Kunden:ZHAW:2015:00067.03-Titelblatt-Bachelorbuch:Versionen:Institute:wmf:2015-04-27-ZHAW-Titelblatt-Bachelorbuch-IDP.wmf

**Bachelorarbeit (Informatik)**

Analyse von Umsatzzahlen aus dem Gastronomiebereich

|  |  |  |
| --- | --- | --- |
| **Autoren** |  | Manu Paul Kunnumpurathu  Muhammed Fatih Solmaz |
| **Hauptbetreuung** |  | Dr. Martin Frey |
| **Nebenbetreuung** |  | Dr. Reto Bürgin |
| **Industriepartner** |  | Prognolite |
| **Externe Betreuung** |  | Marco Wirthlin |
| **Datum** |  | 10.06.2022 |

**Erklärung betreffend das selbstständige Verfassen einer Bachelorarbeit an der School of Engineering**

**Erklärung betreffend das selbstständige Verfassen einer Bachelorarbeit an der School of Engineering**

Mit der Abgabe dieser Bachelorarbeit versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat. (Bei Gruppenarbeiten gelten die Leistungen der übrigen Gruppenmitglieder nicht als fremde Hilfe.)

Der/die unterzeichnende Studierende erklärt, dass alle zitierten Quellen (auch Internetseiten) im Text oder Anhang korrekt nachgewiesen sind, d.h. dass die Bachelorarbeit keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

Bei Verfehlungen aller Art treten die Paragraphen 39 und 40 (Unredlichkeit und Verfahren bei Unredlichkeit) der ZHAW Prüfungsordnung sowie die Bestimmungen der Disziplinarmassnahmen der Hochschulordnung in Kraft.

**Ort, Datum:** **Name Studierende:**

**Zusammenfassung**

Die Nachfrage nach einem einfachen und bequemen Zugang zu relevanten Informationen im World Wide Web wächst exponentiell. Aus diesem Grund müssen neue Methoden für den effizienten Zugriff auf solchen Informationen erforscht werden. Herkömmliche Information Retrieval-Systeme stützen sich auf themenbezogene Ähnlichkeiten zwischen Wörtern und der vom Benutzer eingegebenen Anfrage. Jedoch bleiben bei diesem Ansatz viele semantische Komponenten unberücksichtigt. In diesem Projekt haben wir die Experimente der Masterarbeit von Anastasia Taranova für die deutsche Sprache nachgebaut. Das Ziel ist es herauszufinden, ob eine Unterscheidung der Dokumente hinsichtlich ihrer textuellen Komplexität die Suche nach relevanten Dokumenten für eine gegebene Anfrage verbessern würde. Wir haben festgestellt, dass aufgrund der schlechten Performance des Classifiers bei Dokumenten, die nicht Teil des Trainingsdatensatzes waren, das Experiment keine brauchbaren Ergebnisse lieferte. Daher kann keine Aussage über den Einfluss der Komplexität auf das Information Retrieval-System gemacht werden.

**Abstract**

The demand for accessing relevant information on the World Wide Web in an easy and comfortable way is growing at an exponential rate. Therefore, there is a necessity in the exploration of new ways to efficiently access such information. Conventional information retrieval systems rely on topical similarities between words and the query given by the user. However, there are many semantical components left out in this process. In this project we rebuilt the work by Anastasia Taranova for the German language. The idea is to conclude whether differentiating documents in terms of textual complexity would enhance the retrieval of relevant documents for a given query. In our experiments we tried similar approaches for the German language. We have found that as a consequence of the poor performance of the classifier on documents which were not part of the training dataset, the experiment did not produce any viable results. Therefore, no conclusion could be drawn.

**Vorwort**

Während den ersten zwei Jahren des Informatikstudiums an der ZHAW wurden diverse Themengebiete abgedeckt. Es wurden grundlegende Konzepte und Methoden vermittelt und in Projektarbeiten angewandt. Für das letzte Studienjahr hatten wir die Möglichkeit, verschiedene Wahlmodule nach eigenem Ermessen zu belegen.

Eines der Module, das uns angesprochen hat, war Information Engineering. Dieses Modul beschäftigt sich unter anderem mit der Thematik Information Retrieval, was zu den zentralen Aufgaben einer Suchmaschine gehört.

Aufgrund des gemeinsamen Interesses für dieses Themengebiet, entschieden wir uns, diese Projektarbeit in Zusammenarbeit mit Herr Prof. Dr. Martin Braschler durchzuführen und dabei unser Wissen im Bereich Information Engineering zu vertiefen.

Herzlich gedankt sei an dieser Stelle unserem Betreuer Prof. Dr. Martin Braschler, der uns diese Arbeit ermöglicht hat. Während der gesamten Arbeit hat er uns unterstützt und stellte sich bei Fragen und Unklarheiten zur Verfügung. Unser Dank gilt auch Frau Anastasia Taranova für das Zusenden der relevanten Dateien für die Projektarbeit, sowie für Ihre Masterarbeit, auf der unsere Projektarbeit basiert. Schliesslich möchten wir uns bei Remo Maurer für den Zugriff auf den BTS-Server, wo wir unsere Experimente durchführen konnten, bedanken.

Inhaltsverzeichnis

[1 Einleitung 1](#_Toc91180968)

[1.1 Ausgangslage 1](#_Toc91180969)

[1.2 Zielsetzung und Aufgabenstellung 1](#_Toc91180970)

[2 Theoretische Grundlagen 3](#_Toc91180971)

[2.1 Information Retrieval (IR)-System 3](#_Toc91180972)

[2.1.1 Indexierung 3](#_Toc91180973)

[2.1.2 Index 4](#_Toc91180974)

[2.1.3 Vergleich 4](#_Toc91180975)

[2.2 Evalutationsmassen 4](#_Toc91180976)

[2.2.1 MAP 4](#_Toc91180977)

[2.2.2 nDCG 4](#_Toc91180978)

[2.3 Klassifizierung 5](#_Toc91180979)

[2.3.1 Extrahierung von Merkmalen aus einem Dokument 5](#_Toc91180980)

[2.3.2 Probabilistische Klassifizierung eines Dokuments 5](#_Toc91180981)

[3 Aufbau des Experiments 6](#_Toc91180982)

[3.1 Hardware 6](#_Toc91180983)

[3.2 Software 6](#_Toc91180984)

[3.2.1 Anserini 6](#_Toc91180985)

[3.2.2 Pyserini 6](#_Toc91180986)

[3.2.3 TREC-Evalutionstool 6](#_Toc91180987)

[3.3 Daten 7](#_Toc91180988)

[3.3.1 CLEF 7](#_Toc91180989)

