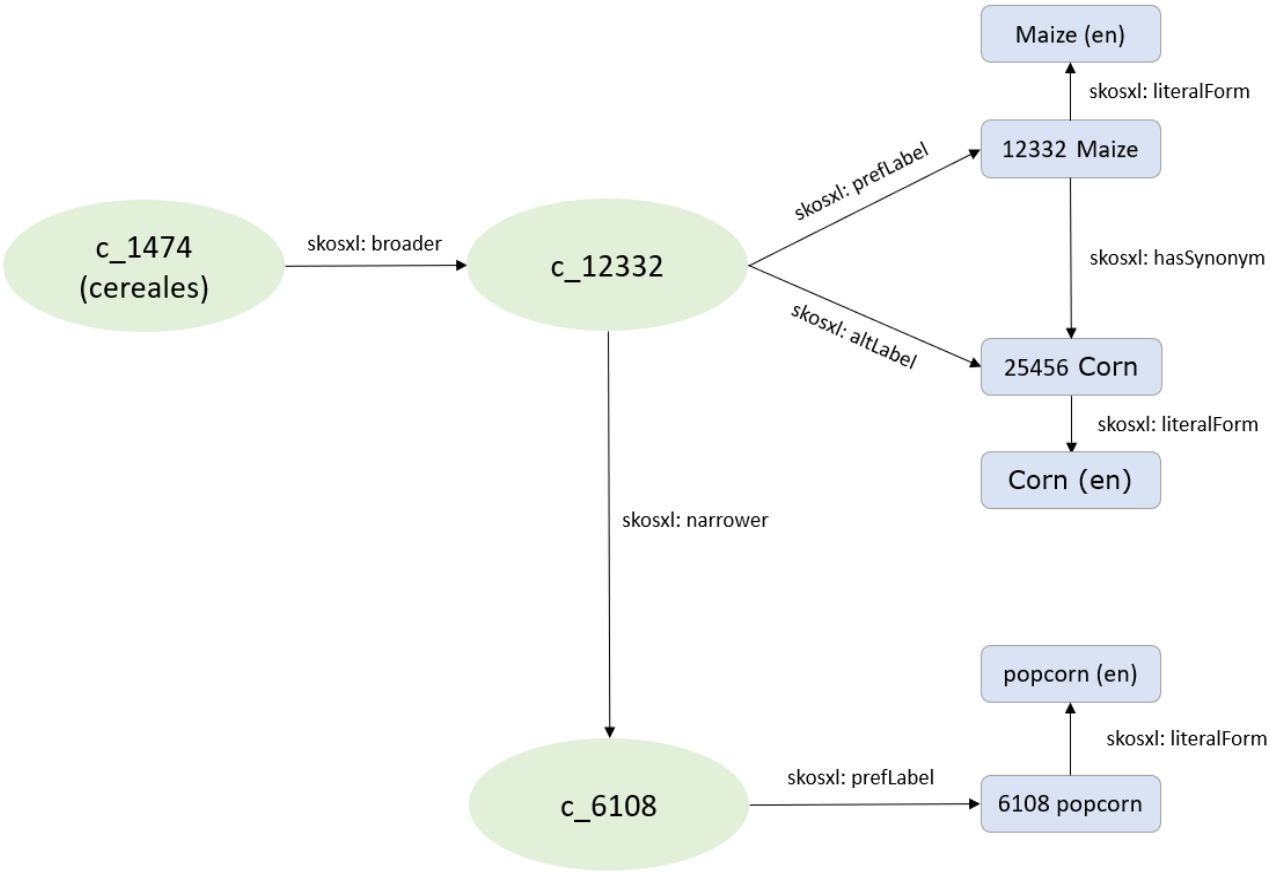

Abbildung 2 - Aufbau eines Konzeptes im Thesaurus

In Abbildung 2 wird anhand des englischen Terms „maize“ die Struktur eines Konzepts in einem Thesaurus verdeutlicht. In der Mitte ist die entsprechende ID des Terms (c_12332) zu erkennen. Der übergeordnete Begriff des Terms ist mit „broader“ gekennzeichnet und beinhaltet die ID, die dem Wort „cereales“, zu Deutsch Getreide zugeordnet ist. Die Verknüpfung mit dem Namen „prefLabel“ führt zu dem Tripel, welches über die Verbindung „literalForm“ das ausgeschriebene Wort des Terms zurückgibt. Im Weiteren kann ein Konzept verschiedene Synonyme der Begriffe beinhalten. In diesem Beispiel wurde ein Synonym aufgezeigt, welches die Bezeichnung „corn“ trägt. Ein letzter Aspekt eines Konzeptes kann ein untergeordneter spezifischer

[13] Vgl. Subirats-Coll et al., „AGROVOC: The linked data concept hub for food and agriculture.”

Begriff sein. Im obigen Beispiel wird die Verknüpfung mit „narrower“ gekennzeichnet und beinhaltet den Term „popcorn“.

2.3. Topic Modeling und Klassifizierungsmodelle

In diesem Abschnitt wird auf eine kurze Definition von Topic Modeling eingegangen und die für die Klassifizierung von Publikationen verwendeten Modelle erläutert. Die Bezeichnung Topic Modeling beschreibt die automatische Erkennung von Themen in einer Sammlung von Dokumenten. Hier kommen Methoden zum Einsatz, die Beziehungen zwischen textuellen Daten finden und diese in thematische Strukturen einordnen. Ziel des Topic Modeling ist es, die wichtigsten Themen in einem Korpus zu identifizieren.[14]

Es gibt unterschiedliche Anwendungsmethoden und Algorithmen, die in diesem Themenbereich angewandt werden. Im Rahmen dieser Arbeit werden zwei Modelle eingesetzt. Dies ist zum einen ein probalistisches Modell, hier repräsentiert durch die *Latent Dirichlet Allocation* (LDA). Zum anderen handelt es sich um ein Vektorraummodell, welches durch den *Stochastic Gradient Descent-Classifier* (SGDC) eingebunden wird.

2.3.1. Latent Dirichlet Allocation

Die LDA ist ein generatives probabilistisches Modell. Dies bedeutet, dass Zufallsvariablen und Wahrscheinlichkeitsverteilungen zur Relevanzeinschätzung der Themen verwendet werden. Somit werden im Vorhinein keine Themen vorgegeben, sondern eigene Themen aus der Dokumentsammlung generiert. Die Idee hinter dem Modell basiert auf der Annahme, dass die Texte eine vordefinierte Anzahl von Themen repräsentieren und jedes Dokument diese Themen in unterschiedlichem Ausmaß darstellt. Ein Thema wird durch eine Wahrscheinlichkeitsberechnung über die Verteilung der Wörter definiert. Um diese Verteilung berechnen zu können, sind im Vorhinein verschiedene Schritte zwingend notwendig. Eines dieser Schritte ist die Anwendung der *Bag-of-Words*-Methode, welche die Häufigkeit der einzelnen Wörter in den Dokumenten beschreibt.[15]

[14] Vgl. Pooja Kherwa und Poonam Bansal, „Topic Modeling: A Comprehensive Review,“ *ICST Transactions on Scalable Information Systems* 0, Nr. 0 (2018), zuletzt geprüft am 30.06.2023, https://doi.org/10.4108/eai.13-7-2018.159623, https://eudl.eu/pdf/10.4108/eai.13-7-2018.159623.

[15] Vgl. Ebenda

Ein großer Vorteil dieses Verfahrens ist, dass keine vorherige manuelle Annotation der Texte erforderlich ist, da der Algorithmus die Themen selbst zusammenstellt. Daher ist der Algorithmus einfach zu implementieren und eignet sich gut für eine erste Analyse der vorzufindenden Themen.

Zur Evaluierung der LDA wird zum einen die Metrik der probalistischen Kohärenz verwendet. Die Kohärenz misst die Verwandtschaft der wichtigsten Wörter in einem Thema unter Berücksichtigung ihrer statistischen Unabhängigkeit. Sie besagt also wie stimmig die Wörter in den Themen sind, die das Modell definiert hat. Eine weitere Metrik ist die Komplexität, die aussagt, wie unerwartet neue Daten für ein Modell sind.[16] Zusätzlich wird im Rahmen dieser Arbeit eine Evaluierungsform für das Modell entwickelt, welche eine Zuordnung der vorhergesagten numerischen Themen zu den alphabetischen Klassen von Averbis ermöglicht. Diese wird im Abschnitt 4.2. Evaluierung Klassifikationsmodelle näher erläutert.

2.3.2. Stochastic Gradient Descent Classifier

Der Stochastic Gradient Descent Classifier ist streng genommen kein eigenständiges Klassifizierungsmodell, sondern basiert auf anderen Algorithmen, die statistische Analysen einbeziehen. Dies können zum Beispiel eine *Support Vector Machine* (SVM) oder eine logistische Regression sein. Das SGD ist in diesem Zusammenhang ein Optimierungsverfahren, welches aufbauend auf den linearen Modellen den stochastischen Gradienten-Abstieg implementiert. Das Lernverfahren basiert auf einer Verlustfunktion, die den Abstiegsgradienten für ein zufällig ausgewähltes Trainingsmuster einschätzt und das Modell bei jeder Iteration aktualisiert. [17]

Der hier angewandte Klassifikator basiert auf der Support Vector Machine. Das Modell beruht auf überwachtem maschinellem Lernen und benötigt daher als Input einen bereits klassifizierten Testdatensatz, um dies auf weitere Datensätze transferieren zu können. Das Ziel des Algorithmus ist es, in einem N-dimensionalen Raum eine Hyperebene zu finden, welche die Datenpunkte eindeutig klassifiziert. N steht dabei für die Anzahl der Merkmale im Korpus. Hier ist es wichtig, die Hyperebene zu finden, die den größten Abstand zwischen den Datenpunkten in den Klassen aufweist. Die Datenpunkte werden mit dieser Ebene in dem Raum abgetrennt und können daher in unterschiedliche Klassen

[16] Vgl. Kherwa und Bansal, „Topic Modeling: A Comprehensive Review."

[17] Vgl. scikit-learn, „1.5. Stochastic Gradient Descent," zuletzt geprüft am 25.07.2023, https://scikit-learn.org/stable/modules/sgd.html.

eingeteilt werden. Je nach Anzahl der Merkmale kann die Dimension der Hyperebene steigen oder sinken. Die ausschlaggebendsten Datenpunkte sind die sogenannten Stützvektoren, die näher an der Hyperebene liegen und somit die Position dieser stärker beeinflussen. Das Ziel des Algorithmus ist es, den Abstand zwischen den Datenpunkten und dieser Hyperebene zu maximieren. Hier kommt nun die Verlustfunktion zum Tragen, die zur Maximierung dieses Abstands beiträgt.[18]

Um die Datenpunkte in dem Vektorraummodell abbilden zu können, müssen Term Gewichte gebildet werden. Zur Bestimmung der Gewichtungen gibt es verschiedene Verfahren. Dabei kann zwischen Verfahren unterschieden werden, die sich auf einzelne Dokumente beziehen, und solche, die die Gesamtheit der Texte in Betracht ziehen. Hier sind auch Verfahrenskombinationen möglich. Eine sehr häufige Kombination ist das TF-IDF-Maß. TF-IDF steht für *term frequency-inverse document frequency* und ist ein statistisches Maß, welches die Wichtigkeit eines Wortes für ein Dokument in einem Korpus misst. Der Gewichtungswert wird berechnet mittels zwei Komponenten. Zum einen wird die Worthäufigkeit eines Terms (TF) in dem Dokument gemessen. Da ein Wort in einem längeren Text öfter vorkommen kann als in einem kürzeren Text, wird diese Worthäufigkeit durch die Gesamtanzahl der Wörter im Dokument geteilt. Die inverse Dokumentenhäufigkeit (IDF) misst im zweiten Schritt die Relevanz des Terms. Denn meist treten die unrelevanten Wörter wie z.B. „der“, „die“ oder „das“ in einem Text sehr häufig auf, sind aber nicht aussagekräftig. Hier wird also die Spezifität des Terms für die Gesamtmenge der Dokumente gemessen.[19]

Dieses Klassifizierungsmodell ist sehr geeignet für große Datensätze, da es hochdimensionale Merkmale effizient bearbeiten kann. Dies liegt daran, dass die Modellparameter während des Trainings für die Teilmengen der Daten immer wieder aktualisiert werden, anstatt den gesamten Datensatz zu verarbeiten. Das ist außerdem der Grund dafür, dass dieses Modell nur einen minimalen Speicherbedarf benötigt. Daher wird es oft wegen der hohen Genauigkeit bei gleichzeitig geringem Bedarf an Rechenleistung bevorzugt.[20]

[18] Vgl. Rohith Gandhi, „Support Vector Machine — Introduction to Machine Learning Algorithms,” *Towards Data Science*, 07.06.2018, zuletzt geprüft am 25.07.2023, https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

[19] Vgl. „TF-IDF — Term Frequency-Inverse Document Frequency – LearnDataSci,” zuletzt geprüft am 27.07.2023, https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/.

