Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Hasan Nour Alhuda

Eine vergleichende Untersuchung zum Clustering von Textdokumenten

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Hasan Nour Alhuda

Eine vergleichende Untersuchung zum Clustering von Textdokumenten

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Wirtschaftsinformatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Olaf Zukunft
Zweitgutachter: Prof. Dr. Marina Tropmann-Frick

Eingereicht am: 20. March 2022

Hasan Nour Alhuda

Thema der Arbeit

Eine vergleichende Untersuchung zum Clustering von Textdokumenten

Stichworte

Künstliche Intelligenz, Data-Mining, Dokument-Clustering, Web-Mining, Suchmaschinen, Information-Retrieval, K-Means, DBScan, TF-IDF, Word2Vec, BERT

Kurzzusammenfassung

Clustering-Analyse ist eines der Hauptforschungsgebiete der Künstlichen Intelligenz und Data-Minings. Ihre Anwendung auf Textdokumente nennt sich Dokument-Clustering, womit sich diese Arbeit insbesondere beschäftigt. Diese Art von Clustering bezeichnet die automatische Einteilung von Dokumenten in Clustern, sodass Dokumente innerhalb eines Clusters eine hohe Ähnlichkeit im Vergleich zu Dokumenten in anderen Clustern aufweisen. Das Fachgebiet hat eine wichtige Rolle in verschiedenen Bereichen wie Web-Mining, Suchmaschinen und Information-Retrieval gespielt. Im Rahmen dieser Arbeit werden zwei Clustering-Algorithmen, K-Means und DBScan, in Kombination mit drei verschiedenen Feature-Extraktionstechniken, TF-IDF, Word2Vec und BERT, eingesetzt bzw. untersucht. Die Leistung dieser Methoden wird anhand drei ausgewählter Datensätze unter Verwendung von Clustering-Bewertungsmetriken gemessen und entsprechend bewertet.

Hasan Nour Alhuda

Title of the paper

A comparative study of clustering of text documents

Keywords

Artificial Intelligence, Data Mining, Document Clustering, Web Mining, Search Engines, Information Retrieval, K-Means, DBScan, TF-IDF, Word2Vec, BERT

Abstract

Clustering analysis is one of the main research areas of artificial intelligence and data mining. Its application on text documents is called document clustering, which is the main focus of this thesis. This type of clustering refers to the automatic classification of documents into clusters, so that documents within one cluster would have high similarity compared to documents in other clusters. This topic has played an important role in various fields such as web mining, search engines and information retrieval. In this work, two clustering algorithms, K-Means and DBScan, are used in combination with three different feature extraction techniques, TF-IDF, Word2Vec and BERT. The performance of these methods is measured and examined based on three preselected data sets using clustering evaluation metrics.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Neben der Verfügbarkeit großer Datenmengen und der Notwendigkeit, diese Daten in aussagekräftiges Wissen und Informationen umzuwandeln, hat Data-Mining in der Informationsbranche und in der Gesellschaft große Aufmerksamkeit erfahren. Dies kann für viele Anwendungen wie Betrugserkennung, Marktanalyse, Kundenbindung und wissenschaftliche Erkundung eingesetzt werden[17]. Da Data-Mining im Prozess der Wissensentdeckung ein wichtiger Schritt ist, können Unternehmen die Leistungsfähigkeit dieser neuen Technologie mit großem Potenzial in ihren Data Warehouses nutzen[17]. Clustering-Analyse ist eines der Hauptforschungsgebiete der Künstlichen-Intelligenz und des Data-Minings. Sie wird verwendet, um Daten in Clustern zu gruppieren, in denen Daten, die sich in einem Cluster befinden, eine hohe Ähnlichkeit aufweisen. Beispielsweise verwenden Websuchmaschinen Clustering, um relevante Dokumente zu einer bestimmten Abfrage in einer Gruppe von Listen mit ähnlichen Dokumenten abzurufen.

Die Anwendung von der Clustering-Analyse auf Textdokumente nennt sich Dokument-Clustering. Dies ist ein unüberwachter ML-Ansatz, der verwendet wird, um eine große Anzahl verstreuter Dokumente in eine kleine Anzahl von signifikanten und konsistenten Clustern zu gruppieren.

Die Motivation des Dokument-Clusterings besteht konkret in der Anwendung zur Untersuchung von Dokumentenbeständen. Eine effiziente Ermittlung der Zusammensetzung eines inhaltlich unbekannten Dokumentenbestandes wird ermöglicht, durch das Aufzeigen der darin beinhalteten Themengruppen und deren Benennung bzw. Bezeichnung anhand der relevantesten Begriffe zu diesen.

Letztendlich setzt die Effizienz der Gruppierung von Dokumenten und derer Qualität viele voneinander abhängenden Faktoren voraus; beginnend von der Auswahl geeigneter Merkmale von Dokumenten, Ähnlichkeitsmaßen und Cluster-Algorithmen bis hin zu der effizienten Implementierung der Clustering-Algorithmen.

1.1 Ziel der Arbeit

Diese Arbeit beschäftigt sich mit den Techniken des maschinellen Lernens, um ähnliche Textdokumente zu identifizieren. Das Hauptziel besteht darin, ausgewählte Clustering-Algorithmen (K-Means und DBScan) in Kombination mit den modernen auf Sprachmodellierung basierenden Ansätzen (Word2Vec und BERT), sowie mit dem klassischen TF-IDF, zu vergleichen. Die Leistung bzw. Qualität dieser Kombinationen wird hinsichtlich drei ausgewählte Datensätze unter Verwendung intrinsischer und extrinsischer Metriken gemessen und bewertet.

1.2 Struktur der Arbeit

Die Arbeit hat folgende Gliederung: Im zweiten Kapitel wird einen Überblick über die verschiedenen Machine-Learning Techniken gegeben. Zudem werden die relevanten Begrifflichkeiten und Prozesse für diese Arbeit erläutert, gefolgt von einer Darstellung der für den gesamten Text-Mining-Prozess notwendigen theoretischen Kenntnisse nach dem aktuellen Stand der Forschung. Darüber hinaus wird auf die wesentlichen Stufen für die Vorverarbeitung von Textdokumenten eingegangen. Das dritte Kapitel enthält eine Einführung und Hintergrundinformationen zu den verschiedenen Feature-Extraktionsmethoden und Clustering-Algorithmen, die im Rahmen dieser Arbeit verwendet werden. Im vierten Kapitel werden ausgewählte Datensätze repräsentiert und ihre qualitative und quantitative Merkmale beleuchtet. Zudem wird argumentiert, warum die Wahl darauf fiel, drei unterschiedliche Datensätze zu verwenden. Der Versuchsaufbau und die Implementierungsdetails - inklusive aller verwendeten Techniken - folgen im fünften Kapitel. Im sechsten Kapitel folgt die Evaluation. Dort wird eine Bewertung der untersuchten Methodenleistungen anhand mehrerer ausgewählter Bewertungsmetriken vorgenommen. Dabei werden die verschiedenen Ergebnisse der eingesetzten Methoden präsentiert und weiter diskutiert. Eine Zusammenfassung mit einem Ausblick zu diesem Thema schließen die Arbeit im siebten Kapitel ab.

2 Grundlagen

Um die Natur des Problems vollständig zu verstehen, ist es notwendig, grundlegende methodische Konzepte zu erläutern. Daher wird in diesem Kapitel die wichtigsten Techniken des maschinellen Lernens vorgestellt, gefolgt von einer Beschreibung des Text-Mining-Prozesses und seine Anwendungsbereiche. Zuletzt wird auf die wesentlichen Stufen für die Vorverarbeitung von Textdokumenten eingegangen.

2.1 Data-Mining

Data-Mining ist eine Problemlösungsmethodik, die eine logische oder mathematische Beschreibung von Mustern und Regelmäßigkeiten, eventuell komplexer Natur, in einem Datensatz findet.[24]. Es gibt verschiedene Definitionen von Data-Mining die sich in Kleinigkeiten unterscheiden[2][3]. Diese Arbeit verwendet die Folgende Definition: "Data-Mining ist das Studium von Sammeln, Bereinigen, Verarbeiten, Analysieren und Gewinnen nützlicher Erkenntnissen aus Daten"[7]. Data-Mining hat also das Ziel, neues Wissen und neue Querverbindungen aus den vorhandenen Daten zu extrahieren[2].

Meistens wird der Begriff Data-Mining auch als “Knowledge Discovery in Databases“ oder “KDD“ bezeichnet. Die grundlegende Definition von KDD ist: „Wissensentdeckung in Datenbanken ist der nichttriviale Prozess der Identifikation gültiger, neuer, potentiell nützlicher und schlussendlich verständlicher Muster in (großen) Datenbeständen“[13].

Es gibt Autoren, die diesen Begriff als Synonym zum Data-Mining verwenden, andere sehen Data-Mining als Kernprozess des KDD im Rahmen der Wissensidentifikation[41].

2.2 Machine-Learning Techniken

In jüngsten Data-Mining-Projekten wurden verschiedene wichtige Data-Mining-Techniken entwickelt und verwendet, damit Rohdaten in umsetzbare Erkenntnisse verwandelt werden können. Darunter sind überwachte und unüberwachte Lernmethoden.

2.2.1 Überwachtes Lernen – Supervised-Learning

Methoden des überwachten Lernens sind Techniken des maschinellen Lernens, die dazu dienen, aus den Trainingsdaten eine Funktion abzuleiten oder einen Klassifikator zu lernen, um Vorhersagen über ungesehene Daten zu treffen. Überwachtes Lernen kann beim Data-Mining in zwei Arten von Problemen unterteilt werden: Klassifizierung und Regression[27][40].

- Klassifizierung verwendet einen Algorithmus, um Testdaten genau bestimmten Kategorien zuzuordnen. Es erkennt bestimmte Entitäten innerhalb des Datensatzes und versucht, einige Schlussfolgerungen darüber zu ziehen, wie diese Entitäten gekennzeichnet oder definiert werden sollten[12]. Gängige Klassifikationsalgorithmen sind: linear-classifiers, support-vector-machines (SVM), decision-trees und k-nearest-neighbor.
- Regression wird verwendet, um Beziehungen zwischen abhängigen und unabhängigen Variablen zu verstehen[12]. Es wird häufig verwendet, um Prognosen zu erstellen, z. B. für den Umsatz eines bestimmten Unternehmens. Lineare Regression, logistische Regression und polynomiale Regression sind beliebte Regressionsalgorithmen.

2.2.2 Unüberwachtes Lernen – Unsupervised-Learning

Im Gegensatz zum überwachten Lernen sind die zu entdeckenden Muster im unüberwachten Lernen nicht bekannt, es sind weder Gruppierungen noch Klassifikationen vorgegeben. Die Lösungen, die durch entsprechende Algorithmen entwickelt werden, werden folglich nicht mit vorliegenden Lösungen abgeglichen[9]. Beispiele für das unüberwachte Lernen sind die Assoziationsanalyse und das Clustering.

- Assoziationsanalyse hat im Wesentlichen das Ziel, Assoziationsregeln zu finden[9]. Diese Regeln sind eine Methode zum Auffinden von Beziehungen zwischen Variablen in einem bestimmten Datensatz. Diese Methode wird beispielsweise häufig für die Warenkorbanalyse verwendet, damit Unternehmen die Beziehungen zwischen verschiedenen Produkten besser verstehen können und somit ein besseres Verständnis von Konsumgewohnheiten der Kunden gewinnen[41].
- Clustering ist eine Data-Mining Technik, die nicht gekennzeichnete Daten basierend auf ihren Ähnlichkeiten oder Unterschieden gruppiert. Clustering-Algorithmen können in einige wenige Typen eingeteilt werden: specifically-exclusive, overlapping, hierarchical, and probabilistic.[41]. Eine ausführliche Definition der Clusteranalyse und ihre Funktionsweise werden im Abschnitt 3.2 behandelt.

2.3 Text-Mining

Als relativ junge Disziplin wurde der Begriff Text-Mining oder Knowledge Discovery from Text (KDT) von [14] erwähnt. Text-Mining ist ein breiter Oberbegriff, der eine Reihe von Technologien zur Analyse und Verarbeitung semistrukturierten und unstrukturierten Textdaten beschreibt, um wertvolle Kenntnisse zu extrahieren[26]. Nach dem Extrahieren werden diese Daten in eine strukturierte Form konvertiert, die weiter analysiert oder direkt mit gruppierten HTML-Tabellen, Mindmaps, Diagrammen usw. präsentiert werden können. Text-Mining verwendet eine Vielzahl von Methoden zur Verarbeitung des Textes, eine der wichtigsten davon ist Natural Language Processing (NLP).

Data-Mining und Text-Mining-Systeme haben auch eine Reihe von Gemeinsamkeiten hinsichtlich ihrer Architektur. Der Unterschied liegt jedoch in der Art der Daten[15].

2.3.1 Anwendungsbereiche

Es gibt sieben verschiedene Text-Mining-Praxisbereiche, d. h. sieben sehr unterschiedliche Gebiete, die infrage kommen, wenn die Rede von Text-Mining ist[26].

Diese sieben Anwendungsbereiche sind in Abbildung 2.1 dargestellt.

Aufgrund der Breite des Bereichs von Text-Mining konzentriert sich diese Arbeit lediglich auf das Dokument-Clustering. Aus diesem Grund stellt der folgende Absatz eine kurze Definition von jedem Anwendungsgebiet in Text-Mining dar.

1. Information-Retrieval (IR): Speicherung und Abruf von Textdokumenten, einschließlich Suchmaschinen und Stichwortsuche.

2. Document-Clustering: Gruppieren und Kategorisieren von Begriffen, Snippets, Absätzen oder Dokumenten mithilfe von Data-Mining-Clustering-Methoden.

3. Document-Classification: Gruppieren und Kategorisieren von Schnipseln, Absätzen oder Dokumenten mithilfe von Data-Mining-Klassifizierungsmethoden, basierend auf Modellen, die an beschrifteten Beispielen trainiert wurden.

4. Web-Mining: Data- und Text-Mining im Internet mit besonderem Fokus auf Umfang und Vernetzung des Webs.

5. Information-Extraction (IE): Identifizierung und Extraktion relevanter Fakten und Zusammenhänge aus unstrukturiertem Text; der Prozess der Erstellung strukturierter Daten aus unstrukturiertem und semistrukturiertem Text.

6. Natural-Language-Processing (NLP): Sprachverarbeitungs- und Verständnisaufgaben auf niedriger Ebene (z. B. Markieren von Wortteilen); wird oft als Synonym von Computerlinguistik verwendet.