[3.3.2 Trainingsdaten: Wikipedia und Klexikon 7](#_Toc91180990)

[3.4 Complexity-Classifier 8](#_Toc91180991)

[3.4.1 Text-Vorverarbeitung 8](#_Toc91180992)

[3.4.2 Doc2Vec-Modell 8](#_Toc91180993)

[3.4.3 Neural Network-Modell 8](#_Toc91180994)

[3.5 IR-System 9](#_Toc91180995)

[3.5.1 Aufbereitung der Dokumente von CLEF für die Indexierung 9](#_Toc91180996)

[3.5.2 Indexierung der Dokumente und Anfragen mit Anserini 9](#_Toc91180997)

[3.5.3 Rangierung der abgefragten Dokumente mittels Pyserini 10](#_Toc91180998)

[3.5.4 Evaluation mit dem TREC-Evaluationstool 11](#_Toc91180999)

[4 Resultate des Experiments 12](#_Toc91181000)

[4.1 Complexity-Classifier Performance 12](#_Toc91181001)

[4.1.1 Learning Curve 13](#_Toc91181002)

[4.1.2 Analyse der Komplexitätswahrscheinlichkeiten 13](#_Toc91181003)

[4.2 Evaluation des IR-Systems 15](#_Toc91181004)

[5 Diskussion und Ausblick 16](#_Toc91181005)

[6 Verzeichnisse 17](#_Toc91181006)

[6.1 Abbildungsverzeichnis 17](#_Toc91181007)

[6.2 Tabellenverzeichnis 17](#_Toc91181008)

[6.3 Literaturverzeichnis 17](#_Toc91181009)

# Einleitung

Das World Wide Web ist über die Jahre stark gewachsen und besteht aus einer grossen Menge an strukturierten und unstrukturierten Daten. Dies ist mit ein Grund, weshalb das Auffinden von relevanten Informationen oft keine leichte Aufgabe ist. Hilfe bieten hier Information Retrieval (IR)-Systeme, die den Nutzer bei der Suche nach Dokumenten und Webseiten entsprechend ihrem Informationsbedürfnis, das in Form einer Anfrage gestellt wird, unterstützt. Das Retrieval-Resultat für eine Anfrage wird als Rangliste von Dokumenten der Relevanz nach sortiert zurückgegeben, wobei sich die relevantesten Dokumente zuoberst befinden. [1]

Die Suche nach Informationen im Internet lässt sich von Personen jeden Alters durchführen. Eine der Herausforderungen bei der Befriedigung des Informationsbedarfs eines Nutzers besteht darin, dem Nutzer ein positives, komfortables Erlebnis bei der Interaktion mit den Informationen zu bieten. Die textuelle Komplexität eines Dokuments ist ein kritischer Aspekt, der die Nutzererfahrung je nach dem positiv oder negativ beeinflusst. In Anbetracht der Relevanz eines Dokuments schätzt eine Person die Komplexität des Dokuments ein und entscheidet basierend auf dieser Einschätzung, ob das Dokument untersucht werden soll oder nicht. [1]

## Ausgangslage

Diese Projektarbeit basiert auf der Masterarbeit „Beyond Topical Similarity: How Query Characteristics Inform Ranking“ von Anastasia Taranova, bei der ein mehrstufiges IR-System aufgebaut wurde. Das Ziel war es, die textuelle Komplexität und Autorität von Dokumenten, die Robustheit eines Retrieval-Systems, und die Diversität von gefundenen Dokumenten, zu untersuchen und zu verstehen, wie diese Aspekte die konventionelle Relevanzbewertung von Dokumenten einer Sammlung beeinflussen. Die Experimente wurden mit englischen Dokumenten aus der ClueWeb09-Kollektion durchgeführt. [1]

## Zielsetzung und Aufgabenstellung

Das Ziel dieser Projektarbeit ist es, das gleiche Experiment mit deutschen Dokumenten aus der CLEF-Kollektion durchzuführen, wobei hier nur die Auswirkung der textuellen Komplexität auf die Performance des IR-Systems untersucht wird.

Die Aufgabenstellung kann in folgende Gebiete unterteilt werden:

1. Datenbeschaffung:
   * Recherche nach komplexen und einfachen deutschen Dokumenten mit denen der Complexity-Classifier trainiert werden kann.
2. Entwicklung eines Complexity-Classifiers:
   * Der trainierte Complexity-Classifier soll in der Lage sein, die deutschen Dokumente aus der CLEF-Kollektion nach der Komplexität zuverlässig zu klassifizieren.
3. Aufbau des IR-Systems:
   * Es soll ein IR-System realisiert werden, das für jede Anfrage aus einer Anfrage-Kollektion eine Rangliste zurückgibt, die nur einfache, schwierige oder alle Dokumente enthalten.
4. Evaluation:
   * Es soll die Performance des Classifiers, sowie des IR-Systems evaluiert werden.

# Theoretische Grundlagen

## Information Retrieval (IR)-System

Information Retrieval beschreibt das Suchen und Finden von relevanten Dokumenten aus einer Dokumentenkollektion (Korpus) basierend auf einer benutzerdefinierten Anfrage, welche in Textform verfasst wird. Das System verarbeitet darauf diese Anfrage und liefert die relevanten Dokumente aus der bestehenden Sammlung von Dokumenten. Diese Dokumente werden dann anhand der Rangierung in absteigender Reihenfolge sortiert dem Benutzer vorgeführt. Der Rang und die Relevanz der zurückgegebenen Dokumente bestimmen, wie adäquat das Ergebnis ist. [2] Nachfolgend ist ein vereinfachtes IR-System abgebildet.

