INSTITUT FÜR ARBEITSMARKT- UND BERUFSFORSCHUNG
Die Forschungseinrichtung der Bundesagentur für Arbeit

IAB-FORSCHUNGSBERICHT

Aktuelle Ergebnisse aus der Projektarbeit des Instituts für Arbeitsmarkt- und Berufsforschung

13|2025 Text Mining mit der „Temi-Box“

Christine Hirmer, Lina Metzger

Finanziert von der Europäischen Union
NextGenerationEU

ISSN 2195-2655

Text Mining mit der „Temi-Box“

Christine Hirmer (IAB),
Lina Metzger (IAB)

In der Reihe IAB-Forschungsberichte werden empirische Analysen und Projektberichte größeren Umfangs, vielfach mit stark daten- und methodenbezogenen Inhalten, publiziert.

The IAB Research Reports (IAB-Forschungsberichte) series publishes larger-scale empirical analyses and project reports, often with heavily data- and method-related content.

In aller Kürze

- Das Python-Paket Temi-Box ermöglicht die Bearbeitung komplexer Text Mining-Aufgaben ohne tiefgehende Programmierkenntnisse, mit einem Fokus auf Textklassifikation und -clustering.
- Die Kombination von vordefinierten und erweiterbaren Standardkomponenten mit Fachdomänen-Komponenten, die projektspezifische Informationen bündeln, vereinfacht die Implementierung und den Vergleich von Methoden erheblich. Diese Pipeline-Architektur schafft Flexibilität und macht die Temi-Box für eine breite Nutzergruppe von Einsteigerinnen und Einsteigern bis hin zu Expertinnen und Experten geeignet.
- Die Temi-Box nutzt TF-IDF (Term Frequency – Inverse Document Frequency) und BERT (Bidirectional Encoder Representations from Transformers) zur Textrepräsentation und bietet eine Vielzahl von Algorithmen für Textklassifikation und -clustering. Verschiedene Evaluationsmetriken ermöglichen einen systematischen Vergleich von Text Mining-Methoden.
- Experimente veranschaulichen die Anwendung der Temi-Box und zeigen, dass BERT-Modelle bessere Ergebnisse als TF-IDF-basierte Modelle erzielen können.
- Über die Verschlagwortung und Themenzuordnung von Publikationen hinaus ist die Temi-Box vielseitig einsetzbar, etwa bei der Analyse von Stellenanzeigen oder Berufs- und Unternehmensbeschreibungen, und erweiterbar um Frage-Antwort-Systeme und Named Entity Recognition.

Inhalt

Zusammenfassung

Die stetig wachsende Menge digital verfügbarer Textdaten und Fortschritte in der natürlichen Sprachverarbeitung (NLP) haben Text Mining zu einer Schlüsseltechnologie gemacht. Die „Temi-Box“ ist ein modularer Baukasten für das Text Mining, der die automatisierte Textklassifikation, Themenzuordnung und Clusterbildung erleichtert, ohne dass tiefgehende Programmierkenntnisse erforderlich sind. Entwickelt anhand der Verschlagwortung und Themenzuordnung von Publikationen für die IAB-Infoplattform und finanziert durch EU-Mittel, steht sie als Open-Source-Projekt zur Verfügung. Dieser Forschungsbericht dokumentiert die Entwicklung und Anwendung der Temi-Box, veranschaulicht ihre Nutzungsmöglichkeiten und interpretiert die erzielten Ergebnisse.

Text Mining extrahiert Wissen aus unstrukturierten Texten durch Methoden wie Klassifikation und Clustering. Die modular aufgebaute Temi-Box macht etablierte Methoden nutzerfreundlich zugänglich und unterstützt Anwenderinnen und Anwender durch eine Pipeline-Architektur, die standardisierte Prozesse wie Datenaufbereitung und Modelltraining vereinfacht. Sie integriert sowohl aktuelle als auch traditionelle Ansätze zur Textrepräsentation, wie BERT und TF-IDF, und bietet eine Vielzahl von Algorithmen zu Textklassifikation und -clustering, darunter K-Nearest Neighbors (KNN), binäre und multinomiale Klassifikatoren als Schichten in neuronalen Netzen sowie K-Means. Verschiedene Evaluationsmetriken ermöglichen es, die Leistung des Modells zu bewerten und unterschiedliche Ansätze miteinander zu vergleichen.

Experimente zur automatisierten Themenzuordnung und zur Identifikation von Themenschwerpunkten veranschaulichen die Nutzung der Temi-Box und die Interpretation der Ergebnisse. Auf Basis eines Datensatzes mit 1.932 IAB-Veröffentlichungen und 105 Themen zeigen die Ergebnisse, dass BERT-basierte Modelle, wie GermanBERT, durchweg die besten Resultate erzielen. Binäre Klassifikatoren erweisen sich als besonders flexibel und präzise, während TF-IDF-basierte Ansätze robuste Alternativen bei geringerer Komplexität bieten. Clustering bleibt eine Herausforderung, insbesondere bei inhaltlichen Überschneidungen.

Die Temi-Box ist vielseitig einsetzbar. Neben der in diesem Forschungsbericht beschriebenen Anwendung für die IAB-Infoplattform kann sie in zahlreichen Bereichen genutzt werden, etwa bei der Analyse von Stellenanzeigen, Berufs- und Unternehmensbeschreibungen, Verschlagwortung von Publikationen oder zur Stimmungsanalyse. Sie ist auch erweiterbar für den Einsatz in Frage-Antwort-Systemen oder zur Named Entity Recognition. Die Temi-Box erleichtert die Anwendung von Text Mining-Methoden für eine breite Nutzerbasis und bietet zahlreiche Anpassungsmöglichkeiten. Sie reduziert den Aufwand für die Entwicklung und den Vergleich von Modellen.

Ihre Open-Source-Verfügbarkeit fördert die Weiterentwicklung und Integration der Temi-Box in verschiedene Forschungsprojekte. Dies ermöglicht es Anwenderinnen und Anwendern, die Plattform an spezifische Bedürfnisse anzupassen und neue Funktionen zu integrieren. Der Bericht zeigt das Potenzial der Temi-Box, die Digitalisierung und Automatisierung der Textdatenanalyse voranzutreiben. Gleichzeitig bleiben Herausforderungen wie die Sicherstellung

der Datenqualität und die Interpretierbarkeit der Modelle. Diese Aspekte erfordern kontinuierliche Validierung und Weiterentwicklung, um die Effektivität und Zuverlässigkeit von Text Mining-Methoden weiter zu verbessern.

Summary

The constantly growing amount of digitally accessible text data and advances in natural language processing (NLP) have made text mining a key technology. The "Temi-Box" is a modular construction kit designed for text mining applications. It enables automated text classification, topic categorization, and clustering without necessitating extensive programming expertise. Developed on the basis of the keywording and topic assignment of publications for the IAB Info Platform and financed by EU funds, it is available as an open source project. This research report documents the development and application of the Temi-Box and illustrates its use and the interpretation of the results obtained.