[20] Vgl. Data Science With Chris, „SGDRegressor and SGDClassifier,” zuletzt geprüft am 27.07.2023, https://datascience-withchris.com/sgdregressor-sgdclassifier/.

2.4. Evaluierungsmetriken

Im folgenden Abschnitt wird auf die verschiedenen Metriken für die Evaluierung der Klassifizierungsmodelle eingegangen. Diese geben Aufschluss über die Performance der Modelle und sind für einen genauen Vergleich mit der Benchmark wichtig. Mit diesen ist es möglich die erforderliche Leistung einzuschätzen und die Forschungsfragen zu beantworten.

Jedes Modell muss bei der Bewertung auf die Richtigkeit der Vorhersage betrachtet werden. Hierzu dient die Konfusionsmatrix, die in einer tabellarischen Darstellung die Vorhersageergebnisse zusammenfasst. Die Matrix ist jedoch für binäre Klassifizierungsprobleme aufgebaut. Da hier eine Mehrklassen-Klassifizierung vorliegt, werden die Ergebnisse der Vorhersage für jedes Klassenlabel einzeln berechnet. Das Modell hat bei der Vorhersage jeweils zwei Möglichkeiten von Erfolg- und Fehlerberechnung: [21]

Erfolg:

- True positives (TP): Das Modell hat die positive Klasse richtig vorhergesagt.
- True negatives (TN): Das Modell hat die negative Klasse richtig vorhergesagt.

Fehler:

- False positive (FP): Das Modell hat die positive Klasse falsch vorhergesagt.
- False negative (FN): Das Modell hat die negative Klasse fälschlicherweise vorhergesagt.

Eine sehr häufig verwendete Metrik ist die Accuracy, zu Deutsch Genauigkeit. Mit dieser wird der Anteil der richtigen Vorhersagen eines Modells angegeben. Es setzt somit die Anzahl der korrekten Vorhersagen und die Anzahl aller gemachten Vorhersagen ins Verhältnis. Hier ist wichtig zu beachten, dass der Datensatz ausgewogen ist und das Verhältnis der Klassen ungefähr gleich ist. Die Formel für die Accuracy lautet: [22]

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

[21] Vgl. Artem Oppermann, „Accuracy, Precision, Recall, F1-Score und Specificity - KI Tutorials," zuletzt geprüft am 17.07.2023, https://artemoppermann.com/de/accuracy-precision-recall-f1-score-und-specificity/.

[22] Vgl. Ebenda

Weitere wichtige Metriken für die Evaluierung sind *Precision* (Relevanz) und *Recall* (Vollständigkeit). Die Precision gibt hier die Genauigkeit der Klassifizierung an und setzt die wahren positiven Ergebnisse, die vom Modell korrekt vorhergesagt wurden, mit der Gesamtanzahl aller positiven Vorhersagen ins Verhältnis. Hierzu gehören auch die fälschlicherweise korrekten Vorhersagen.

Die Formel für Precision wird wie folgt definiert: [23]

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Der Recall hingegen misst, wie gut das Modell die positiven Ergebnisse identifizieren kann. Somit werden hier die wahren positiven Ergebnisse des Modells mit der Gesamtanzahl der tatsächlich wahren Ergebnisse ins Verhältnis gesetzt. Die Formel des Recalls wird wie folgt definiert: [24]

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Precision und Recall sind gegensätzlich. Das bedeutet, wenn ein hoher Precision-Wert vorliegt, wird ein niedriger Recall-Wert zurückgegeben und umgekehrt. Um aus diesen beiden Werten eine Metrik des harmonischen Mittels zu generieren, wird der F1-Wert berechnet. Dieser kombiniert Precision und Recall und wird mit folgender Formel definiert: [25]

$$F1\ Score = \frac{2 * TP}{2 * TP + FP + FN} \tag{4}$$

Zusammenfassend lässt sich hier sagen, dass diese Metriken sehr hilfreich für die Bewertung der Modelle, sowie für den Vergleich untereinander sind. Es ist jedoch zu beachten, dass die Accuracy das Modell nicht immer zuverlässig evaluieren kann. Sie kann das Performance-Ergebnis verfälschen, da die Accuracy bei einem unausgewogenem Datensatz oft hoch ist und es so scheint, als ob das Modell sehr gut klassifiziert. Da in dieser Arbeit darauf geachtet wird, einen ausgewogenen Datensatz zu generieren, werden alle Bewertungskriterien berücksichtigt. [26]

[23] Vgl. Oppermann, „Accuracy, Precision, Recall, F1-Score und Specificity - KI Tutorials."

[24] Vgl. Ebenda

[25] Vgl. Ebd.

[26] Vgl. Oppermann, „Accuracy, Precision, Recall, F1-Score und Specificity - KI Tutorials."

3. Implementierung

Nach der Erläuterung der theoretischen Grundlagen, werden im folgenden Teil die praktischen Umsetzungsschritte der Methodik beschrieben, indem zunächst auf die Datengrundlage eingegangen wird. Darauffolgend werden die einzelnen Umsetzungsschritte im Detail beschrieben.

3.1. Datengrundlage

Eine Datenbank mit ca. 55.000.000 Einträgen bildet die Datengrundlage dieser Arbeit. Die Datenbank umfasst eine Tabelle mit den Metadaten der veröffentlichten mehrsprachigen Publikationen von LIVIVO. Hierzu zählt unter anderem der „Dokumententyp“, „DOI“, „Quelle“, entsprechende Identifikationsnummer, „Titel“ und „Abstract“ der Publikation. Ebenfalls gibt es eine zweite Tabelle, die zusätzlich die definierten Klassen von Averbis beinhaltet.

Da für die Textklassifizierung nur die Spalten „Titel“ und „Abstract“ relevant sind, wird sich auf diese beiden Spalten beschränkt. Zusätzlich wird der Datensatz aus ressourcengründen auf eine Anzahl von 500.000 Einträgen reduziert. Hier wird darauf geachtet, dass der Korpus auf die fünf Averbis-Klassen in gleichen Anteilen aufgeteilt wird, sodass ein ausgewogener Datensatz vorliegt. Ansonsten könnte es zu Verzerrungen bei der Klassifizierung führen. Die Anzahl an Datensätzen wurde gewählt, da sich die Lernkurve des SGD-Modells der Benchmark ab einer Anzahl von ca. 10.000 Datensätzen nicht mehr wesentlich verändert hat.[27] Das bedeutet, dass bereits mit dieser Anzahl eine Beurteilung der Lernkurve und eine Aussage über die Perfomance im Vergleich möglich ist.

Ferner wurden für diese Arbeit Dateien zur Verfügung gestellt, die im Rahmen des Projekts *QuaMedFo* von der ZB MED erstellt wurden. Dazu gehören Pickle-Dateien, die die Lexika für die MeSH Termini enthält. Das Verfahren zur Aufbereitung des Thesaurus zu einem Lexikon wird hier auf den Themenbereich von AGROVOC erweitert.

[27] Vgl. Prantz, „Verbesserung der automatischen Dokument-Klassifikation für den Discovery Service LIVIVO von ZB MED.”

3.2. Methodik und Umsetzung

Die Methodik stützt sich aufbauend auf die Grundlage der festgelegten Benchmark. Es werden methodische Optimierungen zur Aufbereitung des Korpus angewandt, indem sich auf Schlüsselwörter spezialisiert wird. Zur Einbindung der themenspezifischen Schlagwortgewinnung werden Lexika erstellt, die auf der Grundlage von Thesauri basieren. Die Dokumente werden anhand dieser Lexika indexiert und für die Implementierung der Klassifikationsmodelle aufbereitet. Die Schlagwörter der einzelnen Publikationen stellen somit den Input für die Textklassifizierung dar. In Abbildung 3 ist eine Veranschaulichung der Methodik Schritte dargestellt.

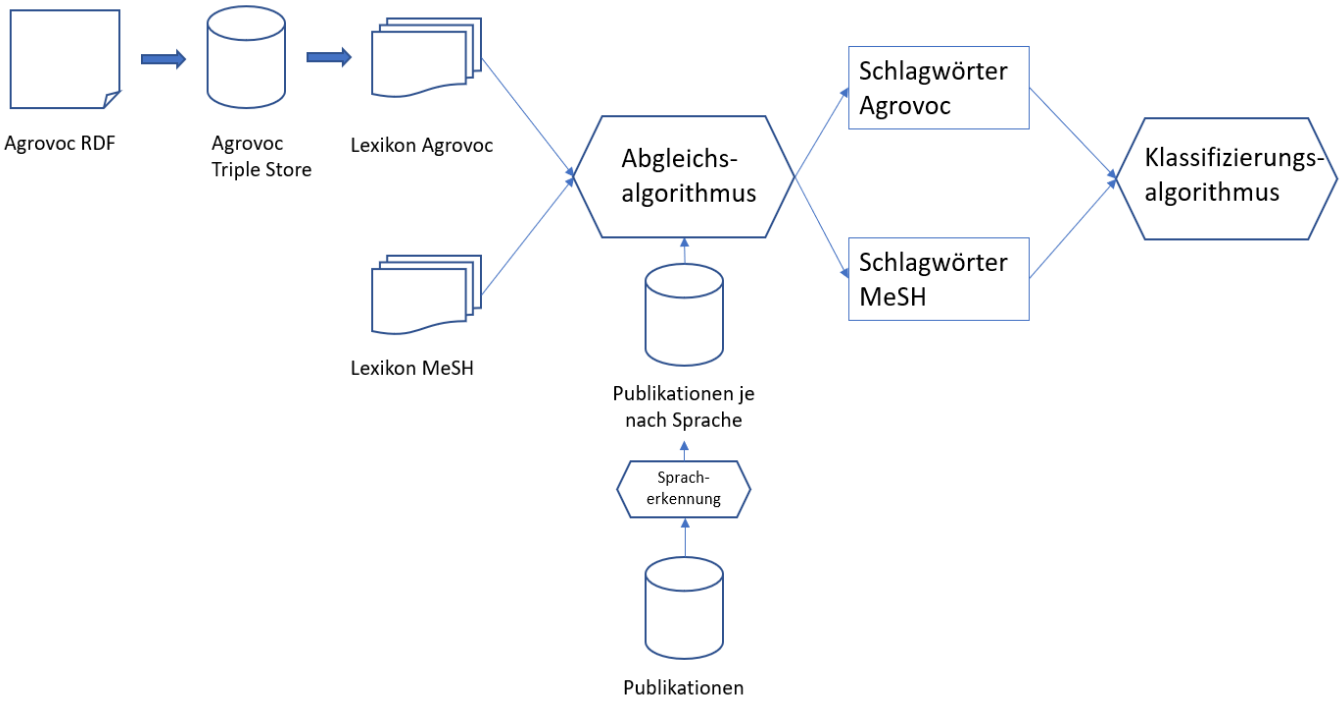

Abbildung 3 - Veranschaulichung der Methodik

Eine detaillierte Beschreibung der Umsetzung erfolgt im folgenden Teil. Die praktische Durchführung erfolgte mithilfe der Programmiersprache Python, sowie die Datenbanksprache PostgreSQL, welche weitgehend auf dem SQL-Standard basiert. In Python werden die Standard-Bibliotheken zur Datenverarbeitung verwendet. Hierzu gehört zum Beispiel die Datenanalyse-Bibliothek *Pandas*. Ein weiteres wichtiges Modul in diesem Zusammenhang ist *SQLAlchemy*, welches ermöglicht, die Daten aus der Datenbank mit dem Python Programm zu verknüpfen. Die zusätzlich verwendeten Bibliotheken werden in den jeweiligen Abschnitten erwähnt.

Im ersten Schritt der Umsetzung wird wie in Abschnitt 3.1. Datengrundlage beschrieben, ein kleineres Datenset für die Testzwecke generiert. Da die Vokabulare der Thesauri sprachspezifisch angewandt werden, ist es im Weiteren notwendig, eine Spracherkennung für die Publikationen durchzuführen. Es ist möglich, dass der Titel und das Abstract in unterschiedlichen Sprachen im Datensatz vorliegen. Daher wird dieser Schritt auf den beiden Spalten einzeln ausgeführt. Für die Sprachbestimmung wird eine Python Bibliothek namens *langdetect* verwendet. Diese basiert auf der Java-Spracherkennungsbibliothek von Google, welche zu Python portiert wurde und kann bis zu 55 verschiedene Sprachen identifizieren. Das Modell klassifiziert den Text mit einem Naive Bayes-Klassifikator und ordnet diesem die Klasse mit der höchsten Wahrscheinlichkeit zu. Als Output wird der ISO 693-Code der entsprechenden Sprache zurückgegeben.[28]

In der nachfolgenden Darstellung ist die TOP5-Verteilung der Sprachen, die im Datensatz vorzufinden sind, dargestellt (vgl. Abbildung 4). Es ist zu erkennen, dass die meisten der Publikationen in englischer Sprache geschrieben sind (über 400.000 Publikationen). In deutscher Sprache sind ca. 30.000 Einträge zu finden. Die weiteren drei Sprachen Spanisch, Portugiesisch und Französisch sind von der Verteilung ungefähr gleich. Da sich auf drei der Sprachen in den TOP 5 im Korpus spezialisiert werden sollte, wurde sich auf die Sprachen Englisch, Deutsch und Französisch beschränkt.