### Indexierung

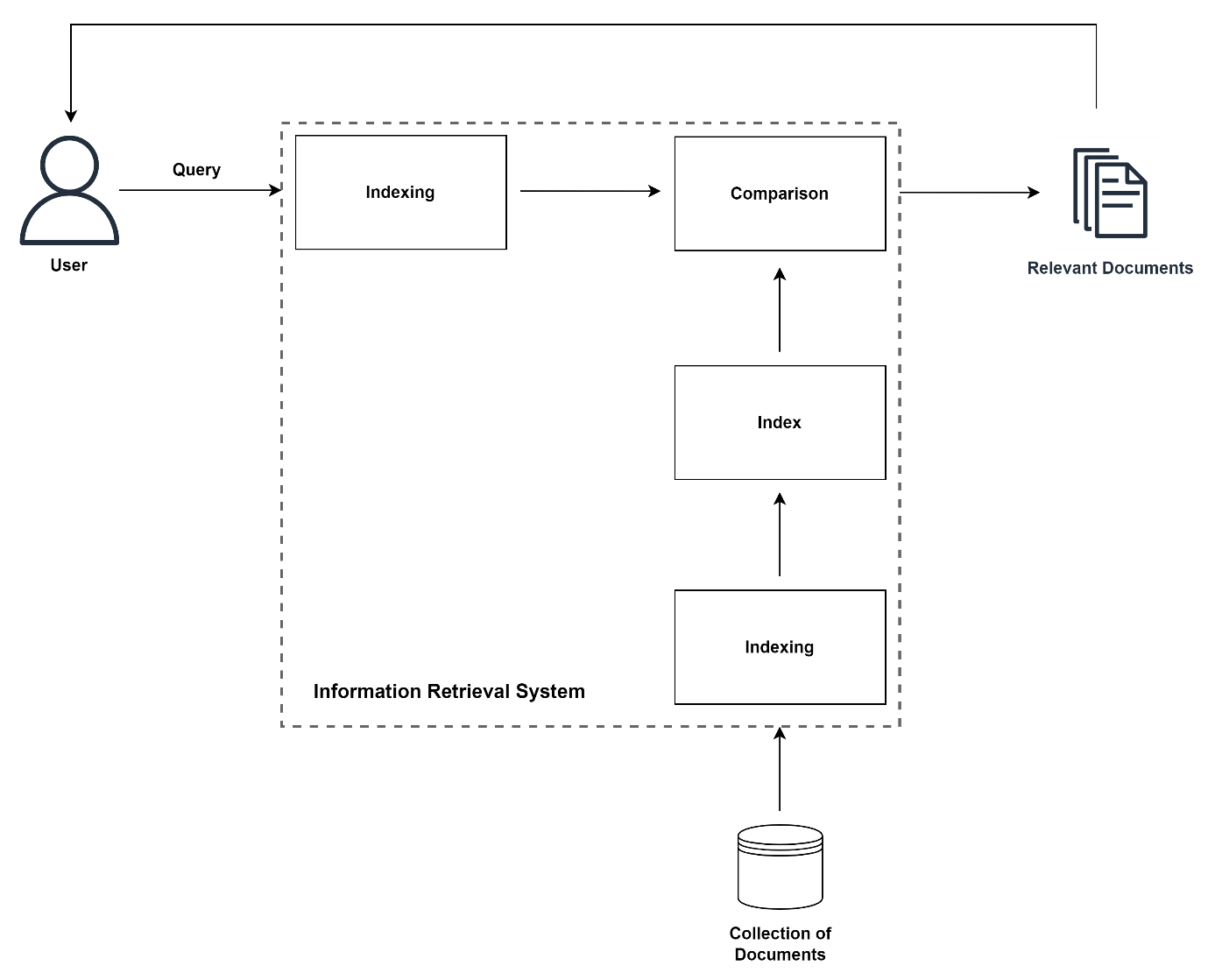


Abbildung 1: Ein vereinfachtes IR-System

Bei der Indexierung (englisch Indexing) werden auf der Anfrage und dem Korpus Tokenisierung, Stoppwortelimination und Stemming angewandt, um diese in einem nächsten Schritt vergleichen zu können. Mit Tokenisierung ist die Extrahierung der einzelnen Wörter aus einem Text gemeint. Die nicht inhaltstragenden Wörter werden mittels Stoppwortelimination entfernt. Schliesslich wird mit dem Stemming-Verfahren, die verschiedenen Varianten eines Wortes auf einen gemeinsamen Stamm zurückgeführt, indem zum Beispiel die Suffixe oder Präfixe entfernt werden. Die beiden Prozesse laufen jedoch nicht parallel ab, wie in Abbildung 1 vereinfacht dargestellt wurde. Zum Beispiel kann sowohl die Indexierung als auch das Erstellen eines invertierten Indexes offline erfolgen und die Bearbeitung einer Anfrage online durchgeführt werden. [3]

### Index

Der invertierte Index wird erzeugt, um die begriffsbasierte Suche effizienter zu gestalten. Es handelt sich hierbei um eine Liste bestehend aus allen in den Dokumenten vorkommenden Begriffen, die nach dem Indexierungsprozess auf das Wesentliche reduziert werden. Für jedes dieser Wörter wird die Dokumentenhäufigkeit, sowie die Position, an der das Wort in einem Dokument vorkommt, festgehalten. [4]

### Vergleich

Die Relevanzbewertung von Dokumenten für eine gegebene Suchanfrage wird mit der BM25 Gewichtungsformel berechnet. BM ist eine Abkürzung für Best Matching. Die Retrieval-Funktion bewertet eine Reihe von Dokumenten auf der Grundlage der Suchbegriffe, die in jedem Dokument vorkommen, unabhängig von deren Nähe zueinander innerhalb des Dokuments und gibt diese in Form einer Rangliste zurück. [5]

## Evalutationsmassen

### MAP

Die mittlere durchschnittliche Präzision über alle Anfragen, auf Englisch: Mean Average Precision oder MAP, ist ein Standardmass für den Vergleich verschiedener Ranking-Methoden. Sie bietet eine ausgeglichene Bewertung der Präzision und Ausbeute von den abgefragten relevanten Dokumenten und ist dabei empfindlich gegenüber dem Rang jedes relevanten Dokumentes. [1]

### nDCG

Der Discounted Cumulative Gain (DCG) ist eine Kennzahl, mit der sich die Qualität der Rankings von Suchergebnissen messen lässt. Der DCG bestimmt den Nutzen eines Dokuments aufgrund seiner Position in der Rangliste. Je nach Position nimmt der Nutzen vom Top-Ergebnis bis hin zum letzten Treffer ab. Dies liefert, wie bei MAP, eine ausgeglichene Bewertung der Präzision und Ausbeute für eine einzelne Anfrage. [6] Um dieses Mass über verschiedene Abfragen hinweg zu vergleichen, wird der Normalized Discounted Cumulative Gain (nDCG) verwendet. [7]

## Klassifizierung

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungDie Komplexitätsschätzung eines Dokuments kann als ein Klassifizierungsproblem formuliert werden, bei dem jedes Dokument in die Klassen „einfach“ oder „komplex“ eingeteilt werden kann. Der Klassifizierungsprozess umfasst zwei Schritte wie in Abbildung 2 ersichtlich: Erstens die Extrahierung von Merkmalen aus einem Dokument und zweitens die Verwendung der extrahierten Merkmale als Input für einen probabilistischen Klassifizierungs-algorithmus. In den folgenden Unterkapiteln werden auf die einzelnen Schritte genauer eingegangen. [1]