Text mining extracts knowledge from unstructured texts using methods such as classification and clustering. The modular Temi-Box provides users with established methods in a user-friendly way and supports users with a pipeline architecture that simplifies standardised processes such as data preparation and model training. It incorporates both current and traditional approaches to text representation, such as BERT and TF-IDF, and offers a variety of algorithms for text classification and clustering, including K-Nearest Neighbors (KNN), binary and multinomial classifiers as layers in neural networks and K-Means. Various evaluation metrics facilitate the assessment of model performance and the comparison of different approaches.

Experiments on automated topic assignment and the identification of key topics illustrate the use of the Temi-Box and the interpretation of the results. Based on a dataset with 1,932 IAB publications and 105 topics, the results show that BERT-based models, such as GermanBERT, consistently achieve the best results. Binary classifiers prove to be particularly flexible and accurate, while TF-IDF-based approaches offer robust alternatives with less complexity. Clustering remains a challenge, especially when content overlaps.

The Temi-Box is a highly versatile instrument. In addition to the application for the IAB Info Platform described in this research report, it can be used in numerous areas, such as the analysis of job advertisements, job and company descriptions, keywording of publications or for sentiment analysis. It can also be extended for use in question-and-answer systems or for named entity recognition. The Temi-Box facilitates the application of text mining methods for a broad user base and offers numerous customization options. It reduces the effort involved in developing and comparing models.

Its open source availability promotes the further development and integration of the Temi-Box into various research projects. This enables users to adapt the platform to specific needs and integrate new functions. The report shows the potential of the Temi-Box to advance the digitization and automation of text data analysis. At the same time, challenges such as ensuring data quality and the interpretability of the models remain. These aspects require continuous

validation and further development in order to further improve the effectiveness and reliability of text mining methods.

Danksagung

Wir möchten uns herzlich bei allen bedanken, die zur Fertigstellung dieses Forschungsberichts beigetragen haben. Unser besonderer Dank gilt Mindaugas Vaitekunas für seine wertvolle technische Unterstützung und Expertise. Ebenso danken wir dem gesamten Fachinformationsteam des IAB, insbesondere Lena Bösel und Judith Bendel-Claus, die mit Geduld und Expertise das komplexe Fachgebiet für uns verständlich gemacht haben. Ihre Bereitschaft, alle Fragen zu beantworten, ihre unermüdliche Unterstützung beim Testen und ihre wertvollen Hinweise und Tipps haben wesentlich zur Qualität dieser Arbeit beigetragen. Darüber hinaus möchten wir der Franconian Data Science League, einem Hackathon für Data Scientists in Franken danken. Sie hat uns wertvolle Impulse insbesondere für die Methodenauswahl der Temi-Box gegeben. Darüber hinaus geht ein herzlicher Dank an Marcel Neunhoeffer für seinen wissenschaftlichen Review und Martin Schludi sowie Renate Perras für ihre Unterstützung bei der redaktionellen Bearbeitung.

1 Einleitung

1.1 Hintergrund und Motivation

In den letzten Jahren hat das Text Mining erheblich an Bedeutung gewonnen und sich zu einem unverzichtbaren Werkzeug in der Datenanalyse etabliert. Diese Entwicklung ist einerseits auf einen starken Anstieg verfügbarer Textdaten zurückzuführen, die aus einer Vielzahl von Quellen wie sozialen Medien, Nachrichtenportalen und online zugänglichen Büchern oder Artikeln stammen. Andererseits haben technologische Fortschritte, insbesondere im Bereich der natürlichen Sprachverarbeitung (Natural Language Processing, NLP) in den letzten Jahren zu einer deutlichen Verbesserung in der Qualität der Ergebnisse im Text Mining geführt.

Abbildung 1: Logo der Temi-Box

Die große Menge an Textdaten birgt wertvolle Informationen, liegt jedoch oft unstrukturiert vor. Dies stellt eine erhebliche Herausforderung für die Analyse und Verarbeitung dar. Während Menschen unstrukturierte Texte intuitiv verstehen und interpretieren können, erfordert die computergestützte Analyse von Textdaten spezialisierte Methoden und Werkzeuge. Moderne NLP-Methoden ermöglichen es, aus großen Textmengen relevante Informationen zu extrahieren, Texte zu klassifizieren, Themen zu identifizieren und Fragen automatisiert zu beantworten. Diese Methoden haben das Potenzial, die Nutzung von Textdaten grundlegend zu verändern.

Viele Schritte zur Aufbereitung und Verarbeitung der Texte verlaufen trotz aller Vielfalt der Anwendungsfälle analog. Um den Zugang zum Text Mining zu erleichtern und den Programmieraufwand für Anwenderinnen und Anwender zu reduzieren, wurde im Rahmen des Projekts „Etablieren eines Standardbaukastens für Text Mining“ die Text Mining Toolbox „Temi-Box“ entwickelt (siehe Abbildung 1). Die Temi-Box bietet eine Vielzahl von Methoden zur Klassifikation und zum Clustern von Texten sowie zur Evaluation der Ergebnisse. Diese innovative Toolbox ist ein leistungsstarkes, benutzerfreundliches Werkzeug, das es ermöglicht, auch ohne tiefgehende Programmierkenntnisse komplexe Text Mining-Aufgaben zu bewältigen. Die Finanzierung der Entwicklung erfolgte durch EU-Mittel (siehe Abbildung 2).

Die Entwicklung der Temi-Box erfolgte anhand der automatisierten Verschlagwortung von Publikationen und deren Themenzuordnung für die IAB-Infoplattform. In dieser Plattform werden der Fachöffentlichkeit Literaturnachweise, Volltexte und Forschungsinformationen zu Themen aus der Arbeitsmarkt- und Berufsforschung zur Verfügung gestellt. Die Ausrichtung an diesen realen Anwendungsszenarien ermöglichte es, die praktischen Herausforderungen und Anforderungen direkt in die Entwicklung der Temi-Box einfließen zu lassen, um ihre Funktionalitäten für den praktischen Einsatz zu optimieren.

Abbildung 2: Finanzierung des Datenlabors aus NextGenerationEU-Mitteln

Der Python-Code der Temi-Box sowie weitere Anleitungen zur Nutzung und zur Weiterentwicklung sind verfügbar unter https://github.com/iab-de/temibox.