7. Concept-Extraction: Gruppierung von Wörtern und Phrasen in semantisch ähnliche Gruppen.

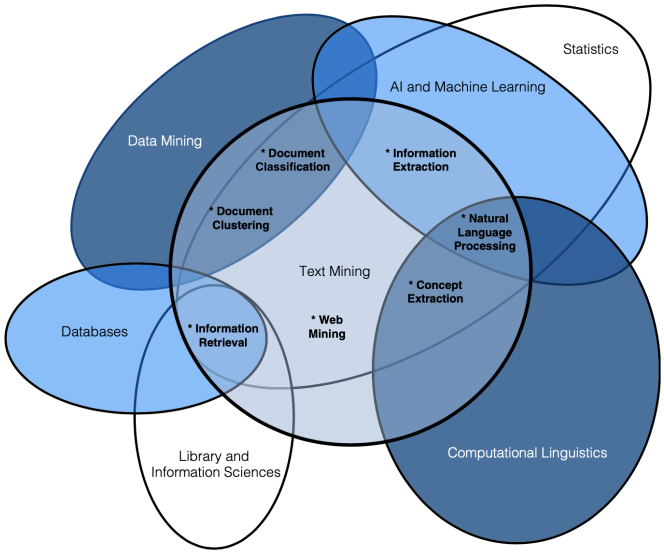


Abbildung 2.1: Anwendungsbereiche in Text-Mining [26].

2.3.2 Text-Mining Prozess

Text-Mining ist ein iterativer Prozess, bei dem die Analyse mit verschiedenen Einstellungen wiederholt und Begriffe für bessere Ergebnisse ein- oder ausgeschlossen werden. Es umfasst eine Reihe von Aktivitäten, die durchgeführt werden müssen, um Informationen effizient zu minen[20]. Diese Aktivitäten umfassen das Auswählen relevanter Daten, die Behandlung fehlender oder problemhafteter Daten, das Generieren von Merkmalen, das Entdecken von Mustern und Beziehungen innerhalb der Daten und letztendlich das Visualisieren und Interpretieren der gefundenen Muster, [9][13][3] wie in Abbildung 2.2 dargestellt. Das Ergebnis dieses Prozesses können Dokumenten-Cluster, Listen von Einzelbegriffen oder Themen mit mehreren Begriffen sein[8].

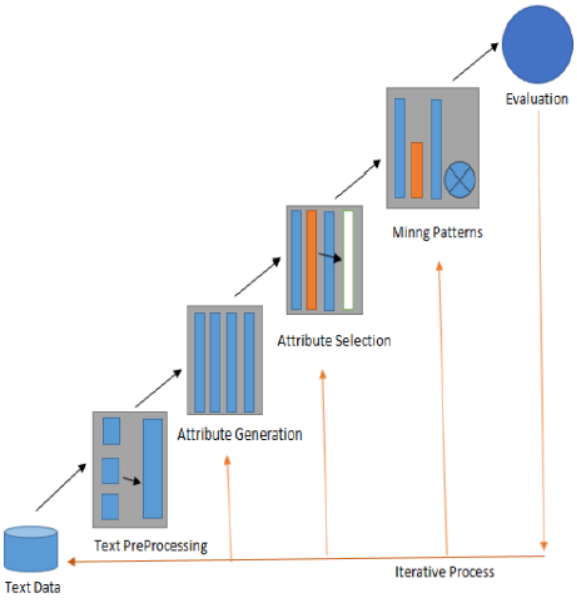


Abbildung 2.2: Activities / Process of Text-Mining [20].

2.4 Vorverarbeitung

Die Datenvorverarbeitung ist die elementarste Phase des Text-Mining, bei der Rohdaten in aussagekräftigere und verständlichere Formate umgewandelt werden. Es ist üblich, dass Textdaten in den zu analysierenden Dokumenten nicht strukturiert und konsistent sind und zahlreiche Störungen enthalten, daher müssen sie für die weitere Verarbeitung normalisiert werden. Zu den Vorverarbeitungsphasen gehören Tokenisierung, Filterung, und Normalisierung[43]. Diese Schritte der Textvorverarbeitung sind für alle Text-Mining-Aufgaben gleich[26].

2.4.1 Tokenisierung

In diesem Schritt handelt es sich um das Aufteilen der Texteinheiten in einzelne Wörter oder Sätze, die Token genannt werden[49]. Für englische Textdaten besteht eine einfache und effektive Tokenisierungsstrategie darin, Leerzeichen und Satzzeichen als Token-Trennzeichen zu verwenden[26].

2.4.2 Filterung

Das Filtern wird normalerweise bei Dokumenten durchgeführt, um Speicherplatz zu sparen und die Verarbeitung zu beschleunigen, indem einige der Wörter entfernt werden. Eine übliche Filterung ist das Entfernen von Stoppwörtern. Stoppwörter sind die Wörter, die häufig im Text vorkommen, ohne viele Inhaltsinformationen zu haben (z. B. Präpositionen, Konjunktionen usw.). Ebenso häufig vorkommende Wörter im Text, die nur wenige Informationen zur Unterscheidung verschiedener Dokumente enthalten. Auch sehr selten vorkommende Wörter sind möglicherweise ohne signifikante Relevanz und können aus den Dokumenten entfernt werden[37][42].

2.4.3 Normalisierung

Textnormalisierungs-Verfahren werden in Natural-Language-Processing zur Umwandlung eines Texts, Wörtern, und Dokumenten in eine kanonische (Wurzel-)Form verwendet. Unter anderem sind die folgende Verfahren hervorzuheben:

Stemming

Stemming ist der Prozess des Entfernens von Affixen (Präfixen und Suffixen) von Merkmalen, d.h. der Prozess, der abgeleitet wird, um flektierte (oder manchmal abgeleitete) Wörter auf ihren Stamm zu reduzieren. Der Stamm muss nicht mit der ursprünglichen morphologischen Wurzel des Wortes identifiziert werden und ist normalerweise durch die Zuordnung von Wörtern zu dem ähnlichen Stamm ausreichend verwandt. Dieser Prozess wird verwendet, um die Anzahl von Merkmalen im Merkmalsraum zu reduzieren und die Leistung der Clusterbildung zu verbessern. Es werden verschiedene Formen von Merkmalen in einem einzigen Merkmal zusammengefasst, um ein einzelnes Feature zu erhalten [19], z.B. wird der Satz von Features (“connect“, “connects“, “connected“ und “connecting“) zu einem einzigen Feature “connect“ zusammengefasst, indem die verschiedenen Suffixe -s, -ed, -ing entfernt werden.

Lemmatisieren

Lemmatisieren ist eine fortgeschrittene Form der Wortstammbildung, die versucht, Wörter basierend auf ihrem Kernkonzept oder Lemma zu gruppieren. Die Lemmatisierung verwendet sowohl den Kontext, welcher das Wort umgibt, als auch zusätzliche grammatikalische Informationen wie die Wortart, um das Lemma zu bestimmen. Folglich erfordert die Lemmatisierung mehr Informationen, um genau ausgeführt werden zu können. Bei Wörtern wie “walk“ führen Stemming und Lemmatisieren zu den gleichen Ergebnissen. Bei Wörtern wie “meeting“

allerdings, die entweder als Substantiv oder Verb dienen können, erzeugt das Stemmen den gleichen Stamm “meet“, das Lemmatisieren jedoch erzeugt “meet“ für das Verb und behält “meeting“ im Nomenfall bei[26].

3 Methoden

Das vorliegende Kapitel stellt die verschiedenen Feature-Extraxtion-Ansätze und Clustering-Algorithmen vor, die im Rahmen dieser Arbeit verwendet werden, indem es einen detaillierten Einblick in ihre Funktionsweise gibt.

3.1 Feature-Extraktion

Die Besonderheit von Textdokumenten als zu gruppierende Objekte ist ihre sehr komplexe und reichhaltige interne Struktur. Da Maschine-Learning-Algorithmen nicht in der Lage sind, Rohinhalte in ihrer Rohstruktur zu verarbeiten bzw. um Textdokumente gruppieren zu können, müssen die Dokumente im Merkmalsraum in wohldefinierten Zahlen fester Länge (Vektoren) umgewandelt werden[15]. Die Wahl der Dokumentdarstellungsmethode hat einen tiefgreifenden Einfluss auf die Gesamtqualität des Clusterings.

In diesem Abschnitt werden einige ausgewählte Techniken zur Abbildung von Texten in numerische Feature-Vektoren vorgestellt, die später im Kombination mit ausgewählten Clustering-Algorithmen verwendet werden. Die Ansätze, die als primäre Feature-Extraktionsmethoden zum Vergleich ausgewählt wurden, sind *term frequency-inverse document frequency* (TF-IDF), *Dence Vector Representations*, (Word2Vec) und *BERT*. Sie werden in zwei häufig verwendete Hauptmethoden grob eingeteilt: Bag-of-Words und Word-Embeddings.

3.1.1 TF-IDF

Tf-idf stammt aus einer traditionellen und einfachen Methode in der Verarbeitung natürlicher Sprache zur Merkmalsextraktion bzw. Darstellung von Textdokumenten basierend auf Worthäufigkeit von Wörtern/Token im Dokument/Korpus nämlich Bag-of-words.

Beim Bag-of-words kann man es sich so vorstellen, dass jedes Wort eine Dimension im Merkmalsraum ist. Jeder Vektor, der ein Dokument in diesem Raum repräsentiert, hat eine Komponente jedes Wortes.

Wenn im Dokument kein Wort vorhanden ist, ist die Wortkomponente des Dokumentvektors null. Andernfalls wird es ein positiver Wert sein, der von der Häufigkeit des Wortes im

Dokument und in der gesamten Dokumentensammlung abhängen kann. Der Name des Modells basiert auf der Tatsache, dass jedes Dokument buchstäblich als eine Tüte mit seinen eigenen Wörtern dargestellt wird. Wenn man z. B. die folgende Wörterliste [“is", “Live", “Permanence", “success”] als Vokabular betrachtet, wird der Satz “Permanence is successäls [1, 0, 1, 1] dargestellt. Auf dieser Weise bleiben die Häufigkeiten der Begriffe erhalten, allerdings berücksichtigt das Modell nicht die Wortreihenfolge und Grammatik innerhalb eines Dokuments.

Bag-of-words Modelle werden häufig beim Computer Vision, Natural Language Processing (NLP), Clustering, Dokumentenklassifizierung und Informationsabruf durch künstliche Intelligenz (KI) verwendet.

Das TF-IDF wiederum ist ein numerisches statistisches Maß, das widerspiegeln soll, wie relevant ein Wort für ein Dokument in einer Sammlung oder einem Korpus ist[46]. Die TF-IDF Transformation berücksichtigt die relative Häufigkeit jedes Wortes im Korpus, sowie die Länge jedes Dokuments. Es verringert das Gewicht von Begriffen, die in vielen Dokumenten vorkommen, und erhöht gleichzeitig das Gewicht von Begriffen, die nur in wenigen Dokumenten vorkommen. Dies verbessert meistens die Vorhersage- und Clustering-Leistung. Diese Methode kann also als eine Form des Bag-of-words Modells angesehen werden, da weder Grammatik noch Reihenfolge dabei berücksichtigt werden.

TF-IDF für ein Wort in einem Dokument wird durch Multiplizieren von zwei verschiedenen Metriken berechnet: Die erste Metrik berechnet, wie oft ein Term in einem Dokument vorkommt. Der Grund hierfür ist, dass Wörter, die häufig in einem Dokument vorkommen, wahrscheinlich wichtiger sind. Das Ergebnis wird dann normalisiert, indem es durch die Anzahl der Wörter im gesamten Dokument dividiert wird. Diese Normalisierung wird durchgeführt, um eine Tendenz zu längeren Dokumenten zu verhindern, sodass die Häufigkeit des Auftretens des Begriffs und nicht nur die Rohzahl des Begriffs erhalten wird[21].

$$tf_{t,d} = \frac{n_{t,d}}{\sum n_{k,d}} \tag{3.1}$$

Wobei $n_{t,d}$ die Häufigkeit ist, mit der der Term t in Dokument d vorkommt, und $n_{k,d}$ die Anzahl der Vorkommen jedes Terms in Dokument d ist.

Um den IDF-Teil der gesamten Formel zu berechnen, wird die Gesamtzahl der Dokumente im Korpus durch die Anzahl der Dokumente dividiert, in denen ein Begriff erscheint. um zu ermitteln wie häufig ein Wort im gesamten Dokumentensatz vorkommt. Das Ergebnis wird dann logarithmiert.

$$idf_t = \log \frac{|D|}{|D|_t} \tag{3.2}$$

Wobei $|D|$ die Gesamtzahl der Dokumente ist, und $|D|_t$ die Anzahl der Dokumente ist, in denen der Begriff t erscheint.
Durch Multiplikation der beiden Teile TF und IDF für einen bestimmten Begriff wird ein Maß dafür erhalten, wie kennzeichnend dieser Begriff ist.

$$tf - idf_{t,d} = tf_{t,d} \times idf_t \tag{3.3}$$

3.1.2 Worteinbettungen

Worteinbettungen sind einfache Vektordarstellungen von Wörtern. Es geht dabei um eine Abbildung von Wörtern in Vektoren reeller Zahlen unter Verwendung des neuronalen Netzes, des Wahrscheinlichkeitsmodells, zur Dimensionsreduktion auf der Wort-Ko-Auftritts-Matrix.

Insbesondere handelt es sich um ein Vektorraummodell (VSM), das Wörter in einem kontinuierlichen Vektorraum darstellt, so dass Wörter, die gemeinsame Kontexte und Semantiken teilen, im Raum dicht nebeneinanderstehen [22]. Im Gegensatz zu Bag-of-words Modellen, die nur die Anzahl der Wortvorkommen in Dokumenten darstellen, ohne Beziehungen oder Kontexte zu erkennen. Zum Beispiel werden Wörter wie [“King", “Queen"] eng verwendet und treten in einem Text eher nahe beieinander auf als Wörter wie [“King", “cosmology"]. Daher würden [“King", “Queen"] im Vektorraum näher beieinander liegen als [“King", “cosmology"]. Aus diesem Grund werden diese Modelle häufig bei Problemen mit Natural Language Processing (NLP) verwendet.

3.1.3 Word2Vec

Word2Vec ist ein flaches, zweischichtiges neuronales Netzwerk, welches von [25] vorgestellt wurde, das trainiert ist, linguistische Kontexte von Wörtern zu rekonstruieren. Es nimmt als Eingabe ein großes Korpus von Wörtern und erzeugt einen Vektorraum, typischerweise von mehreren hundert Dimensionen, wobei jedem eindeutigen Wort im Korpus ein entsprechender Vektor im Raum zugewiesen wird. Wortvektoren werden im Vektorraum so positioniert, dass Wörter, die gemeinsame Kontexte im Korpus haben, im Raum in unmittelbarer Nähe zueinander liegen.

Word2Vec ist ein besonders rechen effizientes Vorhersagemodell zum Erlernen von Worteinbettungen aus Rohtexten.