[28] Vgl. Lahiru Hinguruduwa et al., „Assessing Language Identification over DBpedia," in *2021 IEEE 15th International Conference*, zuletzt geprüft am 13.06.2023, https://www.researchgate.net/profile/Edgard-Marx-2/publication/351010850_Assessing_Language_Identification_over_DBpedia/links/613f9cd46c61e2367c797ef6/Assessing-Language-Identification-over-DBpedia.pdf.

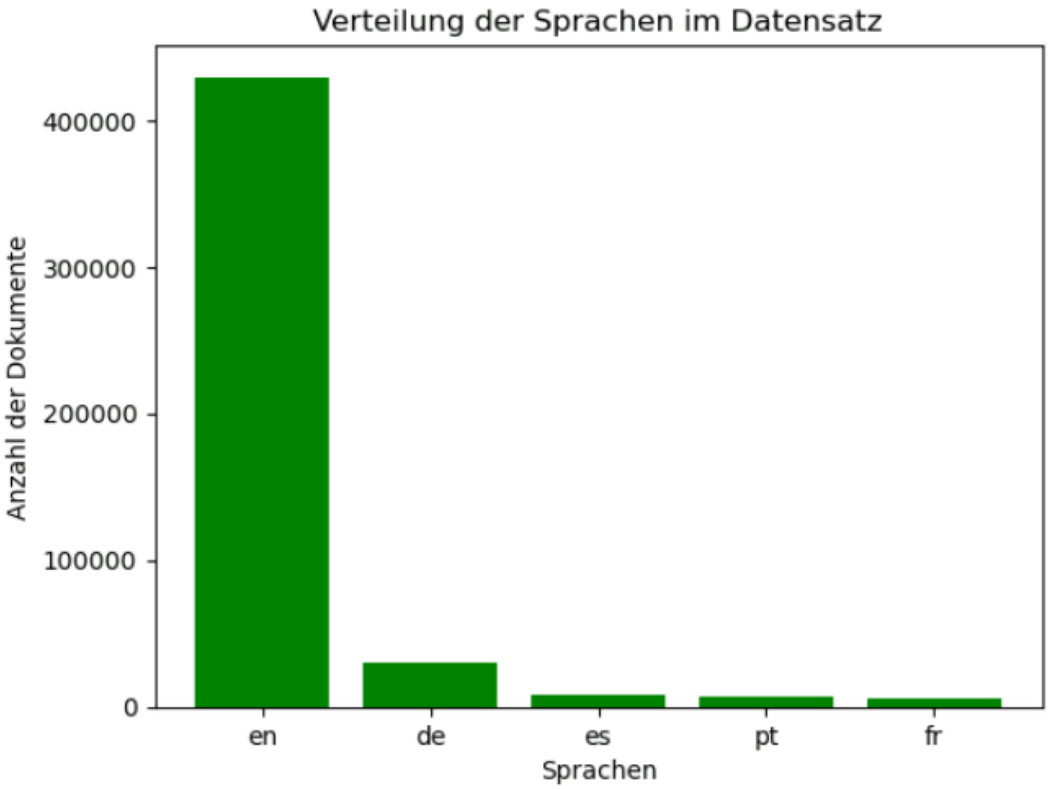

Abbildung 4 - Verteilung der TOP 5 Sprachen im Datensatz

3.2.1. Implementierung Thesauri

Im nächsten Schritt wird der mehrsprachige Thesaurus von AGROVOC aufbereitet, sodass ein Lexikon aus Hauptterm, Synonym und entsprechender ID entsteht. Dieses Lexikon wird benötigt, um die Schlagwörter aus den Texten extrahieren zu können. Der Thesaurus liegt als RDF-Datenset vor, wird mithilfe der Python Bibliothek *rdflib* als N-Tripel-Format eingelesen und in ihre Dreifache zerlegt. Für den Zugriff auf die Daten in den Tripel wird, wie in Abschnitt 2.2.1. beschrieben, die SPARQL-Abfrage verwendet. Innerhalb dieser Abfrage wird mittels SKOS auf das Konzept und mit SKOS-XL auf die Label zugegriffen. Die Abfrage wird so definiert, dass diese für jedes Konzept den Link mit ID des Terms, den Hauptterm, die Synonyme und die jeweilige Sprache ausgibt. Hierbei wird auf die Sprachen Englisch, Deutsch und Französisch gefiltert. Die Daten für die Terme und die Synonyme werden dann in je ein Pandas Dataframe geschrieben. Der Link des Konzeptes wird nachfolgend in Einzelteile zerlegt, sodass nur die Term-ID übrigbleibt. Als nächstes werden die verschiedenen Lexika für die drei Sprachen erstellt. Hierzu wird eine Python Bibliothek, die im Rahmen des Projekts QuaMedFo von der ZB MED entwickelt wurde, verwendet. Die Pipeline erhält als Input das zuvor erstellte Pandas Dataframe mit ID und Term des AGROVOC-Thesaurus. Zuerst erfolgt eine Aufteilung der Terme in Wörter, die nur Großbuchstaben und gemischte Wörter, die

Groß- und Kleinbuchstaben enthalten. Hier muss beachtet werden, dass diese Terme auch als N-Gramm auftreten können. Als N-Gramm werden Sequenzen von einer beliebigen Anzahl an Wörtern bezeichnet, die in dem Kontext zusammengehören. Das N steht für die Anzahl der Wörter in der Sequenz. Ein Beispiel aus dem AGRVOC-Thesaurus wäre „genetic engineering“, zu Deutsch Gentechnik. Dies wird als ein Term angesehen, obwohl es aus zwei Wörtern besteht, die ohneeinander einen anderen Kontext vorweisen. Da diese N-Gramms im Thesaurus bereits berücksichtigt werden, muss hier keine weitere Aktion erfolgen. Als Output gibt die Bibliothek zwei Listen zurück, die die ID und den entsprechenden Term beinhalten. Es gibt eine Liste mit Begriffen, die nur Großbuchstaben enthalten und eine Liste mit Begriffen, die Mischbuchstaben umfassen. Für die spätere Verwendung dieser neu erstellten Lexika werden die Listen für jede Sprache als Pickle-Datei abgespeichert. Das englischsprachige Lexikon umfasst 19.330 Terme, die deutsche Version 13.457 Terme und das französische 13.216 Terme. Um später die Synonyme ihren Haupttermen zuordnen zu können, wird im Anschluss eine neue Datenbank-Tabelle erstellt, die nur die Hauptterme in englischer Sprache beinhaltet. Damit wird erreicht, dass die Schlagwörter im weiteren Teil auf eine Sprache vereinheitlicht werden können und die Klassifizierung nur auf dieser Ebene stattfindet.

3.2.2. Einbindung der Schlagwortextraktion

Für die Integration der Schlagwortextraktion wird im ersten Schritt mittels PostgreSQL eine Datenbanktabelle erstellt, die folgende Inhalte umfasst: „ID“, „Titel“, „Abstract“ und „Sprache“ der Publikation. Dies bildet nun die Datengrundlage für die Schlagwortgewinnung. Dieses Verfahren basiert auf derselben Python Bibliothek, die für die Erstellung der Lexika verwendet wurde. Damit eine genaue Übereinstimmung der einzelnen Wörter in den Texten der Publikationen gewährleistet ist, werden im ersten Schritt alle Sonderzeichen wie Punkte, Klammern, etc. aus den Texten entfernt. Im nächsten Schritt werden mit dem Matching-Algorithmus die Terme, die gleichzeitig in den Publikationstexten und im Lexikon vorkommen in eine Liste extrahiert. Zur selben Zeit erfolgt das Auslesen des Indexes des jeweiligen Begriffs, der die Position dessen kennzeichnet. Zusätzlich wird die Anzahl jedes Terms zurückgegeben. Dies ist notwendig, um später die Worthäufigkeit in den Dokumenten bestimmen zu können. Als Output wird ein Dictionary mit der Wortanzahl, der Position, sowie der Term selbst zurückgegeben. Diese Schlagwortextraktion erfolgt für den „Titel“ und „Abstract“ mithilfe der beiden Lexika MeSH und AGROVOC, sowie für die drei im Vorhinein erwähnten Sprachen.

Wie bereits erwähnt, werden die Schlüsselwörter auf eine Sprache und auf die Hauptterme vereinheitlicht. Das bedeutet, dass die Synonyme den entsprechenden englischsprachigen Haupttermini zugeordnet werden. Dies ist zum einen für die Durchführung des LDA-Modells von Vorteil, da die Wörter, die ein Thema repräsentieren können, eingegrenzt werden. Ein weiterer Vorteil ist, dass die Klassen nur auf einer Sprache als Ausgabewert der Klassifizierung zurückgegeben werden und sie somit unabhängig von Sprachen sind. Somit werden nur Begriffe zurückgegeben, die für eine eindeutige Hauptthematik stehen. Für die Umsetzung werden die einheitlichen IDs der Terme genutzt. Diese sind für alle Sprachen und Synonyme mit den Haupttermen gleich. Hier kommen PostgreSQL gestützte Abfragen zum Einsatz.

Innerhalb der Implementierung aller Methoden kam es zu einem Auslastungsproblem der virtuellen Maschine. Da die Ausführung der Python Programme auf der Maschine für den gesamten Datensatz zu lange dauert, mussten die Daten zunächst, wie in Abschnitt 3.1 beschrieben, auf einen Teildatensatz reduziert werden. Jedoch war auch der reduzierte Datensatz zu groß, sodass die Ressourcen des Computers an die Grenzen kamen. Um den Arbeitsspeicher der Maschine nicht zu überlasten, wurden die Daten aus der Datenbank stückchenweise eingelesen, verarbeitet und wieder in eine neue Datentabelle geschrieben. Eine weitere Zeitersparnis war die Implementierung einer Methode, die zur Parallelisierung der Prozesse dient. Das verwendete Python Modul nennt sich *ProcessPoolExecutor* und dient zur Erstellung mehrerer Prozesse, die durch Verteilung der Aufgaben auf mehreren Kernen gleichzeitig ablaufen können. Es wurde sich für dieses Modul entschieden, da die Integration in den Programmcode wenige Probleme bereitet.
Jedes Python Programm wird als ein Prozess bezeichnet und hat einen sogenannten Main-Thread, der zur Ausführung der Programmanweisungen dient. Wenn mehrere Prozesse erstellt werden, werden diese in einem Prozesspool verwaltet. Dieser steuert wann Prozesse erzeugt und wann sie für eine Aufgabe benötigt werden. Genauso wird gesteuert, was die Prozesse im Wartezustand tun sollen ohne Rechenressourcen zu verbrauchen. Die in Python geschriebenen Funktionen werden also in einer Prozessklasse des ProcessPoolExecutors übergeben. Diese gibt beim Aufruf die Ergebnisse in einem Objekt

zurück. Das bewirkte, dass die Verarbeitung der Daten auf 12 Kernen auf der virtuellen Maschine gleichzeitig ablaufen konnte.[29]

3.2.3. Implementierung Klassifikationsmodelle

Zur Implementierung der Klassifikationsmodelle muss zu Beginn erwähnt werden, dass sich der Datensatz im Laufe der Verarbeitung verändert hat. Aus verschiedenen Gründen sind einige Datensätze verloren gegangen. Dies liegt zum einen daran, dass sich nur auf drei Sprachen beschränkt und somit die weiteren Sprachen entfernt wurden. Darüber hinaus werden nach einer ersten Analyse der Klassifizierung die Datensätze mit der Klasse „Rest“ aufgrund der Annahme, dass eine Verzerrung auftritt, entfernt. Die Begründung hierfür wird im weiteren Verlauf der Arbeit dargestellt. Somit wurden die Modelle nur noch mit einem Datensatz von 386.943 Datensätzen trainiert.