1.2 Zielsetzung des Forschungsberichts

Dieser Forschungsbericht bietet eine umfassende Dokumentation des methodischen Vorgehens bei der Entwicklung und Anwendung der Text Mining Toolbox Temi-Box und benennt mögliche Anwendungsfelder. Konkret geht es um

- die Erläuterung der Verschlagwortung von wissenschaftlichen Publikationen und deren Themenzuordnung für die IAB-Infoplattform als zentrale Anwendungsfälle,
- eine Einführung in das Text Mining und gängige Methoden zur Klassifikation und zum Clustering von Texten,
- eine Beschreibung der technischen Grundlagen und der Implementierung der Temi-Box, insbesondere der Gesamtarchitektur und der einzelnen Komponenten,
- eine umfassende Darstellung der implementierten Methoden,
- eine ausführliche Präsentation der Ergebnisse der beispielhaft durchgeführten Text Mining-Experimente zur automatisierten Themenzuordnung und Clustering sowie um
- das Aufzeigen weiterer potenzieller Anwendungsfelder der Temi-Box, um das breite Spektrum ihrer Einsatzmöglichkeiten darzustellen.

Der Forschungsbericht bietet Einblicke in die Nutzung der Temi-Box und unterstützt deren Einsatz in einer Vielzahl von Projekten. Er liefert detaillierte Informationen und praxisnahe Beispiele, die Forscherinnen und Forscher unterstützen, die sich für die Weiterentwicklung und Anwendung von Text Mining-Technologien interessieren.

2 Anwendungsfälle Verschlagwortung und Themenzuordnung zu Publikationen

Die Temi-Box wurde anhand der automatisierten Verschlagwortung von Publikationen sowie der Zuordnung von Publikationen zu Themen für die IAB-Infoplattform entwickelt.

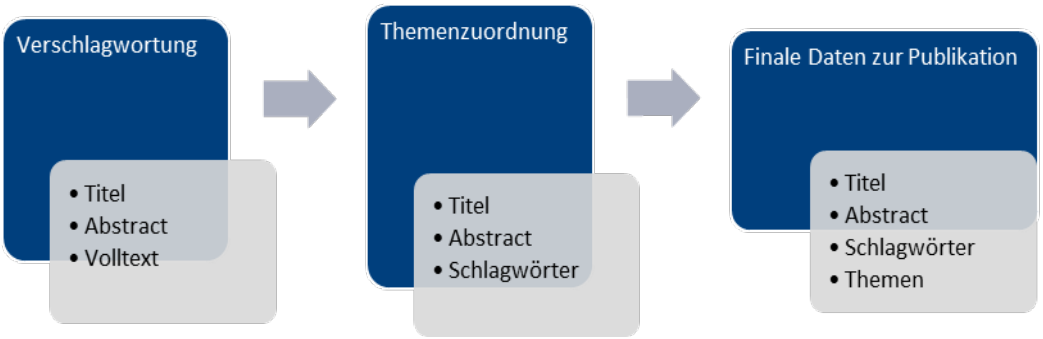
Die IAB-Infoplattform (InfoAB) ist ein thematisch strukturiertes Angebot, das aktuelle arbeitsmarkt- und sozialpolitische Themen aufgreift (IAB 2017) und in Form von Themendossiers präsentiert. Über die URL https://iab.de/publikationen/iab-infoplattform/ bietet sie Wissenschaftlerinnen und Wissenschaftlern sowie der interessierten Fachöffentlichkeit freien

Zugang zu fundierten wissenschaftlichen Fachinformationen aus der Arbeitsmarkt- und Berufsforschung (siehe Abbildung 3).

Abbildung 3: Logo der IAB-Infoplattform

Das Redaktionssystem der InfoAB, die IABinfoplattform, umfasst rund 200.000 bibliografische Nachweise, darunter Aufsätze, Working Papers und Bücher. Diese Sammlung wird seit 1967 im IAB gepflegt und wächst jährlich um etwa 4.000 Titel. Hierzu werden Neuerscheinungslisten sowie zahlreiche wissenschaftliche Zeitschriften und Working Paper-Reihen lektoriert, katalogisiert und auf Basis der bibliografischen Daten, des Titels und Abstracts sowie des Volltextes manuell mit Schlagwörtern aus dem IAB-Thesaurus angereichert (siehe Abbildung 4). Die Verschlagwortung auf Basis des IAB-Thesaurus, der ca. 7.000 Deskriptoren (Stand 2024) aus dem Bereich der Arbeitsmarkt- und Berufsforschung enthält (Oyen/Paulsen 2009) verbessert die thematische Einordnung und Auffindbarkeit der Publikationen.

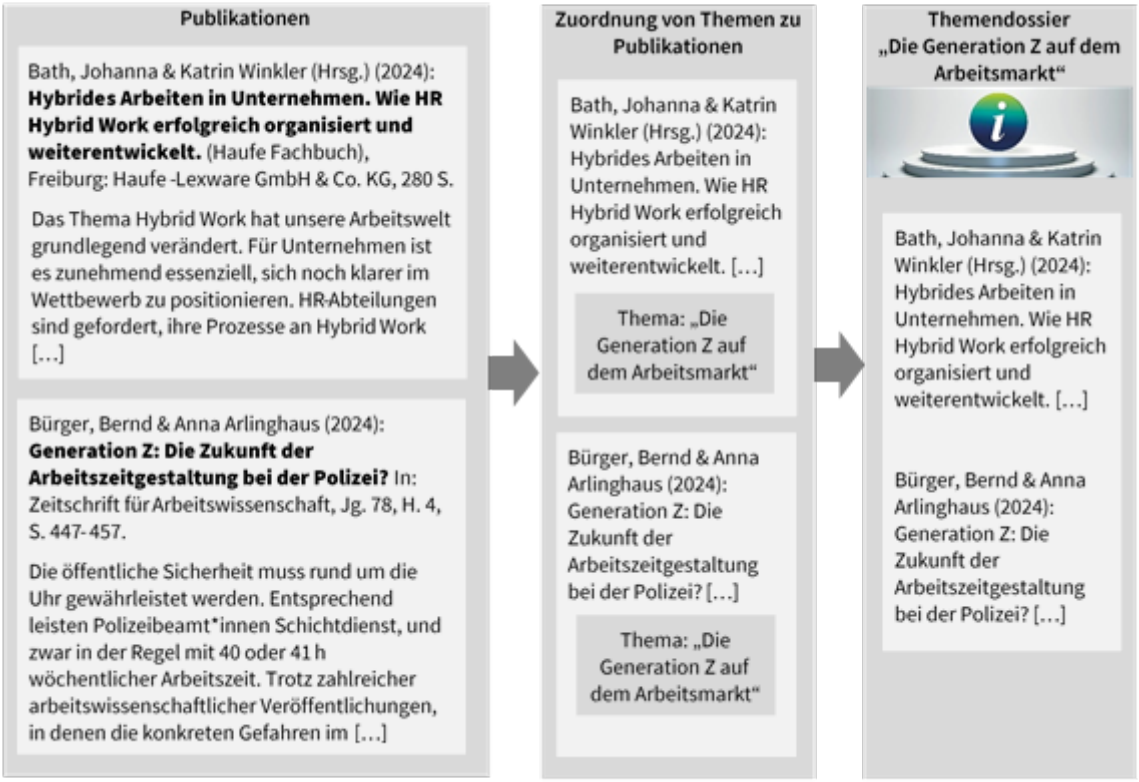
Abbildung 4: Ablauf und Datengrundlagen zur Verschlagwortung und Zuordnung von Themen zu einer Publikation