Das Modell wird gefüttert mit Wörtern als One-Hot-Vektoren, bei denen es sich im Grunde genommen um einen Vektor mit der gleichen Länge wie das Vokabular handelt, der mit Nullen gefüllt ist, außer am Index, der das Wort darstellt, der wird eine 1 zugewiesen. Die versteckte Schicht ist eine standardmäßige, vollständig verbundene (dichte) Schicht, deren Gewichte

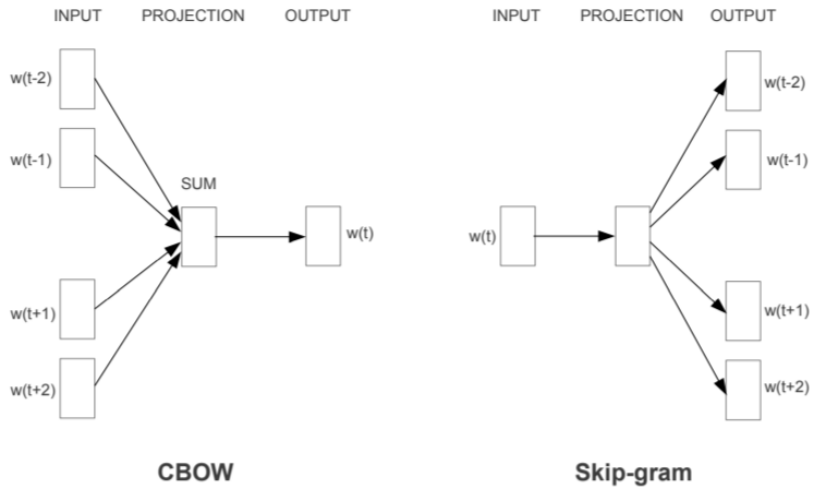


Abbildung 3.1: Word2Vec Training Model Architecture [25].

die Worteinbettungen sind, während die Ausgabeschicht gibt Wahrscheinlichkeiten für die Zielwörter aus dem Vokabular aus.

Wortvektoren können mittels folgender logarithmisch-linearer Modelle ermittelt werden: *CBoW* oder *Skip-Gram*. Wie in der Abbildung 3.1 unten gezeigt. Die Vektordarstellung extrahiert semantische Beziehungen basierend auf dem gemeinsamen Vorkommen von Wörtern im Datensatz.

- CBoW bzw. Continuous Bag of Words versucht im Wesentlichen, ein Zielwort aus einer Liste von Kontextwörtern vorherzusagen
- Skip-Gram-Modell ist das genaue Gegenteil von CBOW-Modell, denn es nimmt das aktuelle Wort als Eingabe und versucht, die Wörter vor und nach diesem aktuellen Wort genau vorherzusagen.

Beide Rechenmodelle haben ihre Vor- und Nachteile. Die Genauigkeit, mit der das Modell die Wörter vorhersagt, hängt davon ab, wie oft das Modell diese Wörter im gesamten Datensatz im selben Kontext sieht. Die versteckte Darstellung wird während des Trainingsprozesses durch mehr Wörter und Kontext-Co-Auftritte verändert, was es dem Modell ermöglicht, mehr zukünftige erfolgreiche Vorhersagen zu haben, somit führt es zu einer besseren Darstellung von Wort und Kontext im Vektorraum. Skip-Gramm ist viel langsamer als CBOW, funktioniert aber bei seltenen Wörtern genauer[25]. Beide word2vec-Varianten erzeugten Worteinbettungen, die

mehrere Ähnlichkeitsgrade erfassen können, einschließlich syntaktischer und semantischer Regelmäßigkeiten[45].

3.1.4 Bert

Das Sprachrepräsentationsmodell BERT steht für Bidirectional Encoder Representation from Transformers und bezeichnet ein Algorithmus-Modell, das auf neuronalen Netzwerken basiert. Es wurde von Google entwickelt, um tiefe bidirektionale Darstellungen von unbeschriftetem Text vorab zu trainieren, indem es sowohl den linken als auch den rechten Kontext gemeinsam konditioniert. Dadurch kann das vortrainierte BERT-Modell mit nur einer zusätzlichen Ausgabeschicht verfeinert werden, um hochmoderne Modelle für eine Vielzahl von NLP-Aufgaben zu erstellen[11].

Im Gegensatz zu direktionalen Modellen, die die Texteingabe sequentiell (von links nach rechts oder von rechts nach links) lesen, liest der Transformer-Encoder die gesamte Wortfolge auf einmal. Daher wird es als bidirektional angesehen. Diese Eigenschaft ermöglicht es dem Modell, den Kontext eines Wortes basierend auf seiner gesamten Umgebung zu lernen[11].

Das Bert-Framework hat zwei Hauptschritte: *Pre-training* und *Fine-tuning*.
Während des Vortrainings wird das Modell auf nicht gekennzeichneten Daten über verschiedene Vortrainingsaufgaben trainiert. Zur Feinabstimmung wird das BERT-Modell zunächst mit den vortrainierten Parametern initialisiert, und alle Parameter werden unter Verwendung von gekennzeichneten Daten aus den nachgelagerten Tasks fein abgestimmt. In Abbildung 3.2 wird das Verfahren zur Beantwortung von Fragen(Question-Answering) als laufendes Beispiel für diesen Abschnitt dienen.

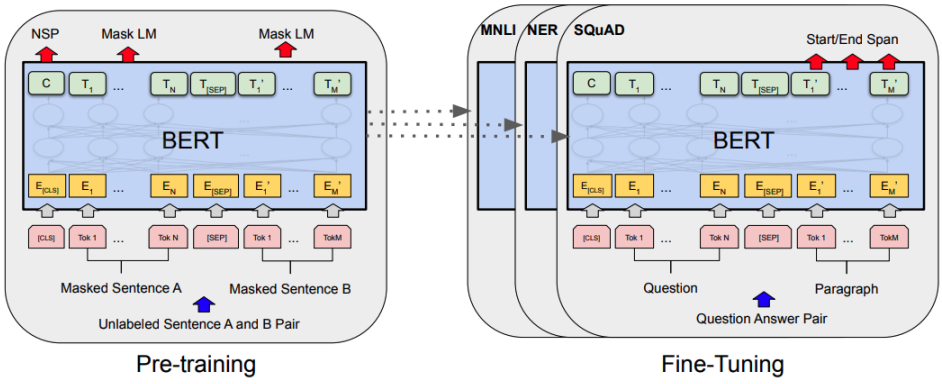


Abbildung 3.2: Allgemeine Vorbereitungs- und Feinabstimmungsverfahren bei BERT [11].

- **Pre-training**

Im Gegensatz zu traditionellen Sprachmodellen, die von links nach rechts oder rechts nach links vorab trainiert, wird Bert trainiert, indem er die folgende zwei unüberwachte Aufgaben, Masked-Language-Modelling (MLM) und Next-Sentence-Prediction (NSP), verwendet. Dieser Schritt ist im linken Teil von Abbildung 3.2 dargestellt.

Masked Language Modelling (MLM): Bevor Wortfolgen in BERT eingegeben werden, werden 15% der Wörter in jeder Folge durch ein [MASK]-Token ersetzt. Das Modell versucht dann, den ursprünglichen Wert der maskierten Wörter basierend auf dem Kontext vorherzusagen, der von den anderen, nicht maskierten Wörtern in der Sequenz bereitgestellt wird.

Next Sentence Prediction (NSP): Beim BERT-Trainingsprozess empfängt das Modell Satzpaare als Eingabe und lernt vorherzusagen, ob der zweite Satz im Paar der nachfolgende Satz im Originaldokument ist. Während des Trainings sind 50% der Eingaben ein Paar, bei dem der zweite Satz der Folgesatz im Originaldokument ist, während in den anderen 50% ein zufälliger Satz aus dem Korpus als zweiter Satz gewählt wird. Die Annahme ist, dass der Zufallssatz vom ersten Satz getrennt wird.

Damit das Modell beim Training zwischen den beiden Sätzen unterscheiden kann, wird die Eingabe vor dem Eintreten in das Modell wie folgt verarbeitet: Zunächst wird ein [CLS]-Token am Anfang des ersten Satzes eingefügt und ein [SEP]-Token am Ende jedes Satzes eingefügt. Jedem Token wird eine Satzeinbettung hinzugefügt, die Satz A oder Satz B angibt. Anschließend wird jedem Token eine Positionseinbettung hinzugefügt, um seine Position in der Sequenz anzuzeigen. Eine Visualisierung dieser Konstruktion ist in Abbildung 3.3 zu sehen.

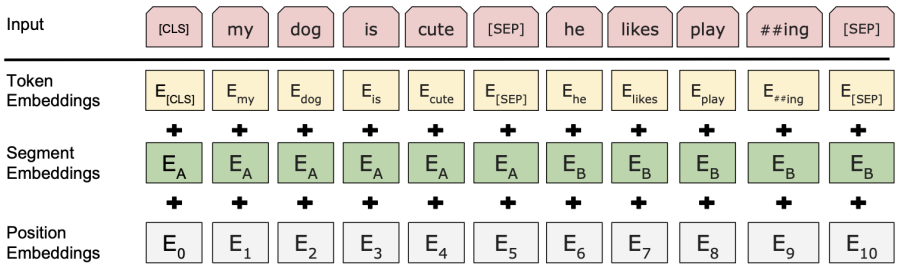


Abbildung 3.3: BERT-Eingabedarstellung [11].

- **Fine-tuning:**

Für jede Aufgabe wird einfach die aufgabenspezifischen Ein- und Ausgänge in BERT eingesteckt und es werden alle Parameter Ende-zu-Ende verfeinert. Bei der Eingabe sind Satz A und Satz B aus dem Pre-training analog zu den Frage-Passage-Paaren bei der QA-Aufgabe und bei der Ausgabe werden die Token-Darstellungen in eine Ausgabeschicht für verschiedene Aufgaben auf Token-Ebene eingespeist, in diesem Fall für QA.

3.2 Clusteranalyse

Clustering oder Clusteranalyse ist der Prozess der automatischen Identifizierung versteckter Strukturen in nicht gekennzeichneten Daten[4]. Clustering ist eine unüberwachte Lernmethode, was bedeutet, dass keine gekennzeichneten Trainingsbeispiele bereitgestellt werden müssen, damit das Clustering erfolgreich wird[38]. Dabei zielen Clustering-Algorithmen darauf ab, latente Muster in Daten zu entdecken, indem sie Funktionen verwenden, um Instanzen in sinnvollerweise unähnliche Gruppen zu organisieren. Unüberwachte Methoden sind weniger leistungsfähig als überwachte Methoden, aber sie können in einem viel größeren Bereich von Problemen eingesetzt werden[26].

3.2.1 Dokument-Clustering

Dokument-Clustering ist ein interessanter Bereich in der Verarbeitung natürlicher Sprache und Textanalyse, der unüberwachte Konzepte und Techniken des maschinellen Lernens anwendet[38]. Die Hauptprämisse des Dokument-Clustering besagt, dass ein ganzes Korpus von Dokumenten, basierend auf einigen charakteristischen Eigenschaften, Attributen und Merkmalen, in verschiedene Clusters aufgeteilt werden muss. Um dies zu erreichen, werden beim Text-Mining Clustering-Algorithmen verwendet. Die Eigenschaften der erstellten Clusters sind so, dass Dokumente innerhalb eines Clusters ähnlicher und miteinander verbunden sind als Dokumente, die zu anderen Clustern gehören. Wenn Dokumente geclustert werden, wird dies normalerweise als Dokumentenclustering oder Textclustering bezeichnet[26]. Nachdem die Dokument-Clusterbildung durchgeführt wurde, wird ein Cluster häufig durch die darin vorkommenden Wörter identifiziert, die am häufigsten vorkommen.

3.2.2 Clustering-Algorithmen

Da sich nun die Vektoren zwei beliebiger Dokumente quantifizieren lässt, kann damit gestartet werden, unüberwachte Algorithmen zum Auffinden ähnlicher Gruppen von Dokumenten zu

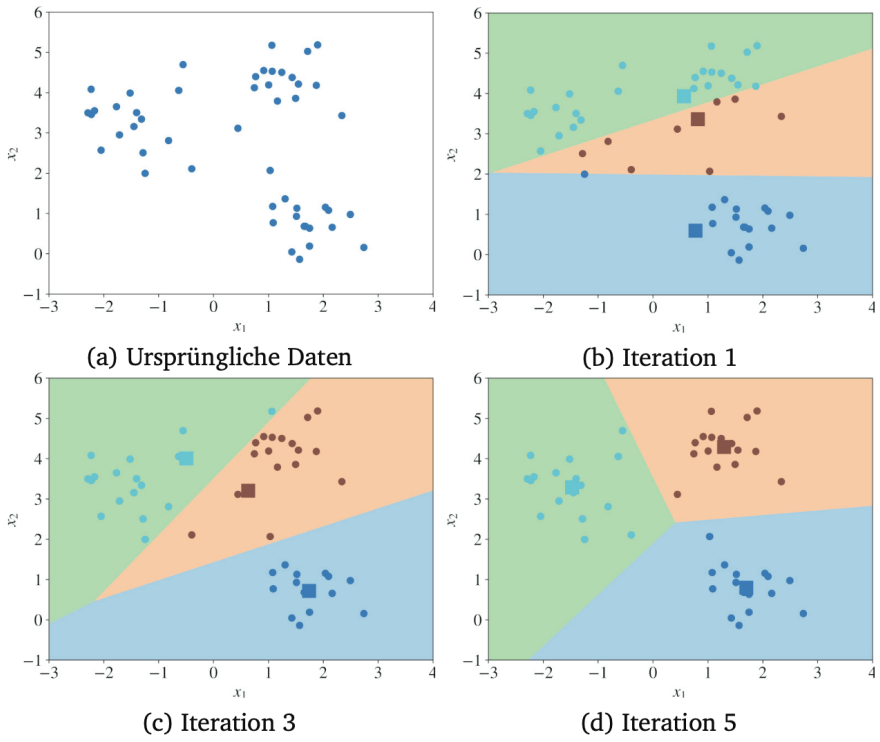
untersuchen. K-Means und DBScan sind im Rahmen dieser Arbeit hauptsächlich verwendet, denn beide teilen Dokumente in Gruppen auf, deren Mitglieder die maximale Ähnlichkeit aufweisen, wie durch eine Distanzmetrik definiert[4]. Darüber hinaus berechnen K-means und DBScan ein sogenanntes hartes Clustering, bei dem jeder Datensatz nur zu einem Cluster gehören kann.

K-Means

Das k-Means Clustering ist der wahrscheinlich am weitesten verbreitete Partitionierungsalgorithmus, da er einfach zu bedienen ist und mit großen Datenmengen skalierbar ist[41]. Er ist ein centroid-based clustering-Modell, das versucht, Daten in Gruppen oder Cluster gleicher Varianz zu gruppieren[38].