Für das Trainieren einer Latent Dirichlet Allocation wird eine Python Bibliothek namens *Gensim* verwendet, die Unterstützung für die Verarbeitung von unstrukturiertem Text bietet. Hier wird ein Algorithmus für die Vektorisierung der Wörter, sowie für das unüberwachte Lernen mit einem LDA-Modell bereitstellt. Der Algorithmus erkennt durch statistische Ko-Okzidenzmuster automatisch die semantische Struktur der Dokumente.[30]

Für die Integration der LDA muss zunächst der Trainingsdatensatz aufbereitet werden. Die LDA benötigt für die Dokumente jeweils eine Liste aller AGROVOC und MeSH Schlagwörter als Input. Nach dem Zusammenstellen der Wörter, wird der Datensatz in Test- und Trainingsdaten aufgeteilt. Hierzu werden 75 % der Daten für Trainingszwecke und 25 % für das darauffolgende Testen verwendet. Für das Trainieren des Algorithmus wird ein Korpus erstellt, indem die Wörter als ID repräsentiert und in einem *Bag-of-Words*-Modell eingebunden werden. Dieses Modell gibt die Häufigkeit der Wörter in dem Dokumentenkorpus zurück. Im nächsten Schritt wird das LDA-Modell mit der Vorgabe der Themenanzahl trainiert. Es liefert im Anschluss die Themenbereiche mit den 10 Wörtern, die für das jeweilige Thema am relevantesten sind. Die Relevanz der Terme wird durch eine Gewichtung festgelegt, die angibt, wie wichtig der Begriff für diese Themenklasse ist. Im ersten Ansatz des LDA-Trainings wurde zunächst eine Anzahl von 5 Themen vorgegeben, da dies der Anzahl der Averbis Themen entspricht. Mithilfe einer

[29] Vgl. Jason Brownlee, „ProcessPoolExecutor in Python: The Complete Guide,” zuletzt geprüft am 10.07.2023, https://super-fastpython.com/processpoolexecutor-in-python/#Python_Processes_and_the_Need_for_Process_Pools.

[30] Vgl. „Gensim: topic modelling for humans,” zuletzt geprüft am 26.07.2023, https://radimrehurek.com/gensim/intro.html.

Intertopic Distance Map konnte das Modell analysiert werden. Diese Darstellung zeigt die TOP 30 Terme in den einzelnen Themengebieten auf. Im Weiteren ist gekennzeichnet, wie oft die Themen im Korpus wiederzufinden sind. Bei der Anwendung des LDA-Modells mit dieser Vorgabe ist eine Überschneidung der Themen 3 und 4 zu erkennen (vgl. Abb. 10 im Anhang). Dies ist ein Hinweis auf die Ähnlichkeit der beiden Themen und sie werden daher als ein Thema betrachtet. Im Weiteren ist im Datensatz zu sehen, dass viele Dokumente nicht in ein Themengebiet klassifiziert werden konnten. Es sind rund 42.000 None Werte im Datensatz vorhanden. Dies bedeutet, dass der Text des Dokuments keines der relevanten Wörter aus der Themenklasse beinhaltet oder der Wahrscheinlichkeitsscore nicht den Schwellenwert von 50 % erreichen konnte. Aufgrund der großen Anzahl nicht klassifizierter Datensätze im Korpus waren die Ergebnisse entsprechend nicht aussagekräftig. Es wurde ein F1-Score von 0,1839 erreicht. Um einen größeren F1-Score zu erzielen, wurde zunächst versucht die None Werte und die Dokumente mit der Klassenwahrscheinlichkeiten unter 60 % zu entfernen. Damit wurde ein erhöhter Score von 0,2371 erreicht. Hieraus kann geschlussfolgert werden, dass die None-Werte und die Klassen mit geringer Wahrscheinlichkeit die Evaluierung verfälschen. Dies liegt daran, dass die None Werte mit den Kombinationswerten nicht verglichen werden können und stehen daher für eine falsche Vorhersage. Ebenso ist es sehr wahrscheinlich, dass die Klassen, die mit einer Wahrscheinlichkeit von weniger als 60 % der Klasse zugeordnet werden konnten, eine falsche Zuordnung vorhergesagt haben. Da jedoch eine Klassifizierung aller Datensätze angestrebt wird, wurde ein anderer Ansatz gewählt. Es ist zu erkennen, dass bereits bei der Themenfindung und somit beim Trainieren des Modells das Problem auftritt, dass die Datensätze nicht klassifiziert werden können. Der neue Ansatz basiert auf der Aussage, dass die Averbis Klasse „Rest“ ein Bias im Datenset verursacht.[31] Dies kann darauf zurückgeführt werden, dass ein sehr großer Teil der Datenbank mit der Klasse „Rest“ klassifiziert wurde und diese somit auch viele verschiedene Wörter beinhaltet. Daher ist davon auszugehen, dass diese Kategorie der Grund für die Überschneidung der beiden Themen ist. Somit werden die Datensätze mit der Averbis Klasse „Rest“ entfernt und das Modell nochmal mit 4 Klassen trainiert. Nun ist auch eine klare Abgrenzung zwischen den 4 Themen zu erkennen (vgl. Abb. 11 im Anhang). Dies belegt noch einmal, dass die Themenanzahl in diesem Kontext sinnvoller ist.

[31] Vgl. Prantz, „Verbesserung der automatischen Dokument-Klassifikation für den Discovery Service LIVIVO von ZB MED.“

Für die Klassifizierung mit dem Stochastic Gradient Descent Classifier wird die Python Bibliothek *scikit-learn* verwendet. Dies ist eine häufig genutzte Bibliothek, welche Algorithmen rund um das Thema maschinelles Lernen bereitstellt. Hierzu gehören die Themen Klassifikation, Regression und Clustering, aber auch Funktionen für die Einbettung der Wörter in zum Beispiel Vektoren.[32]

Auch für dieses Klassifizierungsmodell wird der Trainingsdatensatz aufbereitet. Hier werden als Eingabe alle Schlagwörter in Form einer Zeichenkette benötigt, die durch ein Komma voneinander getrennt sind. Zusätzlich sind die Averbis Klassen jedes Dokuments für das Trainieren erforderlich. Diese Daten werden nun wieder in ein Trainings- (75 %) und Testdatenset (25 %) aufgeteilt und dem Klassifikator mitgegeben. Damit der SGD-Classifier die Daten klassifizieren kann, durchläuft der Input zunächst eine Pipeline. Hier werden zuerst die Wörter in Vektoren umgewandelt und jedes Token wird in seinem Vorkommen gezählt. Dies erfolgt durch die Funktion *CountVectorizer*, die die Dokumente in eine Matrix umwandelt, welche die Anzahl der Tokens beinhaltet.[33] Um in diesem Vektorraummodell die Terme von jedem Dokument im Verhältnis zum Gesamtvokabular zu gewichten, wird die TF-IDF Gewichtung angewandt. Hier wird die Matrix mittels der Funktion *TfidfTransformer* in eine normalisierte TF-IDF Repräsentation transformiert.[34] Dem Algorithmus SGD-Classifier wird der Parameter „hinge“ mitgegeben, welcher die Klassifizierung mit der SVM als Grundlage durchführt. Im Anschluss wird das Modell auf die Testdaten angewandt. Für die Evaluierung des Modells wird eine Tabelle mit den Bewertungsmetriken Precision, Recall, F1-score und Support für jede Klasse einzeln ausgegeben.

[32] Vgl. scikit-learn, „sklearn.linear_model.SGDClassifier,” zuletzt geprüft am 17.07.2023, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html.

[33] Vgl. scikit-learn, „sklearn.feature_extraction.text.CountVectorizer,” zuletzt geprüft am 17.07.2023, https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html.

[34] Vgl. scikit-learn, „sklearn.feature_extraction.text.TfidfTransformer,” zuletzt geprüft am 17.07.2023, https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html.

4. Evaluierung der Methodik und Modelle

4.1. Evaluierung der Schlagwortextraktion

Die Evaluierung der Schlagwortextraktion bezieht sich auf die Messung der Leistung des Matching-Algorithmus. Das Ziel besteht hierin, die Qualität und Genauigkeit der extrahierten Schlüsselwörter zu bewerten. Zur Bewertung der automatischen Schlagwortextraktion mittels eines Algorithmus können die extrahierten Schlagworte mit dem sogenannten Golden Record verglichen werden. Der Golden Record dient als Referenz für die tatsächlich relevanten Schlagwörter in dem Text, da er eine intellektuelle Annotation der Schlüsselwörter von Experten beinhaltet. Hier wird eine manuelle thematische Zuordnung durch Personen vorgenommen, die sich mit dem Text auseinandergesetzt haben. Somit ist es möglich durch Vergleich dieser beiden Ergebnismengen mit geeigneten Metriken die Schlagwortextraktion zu bewerten. Es ist wichtig zu beachten, dass die Evaluierung der Extraktion stark von der Qualität und Verfügbarkeit der Referenzmenge sowie den Merkmalen des vorliegenden Datensatzes abhängt. In Fällen, in denen diese Ressourcen begrenzt sind, sind alternative Ansätze erforderlich, um eine angemessene Evaluierung durchzuführen. Dies kann zum Beispiel eine qualitative Beurteilung durch Experten sein. Die automatisierte Evaluierung der Schlagwortextraktion stellt sich in diesem Fall als herausfordernd dar, da hier einige Probleme auftauchen. Zum einen gibt es nur eine Referenzmenge von Schlagworten, die aus der MeSH Terminologie bestehen. Es sind keine AGROVOC Daten vorhanden. Des Weiteren wurde die Referenzmenge nur teilweise manuell erstellt, und es ist nicht klar ersichtlich, welcher Teil manuell und welcher automatisiert stattgefunden hat. Dies erschwert den Vergleich zwischen den extrahierten Schlagworten und der Referenzmenge. Ein weiteres Problem besteht darin, dass die Terme im vorliegenden Datensatz überwiegend leer sind, was bedeutet, dass es nicht genügend Schlagwörter gibt, um einen genauen Vergleich durchzuführen. Ohne eine ausreichende Anzahl von Schlagworten im Datensatz ist es schwierig, die Leistung der Schlagwortextraktion angemessen zu bewerten. Das Ergebnis dieser beiden Ergebnismengen unterstützt nochmal die Probleme der Bewertung. Es sind lediglich ca. 19 % der Terme in beiden Mengen zu finden. Es ist jedoch hervorzuheben, dass der in dieser Arbeit angewandte automatisierte Ansatz Schlagwörter extrahiert hat, bei denen die Averbis Extraktion keine Terme geliefert hat. Dies zeigt die Fähigkeit zusätzliche Informationen zu erfassen, die für die darauffolgende Textklassifizierung einen hohen Stellenwert haben. Somit war es

nicht möglich, die erste Forschungsfrage (F1), inwiefern eine automatische Schlagwortextraktion mit einer intellektuellen Themenanalyse vergleichbar ist, konkret zu beantworten. Dies ist darauf zurückzuführen, dass die Datengrundlage nicht gegeben ist.

4.2. Evaluierung Klassifikationsmodelle

Im Anschluss an das Modelltraining erfolgt die Evaluierung anhand der Testmengen. Die Qualität der Modelle wird mit den in Abschnitt 2.4 beschriebenen Evaluierungsmetriken gemessen. Hier werden zunächst die Modelle auf ihre Performance geprüft, indem sie mit dem Golden Record, das heißt den Averbis Klassen verglichen werden.

Die Evaluierung eines LDA-Modells ist nicht einfach durchzuführen, da das Modell unüberwacht lernt. Dies bedeutet, dass es keine eindeutigen Referenzwerte wie die Averbis Klassen gibt. Es wird als Ergebnis eine Zahl, welche das Thema repräsentiert, ausgegeben. Im Weiteren sind die in der Klasse zugehörigen, gewichteten Wörter bekannt. Hier wird auf den ersten Blick deutlich, dass eine intellektuelle Zuordnung erforderlich sein muss. Da jedoch die Averbis Klassenzuordnung zu den Dokumenten vorliegt, wird in diesem Teil der Arbeit eine Python Funktion für eine automatisierte Zuordnung entwickelt. Diese wird mittels der Bewertungsmetrik F1-Score die Klassenzuordnung bewerten. Die Idee hinter dieser Funktion besteht darin, mittels Permutation die numerische Klasseneinteilung mit den alphabetischen Averbis Klassen zu vergleichen. Permutation ist die Bezeichnung für die verschiedenen Möglichkeiten der Anordnung einer Menge oder Zeichenkette.[35] Voraussetzung hierfür ist, dass alle Elemente aufgrund ihrer Position eine Eindeutigkeit erhalten. Hierfür wurde das Python-Modul *Itertools* verwendet, welche verschiedene Funktionen für das Arbeiten mit Iteratoren bietet.[36] Mit dieser Bibliothek wird die Permutation für die Averbis Klassen und die Zahlen, die das vorhergesagte Thema der LDA repräsentieren, erstellt. Im zweiten Schritt wird für jede Anordnung der F1-Score berechnet, indem die vorhergesagte Klasse der LDA und das Averbis-Thema als wahre Klasse mitgegeben wird. Somit wird ein Wert zurückgegeben, der eine Aussage über den Anteil der korrekt vorhergesagten Klassen geben kann. Die Anordnung mit dem höchsten F1-Score gilt in diesem Fall als beste

[35] Vgl. *GeeksforGeeks,* „Python Itertools.Permutations," 19.02.2020, zuletzt geprüft am 21.07.2023, https://www.geeksforgeeks.org/python-itertools-permutations/?ref=lbp.

[36] Vgl. Ebenda

Zuordnung, da sie die höchste Anzahl an korrekt vorhergesagten Werten beinhaltet und wird daher abschließend mit den alphabetischen Klassen ersetzt.