Abbildung 2: Klassifizierungsprozess

### Extrahierung von Merkmalen aus einem Dokument

Unüberwachte Methoden des maschinellen Lernens (englisch unsupervised machine learning methods) ermöglichen Kontextinformationen eines Satzes, Absatzes oder Dokuments als numerischen Vektor fester Grösse darzustellen. Im Englischen wird der Prozess des Ersetzens eines Dokuments durch den entsprechenden Vektor als document embedding bezeichnet. Umgesetzt wird die Merkmalextraktion mittels einem doc2vec-Algorithmus, welcher erheblich zur Verbesserung der Performance eines Machine Learning Algorithmus beiträgt. [1]

### Probabilistische Klassifizierung eines Dokuments

Die Klassifizierung eines Dokuments kann als ein binäres probabilistisches Klassifizierungs-problem betrachtet werden. Hier versucht der Classifier für einen gegebenen Feature-Vektor die Wahrscheinlichkeiten p, dass ein Dokument zur Klasse „komplex“ gehört und 1 – p, dass es zur Klasse „einfach“ gehört, zu ermitteln. Dabei entspricht der Wahrscheinlichkeitswert der Klasse „komplex“, der im Bereich zwischen Null und Eins liegt, die Komplexität eines Dokuments. Bei einer Komplexität, die gegen Null tendiert, handelt es sich um ein einfaches Dokument, während eine Komplexität gegen Eins auf ein komplexes Dokument hindeutet (siehe Abbildung 3). Der implementierte Complexity-Classifer basiert auf Neural Network. [1]

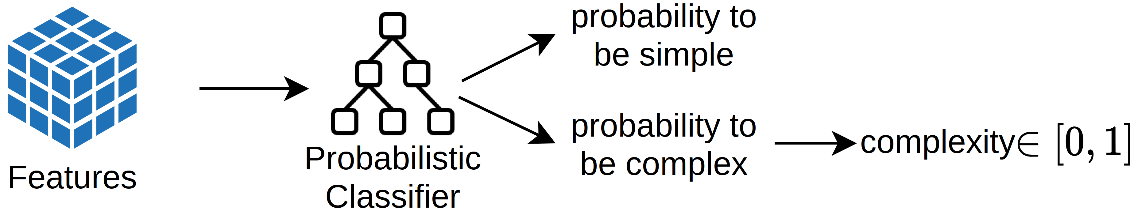


Abbildung 3: Probabilistische Klassifizierung [1]

# Aufbau des Experiments

## Hardware

Alle Experimente wurden auf einem Server mit Intel Core Prozessor (Broadwell-Mikroarchitektur) mit 2.5 GHz, 16 Kernen und 128 GB Hauptspeicher (RAM) durchgeführt. Auf dem Server läuft der Linux-Kernel 4.15. [1]

## Software

### Anserini

Für die Indexierung der Dokumente aus der CLEF-Kollektion wird das Java-basierte Open-Source-IR-Toolkit Anserini eingesetzt, das auf Lucene aufbaut und darauf abzielt, dass Forschung und Praxis im Bereich Information Retrieval besser aufeinander abgestimmt werden. In der Industrie hat sich Lucene zur de facto Plattform für den Einsatz von Suchanwendungen entwickelt. [1] [8]

### Pyserini

Mit dem Python-Toolkit Pyserini wird das Retrieval-Resultat für eine Anfrage, das in Form einer Rangliste von Dokumenten wiedergegeben wird, erstellt. Pyserini wurde in erster Linie entwickelt, um effektive, reproduzierbare und einfach zu handhabende Abfragen in einer mehrstufigen Ranking-Architektur zu ermöglichen. Des Weiteren ist in diesem Toolkit Anserini integriert, das wiederum auf Lucene aufbaut. Beide Module arbeiten eng miteinander und haben den Prozess der Indexierung und des Retrievals in diesem Projekt sehr vereinfacht. [9]

### TREC-Evalutionstool

Das IR-System wird mit dem offiziellen trec\_eval Tool evaluiert. Benötigt werden hierzu die erstellte Rangliste, sowie das qrels-File. Das letztere wird im Kapitel 3.3.1 genauer beschrieben. [1]

## Daten

### CLEF

Um ein Retrieval-Experiment durchzuführen, wird ein grosser Datensatz benötigt, der folgendes enthalten muss: Eine Sammlung von Dokumenten, eine Liste echter Benutzeranfragen, sowie eine echte Relevanzbewertung für Dokumente aus der Kollektion, die vom IR-System abgefragt werden. [1]

Hierzu eignet sich der kostenpflichtige CLEF-Datensatz, welcher auf Deutsch und Englisch erhältlich ist. Erstellt wurde der Datensatz von der CLEF-Initiative (Conference and Labs of the Evaluation Forum), deren Hauptaufgabe darin besteht, Forschung, Innovation und Entwicklung von Systemen für Informationszugang zu fördern. [10]

Der deutsche CLEF-Datensatz besteht aus 124 Anfragen (Topics 101 bis 125), 151‘319 Dokumenten – davon effektiv 145‘941 genutzt werden, da einige Dokumente unvollständig sind – und einem qrels-File, das für die Evaluierung des IR-Systems verwendet wird. Diese Datei enthält eine Liste von Dokumenten, die für jede Abfrage als relevant erachtet werden. Die Relevanzbeurteilung wurde von den an der CLEF-Initiative teilnehmenden Personen vorgenommen. [11]

### Trainingsdaten: Wikipedia und Klexikon

Damit der Classifier die Komplexität eines Dokuments schätzen kann, muss er vorerst mit einem geeigneten Datensatz trainiert werden. Der zuvor beschriebene CLEF-Datensatz erweist sich als ungünstig, da die Dokumente nicht mit einer Komplexitätsbezeichung versehen sind.

Für eine zuverlässige Klassifizierung nach der Komplexität müssen zwei Datensätze erstellt werden: Ein Datensatz mit komplexen Dokumenten, sowie ein weiterer Datensatz, der die entsprechenden einfachen Versionen der komplexen Dokumente beinhaltet.

Der komplexe Datensatz besteht aus Artikeln der deutschen Wikipedia-Enzyklopädie, die nur in „normalem“ Deutsch verfügbar sind, anders als bei der englischen Wikipedia, bei der für jeden Artikel sowohl eine Version in normalem als auch eine in einfachem Englisch existiert.

Aus diesem Grund wird der einfache Datensatz mit den Artikeln des Klexikons erstellt. Das Klexikon ist ein Online-Lexikon für Kinder, dass ähnlich aufgebaut ist wie Wikipedia und zurzeit über 3000 Artikel aufweist. Im Gegensatz zur Wikipedia sind alle Artikel speziell für Kinder geschrieben. [12]

Bemerkung: Für das Trainieren werden nur die Artikel berücksichtigt, die in beiden Datensätzen vorkommen.