Jedes der k Cluster Ci wird durch den Durchschnittswert ci seiner Punkte definiert, ci heißt dann auch Zentroid. Das Ziel dabei ist, die durchschnittliche Distanz zwischen Texten und ihren Zentroiden zu minimieren bzw. die Ähnlichkeit zwischen Texten und ihren Zentroiden zu maximieren. Zuerst werden zufällig k Punkte als initiale Zentroide bestimmt und von diesen ausgehend die anderen Punkte um die Zentroide angeordnet. Für die sich so ergebenen Cluster werden dann die Zentroiden neu bestimmt. Auf dieser Basis wird der Prozess rekursiv solange über den k Clustern fortgeführt, bis keine Veränderungen bzw. Bewegungen der Zentroiden mehr stattfinden. In Abbildung 3.4 ist die Ausführung des k-Means-Algorithmus dargestellt. Die Kreise in der Abbildung sind zweidimensionale Merkmalsvektoren, die Quadrate sind die umherwandernden Zentroiden. Die verschiedenen Hintergrundfarben repräsentieren Bereiche, in denen alle Datensätze zum selben Cluster gehören[6].

K-means Verfahren sind einfach, schnell und funktionieren am besten bei sphärischen, nicht-überlappenden Clustern[44], haben allerdings den Nachteil, dass sie nur numerische Merkmale für die Distanzberechnung verwenden können.



(a) Ursprüngliche Daten (b) Iteration 1

(c) Iteration 3 (d) Iteration 5

Abbildung 3.4: Der Fortschritt des k-Means-Algorithmus für k=3 [6].

DBScan

DBScan (Density-Based Spatial Clustering of Applications with Noise) ist ein dichtebasierter Clustering-Algorithmus, der für beliebig geformte Cluster mit Rauschen geeignet ist[31]. Es basiert auf den Konzepten der density-reachability und density-connectivity unter gegebenen Eingabeparametern (eps) und $(minPts)$[32]. Die beiden Konzepte des DBScan und die Parameter können wie folgt definiert werden:

eps: ist ein Abstandsmaß, das verwendet wird, um die Punkte in der Nähe eines beliebigen Punktes zu lokalisieren.

$minPts$: Ist die Mindestanzahl von Punkten (ein Schwellenwert), die zusammengefasst sind, damit eine Region als dicht angesehen werden kann. Wobei Punkte, die sich innerhalb einer dichten Region befinden, werden als Kernpunkte bezeichnet.

Density-Reachability besagt, dass einen Punkt festgelegt ist, der von einem anderen aus erreichbar ist, wenn er innerhalb eines bestimmten Abstands (eps) von ihm liegt. Density-Connectivity hingegen beinhaltet einen transitivitätsbasierten Verkettungsansatz, um festzustellen, ob sich Punkte in einem bestimmten Cluster befinden. Zum Beispiel könnten p- und

q-Punkte verbunden werden, wenn $(p- > r- > s- > q)$, wobei $(a- > b)$ bedeutet, dass b sich in der Nachbarschaft von a befindet.

Die Funktionsweise wie DBScan mit verschiedenen Eingabeparametern reagiert, ist in Abbidlung 3.5 näher beigebracht. Der Algorithmus fährt fort, indem er willkürlich einen Punkt im Datensatz aufnimmt (bis alle Punkte besucht wurden). Wenn sich mindestens $(minPts)$ in einem Radius von (eps) um den Punkt befinden, betrachtet er alle diese Punkte als Teil desselben Clusters. Die Cluster werden dann erweitert, indem die Nachbarschaftsberechnung für jeden Nachbarpunkt rekursiv wiederholt wird.

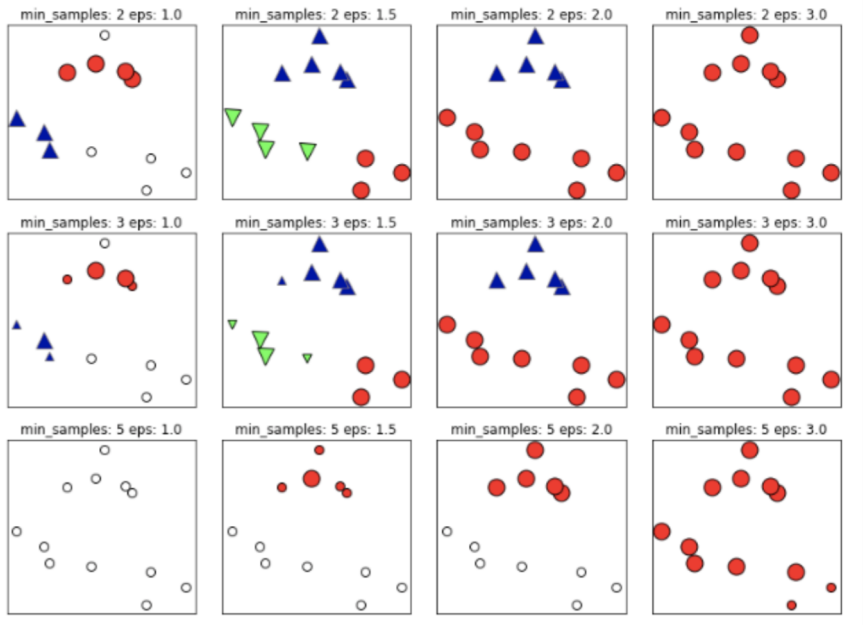


Abbildung 3.5: DBScan-Ergebnisse mit verschiedenen Eingabe-Parameter dargestellt.

Wobei das Erhöhen von eps (von links nach rechts in der Abbildung 3.5) bedeutet, dass mehr Punkte in einem Cluster enthalten sind. Dies lässt Cluster wachsen, kann aber auch dazu führen, dass sich mehrere Cluster zu einem zusammenschließen. Eine Erhöhung von minPts (in der Abbildung 3.5 von oben nach unten) bedeutet, dass weniger Punkte Kernpunkte sind und mehr Punkte als Rauschen gekennzeichnet werden.

3.2.3 Dimensionsreduzierung

Die Dimensionsreduzierung ist ein Prozess, bei dem die Informationen aus einem riesigen dimensionalen Merkmalsraum unter Verwendung weniger wesentlicher Dimensionen gewonnen

werden. Das Reduzieren der Dimensionalität von hochdimensionalen Daten ist gut für eine verbesserte Klassifizierung, Clusterbildung, Regression und Visualisierung von Daten [39]. Die umfangreichste und am weitesten verbreitete Technik ist die Principal-Component-Analysis bzw. Hauptkomponentenanalyse und wird als PCA bezeichnet [39]. PCA komprimiert den Datenrahmen durch orthogonales Umwandeln der gegebenen Daten als eine Anzahl von Hauptkomponenten. Die anfängliche Hauptkomponente beschreibt den Großteil der Datenvariation in einer einzelnen Komponente. Die zweite Komponente beschreibt die zweithöchsten Beträge der Abweichung usw. Durch sorgfältige Auswahl der obersten Hauptkomponenten, die 80–90 % der Datenvariation beschreiben, können die verbleibenden Komponenten verworfen werden, da sie dem Modell nicht wesentlich helfen. Dieser Prozess bewahrt einige latente Informationen in den Hauptkomponenten, die hilfreich sind, um bessere Modelle zu konstruieren [39].

4 Datensätze

Das folgende Kapitel beleuchtet die Datensätze, die in dieser Arbeit verwendet werden. Zuerst werden grundlegende Merkmale der Sätze besprochen. Außerdem wird erklärt, warum gerade drei verschiedene Datensätze für diese Arbeit ausgewählt wurden. Der folgende Teil gibt tiefere Einblicke in qualitative und quantitative Merkmale der Datensätze.

4.1 Datenbeschreibung

Die in dieser Arbeit verwendeten Datensätze wurden von [10][16][1] erstellt und infolge von der Kaggle-Online-Community extrahiert, die hauptsächlich als eine Wettbewerbsplattform für Data Science und Machine Learning bezeichnet wird. Es wurden drei verschiedene Datensätze ausgewählt basierend auf der Tatsache, ob gerade dieser Satz für Clustering-Verfahren gut geeignet ist. Dies lässt sich so prüfen, indem eine schnelle Analyse der Datenverteilung im Vektorraum durchgeführt wird. Durch eine einfache Visualisierung der Datenpunkte mit einem Streudiagramm, kann festgestellt werden, ob dieser Datensatz für Clustering gut geeignet ist. Ein Datensatz ist dann gut geeignet, wenn die Daten eindeutige und sinnvolle Gruppen bilden.

Abbildung 4.1 bietet einen groben Überblick über die Merkmale der einzelnen Datensätze. Die Spalten *Dataset Size* und *Number of Category* beschreiben, wie groß jeder Satz ist und in wie vielen Kategorien die Dokumente eingeteilt sind. Außerdem zeigt die Spalte *Word Count* die durchschnittliche Länge der enthaltenen Dokumente in jedem Satz. Ein wichtiges Merkmal ist die Spalte *Feature*, denn damit wird jeder Satz als für Clustering *geeignet*, *semi geeignet* oder *nicht geeignet* bezeichnet, abhängig von der Verteilung der Datenpunkte, wie bereits beschrieben.

	Dataset Size	Number of Categories	Word Count	Feature
First Dataset	2225	5	1400	suitable
Second Dataset	7628	4	416	semi suitable
Third Dataset	7600	4	129	not suitable

Abbildung 4.1: Einblick in die Datensätze

In der Abbildung 4.2 wird der allgemeine Aufbau der drei Datensätze vorgestellt. Die Spalte ‘Story’ stellt einen Teil des Hauptinhalts des Artikels dar, der als Nachricht veröffentlicht werden soll. Die Spalte *Category* bezeichnet, wie der Name schon sagt, das Genre bzw. die Kategorie, zu der die Nachrichten gehören. Als Beipsiel, teilt der zweite Datensatz die Dokumente in vier Kategorien und lässt sich in Abbildung 4.2b in der Spalte *Section* folgendermaßen kennzeichnen: (*Politics: 0 Technology: 1 Entertainment: 2 Business: 3*).

In der Spalte *Words-count* wird angezeigt, wie viele Wörter jedes Dokument beinhaltet. Damit wird ein besserer Blick in die Struktur des Datensatzes gewonnen. Die Spalte Cleaned-Text beinhaltet bereits bereinigte Dokumente nach Durchführung der Vorverarbeitungs-Schritte, wie im Kapital 2.4 beschrieben wurde. Dabei werden häufig verwendete Wörter bzw. Stopwörter, Zahlenangaben und Interpunktionszeichen entfernt, damit den Wörtern, die die Bedeutung des Textes definieren, mehr Aufmerksamkeit geschenkt werden kann.

	Story	Section	Category	Words_count	Cleaned_text
0	tv future in the hands of viewers with home th...	0	tech	737	future hands viewers home theatre systems plas...
1	worldcom boss left books alone former worldc...	1	business	300	worldcom boss left books alone former worldcom...
2	tigers wary of farrell gamble leicester say ...	2	sport	246	tigers wary farrell gamble leicester rushed ma...
3	yeading face newcastle in fa cup premiership s...	2	sport	341	yeading face newcastle cup premiership side ne...
4	ocean s twelve raids box office ocean s twelve...	3	entertainment	260	ocean twelve raids box office ocean twelve cri...

(a) Erster Datensatz

	Story	Section	Category	Words_count	Cleaned_text
0	But the most painful was the huge reversal in ...	3	Business	148	painful huge reversal fee income unheard priva...
1	How formidable is the opposition alliance amon...	0	Politics	17	formidable opposition alliance congress jharkh...
2	Most Asian currencies were trading lower today...	3	Business	58	asian currencies trading lower south korean ch...
3	If you want to answer any question, click on '...	1	Technology	103	answer question click answer clicking answer c...
4	In global markets, gold prices edged up today ...	3	Business	46	global markets gold prices edged disappointing...

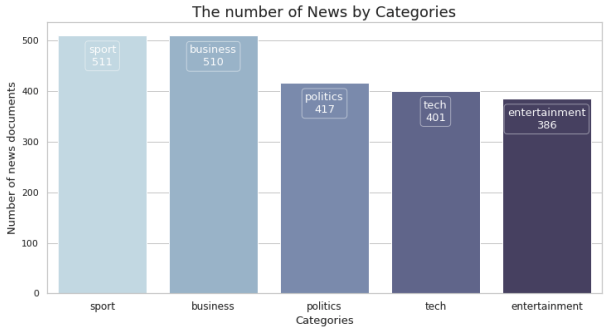
(b) Zweiter Datensatz

	Story	Section	Category	Words_count	Cleaned_text
0	Unions representing workers at Turner Newall...	3	Business	18	unions representing workers turner newall disa...
1	SPACE.com - TORONTO, Canada -- A second\team o...	4	Sci/Tech	34	space com toronto canada team rocketeers compe...
2	AP - A company founded by a chemistry research...	4	Sci/Tech	37	company founded chemistry researcher universit...
3	AP - It's barely dawn when Mike Fitzpatrick st...	4	Sci/Tech	49	barely dawn mike fitzpatrick starts shift blur...
4	AP - Southern California's smog-fighting agenc...	4	Sci/Tech	27	southern californias smog fighting agency went...

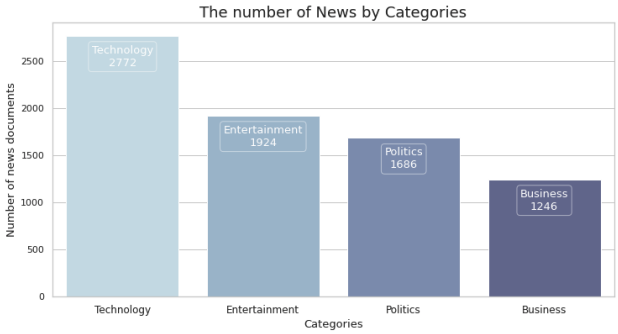
(c) Dritter Datensatz

Abbildung 4.2: Die erste fünf Zeilen der verwendeten Datensätze.

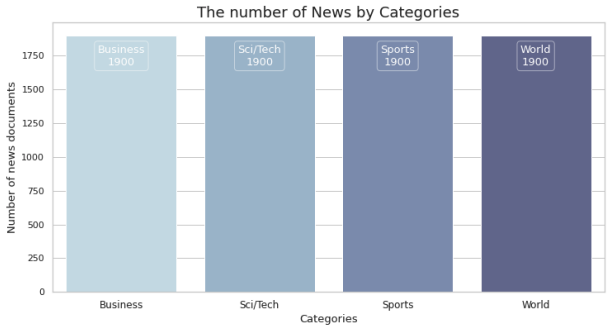
In Abbildung 4.3 wird die Anzahl der Dokumente, verteilt nach Kategorien, in jedem Datensatz dargestellt. Es ist erkennbar, dass der dritte Datensatz eine ausgeglichene Themenverteilung aufweist, während der zweite Satz eine ungleichmäßige Verteilung aufweist.