Zur Veranschaulichung ist nachfolgend eine Abbildung der Funktion dargestellt:

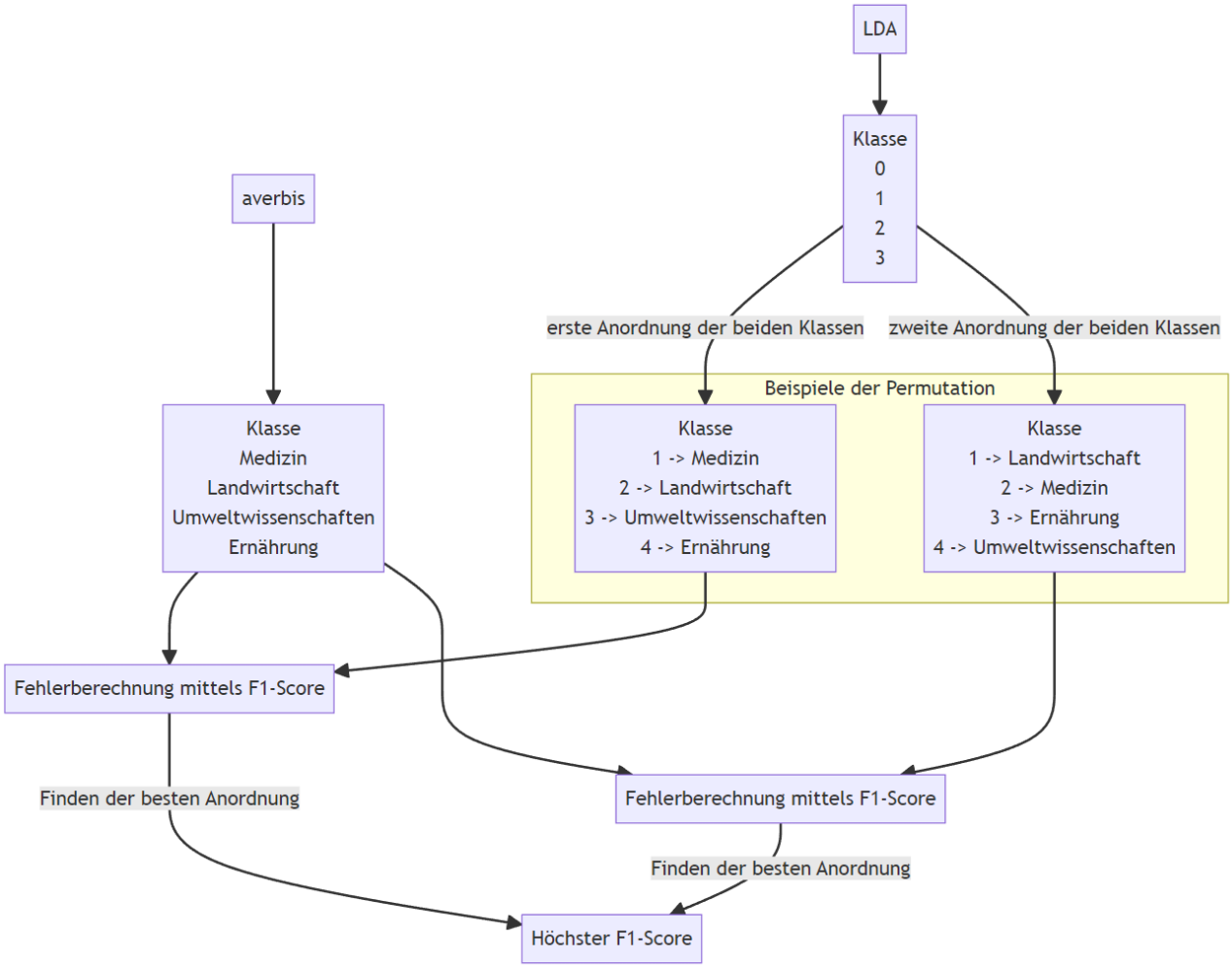

Abbildung 5 - Darstellung der Evaluierungsfunktion der LDA

Vor der Implementierung dieser Funktion erfolgt die Modellanwendung auf die Testdaten. Die einzelnen Datensätze werden nun mit Hilfe des Modells in ihre Themenbereiche eingeteilt. Hierfür gibt es für jedes Dokument einen Integer zwischen 0 und 3 und die entsprechende Wahrscheinlichkeit zurück. Die Wahrscheinlichkeitsangabe drückt aus, wie wahrscheinlich es ist, dass das Dokument dieser Klasse zugeordnet werden kann. Um die Modelle in ihrem Lernprozess analysieren und bewerten zu können, wurde ein Training des Algorithmus mit unterschiedlichen Datensatzgrößen durchgeführt (vgl. Abb. 6).

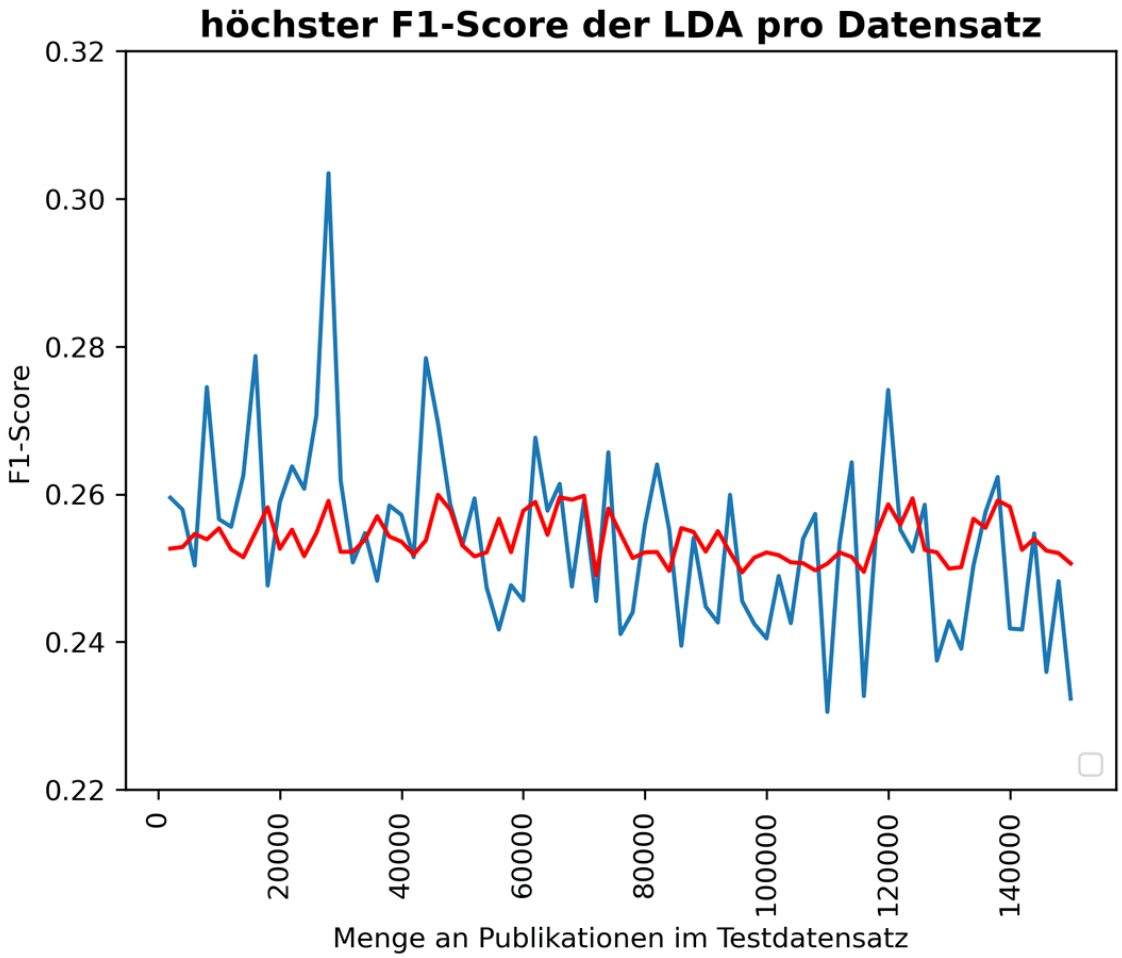

Abbildung 6 - Höchster F1-Score der LDA pro 2.000 Datensätze

Wie in der oberen Abbildung zu sehen ist, oszilliert der F1-Score sehr stark zwischen den Werten 0,23 und 0,28. Bei einer Datenbasis von ca. 20.800 Datensätzen ist ein Ausreißer mit einem Wert über 0,3 festzustellen. Da die Werte so stark schwanken, ist insgesamt keine Tendenz zu erkennen, wo sich der F1-Score einpendeln könnte. Um die Ausreißer und die stark schwankenden Werte interpretieren zu können, wurde eine geglättete Kurve in roter Farbe eingebunden. Die Ausreißer werden im Verlauf um die geglättete Kurve herum etwas kleiner. Allerdings tendiert der F1-Score mit zunehmendem Publikationsaufkommen eher zu niedrigeren Werten. Da der Score mit der Zahl 1 das beste Ergebnis darstellt, sind die von dem Modell erreichten Werte insgesamt nicht als hoch einzustufen und können daher nicht als zuverlässig angesehen werden. Die starken Schwankungen können darauf zurückgeführt werden, dass die LDA in ihrer Funktionsweise anders abläuft als zum Beispiel der SGD-Classifier. Die LDA definiert die Themenklassen bei den verschiedenen Datensatzgrößen stets anders, da sie aus den verschiedenen Dokumenten generiert werden und alle Themen in jedem Dokument

vorkommen sollen. So kann die Auswahl der Wörter je nach Anzahl der Dokumente in den einzelnen Themen unterschiedlich strukturiert und gewichtet sein. Der weitere Verlauf ist mit diesen wenigen Datensätzen daher schwer abzuschätzen. Es ist jedoch möglich, dass sich der Wert mit allen Datensätzen verbessert, da sich die Themen in ihrer Definition nochmals stark verändern werden. Es wäre daher ein logischer nächster Schritt, dieses Modell mit einer höheren Anzahl an Datensätzen zu testen.

Mit allen vorhandenen Datensätzen von 385.943 Einträgen kann folgende Themenzuordnung aus den vorhergesagten Klassen und den Averbis Klassen mit einem Wert von 0,2675 vorhergesagt werden:

Vorhergesagte Klasse (LDA)	**Averbis-Klasse in Wort**
0	Medizin
1	Landwirtschaft
2	Umweltwissenschaften
3	Ernährung

Tabelle 1 - Themenzuordnung der vorhergesagten und der Averbis Klasse

Die Evaluierung der Klassifizierung mittels des SGD-Classifier ist deutlich einfacher im Vergleich zur LDA, da es sich bei diesem Modell um überwachtes Lernen handelt. Somit ist hier die Vergleichbarkeit mit den Averbis Klassen direkt gegeben.

Nach der Anwendung des Modells auf den Testdatensatz wird eine Konfusionsmatrix ausgegeben, die die Metriken Accuracy, Precision, Recall und das Gleichgewichtsmaß F1-Score beinhaltet.

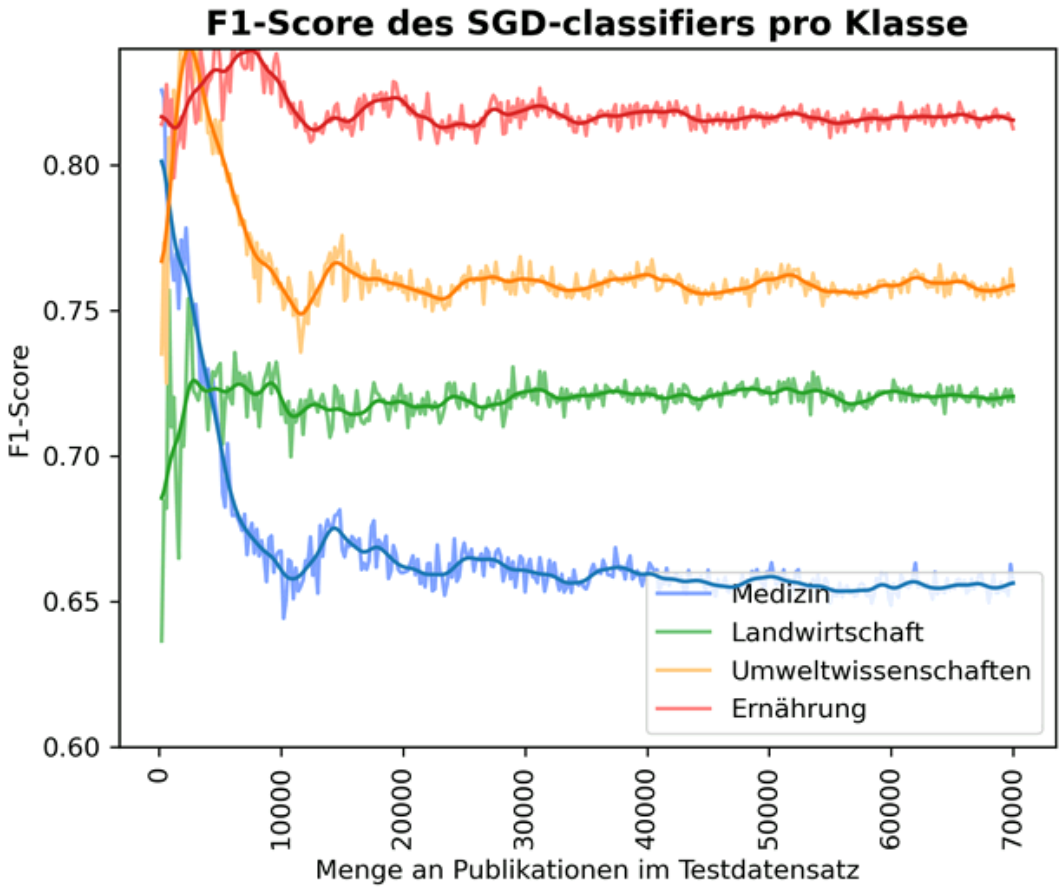

Abbildung 7 - F1-Score des SGD-Modells pro Klasse

In Abbildung 7 ist zu erkennen, dass das Modell zu Beginn schnell einen hohen Anstieg des F1-Scores erreicht. Auffällig ist jedoch, dass die Klassen „Medizin“ und „Umweltwissenschaften“ wieder stark abfallen und sich im weiteren Verlauf auf deutlich niedrigeren Werten einpendeln. Nach den ersten 15.000 Datensätzen läuft das Modell aber insgesamt sehr stabil. Anzumerken ist, dass die Klasse „Ernährung“ mit einem F1-Score von über 0,8070 am besten klassifiziert werden konnte. Am schlechtesten schnitt die Klasse „Medizin“ mit einem Wert von 0,6546 ab. Dies ist überraschend, da erwartet wurde, dass diese Klasse auf Basis der MeSH Termini sehr gut klassifiziert werden könnte. Hier ist jedoch zu beachten, dass der MeSH Thesaurus die meisten Terme beinhaltet und somit aus überproportional vielen Dokumenten MeSH Terme extrahiert