## Complexity-Classifier

In diesem Kapitel werden die einzelnen Schritte des Klassifizierungsprozesses vorgestellt. Der Klassifizierungsalgorithmus wird analog zum Vorgehen in der Masterarbeit entwickelt. Für die trainierten Doc2Vec- und Neural Network-Modelle, werden dieselben Parameter wie bei der Masterarbeit verwendet, da dies zu guten Resultaten geführt hat. Für mehr Informationen siehe die Arbeit von Frau Anastasia Taranova.

### Text-Vorverarbeitung

In einem ersten Schritt werden alle Artikel selektiert, die sowohl im Wikipedia als auch im Klexikon-Datensatz vorkommen. Dies wird mittels Titel-Matching realisiert. Als nächstes folgt die Vorverarbeitung von Texten mit einer Python-Bibliothek namens Gensim, mit der die Klartexte aus den Artikeln extrahiert werden. Auf den Text der einzelnen Artikel wird Tokenisierung angewandt, um den Text in Wörter (Tokens) zu zerlegen und die Interpunktionszeichen zu entfernen. In einem nächsten Schritt werden die Stoppwörter eliminiert – gemeint sind hier häufig vorkommende Wörter in Texten, die nicht inhaltstragend sind und für den IR-Prozess von geringem Wert sind. Mittels Token-Lemmatisierung werden die Wörter in die Grundform umgewandelt. Danach wird der vorverarbeitete Text, der nur noch aus Tokens besteht, mit einem entsprechenden Komplexitätswert versehen. Dieser Vorgang wird für jeden Artikel wiederholt. Die daraus resultierenden vorverarbeiteten Texte werden zu einem neuen Datensatz zusammengeführt. [1]

### Doc2Vec-Modell

Basierend auf dem doc2vec-Algorithmus wird ein Doc2Vec-Modell trainiert. Als Input dient der Datensatz, der durch den Text-Vorverarbeitungsprozess erstellt wurde. Hierbei werden für jedes Dokument im Datensatz wie im Kapitel 2.3.1 beschrieben, die Merkmale extrahiert. Das trainierte Doc2Vec-Modell wird zudem für die Merkmalextraktion eines CLEF-Dokuments genutzt, da der Complexity-Classifier nur anhand der Features eines Dokuments, die Komplexität vorhersagen kann.

### Neural Network-Modell

Mit dem Doc2Vec-Modell wird der Complexity-Classifier trainiert. Als Classifier wird ein Neural Network-Modell gewählt, da es in der Regel sehr gut für Klassifizierungsprobleme geeignet ist. Mit einem gut trainierten Modell lässt sich die Komplexität eines Dokuments ermitteln. [1]

## IR-System

Für diese Arbeit ist eine parametrisierte, wiederholbare Pipeline notwendig, um die Integrität der Evaluationen der verschiedenen Experimente zu bewahren.

In der Abbildung 4 ist die vollständige Pipeline der Experimente ersichtlich. In den nächsten Abschnitten werden die folgenden Schritte der Pipeline beschrieben:

1. Aufbereitung der Dokumente von CLEF für die Indexierung
2. Indexierung der Dokumente und Anfragen mit Anserini
3. Rangierung der abgefragten Dokumente mittels Pyserini
4. Ein Bild, das Text, Parkplatz enthält.

   Automatisch generierte BeschreibungEvaluation der Ranglisten mit dem TREC-Evaluationstool

Abbildung 4: Pipeline des aufgebauten IR-Systems

### Aufbereitung der Dokumente von CLEF für die Indexierung

Für das Information Retrieval werden hauptsächlich die Datensätze der CLEF Initiative verwendet. Da ein minimaler Anteil der Daten unvollständig und in verschiedene Einheiten aufgeteilt sind, werden sie in einem ersten Schritt zusammengeführt und vereinheitlicht. Als Nächstes werden die einzelnen CLEF-Dokumente nach der Komplexität klassifiziert und mit einer Bezeichnung versehen.

### Indexierung der Dokumente und Anfragen mit Anserini

Die aufbereiteten Dokumente sowie die Anfragen werden mit Anserini indexiert. Zusätzlich wird mit Anserini ein invertierter Index erzeugt, um die begriffsbasierte Suche effizienter zu gestalten. [13] Beim Indexierungsprozess wird der GermanAnalyzer des Lucene-Frameworks eingesetzt, da die Experimente auf deutschen Datensätzen basieren. [14]

### Rangierung der abgefragten Dokumente mittels Pyserini

Nach der Indexierung folgt die Rangierung der abgefragten Dokumente. Hierzu erstellt Pyserini mit einer BM25 Gewichtungsformel für jede Anfrage aus der Anfragen-Kollektion eine Rangliste. [15] Insgesamt werden drei Rankings erstellt, diese sind: Ranking with Pyserini (No Filter), Ranking with Pyserini (Simple Filter) und Ranking with Pyserini (Complex Filter).

Die Ranglisten, die mit der No Filter-Variante generiert werden, beinhalten sowohl einfache als auch komplexe Dokumente und haben das folgende Format (siehe Abbildung 5).

Die Spalte Complexity gibt die Komplexität eines Dokuments an: die Einsen repräsentieren komplexe Dokumente, die Nullen hingegen die Einfachen. Die weiteren erstellten Rankings weisen einen ähnlichen Aufbau auf: das Simple Filter-Ranking beinhaltet Ranglisten, die nur einfache Dokumente enthalten, mit dem Complex Filter werden Ranglisten erzeugt, die nur komplexe Dokumente aufführen.Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Abbildung 5: Aufbau der Ranglisten für jede Anfrage

Leider ist es nicht im Voraus klar, welche Anfrage welche Komplexität erfordert, um die beste Rangliste von relevanten Dokumenten zu erhalten. Das Aufteilen in zwei Kategorien von Ranglisten, Simple und Complex, reicht nicht aus, um eine Aussage darüber zu machen, ob diese Methodik eine positive Auswirkung auf die Präzision und Ausbeute von relevanten Dokumenten hat. Um dieser Problematik zu begegnen, werden die evaluierten durchschnittlichen Präzisionswerte (Average Precision) einzelner Anfragen des Simple und Complex Rankings verglichen, wie in Abbildung 6 zu erkennen ist.

Ein Bild, das Tisch enthält.