(a) Erster Datensatz



(b) Zweiter Datensatz

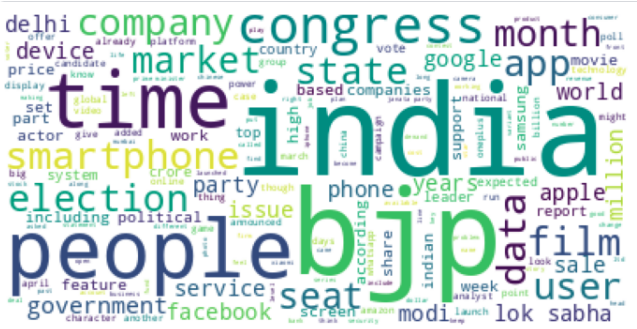


(c) Dritter Datensatz

Abbildung 4.3: Anzahl der Dokumenten nach Kategorie.

In Abbildung 4.4 werden die am häufigsten aufgetretenen Wörter in jedem Datensatz visualisiert. Während offensichtliche neutrale Wörter, wie zum Beispiel “*years*“, “*time*“, “*people*“, im zweiten Datensatz in Abbidlung 4.4b, zu mehreren Clustern zugeordnet werden können, lassen sich Begriffe wie (“*goverment*", “*state*") eindeutig der Kategorie *Politics* zuordnen.

(a) Erster Datensatz

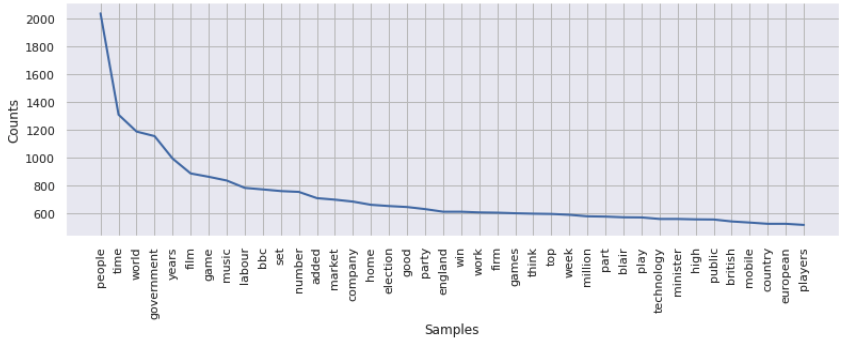


(b) Zweiter Datensatz

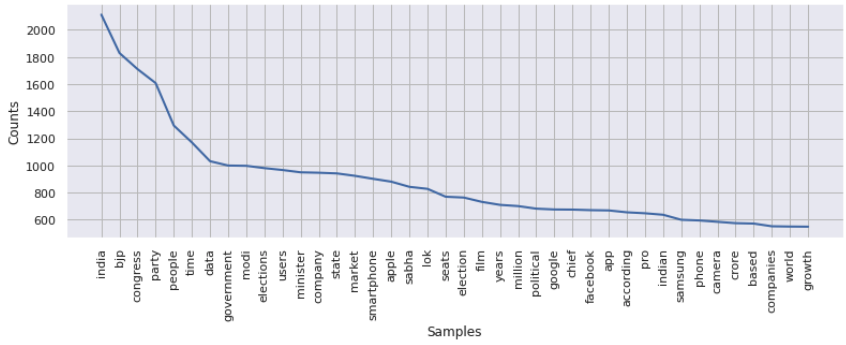
(c) Dritter Datensatz

Abbildung 4.4: Word Cloud.

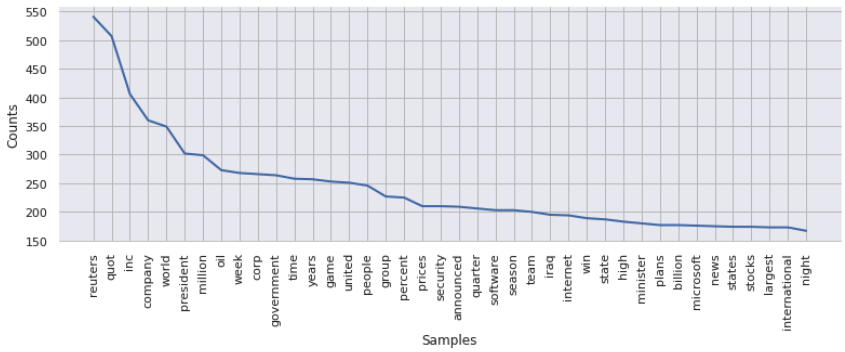
Abbildung 4.5 zeigt wiederum wie häufig die, in der Abbildung 4.4, visualiserten Begriffe vorgekommen sind. Zum Beispiel ist der Begriff ("*India*") ca. 2100 mal im zweiten Datensatz aufgetreten, während der Begriff ("*reuters*") im dritten Datensatz nicht mehr als ca. 545 mal vorgekommen ist.



(a) Erster Datensatz



(b) Zweiter Datensatz



(c) Dritter Datensatz

Abbildung 4.5: Verteilung der 40 am häufigsten vorkommenden Token in jedem Datensatz.

5 Experimente

Basierend auf dem Hauptziel dieser Arbeit, die Leistung unterschiedlicher Ansätze im Kombination mit ausgewählten Clustering-Algorithmen zur Gruppierung von ähnlichen Dokumenten zu bewerten, wird eine Reihe von Experimenten durchgeführt, um das Dokument-Clustering als eines der wichtigsten Probleme von NLP weiter zu untersuchen. Infolgedessen gibt dieses Kapitel einen Überblick über das experimentelle Konzept und Design. Darüber hinaus werden die technischen Details zu den verwendeten Python- Bibliotheken, -Tools und -Modellen sowie die konkrete Umsetzung dieser Experimente näher beschrieben.

5.1 Versuchsaufbau

Alle Experimente haben den Zweck, Kombinationen verschiedener Feature-Extraktionsmodelle und ausgewählte Clustering-Algorithmen zu vergleichen. Jeder Test folgt daher die gleiche Pipeline, die in Abbildung 5.1 dargestellt ist.

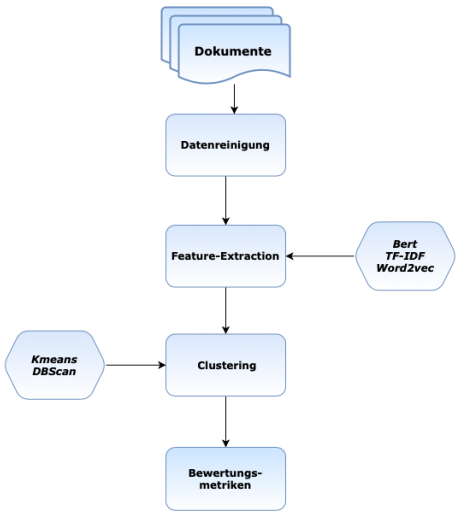


Abbildung 5.1: Experiment-Pipeline.

5.1.1 Datenreinigung

Die Bereinigung und Vorverarbeitung von Daten ist eins der wesentlichen Schritte, da die Zuverlässigkeit des zu erstellenden Modells stark von der Qualität der Daten abhängt. In diesem Schritt werden die rohen Textdokumente einem Datenbereinigungsmodul übergeben, welches die Daten für die nächste Schritte vorbereitet. Dies umfasst die Kleinschreibung alle Texte, Tokenisierung der Textdaten in Wörtern und die Entfernung von Stopwörter, Zahlenangaben und Interpunktionszeichen.

5.1.2 Feature-Extraktion

Zunächst folgt die Feature-Extraktion, welche ein entscheidender Prozess ist, um eine effiziente Darstellung von Textdaten zu erhalten. Im Rahmen dieser Arbeit werden drei Ansätze aus unterschiedlichen Bereichen ausgeführt:

- Ein TF-IDF-Modell wird auf die vorverarbeiteten Textelemente angewendet, um die Wörter in diesen Texten entsprechend zu gewichten
- Es werden zwei vorab trainierte Feature-Extraktionsmodelle (Word2Vec/BERT) verwendet, um Vektordarstellungen von den Textelementen direkt zu erstellen

Aus der Vielzahl der verfügbaren Methoden wurde TF-IDF ausgewählt, da es sich um eine beliebte und aktuelle Strategie im Bereich Text-Mining handelt, die den Extraktionsprozess erheblich vereinfacht. Allerdings wird beim Extrahieren von Wörtern mit TF-IDF jedes Wort im Text als separate Einheit behandelt, sodass die semantischen Merkmale des Textes nicht effektiv erhalten werden können.

Word-Embeddings hingegen erzeugt eine Wortvektordarstellung, die syntaktische und semantische Aspekte erfasst [22]. Daher besitzen Wörter, die eine ähnliche Bedeutung und eine enge Korrelation haben, auch eine ähnliche Wortvektordarstellung. Viele frühere Studien zur Verarbeitung natürlicher Sprache (NLP) haben festgestellt, dass die Darstellung von Worteinbettungen die NLP-Aufgabenleistung erhöht. [22]. Die Verwendung der Worteinbettung hat deswegen sicherlich einen Vorteil für den Dokumenten-Clustering-Prozess, da die gleichen Wörter in Dokumenten mit ähnlichem Thema dazu neigen, eine ähnliche Wortvektordarstellung zu haben.

BERT wurde letztendlich ausgewählt, da dieses Modell in der Lage ist, hochgradig kontextspezifische Repräsentationen von Wörtern zu erstellen, wodurch es möglich wird, korrekte Repräsentationen für Wörter zu generieren, die in mehreren Kontexten im selben Korpus existieren [11]

5.1.3 Clustering

In diesem Schritt werden die ausgewählten Clustering-Algorithmen erläutert und basierend auf die früheren Schritte angewendet. Aus der breiten Auswahl gängiger Clustering-Verfahren und ihren entsprechenden Derivaten wurden im Rahmen dieser Arbeit zwei unterschiedliche Verfahren zur Anwendung und Evaluation gewählt: 1. K-Means-Algorithmus 2. DBScan-Algorithmus, die bereits im Kapital 3.2.2 in ihrer theoretischen Funktionsweise und ihren Eigenschaften erklärt wurden.

Das Motiv zur Wahl dieser zwei Algorithmen aus der Menge der gängigen Verfahren ist unter anderem darin zu finden, dass sich jedes dieser Verfahren im Konzept seiner Funktionsweise grundlegend von den anderen unterscheidet. Für jedes der zwei Verfahren existieren zudem Erweiterungen bzw. Derivate, z.B. der K-Median-Algorithmus, K-Means++Algorithmus, die in ihrer Arbeitsweise und in ihren Ergebnissen meist der Grundvariante ähneln. Deshalb ist anzunehmen, dass die Anwendung und Evaluation dieser Grundvarianten auch repräsentiv für diese Derivate sein könnten und damit eine Einschätzung ihrer möglichen Eingung zulässt.

Die Beschränkung auf nur zwei Verfahren ergibt sich außerdem daraus, dass der Schwerpunkt dieser Arbeit auf die Anwendung und Evaluation verschiedener Feature-Extractions-Verfahren gelegt wurde. So ergibt sich durch die Anwendung der zwei Algorithmen auf alle Kombinationen der Features bereits eine hohe Anzahl von Durchführungen, deren Ergebnisse manuell evaluiert werden müssen, weswegen es im Rahmen dieser Arbeit nicht möglich ist, alle der möglichen Verfahren anzuwenden.

5.1.4 Bewertungsmetriken

Die Cluster-Evaluierungsmessung eines Clustering-Algorithmus ist ebenso wichtig wie der Algorithmus selbst. Eine Clustering-Evaluation erfordert ein unabhängiges und zuverlässiges Maß zur Bewertung und zum Vergleich von Clustering-Experimenten und -Ergebnissen. Außerdem sollte bei der Cluster-Auswertung keine Bewertungsmetrik verwendet, die die absoluten Werte der Clusterbezeichnungen berücksichtigt. Die Cluster-Evaluierung dient jedoch dazu, Trennungen der Daten zu definieren, d. h. Mitglieder, die zu dem selben Cluster gehören, sind gemäß einer Ähnlichkeitsmetrik ähnlicher als Mitglieder verschiedener Clusters. Beim Clustering von Dokumenten gibt es zwei Arten von Methoden, nämlich intrinsische und extrinsische Maße. In dieser Arbeit wurden folgende Maße verwendet:

- Silhouette-Coefficient
- Adjusted Rand Index

- Adjusted Mutual Information
- V-Measure einschließlich Homogenity und Completeness

In Kapital 6.1 wird auf die Arten, Definitionen und Formeln dieser Maße näher eingegangen.

5.2 Technische Implementierung

Python gilt unter Forschern als die beste Wahl für die Analyse von Big Data. Sie bietet eine umfangreiche Sammlung von NLP-Tools und -Bibliotheken, die es Entwicklern ermöglichen, eine große Anzahl von NLP-bezogenen Aufgaben zu erledigen. Darüber hinaus bekommen Entwickler eine hervorragende Unterstützung für die Integration mit anderen Sprachen und Tools, die sich für Techniken wie maschinelles Lernen als nützlich erweisen. Aus diesem Grund wurden alle Experimente in Python 3 mithilfe mehrerer Bibliotheken implementiert, die allgemein über den pip-Befehl auf der Shell installiert werden können.

5.2.1 Python-Bibliotheken

Zum Aufbauen der geplanten Experimente werden die folgenden Standardbibliotheken von Python verwendet:

Gensim

Genism [50] steht für Generate Similarity. Es ist eine Open-Source-Python-Bibliothek für die unüberwachte Themenmodellierung und die Verarbeitung natürlicher Sprache unter Verwendung modernen statistischen maschinellen Lernens. Gensim wurde entwickelt, um große Textsammlungen mithilfe von Datenstreaming und inkrementellen Online-Algorithmen zu verarbeiten, was es von den meisten anderen Softwarepaketen für maschinelles Lernen unterscheidet, die nur auf In-Memory-Verarbeitung abzielen.

NumPy

NumPy steht für Numerical-Python[18] und ist eine Python-Bibliothek, die eine einfache Handhabung von Vektoren, Matrizen oder generell großen mehrdimensionalen Arrays ermöglicht. Neben den Datenstrukturen bietet NumPy auch effizient implementierte Funktionen für numerische Berechnungen an.

Scikit-learn

Scikit-learn [30] ist eine Bibliothek in Python, die viele unüberwachte und überwachte Lernalgorithmen bereitstellt und wird zusammen mit NumPy eingesetzt.