worden sind. Diese Terme können daher nicht nur ein Merkmal für die Klasse „Medizin“ darstellen, sondern auch Merkmale anderer Klassen sein. Hier ist wichtig zu erwähnen, dass Überschneidungen in den beiden eingebundenen Thesauri auftreten. Man könnte darauf schließen, dass hier ein Bias entsteht, da sich die Merkmale in den Klassen überschneiden und es für das Modell nicht einfach ist, diese Merkmale zu unterscheiden. Daher wäre es von Vorteil an diesem Ansatz noch weitere Optimierungen durchzuführen. Beim Betrachten der Precision-Werte, ist zu erkennen, dass bei den Klassen „Ernährung“ und „Medizin“ ein hoher Anteil an tatsächlich wahren Fällen zu verzeichnen ist (vgl. Abb. 12 im Anhang). Hier werden jedoch nicht die falschen Werte in Betracht gezogen. Somit wird sich im zweiten Schritt die Recall-Werte angeschaut. Hier ist zu erkennen, dass das Modell bei der Klasse „Medizin“ auch sehr viele fälschliche richtige Fälle zuordnet (vgl. Abb. 13 im Anhang). Daher kann auch der F1-Score kein gutes Ergebnis für diese Klasse verzeichnen. Die Recall-Werte lassen im Weiteren erkennen, dass bei den Klassen „Ernährung“ und „Umweltwissenschaften“ tatsächlich mehr wahre positive Fälle richtig zugeordnet wurden. Insgesamt erzielt das Modell eine Accuracy von 73,47 %, welches als ein solides Ergebnis zu betrachten ist. Da die Klassen alle mit einem Wert über 0,65 eingeordnet wurden, ist das Modell als gutes Mittelmaß anzusehen. Wie bei der Klassifizierung der LDA wurde auch bei diesem Modell das Klassifizieren mit der Averbis Klasse „Rest“ eingebunden. Hier erreicht das Modell eine Accuracy von 64,66 % und klassifiziert die Klasse „Rest“ mit dem höchsten F1-Score am besten. Dies unterstreicht nochmal die Aussage, dass diese Kategorie einen Bias im Datensatz hinterlässt und eine separate Analyse dieser erforderlich ist.

5. Diskussion der Ergebnisse

In diesem Teil wird die angewandte Methodik und die Performance der Klassifizierungsmodelle mit der Benchmark verglichen. Hierzu wurden die erstellten Rohdaten mit den Vorverarbeitungsschritten des Vorprojekts aufbereitet und auf die Klassifizierungsmodelle angewandt. Somit ist die Vergleichbarkeit der unterschiedlichen Verarbeitungsschritte gewährleistet, da dieselbe Datengrundlage verwendet wird.

Die Benchmark der LDA war mit einem errechneten F1-Score von 0,0456 nicht hoch angesetzt.[37] Dieser ist jedoch händisch berechnet worden und lässt sich schwer mit dem erzielten Ergebnis in dieser Arbeit vergleichen. Daher wurde ein Vergleich mit der Evaluierungsfunktion eingebunden und im nachfolgenden Diagramm dargestellt:

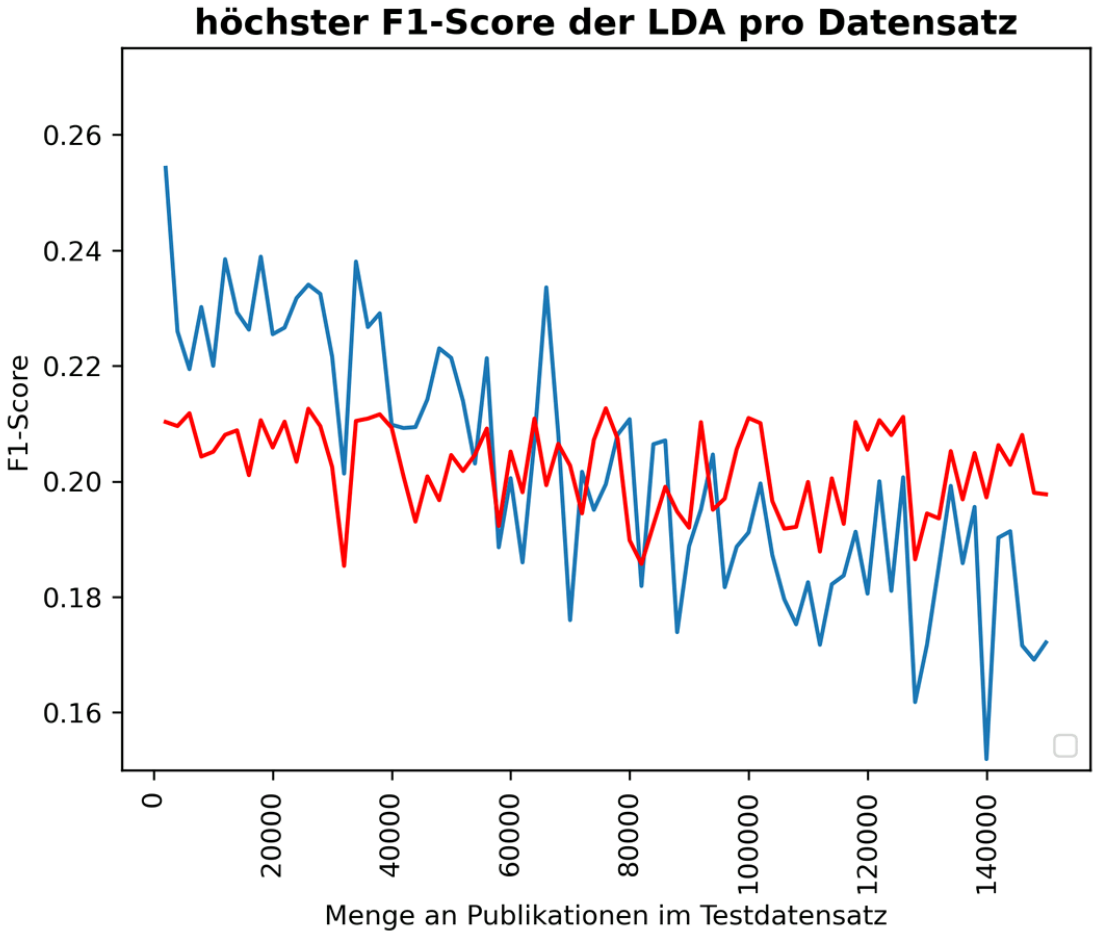

Abbildung 8 - höchster F1-Score der LDA pro Datensatz der Benchmark

[37] Vgl. Prantz, „Verbesserung der automatischen Dokument-Klassifikation für den Discovery Service LIVIVO von ZB MED."

Im Vergleich zum F1-Score der Benchmark liegt das in dieser Arbeit trainierte Modell vorne und konnte die Datensätze besser klassifizieren. Wie in Abbildung 8 zu sehen, schwankt der F1-Score der Benchmark zwischen den Werten 0,16 und 0,26. Auch die geglättete Kurve schwankt hier um einiges mehr und lässt viele Ausreißer zu. Insgesamt zeigt das Modell einen absteigenden Trend auf. Dies unterstützt auch nochmal der F1-Score, der mit allen Benchmark-Datensätzen berechnet wurde. Dieser liegt bei 0,176. Somit konnte das LDA-Modell durch die Aufbereitung mittels Schlagwortextraktion eine Verbesserung um 0,09 Punkte verzeichnen.

Außerdem wurde mit dem optimierten Korpus ein Kohärenz Wert von 0,3364 und eine Komplexität von -7,417 erreicht. Im Vergleich zur Benchmark, mit einem Kohärenz-Wert von 0,3646 und einer Komplexität von -9,1060, schnitt das Modell etwas schlechter ab. Das bedeutet, dass das LDA-Modell der Benchmark schlüssigere Themen vorweisen kann. Die Abweichung dieser Werte ist jedoch sehr gering und wird daher nicht als bedeutend bewertet.

Es lässt sich also sagen, dass die Schlagwortextraktion eine Verbesserung für das Klassifizieren mit dem LDA-Modell erreichen konnte. Daher kann davon ausgegangen werden, dass das Modell mit spezifischeren Wörtern besser umgehen kann und eine sinnvollere Klasseneinteilung durchführen konnte.

Wie bereits beschrieben, hat das SGD-Modell mit den Schlüsselwörtern als Input im Großen und Ganzen keine schlechte Klassifizierung erzielt. Doch nach einem Vergleich mit der Benchmark ist zu erkennen, dass das Modell die Klasseneinteilung nach dem F1-Score zu urteilen, schlechter durchführen konnte. In Abbildung 9 ist eine Darstellung des F1-Scores mit den Daten der Benchmark für ein Vergleich dargestellt.

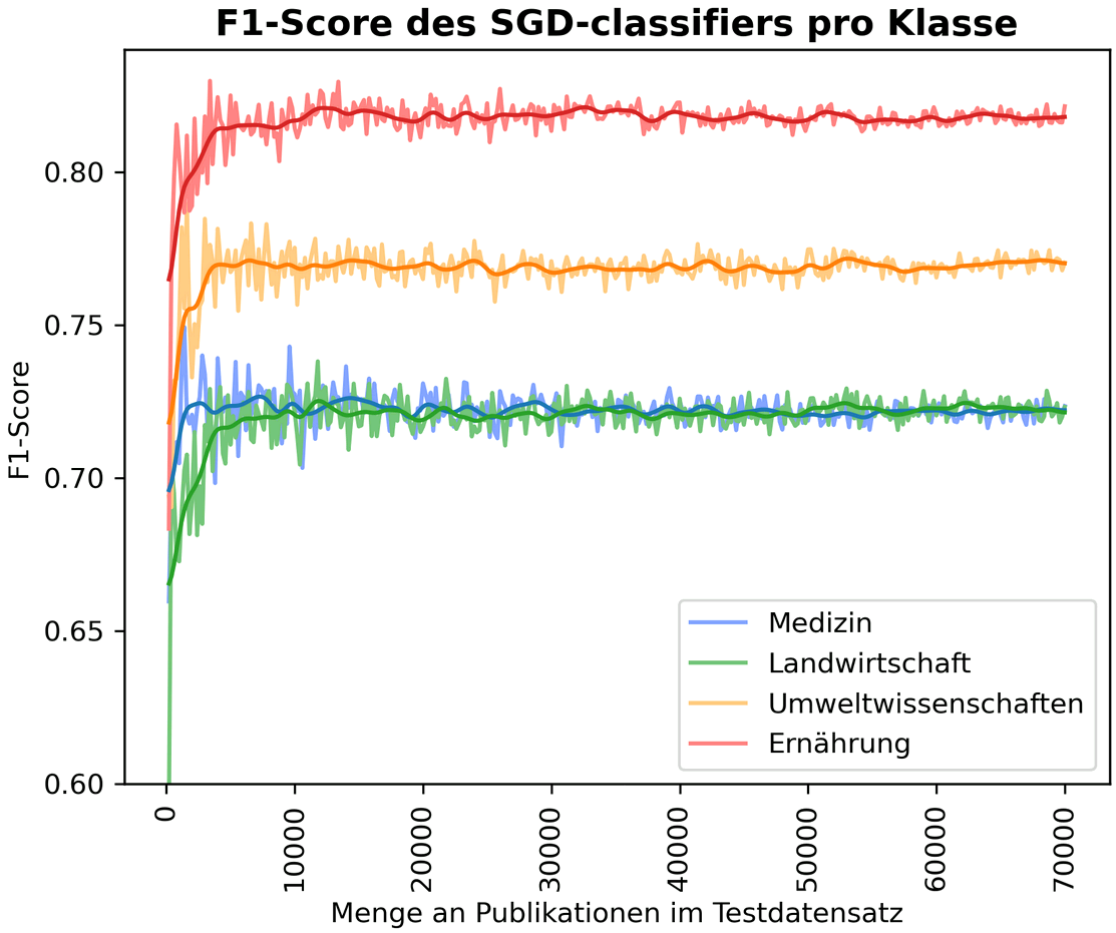

Abbildung 9 - F1-Score des SGD-Classifiers der Benchmark

Das SGD-Modell der Benchmark klassifiziert im Vergleich viel stabiler über die unterschiedlichen Datensatzmengen hinweg. Die Klassen „Ernährung“, „Umweltwissenschaften“ und „Landwirtschaft“ weisen im Vergleich sehr ähnliche F1-Scores auf. Jedoch schneidet die Klasse „Medizin“ hier um einiges besser ab. Sie kann um 0,07 Punkte besser erkannt werden. Es ist somit ein signifikanter Unterschied in der Einteilung der Klassen im Bereich Medizin zu erkennen. Das Modell erreicht insgesamt einen Accuracy-Score von 76,16 %. Somit liegt das Modell mit den Input-Daten der Benchmark um 2,69 % besser. Also muss hier gesagt werden, dass die neuen Input-Daten