Quelle: Eigene Darstellung

Die Zuordnung der Publikationen zu Themen für die Themendossiers der IAB-Infoplattform erfolgt manuell anhand der bibliografischen Daten und Schlagwörter, wie in **Fehler! Verweisquelle konnte nicht gefunden werden.** dargestellt. Die Publikationen werden in den entsprechenden Themendossiers auf der IAB-Infoplattform veröffentlicht und ermöglichen den Nutzerinnen und Nutzern der IAB-Infoplattform einen thematischen Zugang zu den Inhalten. Ein Beispiel für die Zuordnung von Publikationen zu einem Themendossier zeigt Abbildung 5.

Abbildung 5: Beispielhafte Zuordnung von Publikationen zu einem Themendossier



Quelle: Eigene Darstellung

Die manuell gelabelten Daten aus der Themenzuordnung der Publikationen dienen als Grundlage für die Bewertung der Qualität der automatisch vorgeschlagenen Themen.

3 Text Mining im Überblick

3.1 Definition des Text Mining

Text Mining bezieht sich auf den Prozess, interessante Informationen und Wissen aus unstrukturiertem Text zu extrahieren (Hotho/Nürnberger/Paaß 2005). Dazu werden Methoden und Techniken aus mehreren Forschungsgebieten genutzt (Allahyari et al. 2017; Hotho/Nürnberger/Paaß 2005):

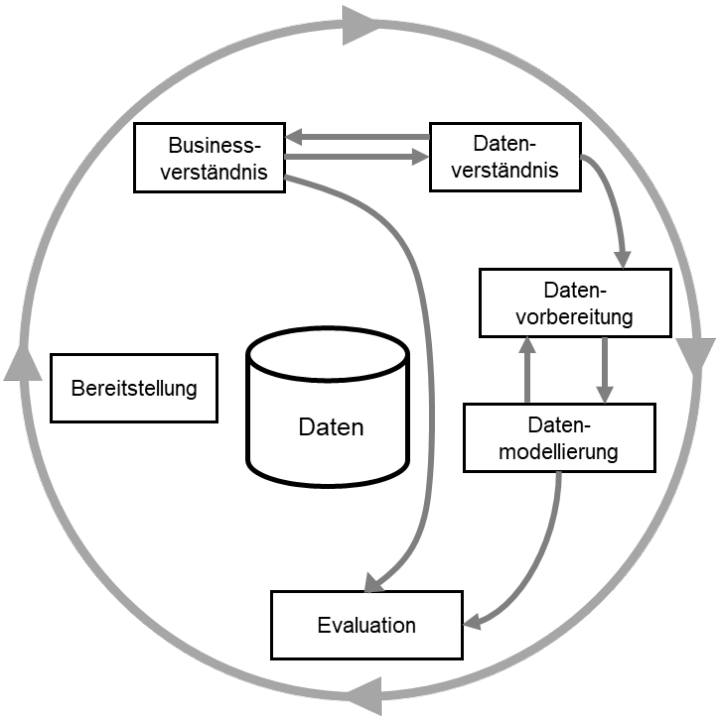
- Data Mining: Data Mining beschäftigt sich mit dem Extrahieren von Mustern aus großen Datensätzen. Text Mining ist eine Sonderform des Data Mining, es werden Data Mining-Algorithmen für die Verarbeitung von unstrukturierten Textdaten verwendet. Dazu müssen Textdaten zunächst bereinigt und in ein geeignetes Format umgewandelt werden.
- Information Retrieval: Information Retrieval konzentriert sich darauf, relevante Dokumente zu einem Informationsbedarf zu finden. Information Retrieval betont nicht die Analyse von Informationen zur Entdeckung verborgener Muster, sondern erleichtert den Informationszugang. Ein Beispiel für Information Retrieval Systeme sind Suchmaschinen.

- Natural Language Processing (NLP): NLP hat zum Ziel, mit Hilfe von Computern die menschliche Sprache zu verstehen und zu verarbeiten. Die Möglichkeiten des Text Mining sind eng verbunden mit den technischen Möglichkeiten zur Verarbeitung natürlicher Sprache. Deshalb werden NLP-Techniken, wie syntaktische Analyse und Part-of-Speech-Tagging, im Text Mining intensiv genutzt, um Texte zu analysieren und zu interpretieren.

3.2 Prozess des Text Mining

Text Mining ist ein mehrstufiger Prozess, der in verschiedene Phasen unterteilt werden kann, wobei jede Phase spezifische Aufgaben und Techniken umfasst. Dieser Prozess ist ähnlich aufgebaut wie ein klassischer Data Mining Prozess, der in der Datenaufbereitung Besonderheiten aufweist (Hippner/Rentzmann 2006).

Abbildung 6: Prozess des Text Mining „CRoss Industry Standard Process for Data Mining (CRISP-DM)“



Quelle: Eigene Darstellung nach Hotho/Nürnberger/Paaß (2005)

In Anlehnung an den weitverbreiteten CRoss Industry Standard Process for Data Mining (CRISP-DM) lässt sich der Text Mining Prozess in sechs Phasen gliedern, die häufig iterativ durchlaufen werden, um optimale Ergebnisse zu erzielen (Hotho/Nürnberger/Paaß 2005; Wirth/Hipp 2000). Abbildung 6 stellt die verschiedenen Phasen und ihre Interaktionen im Überblick dar.

3.2.1 Businessverständnis

In der ersten Phase *Businessverständnis* geht es darum, Projektziele und -anforderungen aus Geschäfts- oder Forschungsperspektive zu verstehen. Dieses Wissen wird in eine Text Mining-Aufgabenstellung umgewandelt. Die wichtigsten Aufgabenstellungen sind (Allahyari et al. 2017)

- Textklassifikation, also die Zuordnung von vordefinierten Klassen zu Textdokumenten,

- Clustern von Texten, also die Aufgabe, in einer Sammlung von Dokumenten Gruppen ähnlicher Dokumente zu finden, sowie
- Informationsextraktion, also die automatische Gewinnung von bekannten Informationen oder Fakten aus unstrukturierten oder halbstrukturierten Dokumenten.

3.2.2 Datenverständnis

Die Phase *Datenverständnis* beinhaltet die Sammlung der Daten und eine erste explorative Analyse der Texte, um Einblicke in Inhalte, Strukturen und mögliche Qualitätsprobleme zu erhalten.