NLTK

Das Natural Language Toolkit [5] ist ein Toolkit für die Arbeit mit NLP in Python. Es stellt eine benutzerfreundliche Schnittstellen zu über 50 Korpora zusammen mit einer Reihe von verschiedene Textverarbeitungsbibliotheken zur Verfügung. Mit NLTK können eine Vielzahl von Aufgaben ausgeführt werden, z. B. Tokenisierung, Stemming, Lemmatisieren Wrapper für industrietaugliche NLP-Bibliotheken usw.

Pandas

Pandas [23] ist eine weit verbreitete Python-Bibliothek zur Datenanalyse und -manipulation. Sie bietet zahlreiche Funktionen und Methoden, die die Datenanalyse und die Vorverarbeitungsschritte beschleunigen z. B. Importieren und Speichern von Daten, Bereinigen, Normalisieren, Visualisieren usw.

5.2.2 Feature-Extraktion-Modelle

Als Nächstes werden die technische Aspekte diskutiert, die bei der Umsetzung der Feature-Extraktion-Modelle aufgesetzt wurden.

TF-IDF

Für die Implementierung von TF-IDF wurde die TfidfVectorizer-Klasse aus der Scikit-Learn-Bibliothek eingesetzt.

Word2Vec

Wie bereits erwähnt, wurde Für die Word2Vec-Einbettung ein vorab trainiertes Modell von Google verwendet. Dabei wird die Gensim-Bibliothek zum Importieren gebraucht.

Bert

Für die Nutzung von Bert, wurde das vorab trainierte BERT-Modell von Huggingface [29] aus dem sentence-transformers-Repository verwendet. Dieses Repository ermöglicht das

Trainieren und Verwenden von Transformer-Modellen zum Generieren von Satz- und Texteinbettungen.

5.2.3 Vorbereitung und Durchfühung des Clusterings

Die entstehenden Vektoren aus jeder Feature-Extraction-Methode bilden die Voraussetzung zur Anwendung der Clustering-Algorithmen. Für die Implementation des K-Means-Algorithmus wurde die KMeans-Klasse aus dem Cluster-Modul von der Scikit-Learn-Bibliothek verwendet. Dabei ist die Angabe der Anzahl der zu bildenden Cluster einzugeben. Dies wurde basierend auf Domänenkenntnissen der drei verwendeten Datensätze gewählt. Es kann auch eine von einer Vielzahl von Techniken verwendet werden, um eine „gute“ Anzahl von Clustern zu finden, wie z.B. die Verwendung der Gap-Statistik oder der Elbow-Methode. Die Elbow-Methode ist eine einfache, aber effektive Möglichkeit, den $k - Parameter$ abzustimmen, falls keine Kenntnisse über die Datensätze vorliegen.

DBScan wurde implementiert durch die bereitgestellte Sklearn-Cluster-API aus der Scikit-Learn-Bibliothek. Im Gegensatz zu dem K-Means-Algorithmus benötigt der DBScan-Algorithmus keine Angabe der zu bildenden Cluster. Jedoch besteht bei diesem Algorithmus die Herausforderung, eine geeignete Wahl für die Parameter eps und $minPts$ zu bestimmen. Für die Wahl des Parameters eps kann ein k-nearest-neighbour-Histogramm erstellt werden. Dafür werden für jedes Objekt die Abstände der k-nächsten Nachbarn ermittelt. Die ermittelten Abstände werden dann in aufsteigender Reihenfolge in einem Diagramm dargestellt. Für die Ermittlung der k-nächsten Nachbarn ist die Angabe eines k nötig, die dem Wert $minPts$ entsprechen sollte. In dem erstellten Diagramm sollte ein scharfer Übergang zu sehen sein, der eine gute Wahl für eps darstellt. In Abbildung 5.2 ist ein solches Diagramm zu sehen. Dort ist ein Richtwert von ungefähr 0.08 für eps zu sehen. Für eine gute Wahl des Parameters $minPts$ sollte mit verschiedenen Werten experimentiert werden. Darüber hinaus wurde vor der Durchführung des DBScan-Algorithmus eine Dimensionsreduktion durch Anwendung der Hauptkomponentenanalyse (PCA) ausgeführt. Eine Dimensionsreduktion kann eine starke Unterstützung zur Verbesserung der Clustervalidität in Clustering-Algorithmen wie DBScan bereitstellen. Dies ist in der Vergleichenden Untersuchung von [28] zu entnehmen.

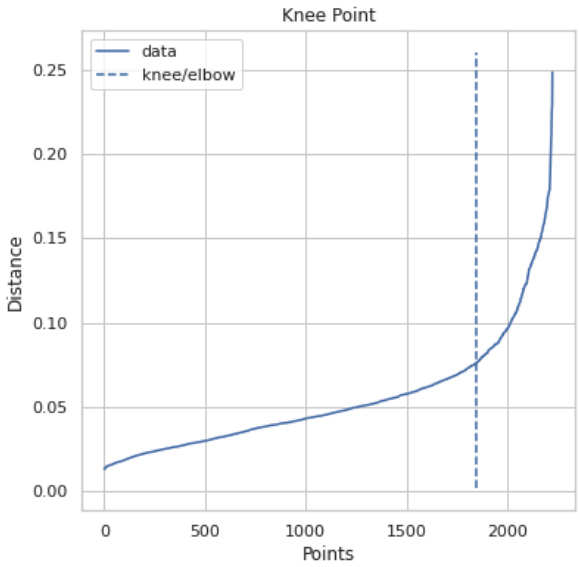


Abbildung 5.2: Ermittlung von Epsilon für DBScan.

6 Ergebnisse

In diesem Kapitel werden alle Clustering-Experimente und -Ergebnisse präsentiert und entsprechend diskutiert. Zunächst werden die zur Bewertung der Clustering-Algorithmen Leistung verwendeten Metriken vorgestellt. Dann wird ein Überblick über die Methodenleistung gegeben.

6.1 Qualitätsmetriken

Wie bereits erwähnt, werden für das Clustering zwei Maße für die „Güte“ oder Qualität des Clusters verwendet. Eine Art von Maß ermöglicht es, verschiedene Gruppen von Clustern ohne Bezugnahme auf externes Wissen zu vergleichen, dies wird als internes Qualitätsmaß bezeichnet. Mit dem zweiten Maß lässt sich bewerten, wie gut das Clustering funktioniert, indem die durch die Clustering-Techniken erzeugten Gruppen mit bekannten Klassen verglichen werden. Diese Art von Maßen wird als externe Qualitätsmaßnahme bezeichnet. Die Leistung und die relative Rangordnung verschiedener Clustering-Algorithmen können je nach verwendetem Qualitätsmaß erheblich variieren. Wenn jedoch ein Clustering-Algorithmus bei vielen dieser Maßnahmen besser abschneidet als andere Clustering-Algorithmen, kann darauf verlassen werden, dass es sich wirklich um den besten Clustering-Algorithmus für die zu bewertende Situation handelt.

6.1.1 Interne Qualitätsmaße

Intrinsische oder interne Qualitätsmaße geben an, wie gut ein Algorithmus eine bestimmte Darstellung optimiert hat. Intrinsische Vergleiche sind inhärent durch die gegebene Repräsentation begrenzt, mit anderen Worten abhängig von den Merkmalsdarstellungen, daher können intrinsische Vergleiche nicht zwischen verschiedenen Repräsentationen verglichen werden [48]. Intrinsische Maßnahmen berechnen die Clustertrennung und Kohäsion. Der Vorteil dieser Methode besteht darin, dass kein Ground-Truth-Label erforderlich ist. Das Beispiel dieser Methode ist der Silhouette-Koeffizient.

Silhouette-Koeffizient

Der Silhouette-Koeffizient[36] berechnet, wie ähnlich ein Objekt seinem eigenen Cluster (Kohäsion) im Vergleich zu den verschiedenen Clustern (Trennung) ist. Mit anderen Worten, der Silhouette-Koeffizient (S) wird unter Verwendung des mittleren Intra-Cluster-Abstands (a) und des mittleren Abstands des nächsten Clusters (b) für jede Probe in dem Cluster (C) berechnet. Der beste Wert ist 1 und der schlechteste Wert ist -1. Werte nahe 0 weisen auf überlappende Cluster hin. Negative Werte weisen im Allgemeinen darauf hin, dass eine Probe dem falschen Cluster zugeordnet wurde, da ein anderer Cluster ähnlicher ist.

6.1.2 Externe Qualitätsmaße

Das extrinsische oder externe Maß ist in der Lage, das Clustering-Ergebnis aus verschiedenen Methoden der Naturrepräsentation zu vergleichen. Diese Qualitätsmaßstäbe vergleichen ein Clustering mit einer externen Wissensquelle, wie z. B. Ground-Truth-Labels. Folgende Maße fallen darunter:

Adjusted-Rand-Index

Das Rand-Index [33] berechnet ein Ähnlichkeitsmaß zwischen zwei Clustern, indem es alle Musterpaare berücksichtigt und Paare zählt, die in den vorhergesagten und wahren Clusterings demselben oder unterschiedlichen Clustern zugeordnet sind. Die Formel des Rand-Index lautet:

$$RI = \frac{\text{Correct Similar Pairs + Correct Dissimilar Pairs}}{\text{Total Similar Pairs}} \tag{6.1}$$

Der RI-Rohwert wird dann unter Verwendung des folgenden Schemas für den Zufall in den ARI-Wert angepasst:

$$ARI = \frac{RI - \text{Expected RI}}{\text{Max(RI)} - \text{Expected RI}} \tag{6.2}$$

Die ARI-Werte liegen zwischen null und eins, wobei null einer zufälligen Kennzeichnung entspricht und eins identische Partitionen darstellt.

Adjusted-Mutual-Information

Adjusted-Mutual-Information ist ein Maß für die Ähnlichkeit zwischen zwei Labels mit denselben Daten. Wo |Ui|, die Anzahl der Samples in Cluster Ui ist, und |Vj|, die Anzahl der Proben

in Cluster Vj ist, wird die gegenseitige Information zwischen den Clustern U und V wie folgt angegeben:

$$MI(U,V) = \sum_{i=1}^{R}\sum_{j=1}^{C} Puv(i,j)\log\frac{Puv(i,j)}{Pu(i)Pv(j)} \tag{6.3}$$

Adjusted-Mutual-Information [47] korrigiert den Effekt der Übereinstimmung ausschließlich aufgrund des Zufalls zwischen Clusterings, ähnlich wie der angepasste Rand-Index den Rand-Index korrigiert.

Das adjusted-Maß für die gegenseitige Information kann dann wie folgt definiert werden:

$$AMI = \frac{MI(U,V) - E\{MI(U,V)\}}{max\{H(U),H(V)\} - E\{MI(U,V)\}} \tag{6.4}$$

Die Maße ARI und AMI werden häufig verwendet, um Clusterings desselben Datensatzes zu vergleichen und sind in der Clustering-Community sehr beliebt[34].

In [34] wurde die Verwendung von ARI als externe Validierungsindizes vorgeschlagen, wenn das Referenz-Clustering die gleiche Größe von Clusters aufweist. Beim unbalancierten Referenz-Clustering ist AMI zu verwenden. Da in dieser Arbeit drei unterschiedliche Datensätze in Betracht genommen werden, wird auf beide Maße ARI und AMI zurückgegriffen.

Homogeneity

Ein Clustering-Ergebnis erfüllt Homogeneity, wenn alle seine Cluster nur Datenpunkte enthalten, die Mitglieder einer einzigen Klasse sind[35]. Es ist definiert als:

$$h = 1 - \frac{H(Ytrue|Ypred)}{H(Ytrue)} \tag{6.5}$$

Es ist zwischen 0 und 1 begrenzt, wobei niedrige Werte eine geringe Homogenität anzeigen. Tatsächlich wird H(Ytrue|Ypred) kleiner (h $\rightarrow$ 1) und umgekehrt, wenn die Kenntnis von Ypred die Unsicherheit von Ytrue reduziert.

Completeness

Das Ergebnis von Completeness dient dazu, eine Information über die Zuordnung von Proben derselben Klasse zu geben[35]. Genauer gesagt sollte ein guter Clustering-Algorithmus alle

Samples mit demselben True-Label demselben Cluster zuweisen. Die Definition ist symmetrisch zum Homogenität-Score:

$$c = 1 - \frac{H(Ypred|Ytrue)}{H(Ypred)} \tag{6.6}$$

Wenn H(Ypred|Ytrue) niedrig ist (c $\rightarrow$ 1), bedeutet dies, dass die Kenntnis der Grundwahrheit abnimmt.

V-measure

Das V-measure ist ein entropie-basiertes Maß, das explizit misst, wie erfolgreich die Kriterien Homogeneity und Completeness erfüllt wurden. Das V-Maß wird als harmonischer Mittelwert unterschiedlicher Homogeneity- und Completeness-Bewertungen berechnet[35]. Damit ist das gewichtete V-Maß wie folgt gegeben:

$$v = \frac{(1+\beta) \times h \times c}{(\beta \times h + c)} \tag{6.7}$$

Der V-Measure-Wert wird im Rahmen dieser Arbeit ebenfalls zur Bewertung beider Modelle, K-Means und DBScan eingesetzt. Somit lässt sich überprüfen, ob die gebildeten Gruppen tatsächlich aussagekräftig sind.

Für die Implementierung der oben verwendeten Metriken wurde die Funktion *sklearn.metrics* aus der Scikit-learn-Bibliothek verwendet.

6.2 Methodenleistung

In den Tabellen 6.1, 6.2 bis 6.3 sind alle Messwerte bezüglich des jeweiligen Datensatzes mit allen möglichen Kombinationen dargestellt. Somit lässt sich die Leistung beider Algorithmen, K-Means und DBScan, hinsichtlich aller verwendeten Feature-Extraktionsmethoden anschaulicher vergleichen und interpretieren. In den nächsten Schritten werden zur Bewertung und Diskussion der Algorithmen-Leistung ausschließlich die AMI-, V-Measure-Metriken hervorgehoben.

Erster Versuch - Datensatz 1

Wenn man sich die Ergebnisse des ersten Datensatzes 6.1 ansieht, kann man sofort erkennen, dass beide Algorithmen, K-Means und DBScan, in Kombination mit Word2Vec ziemlich gute Ergebnisse erzielen konnten. Dabei betrugen die AMI-Werte 0.849 bei K-Means und 0.515

bei DBScan. Die V-Measure-Werte lagen hingegen bei 0.820 und 0.590. Diese Werte zeigen, dass K-Means hierbei eine bessere Leistung erbracht hat. Darüber hinaus hat K-Means in Kombination mit BERT und TF-IDF einen deutlich besseren Leistungsunterschied als DBScan aufweisen können.