mit diesem Modell keine Verbesserung der Klassifizierung aufzeigen konnte. In Anbetracht der Precision- und Recall-Werte ist auch hier zu sehen, dass das Benchmark-Modell die Klassen „Ernährung“ und „Umweltwissenschaften“ häufiger korrekt klassifiziert und die Klassen „Medizin“ und „Landwirtschaft“ schlechter abschneiden. Daher muss die erste Vermutung, dass dies einen Zusammenhang mit den Überschneidungen in den Thesauri sein könnte, revidiert werden. Der Bias gegenüber den beiden Klassen bleibt auch ohne die Schlagwörter erhalten und konnte mit der Korpus Optimierung nicht verbessert werden.

Nun kann abschließend auf die Forschungsfrage 2 eingegangen werden, welche die Performance der Klassifizierungsmodelle infrage stellt. Es war möglich die Performance der LDA mit der Optimierung des Korpus zu verbessern. Dennoch können bei beiden Modellen keine guten Ergebnisse erzielt werden, da der F1-Score im Allgemeinen nicht sehr hoch ist. Hier ist es sinnvoll das Modell noch einmal für alle Datensätze zu trainieren und zu evaluieren, um abschließend ein aussagekräftiges Analyseergebnis zu erheben und die Performance bewerten zu können. Das SGD-Modell schnitt mit den Schlagwörtern als Input schlechter ab. Die Performance verschlechterte sich insgesamt und gerade in der Klasse Medizin ist ein Bias zu erkennen. Wie bereits beschrieben, wurde dies darauf zurückgeführt, dass die meisten der Dokumente MeSH Terme beinhalten und somit nicht nur für eine Klasse „Medizin“ stehen, sondern in mehreren Klassen auftauchen. Es entsteht die Vermutung, dass hier ein Ungleichgewicht durch die unterschiedlichen Größen der Thesauri entstanden sein könnte.

6. Zusammenfassung und Ausblick

Zusammenfassend lässt sich sagen, dass die Methoden der Benchmark um einige Parameter erweitert werden konnten. Es wurde eine Spracherkennung der Texte implementiert, die zu einer sprachspezifischen Schlagwortextraktion mittels Thesauri führte. Darüber hinaus wurde eine automatisierte LDA-Evaluierung eingebunden. Es lässt sich feststellen, dass das SGD-Modell mit einer Accuracy von 73,47 % bereits eine gute Performance erreicht. Jedoch konnte hier der Wert der Benchmark nicht erreicht werden. Die Korpus Optimierung konnte für dieses Modell keine positiven Auswirkungen verzeichnen. Die Performance der LDA konnte mit einem F1-Score von 0,2675 durch die Schlüsselwortextraktion verbessert werden. Jedoch muss auch hier erwähnt werden, dass dies kein hoher Wert ist und es sinnvoll wäre, die Datensatzgröße zu erhöhen und zu betrachten, ob sich die Performance verbessert. Natürlich kann im Weiteren auch ein Hyperparameter Tuning die Modelle noch optimieren, jedoch sind dazu mehr Ressourcen in Bezug auf Rechenleistung und Zeit notwendig.

Um dem Ungleichgewicht der Schlagwortextraktion entgegenzuwirken, könnten die Themen „Ernährung“ und „Umweltwissenschaften“ noch weiter spezifiziert werden. Diese Themen sind im AGROVOC-Thesaurus nur teilweise vertreten und eine Ausweitung dieses Vokabulars könnte eine bessere Klassifizierung möglich machen. Daher wäre eine Erweiterung an Wissensorganisationssystemen zu diesen Themen hilfreich. Im Weiteren gibt es hier auch ein weiteres Modell von AGROVOC, welches mehr Terme umfasst und eingebunden werden könnte, sofern die Ressourcen hier zur Verfügung stünden. Genauso könnte die Überschneidung der bereits eingebundenen Thesauri analysiert werden, um zu prüfen, ob hierdurch Verzerrungen entstehen.

Ein weiterer Versuch einer gesteigerten Performance der Klassifizierungsmodelle wäre, das SGDC-Modell mit den beiden Preprocessing Methoden zu verbinden. Dabei könnten zuerst die Preprocessing Schritte der Benchmark implementiert werden und die Schlüsselwörter als zweites Feature in einen zweiten Input-Layer integriert werden. Durch die Spracherkennung kann im Vorhinein eine sprachspezifische Stopwort-Entfernung und Tokenisierung erfolgen, um diesen Bias zu vermeiden. Weitere mögliche Schritte könnten das Annotieren der Texte für Named Entity Recognition (NER) sein, um die Modelle auf spezifische Entitäten trainieren zu können. Außerdem könnte es sinnvoll sein, weitere Klassifikationsmodelle auszuprobieren, um die Leistung der Klassifizierung zu verbessern.

Literaturverzeichnis

Averbis GmbH. „Informationen zum Unternehmen Averbis GmbH." Zuletzt geprüft am 02.08.2023. https://averbis.com/de/unternehmen/#1592585215750-120d0d24-b750.

Brownlee, Jason. „ProcessPoolExecutor in Python: The Complete Guide." Zuletzt geprüft am 10.07.2023. https://superfastpython.com/processpoolexecutor-in-python/#Python_Processes_and_the_Need_for_Process_Pools.

Data Science With Chris. „SGDRegressor and SGDClassifier." Zuletzt geprüft am 27.07.2023. https://datasciencewithchris.com/sgdregressor-sgdclassifier/.

Eisenstein. *Introduction to natural language processing.* Adaptive computation and machine learning. Cambridge, Massachusetts, London: The MIT Press, 2018.

Gandhi, Rohith. „Support Vector Machine — Introduction to Machine Learning Algorithms." *Towards Data Science*, 07.06.2018. Zuletzt geprüft am 25.07.2023. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

„Gensim: topic modelling for humans." Zuletzt geprüft am 26.07.2023. https://radimrehurek.com/gensim/intro.html.

Hinguruduwa, Lahiru, Edgard Marx, Tommaso Soru, und Thomas Riechert. „Assessing Language Identification over DBpedia." In *2021 IEEE 15th International Conference*, 296–97. Zuletzt geprüft am 13.06.2023. https://www.researchgate.net/profile/Edgard-Marx-2/publication/351010850_Assessing_Language_Identification_over_DBpedia/links/613f9cd46c61e2367c797ef6/Assessing-Language-Identification-over-DBpedia.pdf.

Kherwa, Pooja, und Poonam Bansal. „Topic Modeling: A Comprehensive Review." *ICST Transactions on Scalable Information Systems* 0, Nr. 0 (2018): 159623. Zuletzt geprüft am 30.06.2023. https://doi.org/10.4108/eai.13-7-2018.159623. https://eudl.eu/pdf/10.4108/eai.13-7-2018.159623.

„Medical Subject Headings - Home Page." Zuletzt geprüft am 25.07.2023. https://www.nlm.nih.gov/mesh/meshhome.html.

Oppermann, Artem. „Accuracy, Precision, Recall, F1-Score und Specificity - KI Tutorials." Zuletzt geprüft am 17.07.2023. https://artemoppermann.com/de/accuracy-precision-recall-f1-score-und-specificity/.

Prantz, Max. „Verbesserung der automatischen Dokument-Klassifikation für den Discovery Service LIVIVO von ZB MED." Technische Hochschule Köln, 2023. Zuletzt geprüft am 02.08.2023.

GeeksforGeeks. „Python Itertools.Permutations." 19.02.2020. Zuletzt geprüft am 21.07.2023. https://www.geeksforgeeks.org/python-itertools-permutations/?ref=lbp.

scikit-learn. „sklearn.linear_model.SGDClassifier." Zuletzt geprüft am 17.07.2023. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html.

scikit-learn. „sklearn.feature_extraction.text.CountVectorizer." Zuletzt geprüft am 17.07.2023. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html.

scikit-learn. „sklearn.feature_extraction.text.TfidfTransformer." Zuletzt geprüft am 17.07.2023. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html.

scikit-learn. „1.5. Stochastic Gradient Descent.” Zuletzt geprüft am 25.07.2023. https://scikit-learn.org/stable/modules/sgd.html.

Staab, Steffen. „The Semantic Web Revisited.” 2006. Zuletzt geprüft am 25.07.2023. https://web.archive.org/web/20130320130521/http:/eprints.soton.ac.uk/262614/1/Semantic_Web_Revisted.pdf.

Subirats-Coll, Imma, Kristin Kolshus, Andrea Turbati, Armando Stellato, Esther Mietzsch, Daniel Martini, und Marcia Zeng. „AGROVOC: The linked data concept hub for food and agriculture.” *Computers and Electronics in Agriculture* 196 (2022): 105965. https://doi.org/10.1016/j.compag.2020.105965.

„TF-IDF — Term Frequency-Inverse Document Frequency – LearnDataSci.” Zuletzt geprüft am 27.07.2023. https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/.

W3C. „Web Standards.” Zuletzt geprüft am 25.07.2023. https://www.w3.org/standards/semanticweb/ontology.

ZB MED - Informationszentrum Lebenswissenschaften. „LIVIVO-Suchportal.” Zuletzt geprüft am 25.07.2023. https://www.zbmed.de/recherchieren/livivo.

Anhangsverzeichnis

Anhang

Nachfolgender Link führt zum Code-Repository, welches alle im Rahmen dieser Arbeit entwickelten Code-Dateien enthält. Die zugehörige Dokumentation ist in der README.md-Datei zu finden.