3.2.3 Datenvorbereitung

In der *Datenvorbereitung* werden die Daten in eine Form gebracht, die für die Modellierung geeignet ist. Diese Phase ist für das Text Mining eine besondere Herausforderung, weil Modellierungsalgorithmen in der Regel Texte nicht direkt verarbeiten können, sondern numerische Werte voraussetzen. In der Datenvorbereitung müssen die Daten daher zunächst vorverarbeitet und in eine numerische Textrepräsentation umgewandelt werden.

Zu den häufigsten Vorverarbeitungsschritten gehören (Allahyari et al. 2017):

- Tokenization: Es wird ein Text in einzelne Wörter oder Sätze, die sog. Tokens, aufgeteilt, wobei bestimmte Zeichen, wie Satzzeichen, entfernt werden.
- Filtering: Beim Filtern werden Stopp-Wörter entfernt, also häufig vorkommende, aber inhaltsarme Wörter wie „und“, „oder“ und „aber“, sowie Wörter, die in den Dokumenten sehr häufig oder sehr selten vorkommen und deshalb wenig zur Unterscheidung verschiedener Dokumente beitragen oder möglicherweise nicht wichtig sind. Zudem wird das Rauschen entfernt, das beispielsweise durch HTML-Tags, Sonderzeichen oder Zahlen entsteht.
- Lemmatization: Es werden verschiedene Formen von Verben und Substantiven auf eine Grundform, das sogenannte Lemma, abgebildet.
- Stemming: Beim Stemming werden Wörter auf ihre Wurzel- oder Stammform reduziert.

Nach der Vorverarbeitung besteht der Text aus einer Sequenz von Tokens, die in eine numerische Form umgewandelt oder enkodiert werden müssen. Für die Erzeugung der Embeddings für die Tokens gibt es drei Techniken (Birunda/Devi 2020):

- Traditionelle Word-Embeddings: Traditionelle Word-Embeddings ermitteln, wie häufig oder selten ein Token in einem Dokument vorkommt oder mit welchen Tokens es gemeinsam auftaucht. Beispiele dafür sind Bag-of-Words oder TF-IDF. Bag-of-Words wandelt Texte in Vektoren um, indem die Häufigkeit jedes Tokens in einem Text gezählt wird. Traditionelle Word-Embeddings sind einfach zu verstehen, berücksichtigen allerdings die Reihenfolge der Tokens nicht. Sie verlieren dadurch wichtige Kontextinformationen, was die Leistung der Klassifikatoren in komplexen Aufgaben einschränken kann.
- Statische Word-Embeddings: Statische Word-Embeddings wandeln Tokens in feste Vektoren um, indem Nachschlagetabellen trainiert werden. Die Einbettungen heißen statisch, weil sie unverändert bleiben, unabhängig vom Kontext oder den Sätzen, in denen die Tokens vorkommen. Die wichtigsten Vertreter sind Word2Vec, Glove oder Fast Text.

- Kontextabhängige Word-Embeddings: Kontextabhängige Word-Embeddings hängen vom Token selbst und den benachbarten Tokens im jeweiligen Dokument ab. Deshalb besitzen gleiche Wörter in verschiedenen Sätzen des Dokuments unterschiedliche Embeddings. Beispiele für diese Technik sind ELMo, GPT oder BERT. Die Erzeugung kontextabhängiger Word-Embeddings ist komplex und rechenintensiv, ermöglicht aber auch eine deutlich bessere Leistung bei vielen Aufgaben zur Verarbeitung natürlicher Sprache.

3.2.4 Modellierung

In dieser Phase werden Modelle entwickelt, die auf den vorbereiteten Daten trainiert werden. Es werden Algorithmen zur Modellierung ausgewählt, Modelle erstellt und validiert. Abhängig von der Aufgabenstellung stehen im Text Mining eine Vielzahl von Modellierungsmethoden zur Verfügung (Allahyari et al. 2017; Gasparetto et al. 2022; Li et al. 2022; Petukhova/Matos-Carvalho/Fachada 2025), auf die in Abschnitt 3.3 eingegangen wird.

Nach der Erstellung der Modelle erfolgt eine gründliche Validierung anhand von Metriken wie Precision oder Recall, um die Leistungsfähigkeit der Modelle zu bewerten. Bei Bedarf werden die Modelle überarbeitet oder neu erstellt, um ihre Leistung zu verbessern.

Zwischen Modellierungs- und der Datenaufbereitungsphase besteht eine enge Verbindung. Häufig treten während der Modellierung Datenprobleme auf, die zuvor nicht erkennbar waren. Dies kann den Bedarf an zusätzlichen Daten oder an einer erneuten Datenaufbereitung nach sich ziehen, um die Qualität der Inputdaten und damit auch die Modellleistung zu verbessern.

3.2.5 Evaluation

In der Modellierungsphase werden ein oder mehrere Modelle entwickelt, die aus statistischer Sicht eine hohe Qualität aufweisen. Ziel der Evaluationsphase ist es, die Modellleistung im Hinblick auf die zu Beginn definierten Projektziele zu überprüfen und festzustellen, ob wichtige Aspekte aus Geschäfts- oder Forschungsperspektive möglicherweise nicht ausreichend berücksichtigt wurden. Am Ende der Evaluationsphase steht eine fundierte Entscheidung über die Nutzung der Text Mining-Ergebnisse.

3.2.6 Bereitstellung

In der Bereitstellungsphase werden die Ergebnisse des Text Minings zur Verfügung gestellt. Je nach Projektzielen kann es sich dabei um die Erstellung eines Berichts handeln oder auch um die Integration des Modells in eine produktive Umgebung.

3.3 Zentrale Aufgabenstellungen im Text Mining

Text Mining besitzt ein breites Spektrum an Anwendungsmöglichkeiten, die sich über zentrale Aufgabenstellungen abbilden lassen. Die Datenvorbereitung verläuft für diese Aufgabenstellungen weitgehend gleich, Unterschiede gibt es im Wesentlichen in der Modellierung. Für die Verschlagwortung und Themenzuordnung sind die Aufgabenstellungen *Textklassifikation* und *Textclustering* besonders relevant, deshalb gehen die folgenden Abschnitte genauer darauf ein.