Automatisch generierte BeschreibungFür die neue Liste Perfect Baseline wird für jede Anfrage die höhere Average Precision genommen und darauffolgend der Durchschnitt über all diese Werte berechnet, damit man die MAP erhält. Bei genauer Betrachtung der Abbildung ist festzustellen, dass die MAP der Perfect Baseline aufgrund der gewählten Average Precision-Werte, stets grösser ist als die der beiden Ranking-Evaluationen. Durch diese Methodik, wüsste somit ein Nutzer, bei welcher Anfrage welche Unterteilung der Ranglisten, in Simple oder Complex, die beste MAP bietet. Dies wäre das Best-Case-Szenario des Experiments.

Abbildung 6: Perfect Baseline

### Evaluation mit dem TREC-Evaluationstool

Anschliessend werden die erstellten Ranglisten, zusammen mit den qrels-Daten ins trec\_eval Tool eingefügt. Dieses Tool liest die beiden Dateien ein und evaluiert die erhaltenen Ranglisten, bezüglich der Relevanz der erhaltenen Dokumente, abgeschnitten haben. Die vier Ranking-Ansätze werden anhand den Bewertungsmetriken: MAP und nDCG untereinander verglichen, da diese Grössen die Präzision und die Ausbeute balanciert beurteilen.

# Resultate des Experiments

## Complexity-Classifier Performance

Das Trainingsset beinhaltet um die 3000 Dokumentenpaare aus der einfachen und komplexen Kollektion, was wesentlich tiefer ist als die 100‘000 Dokumentenpaare, die bei der Masterarbeit verwendet wurden. [1]

Zu Beginn waren es zirka 2900 Dokumentenpaare, da die Titel einiger Artikel aus der einfachen Kollektion leicht von denen aus der komplexen Kollektion abwichen und infolgedessen vom Vorverarbeitungsalgorithmus nicht berücksichtigt wurden, wie beispielsweise der Artikel im Klexikon-Datensatz „Aargau“ und das entsprechende Gegenstück „Kanton Aargau“. Der auf diesen Dokumenten trainierte Complexity-Classifier lieferte eine Genauigkeit von 0.4917 und einen F1-Score von 0.6170.

Aufgrund dieses Resultats wurden die Trainingsdaten genauer analysiert. Im Wiki-Datensatz konnte festgestellt werden, dass es sich bei gewissen Dokumenten nicht um tatsächliche Artikel handelte, sondern um Begriffserklärungsseiten (siehe Abbildung 7). Diese mussten durch richtige Artikel ersetzt werden.

Aufbauend auf diesen Erkenntnissen wurden die Trainingsdaten bereinigt und die Anzahl Dokumentenpaare auf 3163 erweitert. Dies führte zu einer erheblichen Verbesserung des Complexity-Classifiers, welcher neu eine Genauigkeit von 0.9052 und einen F1-Score von 0.9046 erreichte.

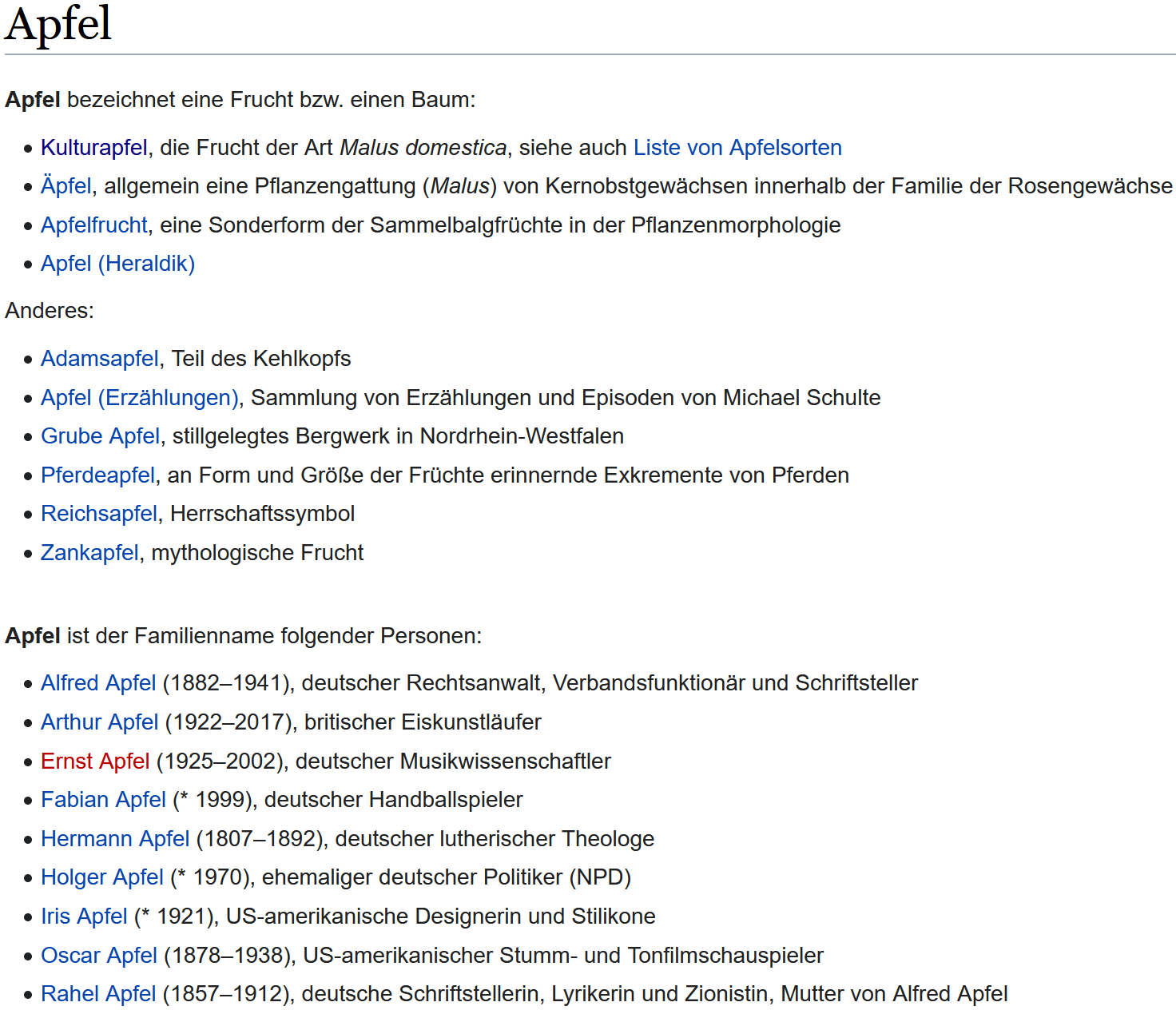


Abbildung 7: Begriffserklärungsseite für den Begriff Apfel: https://de.wikipedia.org/wiki/Apfel