Zweiter Versuch - Datensatz 2

Bei dem zweiten Datensatz konnten auch beide Algorithmen in Kombination mit Word2Vec gute Leistungen erbringen. Diesmal hat DBScan jedoch einen höheren AMI-Wert aufgewiesen, mit 0,675. Wobei der AMI-Wert für K-Means bei 0,562 lag. Allerdings hat K-Means im Vergleich zu DBScan wieder, sowohl mit BERT als auch mit TF-IDF deutlich bessere Werte aufgezeigt. Dabei lag der AMI-Wert der Kombination K-Means & BERT bei 0.584, während der AMI-Wert der Kombination DBScan & BERT nur bei 0.177 lag.

Dritter Versuch - Datensatz 3

Für den dritten Datensatz wurde der höchste Ergebniswert von K-Means und TFIDF erzielt. Beide Werte, AMI und V-Measure, betrugen dabei 0.334. Außerdem hat die Word2Vec-Methode sowohl mit dem K-Means- als auch mit dem DBScan-Algorithmus im Vergleich zu der erbrachten Leistung bei dem ersten und zweiten Datensatz wenig gute AMI- bzw. V-Measure Werte erzielt.

Ein tieferer Einblick in die Struktur der Datensätze und die gebildeten Clusters nach Durchführung aller Methoden-Kombinationen sind in Abbildungen 6.1, 6.2 bis 6.3 veranschaulicht.

6.3 Diskussion

Anhand der durchgeführten Experimente ist daraus zu schließen, dass die Clustering-Leistung sowohl von der Auswahl eines für das Clustering gut strukturierten Datensatzes als auch einer geeigneten Feature-Extraktionsmethode abhängt.

Ein gutes Beispiel lässt der erste Datensatz darstellen. Zum einen bilden die in diesem Datensatz enthaltenen Dokumente abgrenzbare Gruppen, die es einem Clustering-Algorithmus generell ermöglichen, ähnliche Dokumente besser und effizienter zu erkennen. Zum anderen ist die Word2Vec-Methode in der Lage, die semantische Bedeutung von Dokumenten zu erfassen und somit die wichtigsten Merkmale daraus zu extrahieren. Diese beiden Faktoren haben in dem ersten Versuch dazu beigetragen, dass K-Means in Kombination mit Word2Vec den besten Bemessungswert aufgewiesen hat. Allerdings lässt sich hier feststellen, dass DBScan mit TF-IDF

und BERT trotz gut strukturiertem Datensatz unerwartet gescheitert hat. Das könnte daran liegen, dass weitere Hyperparameter-Einstellungen in Betracht genommen werden sollten.

Im Gegensatz dazu sind beiden Algorithmen, K-Means und DBScan, nicht gelungen, den dritten Datensatz in gut voneinander separierte Cluster einzuteilen, wie in Abbildungen 6.3e und 6.3f zu erkennen ist. Dies hängt stark von der unklaren Datenverteilung im Vektorraum ab, wie in Abschnitt 4.1 bereits beschrieben wurde. Basierend auf den erzielten Ergebnissen im dritten Datensatz, und aufgrund der Tatsache, dass TF-IDF sehr viele Merkmale erfassen und damit einen spärlichen Vektorraum bilden kann, lässt sich daraus schließen, dass die Kombination K-Means & TF-IDF in diesem Fall am besten geeignet ist.

Abschließend haben die Experimente gezeigt, dass der K-Means-Algorithmus unerwarteterweise bessere Leistung als DBScan in den meisten Kombinationen erzielt hat. Das könnte darauf zurückzuführen sein, dass DBScan mit spärlichen Datensätzen bzw. Datensätzen mit unterschiedlicher Dichte nicht sehr gut funktioniert.

		Feature-Extraction	ARI	AMI	Homogenity	Completeness	V-measure	Silhouette
Dataset 1	K-Means	TF-IDF	0.530963	0.657942	0.621460	0.700822	0.658759	0.010970
		Word2vec	0.849370	0.820304	0.820673	0.820743	0.820708	0.424512
		BERT	0.474095	0.533532	0.525597	0.543918	0.534600	0.082686
	DBScan	TF-IDF	0.178907	0.291137	0.235543	0.384489	0.292126	0.024085
		Word2vec	0.515713	0.588687	0.613381	0.568828	0.590265	0.200645
		BERT	0.166026	0.237659	0.222529	0.262803	0.240995	-0.133354

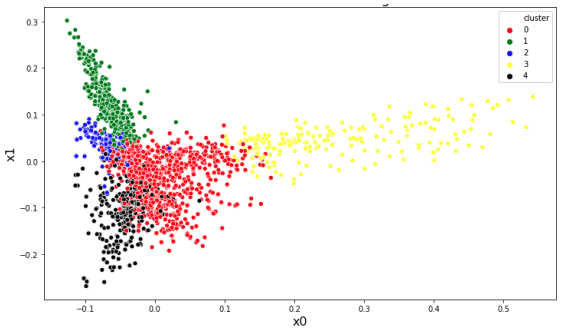
Tabelle 6.1: Vergleich der Leistung von K-menas und DBScan für den ersten Datensatz.

		Feature-Extraction	ARI	AMI	Homogenity	Completeness	V-measure	Silhouette
Dataset 2	K-Means	TF-IDF	0.156679	0.359097	0.288601	0.476395	0.359448	0.008628
		Word2vec	0.419821	0.562613	0.547258	0.579273	0.562811	0.381164
		BERT	0.603377	0.584443	0.589148	0.580170	0.584624	0.080522
	DBScan	TF-IDF	0.16936	0.209801	0.15908	0.308665	0.209954	0.264182
		Word2vec	0.683695	0.675326	0.746772	0.616794	0.675588	0.328014
		BERT	0.079965	0.177898	0.185498	0.173983	0.179556	-0.354316

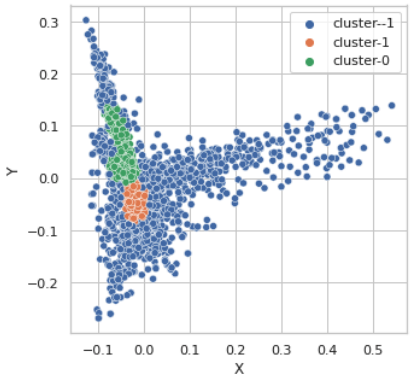
Tabelle 6.2: Vergleich der Leistung von K-menas und DBScan für den zweiten Datensatz.

		Feature-Extraction	ARI	AMI	Homogenity	Completeness	V-measure	Silhouette
Dataset 3	K-Means	TF-IDF	0.196918	0.334325	0.301744	0.375589	0.334641	0.002228
		Word2vec	0.010508	0.014681	0.014782	0.015457	0.015112	0.515549
		BERT	0.221564	0.249993	0.246042	0.254747	0.250319	0.060538
	DBScan	TF-IDF	0.044408	0.074958	0.052164	0.134335	0.075148	0.533367
		Word2vec	0.015236	0.014141	0.012442	0.01728	0.014468	0.654443
		BERT	0.057378	0.071363	0.058744	0.094174	0.072355	-0.297151

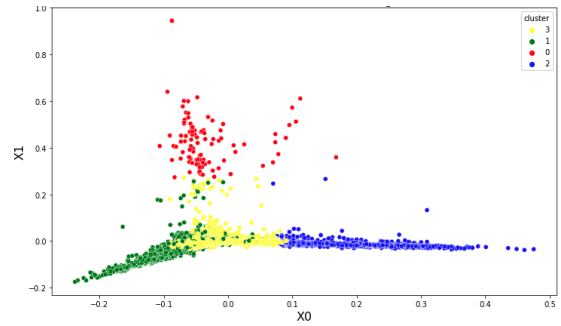
Tabelle 6.3: Vergleich der Leistung von K-menas und DBScan für den dritten Datensatz.



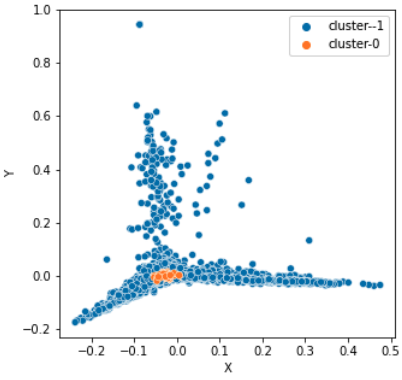
(a) Dataset1 with K-Means & TF-IDF



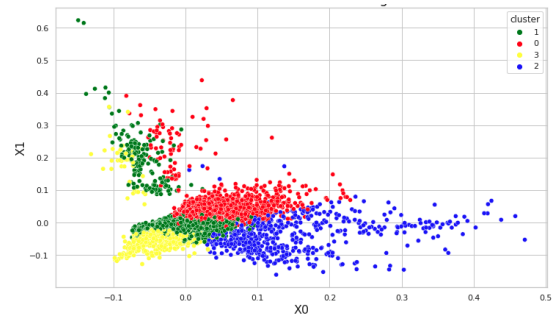
(b) Dataset1 with DBScan & TF-IDF



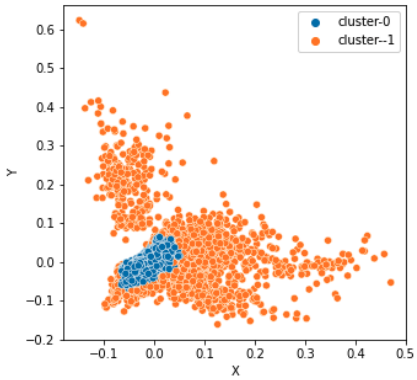
(c) Dataset2 with K-Means & TF-IDF



(d) Dataset2 with DBScan & TF-IDF

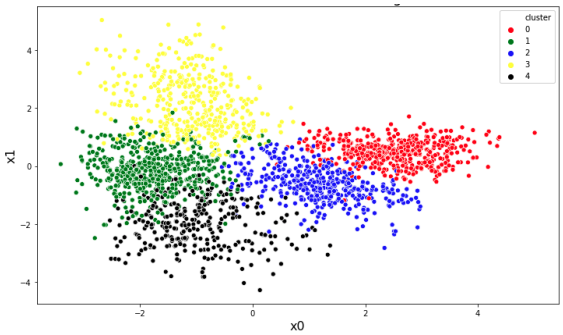


(e) Dataset3 with K-Means & TF-IDF

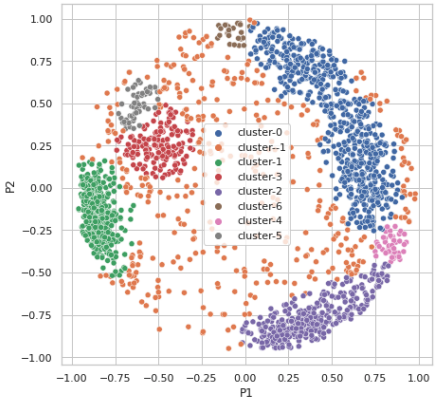


(f) Dataset3 with DBScan & TF-IDF

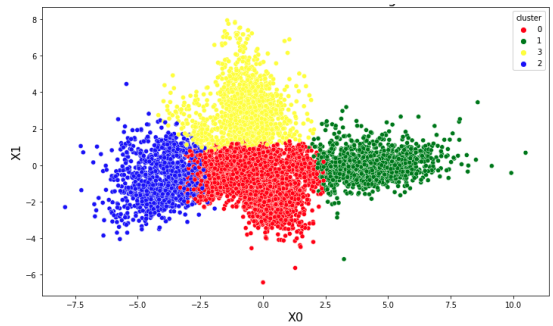
Abbildung 6.1: Datenverteilung für alle Datensätze mit TF-IDF.



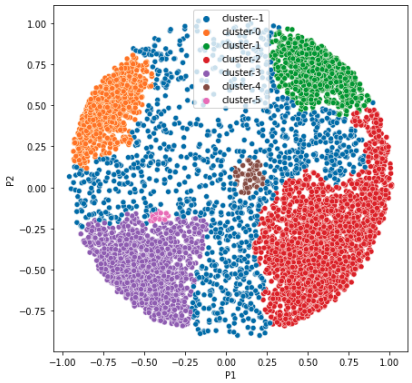
(a) Dataset1 with K-Means & Word2Vec



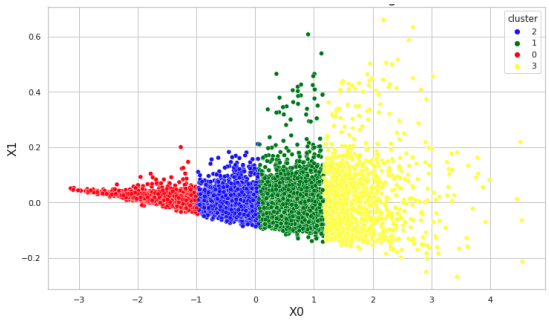
(b) Dataset1 with DBScan & Word2Vec



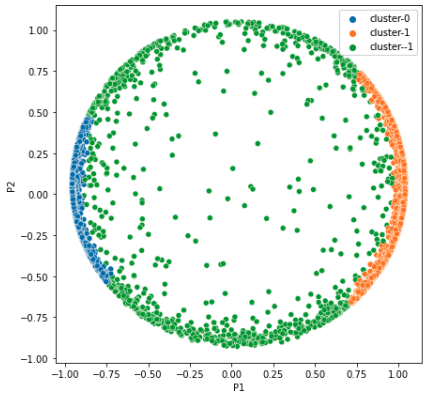
(c) Dataset2 with K-Means & Word2Vec



(d) Dataset2 with DBScan & Word2Vec

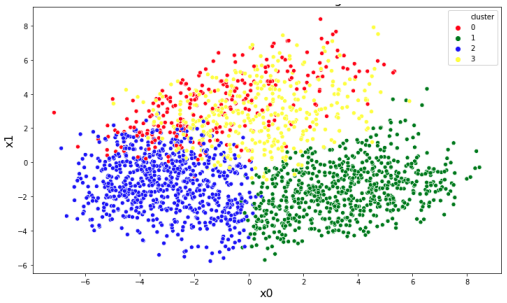


(e) Dataset3 with K-Means & Word2Vec

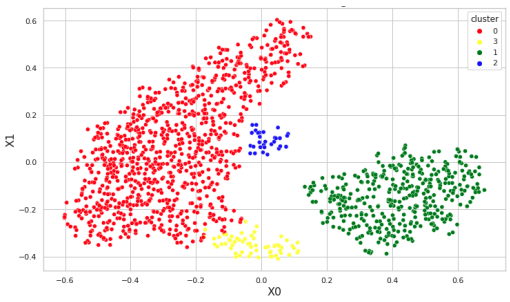


(f) Datasete with DBScan & Word2Vec

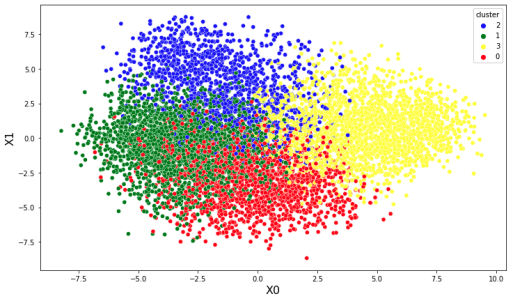
Abbildung 6.2: Datenverteilung für alle Datensätze mit Word2Vec.