- https://github.com/mullrich192/BA_text_classification

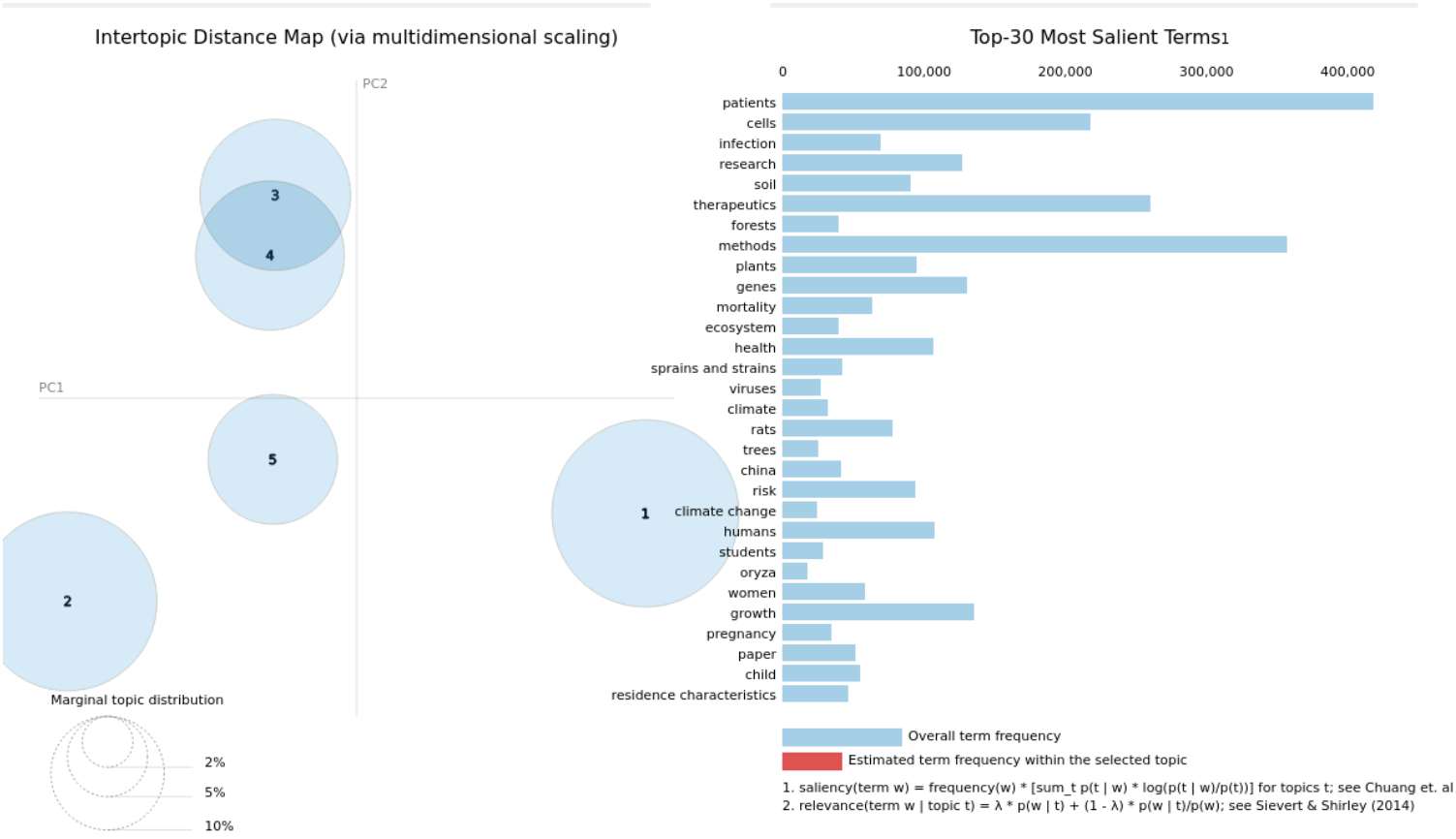

Abbildung 10 - Intertopic Distance Map (Training mit Vorgabe von 5 Klassen)

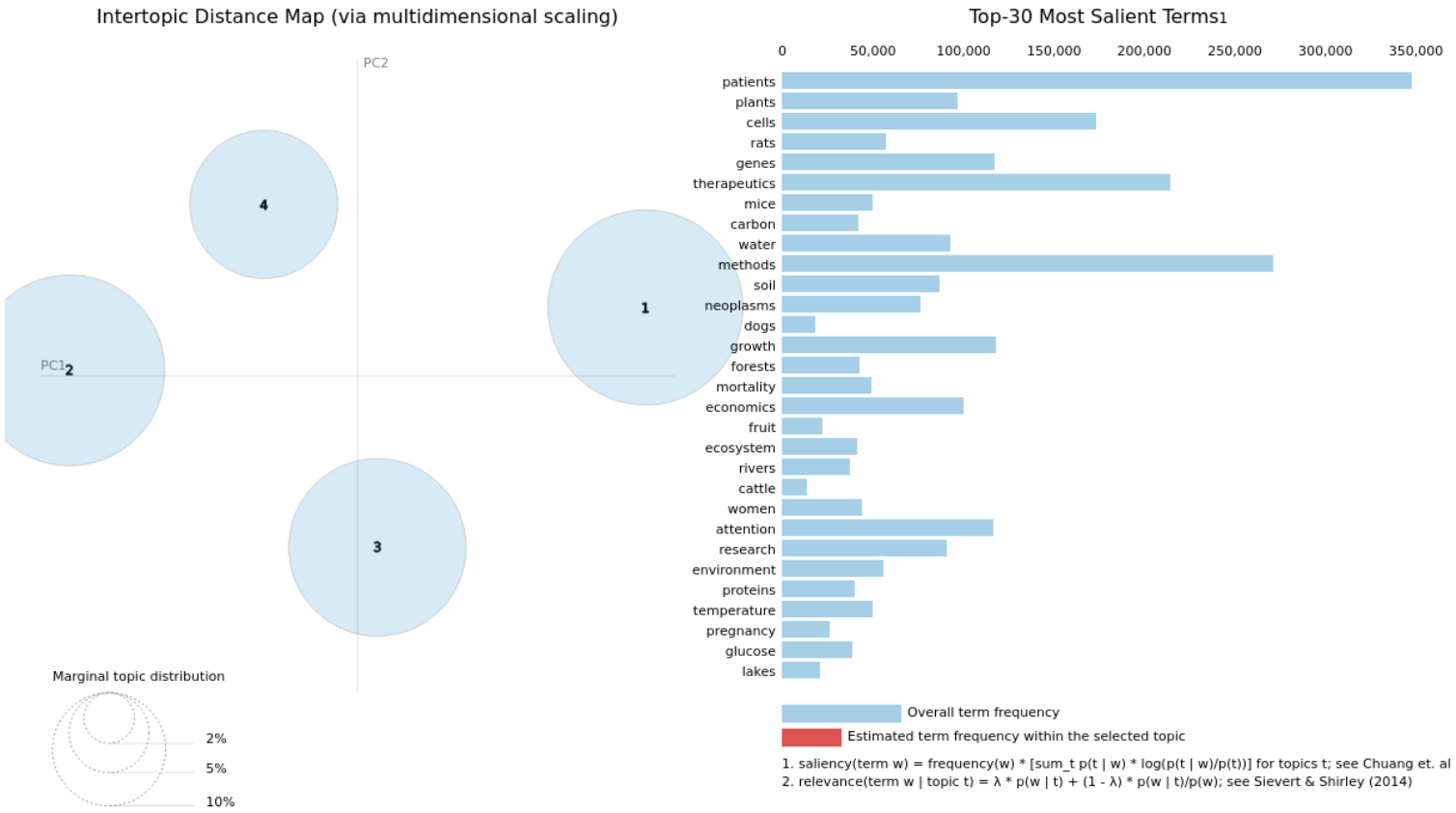

Abbildung 11 - Intertopic Distance Map (Training mit Vorgabe von 4 Klassen)

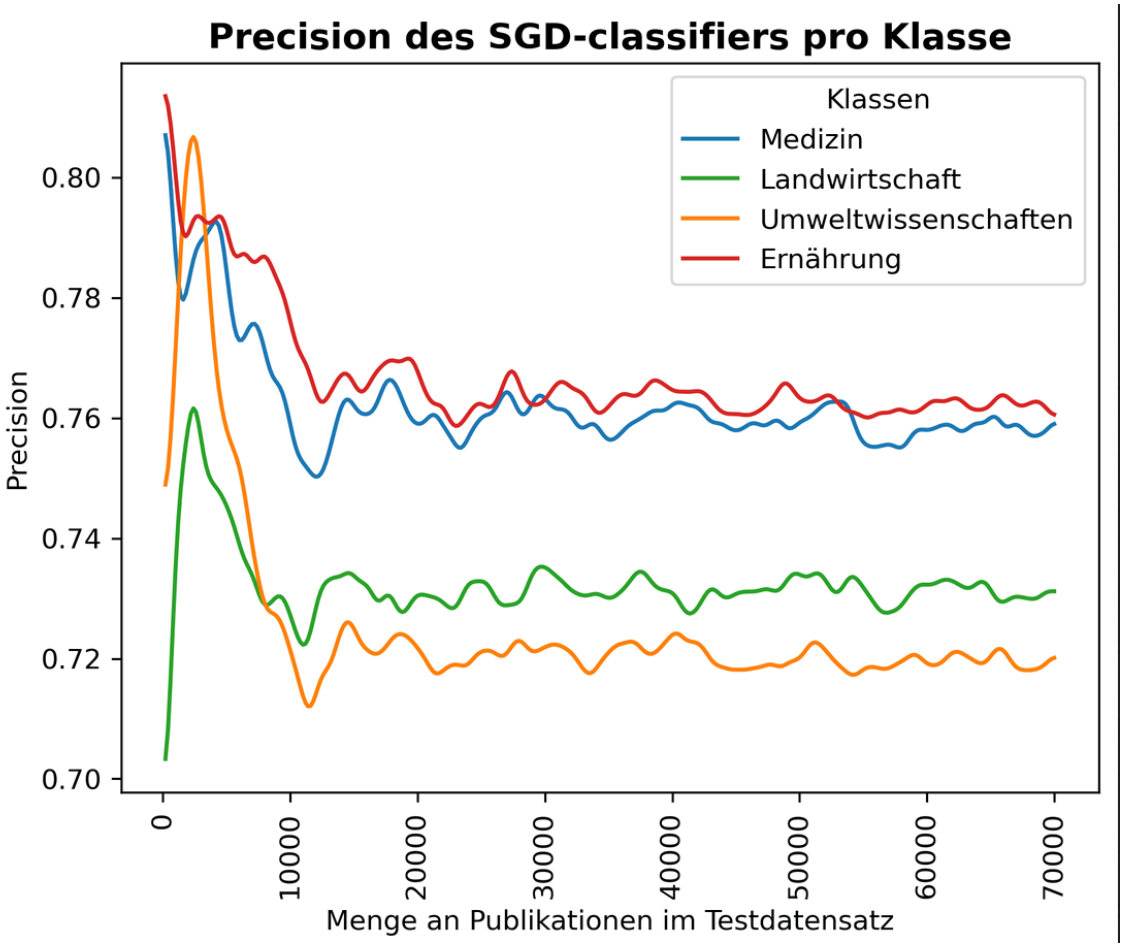

Abbildung 12 - Precision-Werte des SGDC

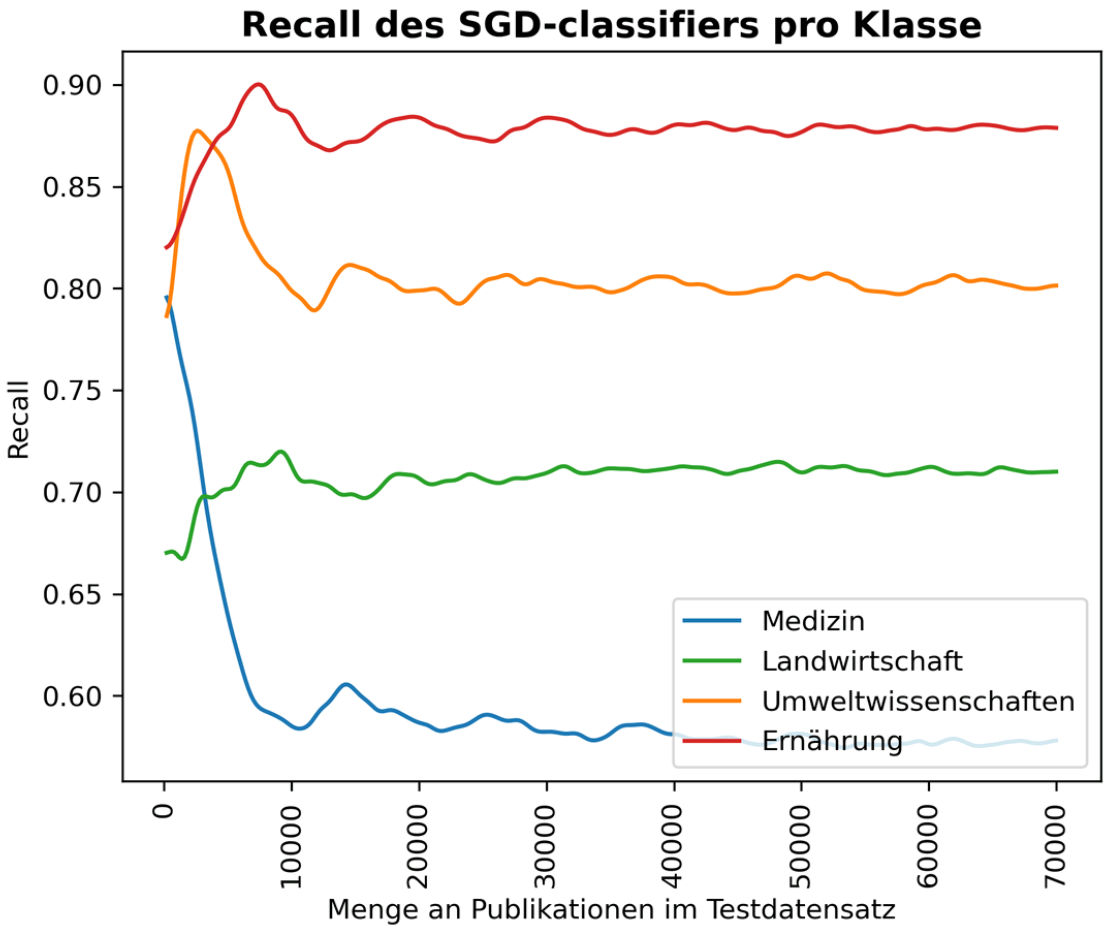

Abbildung 13 - Recall-Werte des SGDC