### Learning Curve

Um die Performance eines Neural Network-Modells zu analysieren, werden sogenannte Learning Curves erstellt, die zur Diagnose von Problemen beim Lernprozess dienen. Zum Beispiel kann aus den Lernkurven abgeleitet werden, ob es sich beim trainierten Modell um ein „underfit“, „overfit“ oder „well-fit“ handelt. Ausserdem lässt sich auch ermitteln, wie sehr ein Modell von zusätzlichen Trainingsdaten profitiert. Es gibt zwei Arten von Learning Curves: eine Trainings-Lernkurve, die anhand Trainingsdaten berechnet wird und vermittelt wie gut das Modell lernt und eine Validierungs-Lernkurve, die mittels Validierungsdatensatz generiert wird und beschreibt, wie gut das Modell generalisiert. [1]

In Abbildung 8 sind die beiden Lernkurven ersichtlich. Beide weisen eine ausreichend hohe Genauigkeit auf, wobei die Scores mit mehr Trainingsdaten erhöht werden könnten. Des Weiteren besteht eine Lücke zwischen den beiden Lernkurven, die darauf hindeutet, dass das Modell nicht gut generalisiert. [1]

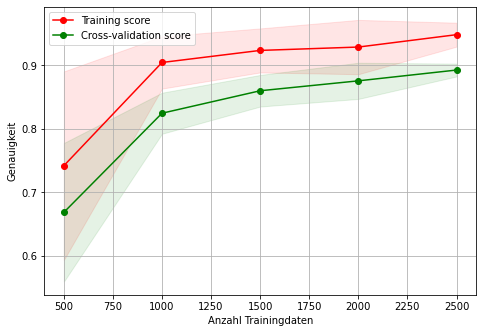


Abbildung 8: Learning Curve für das Neural Network-Modell

### Analyse der Komplexitätswahrscheinlichkeiten

Der Histogramm-Plot in Abbildung 9 veranschaulicht den Unterschied der Wahrscheinlich-keiten zwischen Dokumentenpaaren, die vom Complexity-Classifier vorhergesagt wurden. Die Differenz der Wahrscheinlichkeiten liegt bei vielen Dokumentpaaren nahe bei Eins. Das bedeutet, dass bei diesen Paaren sowohl die einfache als auch die komplexe Version korrekt klassifiziert wurden. Zu bemerken ist, dass die Differenz nur maximal Eins werden kann, da ein echtes komplexes Dokument im Idealfall einen Wahrscheinlichkeitswert von Eins bekommt und das dazugehörende echte einfache Dokument einen Wert Null. Es sind zudem auch einige Dokumentenpaare abgebildet, die eine negative Differenz aufweisen. Hier sind die einfachen Artikel als komplex und die Komplexen als einfach eingestuft worden.

Die zwei weiteren Plots in Abbildung 10 zeigen die Klassifizierung von echten einfachen und echten komplexen Dokumenten. Auch hier werden die Dokumente mehrheitlich vom Classifier richtig identifiziert: Die Komplexitätswahrscheinlichkeiten der meisten einfachen Dokumente streben gegen Null, die der komplexen Dokumente gegen Eins.

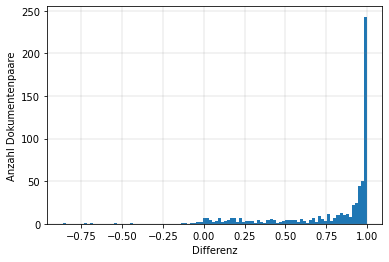


Abbildung 9: Differenz der Wahrscheinlichkeiten von Dokumentenpaaren

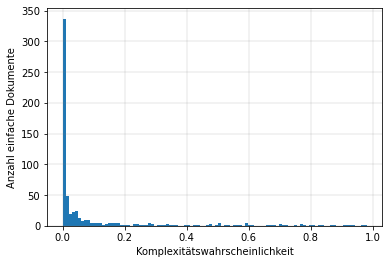
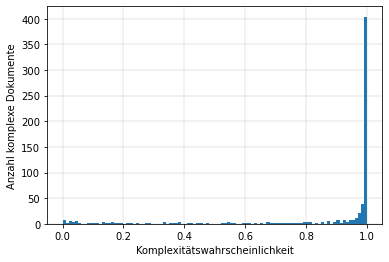


Abbildung 10: Komplexitätswahrscheinlichkeit für einfache und komplexe Dokumente

## Evaluation des IR-Systems

Um die Performance des aufgebauten IR-Systems zu untersuchen, werden die im Kapitel 3.5.3 beschriebenen vier Rankingansätze, mit dem trec\_eval Tool evaluiert. In der nachfolgenden Tabelle sind die Ergebnisse der Messungen aufgeführt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ALL | SIMPLE | COMPLEX | PERFECT BASELINE |
| MAP | 0.2449 | 0.2043 | 0.0465 | 0.2053 |
| nDCG | 0.4578 | 0.4016 | 0.1346 | 0.4026 |

Tabelle 1: Ergebnisse der vier Rankingansätze

Aus der Tabelle lässt sich entnehmen, dass die Ranglisten mit einfachen Dokumenten viel besser abschneiden als die mit Komplexen. Die beiden Messwerte MAP und nDCG vom SIMPLE-Ansatz liegen sehr nahe bei denen des ALL-Ansatzes. Dies kann damit begründet werden, dass der Complexity-Classifier nicht in der Lage ist die Dokumente zuverlässig zu klassifizieren. Denn die Mehrheit der Dokumente wurde als einfach eingestuft. Die schlechte Performance des Neural Network-Modells liegt vermutlich an der unzureichenden Menge an Trainingsdaten. Beim englischen Experiment wurden über 100‘000 Dokumente verwendet, bei dieser Arbeit hingegen waren es nur um die 3000. Ein weiterer Faktor für dieses Ergebnis könnten eventuell die Klexikon- und Wikipedia-Datensätze sein, die für die Klassifizierung nach Komplexität nicht geeignet sind. Aufgrund dieser Tatsachen konnten keine erheblichen Steigerungen bei der Perfect Baseline erreicht und folglich keine Schlussfolgerung für den Nutzen des aufgebauten Experiments gezogen werden.