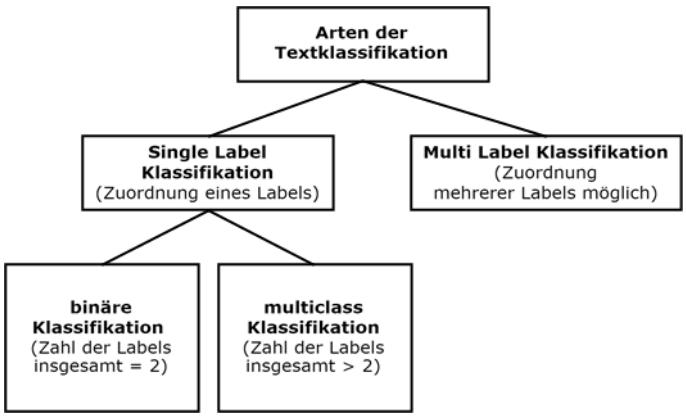
3.3.1 Textklassifikation

Bei der Textklassifikation wird einem kurzen Text oder einem Dokument eine spezifische Klasse, auch als Kategorie oder Label bezeichnet, zugeordnet (Paaß/Giesselbach 2023). Zahlreiche Anwendungsfälle im Text Mining lassen sich als Klassifikationsaufgabe formulieren, wie etwa die Erkennung von Spam-Mails, die Durchführung von Stimmungsanalysen oder die Zuordnung von Dokumenten zu bekannten Themen, dem Anwendungsfall zur Entwicklung der Temi-Box.

Arten der Textklassifikation

Abhängig von der Zahl der Klassen lässt sich die Textklassifikation in verschiedene Arten unterteilen, wie Abbildung 7 zeigt (Paaß/Giesselbach 2023).

Abbildung 7: Arten der Textklassifikation



Quelle: Eigene Darstellung

Bei der binären Klassifikation wird ein Text einer von zwei möglichen Klassen zugeordnet. Die Multiclass-Klassifikation erweitert dieses Konzept, indem sie die Zuordnung eines Textes zu einer von mehr als zwei Klassen ermöglicht. In Szenarien, in denen ein Text mehreren Klassen gleichzeitig zugeordnet werden kann, spricht man von Multi-Label-Klassifikation. Besonders bei der extremen Multi-Label-Klassifikation, die eine sehr große Anzahl von Labels umfasst, sind spezialisierte Algorithmen erforderlich, um die damit verbundene hohe Komplexität effektiv bewältigen zu können.

Traditionelle Methoden der Textklassifikation

Frühe Methoden zur Textklassifikation seit den 1990er Jahren verwendeten klassische maschinelle Lernverfahren (Kowsari et al. 2019; Li et al. 2022; Paaß/Giesselbach 2023). Diese Ansätze erforderten eine manuelle Extraktion von Merkmalen (Features) aus Texten, um sie in eine maschinenlesbare Form zu überführen. Anschließend wurden diese Merkmale genutzt, um einen Klassifikator zu trainieren, der die Texte den vordefinierten Klassen zuordnete.

Zu den häufig verwendeten Klassifikationsalgorithmen zählten Naïve Bayes (Maron 1961), K-Nearest Neighbor (KNN) (Cover/Hart 1967), logistische Regression (Cox 1958), Support Vector Machines (SVM) (Joachims 1998) oder baumbasierte Methoden wie Random Forest (Breiman 2001). Obwohl diese Modelle eine hohe Genauigkeit in vielen Anwendungen erreichen, leiden sie unter der Notwendigkeit aufwändiger Feature-Engineering-Prozesse und der Unfähigkeit,

semantische Informationen direkt aus den Texten zu erfassen. In den letzten Jahren wurden auch traditionelle Algorithmen weiterentwickelt, wie beispielsweise eXtreme Gradient Boosting (XGBoost; Chen/Guestrin (2016)) und die Light Gradient Boosting Machine (LightGBM; Ke et al. (2017)). Diese Methoden bieten eine verbesserte Leistungsfähigkeit und Effizienz, erfordern jedoch nach wie vor manuelle Feature-Engineering-Schritte.

Deep Learning Methoden der Textklassifikation

Seit den 2010er Jahren haben sich Deep Learning Methoden zur Textklassifikation etabliert, weil sie sich als besonders leistungsfähig erwiesen haben (Li et al. 2022; Paaß/Giesselbach 2023). Der Begriff *Deep* verweist auf die Vielzahl von Schichten in Netzwerk-Modellen, die es ermöglichen, Merkmale automatisch zu extrahieren und komplexe Muster in Textdaten zu erkennen, ohne dass manuelles Feature-Engineering erforderlich ist. Diese Modelle erfassen tiefere und semantisch reichhaltigere Textrepräsentationen, was ihre Leistung in verschiedenen Klassifikationsaufgaben erheblich verbessert.

Zu den anfänglich eingesetzten Ansätzen zählen Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) und Attention-Mechanismen, die kontinuierlich weiterentwickelt wurden (Li et al. 2022). Einen bedeutenden Fortschritt in der Textklassifikation markierte die Einführung von BERT-Modellen (Bidirectional Encoder Representations from Transformers), die auf der Transformer-Architektur basieren (Devlin et al. 2019; Minaee et al. 2021; Paaß/Giesselbach 2023). Vortrainierte Sprachmodelle wie BERT werden zunächst auf großen Mengen generischer Textdaten trainiert, um allgemeine Sprachrepräsentationen zu lernen. Anschließend erfolgt das Fine-Tuning, bei dem das Modell an spezifische Aufgaben angepasst wird. Vortrainierte Sprachmodelle wie BERT haben in der Textklassifikation wie auch in anderen Aufgaben der natürlichen Sprachverarbeitung neue Maßstäbe gesetzt (Devlin et al. 2019).

Textklassifikation mit der Temi-Box

Bei den Anwendungsfällen automatisierte Verschlagwortung und Themenzuordnung handelt es sich um Textklassifikationsaufgaben. Deshalb liegt der Schwerpunkt der Temi-Box auf den Möglichkeiten zur Klassifikation von Texten. Standardmäßig kann die Anwenderin oder der Anwender Deep Learning-Methoden auf Basis von BERT einsetzen, weil hier die beste Performance zu erwarten ist. Zusätzlich wurde TF-IDF implementiert, um einen Vergleich mit traditionellen Ansätzen zu ermöglichen oder TF-IDF bei kleineren Datensätzen oder unzureichender Infrastruktur einzusetzen.

3.3.2 Textclustering

Das Clustern von Texten ist eine zentrale Aufgabe im Text Mining, die es ermöglicht, Texte basierend auf ihrer Ähnlichkeit in Gruppen, sogenannte Cluster, zu unterteilen (Allahyari et al. 2017). Diese Methode findet in zahlreichen Bereichen Anwendung, etwa bei der Organisation von Büchern, der Zusammenfassung von Texten, der Klassifizierung von Dokumenten und der Erkennung von Themen (Subakti/Murfi/Hariadi 2022).