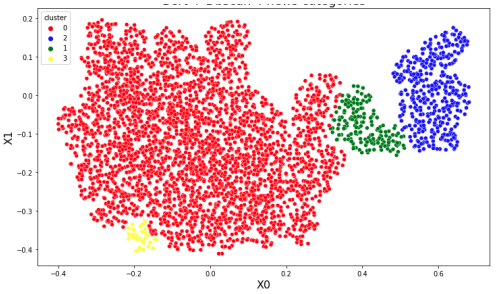
(a) Dataset1 with K-Means & Bert



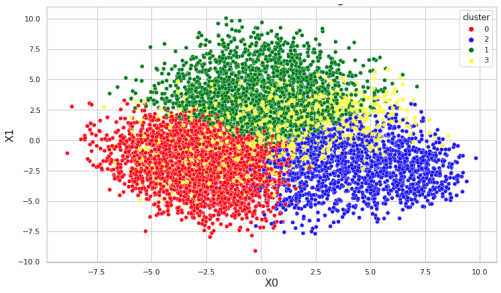
(b) Dataset1 with DBScan & Bert



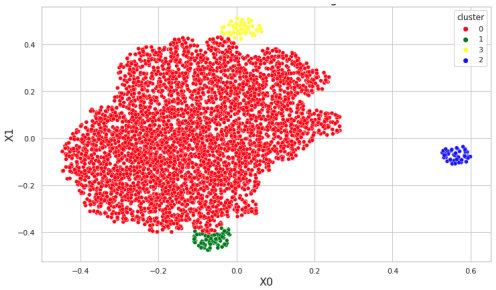
(c) Dataset2 with K-Means & Bert



(d) Dataset2 with DBScan & Bert



(e) Dataset3 with K-Means & Bert



(f) Dataset3 with DBScan & Bert

Abbildung 6.3: Datenverteilung für alle Datensätze mit Bert.

7 Abschlussbetrachtung

Die Abschlussbetrachtung beinhaltet eine Zusammenfassung in Bezug auf die in dieser Arbeit verwendete Vorgehensweise und die erreichten Ergebnisse. Zudem wird ein Ausblick auf mögliche weiterführende Untersuchungen gegeben.

7.1 Zusammenfassung

In dieser Forschungsarbeit wurde die Leistung zwei Clustering-Algorithmen, K-Means und DBSCAN, in Kombination mit drei verschiedenen Feature-Extraktionstechniken, TF-IDF, Word2Vec und BERT, bewertet. Sie wurden anhand drei unterschiedlicher Datensätze in Bezug auf deren Datenstruktur verglichen.

Es ist anhand der vorliegenden Arbeit zu erkennen, dass vor der Anwendung von ausgewählten Clustering-Algorithmen zunächst eine Datensatzanalyse sowie ein systematischer Datenvorverarbeitungsprozess zu erfolgen hat. Somit wird gewährleistet, dass ein tiefgreifendes Datenverständnis vorliegt und der Clustering-Mechanismus aufgrund der gesteigerten Datenqualität effektiver arbeitet. Die Auswahl einer geeigneten Feature-Extraxionsmethode ist einer der wesentlichen Schritte beim Dokument-Clustering, da sie die Leistung des Clusterings stark beeinflusst. Eine ausgezeichnete Feature-Extraxionsmethode führt zu einem guten Clustering-Ergebnis, selbst wenn nur ein einfacher Clustering-Algorithmus wie K-Means verwendet wird.

7.2 Ausblick

Aufgrund des breiten Feldes, das das Dokument-Clustering und seine zahlreichen Anwendungen abdeckt, wären eine Vielzahl von möglichen Weiterentwicklungen und Anwendungsszenarien dafür denkbar.

Neben den zwei verwendeten Clustering-Algorithmen gibt es eine Vielzahl weiterer Algorithmen sowie ihrer Derivate und Erweiterungen. Allein werden viele weitere Algorithmen in der Scikit-learn-Bibliothek angeboten, wie “Mean-Shift“, “Spectral-Clustering“, “Gaussian-Mixtures“ und “Birch“. Wie bereits erwähnt, existieren zu diesen Grundtypen auch noch

Derivate und Erweiterungen, die meist ähnliche, aber dennoch unterschiedliche Ergebnisse liefern und oft für bestimmte Anwendungen entwickelt oder optimiert wurden. Auf Basis dieser Erkenntnisse wäre also eine Verbesserung der Genauigkeit, selbst im Bereich des Dokument-Clusterings, zu erwarten.

Ein weiterer vielversprechender Ansatz könnte die Kombination der eingesetzten Clustering-Algorithmen sein. So könnte zunächst eine Sammlung aller zugehörigen Algorithmen-Clustering-Ausgaben erfolgen, dann lässt sich als endgültiger Optimum den Fall auswählen, in dem die Ergebnisse der beiden Algorithmen am ähnlichsten sind. Andere ergänzende Verwendungen sind auch denkbar, wie die Ermittlung der Anzahl der möglichen Cluster durch DBScan und die Nutzung dieser Information für andere Algorithmen, die eine Angabe oder zumindest Vermutung der Clusteranzahl benötigen.

Eine weitere Möglichkeit besteht darin, eine Data-Mining-Software wie *RapidMiner* bzw. *Knime* zur Bearbeitung der vorliegenden Aufgabenstellung zu verwenden und die Ergebnisse damit zu vergleichen.

Literaturverzeichnis

[1] Anand, Aman: AG News Classification Dataset. (2015). – URL https://www.kaggle.com/amananandrai/ag-news-classification-dataset?select=test.csv

[2] A.Runkler, Thomas: *Data Mining Modelle und Algoorithmen intelligenter Datenanalyse*. Springer, 2015

[3] Bankhofer, Udo ; Vogel, Jürgen: *Datenanalyse und Statistik*. Springer, 2008

[4] Bengfort, Benjamin ; Bilbro, Rebecca ; Ojeda, Tony: *Applied Text Analysis with Python*. O'Reilly Media, Inc., 2018

[5] Bird, Steven ; Loper, Edward: *NLTK: the Natural Language Toolkit*. In Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions, 2004

[6] Burkov, Andriy: *The Hundred-Page Machine Learning Book*. 2019

[7] C.Aggarwal, Charu: *Data Mining The Textbook*. Springer, 2015

[8] Chakraborty, G. ; Pagolu, Murali ; Garla, S.: Text Mining and Analysis: Practical Methods, Examples, and Case Studies Using SAS. (2013)

[9] Cleve, Jürgen ; Lämmel, Uwe: *Data Mining*. De Gruyter Oldenbourg, 2014

[10] dev, Yufeng: BBC articles fulltext and category. (2006). – URL https://www.kaggle.com/yufengdev/bbc-fulltext-and-category

[11] Devlin, J. ; Chang, M. W. ; Lee, K. ; Toutanova, K. B.: Pre-training of Deep Bidirectional Transformers for Language Understanding. (2018)

[12] Fayyad, Usama ; Piatetsky-Shapiro, Gregory ; Smyth, Padhraic: From Data Mining to Knowledge Discovery in Databases. In: *AI Magazine, 17(3), 37* (1996)

[13] Fayyad, Usama ; Piatetsky-Shapiro, Gregory ; Smyth, Padhraic: Knowledge Discovery and Data Mining. In: *KDD-96 Proceedings* (1996)

[14] Feldman, Ronen ; Dagan, Ido: Knowledge Discovery in Textual Databases (KDT). (1995)

[15] Feldman, Ronen ; Sanger, James: *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data.* Cambridge University Press, 2009

[16] Gupta, Akash: News Category Dataset. (2020). – URL https://www.kaggle.com/akash14/news-category-dataset?select=Data_Train.csv

[17] Han, Jiawei ; Kamber, Micheline ; Jian: *Data Mining: Concepts and Techniques 3rd Edition.* Morgan Kaufmann, 2019

[18] Harris, Charles R. ; Millman, K. J. ; Walt, St'efan J. van der ; Gommers, Ralf ; Virtanen, Pauli ; Cournapeau, David ; Wieser, Eric ; Taylor, Julian ; Berg, Sebastian ; Smith, Nathaniel J. ; Kern, Robert ; Picus, Matti ; Hoyer, Stephan ; Kerkwijk, Marten H. van ; Brett, Matthew ; Haldane, Allan ; R'io, Jaime F. del ; Wiebe, Mark ; Peterson, Pearu ; G'erard-Marchant, Pierre ; Sheppard, Kevin ; Reddy, Tyler ; Weckesser, Warren ; Abbasi, Hameer ; Gohlke, Christoph ; Oliphant, Travis E.: Array programming with NumPy. In: *Nature* 585 (2020), S. 357–362. – URL https://doi.org/10.1038/s41586-020-2649-2

[19] Kadhim1, Ammar I. ; Cheah, Yu-N ; Ahamed, Nurul H.: Text Document Preprocessing and Dimension Reduction Techniques for Text Document Clustering. (2014)

[20] Kumar, Lokesh ; Bhatia, Parul K.: TEXT MINING: CONCEPTS, PROCESS AND APPLICATIONS. (2013)

[21] Lee, Sungjick ; Kim, Hanjoon: *News Keyword Extraction for Topic Tracking.* Bd. 2. In: Networked Computing and Advanced Information Management, 2008

[22] Li, Quanzhi ; Shah, Sameena ; Liu, Xiaomo ; Nourbakhsh, Armineh: Data Sets: Word Embeddings Learned from Tweets and General Data. (2017)

[23] McKinney, Wes: *Data Structures for Statistical Computing in Python.* In Stefan van der Walt and Jarrod Millman, editors, Proceedings of the 9th Python in Science Conference, 2010. – 51–56 S

[24] M.Decker, Karsten ; Focardi, Sergio: *Technology Overview: A Report on Data Mining.* CSCS-ETH, 1995

[25] Mikolov, Tomas ; Chen, Kai ; Corrado, Greg ; Dean, Jeffrey: Efficient Estimation of Word Representations in Vector Space. (2013)

[26] Miner, Gary D. ; Elder, John ; Fast, Andrew ; Hill, Thomas ; Nisbet, Robert ; Delen, Dursun: *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. Elsevier Science, 2012

[27] Mitchell, Tom M.: *Machine Learning*. Austin, TX, USA, 1997

[28] Mustakim ; Rahmi, Emi ; Mundzir, Mediantiwi R. ; Rizaldi, Said T. ; Okfalisa ; Maita, Idria: Comparison of DBSCAN and PCA-DBSCAN Algorithm for Grouping Earthquake Area. (2021)

[29] Online: HuggingFace's Transformers: State-of-the-Art Natural Language Processing. (2020). – URL https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens

[30] Pedregosa, F. ; Varoquaux, G. ; Gramfort, A. ; Michel, V. ; Thirion, B. ; Grisel, O. ; M. Blondel, P. P. ; Weiss, R. ; Dubourg, V. ; Vanderplas, J. ; Passos, A. ; Cournapeau, D. ; Brucher, M. ; Perrot, M. ; Duchesnay, E.: Scikit-learn: Machine learning in Python. In: *Journal of Machine Learning Research* (2011)

[31] Radu, Robert-George ; Radulescu, Iulia-Maria ; Truică, Ciprian-Octavian ; Apostol, Elena S. ; Mocanu, Mariana: Clustering Documents using the Document to Vector Model for Dimensionality Reduction. In: *Conference: 2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)* (2020)

[32] Radulescu, Iulia-Maria ; Truică, Ciprian-Octavian ; Apostol, Elena S. ; Boicea, Alexandru ; Mocanu, Mariana ; Popeangă, Daniel-Călin ; Rădulescu, Florin: Density-based Text Clustering using Document Embeddings. In: *Conference: The 36th International Business Information Management Association (IBIMA)At: Granada, Spain* (2020)

[33] Rand, William M.: Objective Criteria for the Evaluation of Clustering Methods. In: *Journal of the American Statistical Association* (1971)

[34] Romano, Simone ; Vinh, Nguyen X. ; Bailey, James ; Verspoor, Karin: Adjusting for Chance Clustering Comparison Measures. In: *Journal of Machine Learning Research 17 (2016) 1-32* (2016)

[35] Rosenberg, Andrew ; Hirschberg, Julia: V-Measure: A conditional entropy-based external cluster evaluation measure. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning

(EMNLP-CoNLL), pages 410–420, Prague, Czech Republic. Association for Computational Linguistics. In: *Association for Computational Linguistics* (2007)

[36] Rousseeuw, Peter J.: Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. In: *Computational and Applied Mathematics. 20, 1987, S. 53–65* (1987)

[37] Saif, Hassan ; Fernandez, Miriam ; He, Yulan ; Alani, Harith: On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter. (2014)

[38] Sarkar, Dipanjan: *Text Analytics with Python.* Apress, 2019

[39] Satapathy, Suresh C. ; Bhateja, Vikrant ; Das, Swagatam: *Smart Intelligent Computing and Applications.* Springer, 2019. – 51–56 S

[40] Sebastiani, Fabrizio: Machine learning in automated text categorization. (2002)

[41] Sharafi, Armin: *Knowledge Discovery in Databases.* Springer, 2012

[42] Silva, Catarina ; Ribeiro, Bernardete: The Importance of Stop Word Removal on Recall Values in Text Categorization. (2003)

[43] Solka, Jeffrey L.: Text Data Mining: Theory and Methods. (2008)

[44] Soo, Kenneth ; Delbrück(Übersetzer), Matthias: *Data Science – was ist das eigentlich?!: Algorithmen des maschinellen Lernens verständlich erklärt.* Springer, 2018

[45] Tekir, Selma ; Sezerer, Erhan: A Survey On Neural Word Embeddings. (2021)

[46] Ullman, Jeffrey D. ; Rajaraman, Anand ; Leskovec, Jure: *Mining of Massive Datasets.* Cambridge University Press, 2020

[47] Vinh, Nguyen X. ; Epps, Julien ; Bailey, James: Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. In: *Journal of Machine Learning Research* (2010)

[48] Vries, Christopher M. D. ; Geva, Shlomo ; Trotman, Andrew: Document Clustering Evaluation: Divergence from a Random Baseline. (2012). – URL arXiv:1208.5654

[49] Webster, Jonathan J. ; Kit, Chunyu: Text Data Mining: Theory and Methods. (1992)

[50] Řehůřek, Radim ; Sojka, Petr: *Software Framework for Topic Modelling with Large Corpora.* Valletta, Malta : In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, 2010

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „— bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] — ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen."

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: _____

Vorname: _____

dass ich die vorliegende Bachelorarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Eine vergleichende Untersuchung zum Clustering von Textdokumenten

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

_____	_____	_____
Ort	Datum	Unterschrift im Original