# Diskussion und Ausblick

Um eine Aussage über die Auswirkung der textuellen Komplexität auf ein IR-System zu machen, muss zuerst die Performance des Classifiers auf andere Datensätze (z.B. CLEF) untersucht werden. Für die Verbesserung des Klassifizierungsproblems gibt es zwei Möglichkeiten. Die eine Möglichkeit ist den bestehenden Trainingsdatensatz zu erweitern oder nach neuen geeigneten Trainingsdaten zu suchen. Die zweite Möglichkeit wäre zum Beispiel die Klassifizierung mittels Support Vector Machine, da dies – verglichen mit dem Neural Network-Modell – für eine kleine Menge an Trainingsdaten gute Resultate liefert. Wenn das Modell einmal zuverlässig klassifiziert, wäre es sinnvoll, eine untere und eine obere Schwelle einzubauen, sodass die Komplexität der Dokumente, die diese Schwellen unter- oder überschreiten sehr wahrscheinlich als korrekt eingestuft werden. Dadurch könnten die Messwerte der Perfect Baseline verbessert werden.

# Verzeichnisse

## Abbildungsverzeichnis

[Abbildung 1: Ein vereinfachtes IR-System 3](file:///D:\ZHAW\01_Informatik\3_Jahr\5_SEM\PA\Projektarbeit-HS21\01_Documentation\Bericht\PA_Bericht_bram01_final.docx#_Toc91132301)

[Abbildung 2: Klassifizierungsprozess 5](file:///D:\ZHAW\01_Informatik\3_Jahr\5_SEM\PA\Projektarbeit-HS21\01_Documentation\Bericht\PA_Bericht_bram01_final.docx#_Toc91132302)

[Abbildung 3: Probabilistische Klassifizierung [1] 5](file:///D:\ZHAW\01_Informatik\3_Jahr\5_SEM\PA\Projektarbeit-HS21\01_Documentation\Bericht\PA_Bericht_bram01_final.docx#_Toc91132303)

[Abbildung 4: Pipeline des aufgebauten IR-Systems 9](file:///D:\ZHAW\01_Informatik\3_Jahr\5_SEM\PA\Projektarbeit-HS21\01_Documentation\Bericht\PA_Bericht_bram01_final.docx#_Toc91132304)

[Abbildung 5: Aufbau der Ranglisten für jede Anfrage 10](file:///D:\ZHAW\01_Informatik\3_Jahr\5_SEM\PA\Projektarbeit-HS21\01_Documentation\Bericht\PA_Bericht_bram01_final.docx#_Toc91132305)

[Abbildung 6: Perfect Baseline 11](file:///D:\ZHAW\01_Informatik\3_Jahr\5_SEM\PA\Projektarbeit-HS21\01_Documentation\Bericht\PA_Bericht_bram01_final.docx#_Toc91132306)

[Abbildung 7: Begriffserklärungsseite für den Begriff Apfel: https://de.wikipedia.org/wiki/Apfel 12](file:///D:\ZHAW\01_Informatik\3_Jahr\5_SEM\PA\Projektarbeit-HS21\01_Documentation\Bericht\PA_Bericht_bram01_final.docx#_Toc91132307)

[Abbildung 8: Learning Curve für das Neural Network-Modell 13](file:///D:\ZHAW\01_Informatik\3_Jahr\5_SEM\PA\Projektarbeit-HS21\01_Documentation\Bericht\PA_Bericht_bram01_final.docx#_Toc91132308)

[Abbildung 9: Differenz der Wahrscheinlichkeiten von Dokumentenpaaren 14](file:///D:\ZHAW\01_Informatik\3_Jahr\5_SEM\PA\Projektarbeit-HS21\01_Documentation\Bericht\PA_Bericht_bram01_final.docx#_Toc91132309)

[Abbildung 10: Komplexitätswahrscheinlichkeit für einfache und komplexe Dokumente 14](file:///D:\ZHAW\01_Informatik\3_Jahr\5_SEM\PA\Projektarbeit-HS21\01_Documentation\Bericht\PA_Bericht_bram01_final.docx#_Toc91132310)

## Tabellenverzeichnis

[Tabelle 1: Ergebnisse der vier Rankingansätze 15](#_Toc91132311)

## Literaturverzeichnis

[1] A. Taranova and M. Braschler, “Beyond Topical Similarity: How Query Characteristics Inform Ranking.”

[2] “Information Retrieval using word2vec based Vector Space Model.” https://www.analyticsvidhya.com/blog/2020/08/information-retrieval-using-word2vec-based-vector-space-model/ (accessed Dec. 22, 2021).

[3] M. Braschler, “Information Engineering Vorlesung: Indexierung/Vergleich.”

[4] “Der Invertierte Index, Speichertechnik, Wort-orientierte Mechanismen, erweitertes boolsches Modell.” http://wissensexploration.de/wissensexploration-web-ir-invertierterindex.php (accessed Dec. 22, 2021).

[5] “Okapi BM25 - Wikipedia.” https://en.wikipedia.org/wiki/Okapi\_BM25 (accessed Dec. 22, 2021).

[6] “Discounted Cumulative Gain (DCG) - SEO-Glossar - SEO Südwest.” https://www.seo-suedwest.de/seo-wissen/seo-glossar/discounted-cumulative-gain.html (accessed Dec. 22, 2021).

[7] “Evaluation measures (information retrieval) - Wikipedia.” https://en.wikipedia.org/wiki/Evaluation\_measures\_(information\_retrieval)#Discounted\_cumulative\_gain (accessed Dec. 22, 2021).

[8] “castorini/anserini: Anserini is a Lucene toolkit for reproducible information retrieval research.” https://github.com/castorini/anserini (accessed Dec. 22, 2021).

[9] “castorini/pyserini: Pyserini is a Python toolkit for reproducible information retrieval research with sparse and dense representations.” https://github.com/castorini/pyserini (accessed Dec. 22, 2021).

[10] “The CLEF Initiative (Conference and Labs of the Evaluation Forum) - Homepage.” http://www.clef-initiative.eu/ (accessed Dec. 22, 2021).

[11] “Learn how to use trec\_eval to evaluate your information retrieval system - Rafael Glater.” http://www.rafaelglater.com/en/post/learn-how-to-use-trec\_eval-to-evaluate-your-information-retrieval-system (accessed Dec. 22, 2021).

[12] “Klexikon – das Kinderlexikon.” https://klexikon.zum.de/ (accessed Dec. 22, 2021).

[13] “Apache Lucene - Index File Formats.” https://lucene.apache.org/core/3\_0\_3/fileformats.html (accessed Dec. 22, 2021).

[14] “GermanAnalyzer (Lucene 6.6.1 API).” https://lucene.apache.org/core/6\_6\_1/analyzers-common/org/apache/lucene/analysis/de/GermanAnalyzer.html (accessed Dec. 22, 2021).

[15] “BM25Similarity (Lucene 7.0.1 API).” https://lucene.apache.org/core/7\_0\_1/core/org/apache/lucene/search/similarities/BM25Similarity.html (accessed Dec. 22, 2021).