Grundprinzip des Clusterns und Algorithmen

Das Ziel des Clusterns ist es, Dokumente so zu gruppieren, dass die Texte innerhalb eines Clusters möglichst ähnlich und die Texte zwischen verschiedenen Clustern möglichst unterschiedlich sind (Allahyari et al. 2017). Dies erfolgt in der Regel durch die Nutzung von Ähnlichkeitsfunktionen. Die verwendete Repräsentation der Texte spielt dabei eine wichtige Rolle, da sie die Grundlage für die Berechnung der Ähnlichkeit bilden. Es gibt eine Reihe von Clustering-Algorithmen, die je nach Anwendungsfall und Datenstruktur unterschiedlich gut geeignet sind. Zu den traditionellen Clustering-Methoden gehören K-Means, hierarchisches Clustering und dichtebasiertes Clustering (Xu/Gu/Ji 2024).

Clustern mit der Temi-Box

Die Temi-Box bietet den K-Means-Algorithmus für das Textclustering, um ähnliche Texte zu gruppieren. Dies kann beispielsweise bei Publikationen die Identifikation übergreifender Themenschwerpunkte erleichtern oder die Literaturrecherche zu verwandten Publikationen vereinfachen. Anwenderinnen und Anwender können dabei zwischen BERT und TF-IDF als Methoden zur Textrepräsentation wählen.

4 Aufbau der Temi-Box

Die Text Mining Toolbox Temi-Box unterstützt Anwenderinnen und Anwender bei der Analyse und Erschließung von Textdaten in Python. Im Text Mining-Prozess sind einige Schritte projekt- oder datenspezifisch, wie etwa das Geschäfts- und Datenverständnis, die Evaluation im Hinblick auf die Projektziele sowie das Deployment. Die Datenvorbereitung und Modellierung verlaufen hingegen für viele Text Mining-Aufgaben ähnlich. Gleichzeitig sind das die Schritte, die sehr viel Wissen zu den Methoden des Text Mining erfordern. Zu diesen Schritten bietet die Temi-Box die notwendige Unterstützung und vereinfacht das Text Mining. Im Folgenden werden die Anforderungen an die Temi-Box als Standardbaukasten für Text Mining detailliert beschrieben. Darüber hinaus werden Architektur und technische Umsetzung der Toolbox im Detail erläutert. Zudem wird die Temi-Box zu bestehenden Frameworks und Bibliotheken abgegrenzt.

4.1 Anforderungen an die Entwicklung

Für die Entwicklung des Standardbaukastens Text Mining galten im Wesentlichen die folgenden Anforderungen aus dem Projektauftrag:

- Eine zentrale Anforderung an den Standardbaukasten war die leichte Zugänglichkeit insbesondere auch für Anwenderinnen und Anwender mit nur grundlegenden Programmierkenntnissen in Python. Zur Benutzerfreundlichkeit sollte auch eine umfassende Dokumentation mit leicht verständlichen Codebeispielen beitragen.
- Ein weiterer wichtiger Aspekt war die Unterstützung verschiedener Methoden aus den Bereichen Natural Language Processing (NLP), Klassifikation und Clustering. Der Baukasten sollte eine Auswahl etablierter Methoden und Techniken bieten, die es den Anwenderinnen und Anwendern erleichtern, unterschiedliche Ansätze zu testen und ihre Ergebnisse

systematisch miteinander zu vergleichen. Hierdurch sollten Nutzerinnen und Nutzer Einblicke in Stärken und Schwächen der verschiedenen Methoden gewinnen können.

- Um die Entwicklungszeiten zu verkürzen, sollte der Baukasten die Möglichkeit bieten, die eingesetzten Methoden in Form einer Pipeline zu integrieren. Eine Pipeline-Architektur ermöglicht es, die verschiedenen Verarbeitungsschritte – wie Datenvorverarbeitung, Modellierung oder Evaluation – nahtlos miteinander zu verknüpfen. Die Bereitstellung von vorgefertigten Pipeline-Komponenten und die einfache Anpassbarkeit an spezifische Anwendungsfälle sollten zudem die Wiederverwendbarkeit des Codes fördern.

4.2 Architektur und technische Umsetzung

Die Temi-Box zeichnet sich durch eine modulare und flexible Architektur aus. Sie bietet eine Vielzahl vorgefertigter Komponenten, die sich in eine Pipeline integrieren lassen. Sowohl die Pipeline als auch die einzelnen Komponenten sind jederzeit anpassbar und erweiterbar, um spezifischen Anforderungen und komplexen Anwendungsfällen gerecht zu werden. Die Temi-Box wurde als Python-Paket konzipiert, um eine intuitive Einrichtung und Anwendung zu ermöglichen. Eine umfassende Dokumentation mit leicht verständlichen Codebeispielen senkt die Einstiegshürde und erleichtert den Nutzerinnen und Nutzern den schnellen Einstieg.

4.2.1 Pipeline-Architektur

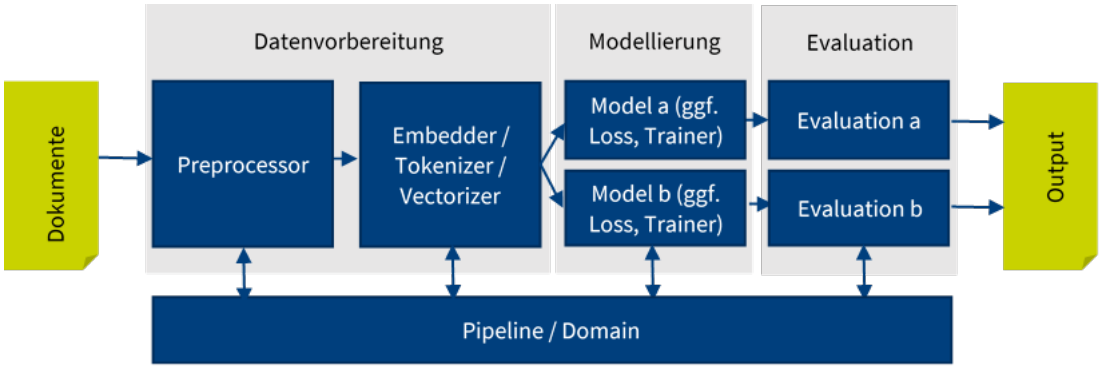
Die Architektur der Temi-Box hat Pipeline-Charakter, d. h. der Gesamtvorgang wird in einzelne Verarbeitungsschritte unterteilt. Diese Schritte sind in Form der Komponenten umgesetzt. Die Komponenten brauchen als Input Daten in einer bestimmten Form, übernehmen dann jeweils eine bestimmte Aufgabe, wie die Bildung der Word-Embeddings oder das Trainieren von Modellen, und stellen ihre Ausgaben wieder anderen Komponenten zur Verfügung.

Die Pipeline-Architektur ermöglicht den Nutzerinnen und Nutzern eine klare Trennung der verschiedenen Aufgaben, vereinfacht den Einsatz und den Vergleich verschiedener Methoden und Algorithmen und macht den gesamten Prozess überschaubar und leicht erweiterbar. Zudem ermöglicht sie, die Abfolge der Verarbeitungsschritte für Textdaten festzulegen, Komponenten in mehreren Pipelines einzusetzen und Abhängigkeiten zwischen ihnen festzulegen.

4.2.2 Standardkomponenten

Abbildung 8 zeigt, mit welchen Standardkomponenten die Temi-Box Pipeline den Text Mining Prozess unterstützt.

Abbildung 8: Schematischer Überblick über eine Temi-Box Pipeline mit Standardkomponenten



Anmerkung: Die Temi-Box unterstützt die Datenvorbereitung, Modellierung und Evaluation im Text Mining.
Quelle: Eigene Darstellung

In der Datenvorbereitung werden die Texte mit dem Preprocessor aufbereitet. Embedder, Tokenizer bzw. Vectorizer bilden eine numerische Repräsentation des Textes. Dieser Schritt ist notwendig, weil die Modelle üblicherweise keine Texte verarbeiten können, sondern numerische Werte voraussetzen (siehe hierfür auch Abschnitt 3.2.3).

In der Modellierungsphase werden Modelle trainiert und mit Klassifikations- oder Clusteralgorithmen Aussagen über den Text getroffen (siehe hierfür auch Abschnitt 3.2.4). In unserem Anwendungsfall wird beispielsweise der Grad der Relevanz eines Schlagworts oder Themas für eine Publikation ermittelt. Schlagwörter oder Themen mit hoher Relevanz werden als zugehöriges Schlagwort oder Thema identifiziert. Die Algorithmen können aus dem Bereich *supervised learning* stammen, also mit Beispieldaten trainiert werden, die bereits bekannte Zielwerte enthalten. Diese Zielwerte, auch Labels genannt, sind in unserem Anwendungsfall die bisher manuell vorgenommenen Zuordnungen von Schlagwörtern und Themen. Stammen die Algorithmen aus dem Bereich *unsupervised learning*, werden Modelle mit Daten ohne Labels trainiert. Als Zielgröße können beispielsweise eine angestrebte Anzahl an Clustern und eine angestrebte Verteilung auf die Cluster herangezogen werden.

In der Evaluation (siehe hierfür auch Abschnitt 3.2.5) werden abschließend diverse Messgrößen ausgegeben, um die Qualität eines Modells zu ermitteln oder verschiedene Modelle zu vergleichen.

4.2.3 Temi-Box Komponenten

Die Temi-Box gliedert die Verarbeitungsschritte zum Text Mining modular in verschiedene Komponenten. Diese Komponenten lassen sich unterteilen in Komponenten der Fachdomäne und anwendungsfallneutrale Komponenten. Komponenten der Fachdomäne bündeln alle anwendungsfallspezifischen Informationen und Logiken. Dazu gehören z. B. die verwendeten Features, Labels, Datenladeprozesse und spezifischen Vorverarbeitungsschritte. Alle übrigen Komponenten sind so konzipiert, dass sie vom konkreten Anwendungsfall unabhängig sind und bei Bedarf auf Informationen aus den Komponenten der Fachdomäne zugreifen. Insbesondere verschiedene etablierte Methoden und Techniken zum Text Mining, wie der Embedder oder Trainer, sind als anwendungsfallneutrale Komponenten umgesetzt. Sie können in verschiedene Pipelines unverändert oder anpasst eingesetzt werden.

4.2.4 Anpassbarkeit und Erweiterbarkeit

Für komplexere Anwendungsfälle kann es erforderlich sein, fortgeschrittene Techniken und Methoden im Bereich des Text Mining einzusetzen. Die Temi-Box bietet hierfür die Möglichkeit, vorhandene Komponenten zu modifizieren oder zu erweitern. Durch bereitgestellte Schnittstellen können Expertinnen und Experten zudem individuelle Komponenten, Modelle oder vollständige Pipelinelogiken implementieren. Dies ermöglicht es, die Temi-Box an spezifische und komplexe Anwendungsfälle anzupassen.

4.2.5 Zielgruppenorientierung

Die Temi-Box stellt spezifische Funktionen für verschiedene Zielgruppen bereit und versucht damit, unterschiedlichen Bedürfnissen und Erfahrungsstufen ihrer Nutzerinnen und Nutzer gerecht zu werden.

Für den Einstieg in das Text Mining stehen vordefinierte Vorlagen, sogenannte Blueprints, zur Verfügung. In den Blueprints sind alle wesentlichen technischen Aspekte bereits vordefiniert, Anwenderinnen und Anwender ergänzen lediglich die projektspezifischen fachlichen Informationen. Das ist für einfache Anwendungsfälle ausreichend und erleichtert den Einstieg in das Text Mining.

Nutzerinnen und Nutzer mit bereits vorhandener Erfahrung im Text Mining können mit der Temi-Box ihre eigenen Pipelines aus Standardkomponenten konfigurieren. Diese Pipelines können aus mehreren Komponenten, einem oder mehreren Prognosemodellen sowie spezifischen fachlichen Themen bestehen. Diese Anpassungsfähigkeit ermöglicht es fortgeschrittenen Nutzerinnen und Nutzern, ihre Text Mining-Prozesse nach individuellen Anforderungen zu gestalten.

Für Expertinnen und Experten bietet die Temi-Box erweiterbare Schnittstellen, die es ermöglichen, individuelle Komponenten, Modelle oder eigene Pipelinelogiken zu implementieren. Diese Funktionalität erlaubt es, auch komplexe Aufgabenstellungen zu bearbeiten und maßgeschneiderte Lösungen zu entwickeln. Damit wird gewährleistet, dass sich auch komplexe Text Mining Aufgaben mit der Temi-Box bearbeiten lassen.

4.3 Abgrenzung zu bestehenden Frameworks und Bibliotheken

In Python existieren verschiedene Bibliotheken, die die Erstellung von Pipelines für das Text Mining ermöglichen. Beispiele hierfür sind scikit-learn (Pedregosa et al. 2011) für allgemeine Machine-Learning-Pipelines, spaCy (Honnibal et al. 2020) für Natural Language Processing (NLP) Pipelines und Transformers für transformer-basierte Pipelines. Die Arbeit mit diesen Bibliotheken setzt jedoch in der Regel umfangreiche Kenntnisse in Python und NLP voraus.

Die Temi-Box baut auf etablierten Machine-Learning-Frameworks und Bibliotheken wie PyTorch (Paszke et al. 2019), Transformers (Wolf et al. 2020) und spaCy auf, löst jedoch die anwendungsfallspezifischen Aspekte aus den Text Mining- und Machine-Learning-Prozessen heraus. Sie bündelt diese in den Komponenten der Fachdomäne und liefert vordefinierte Komponenten und Pipelines. Dadurch wird die Implementierung der weiteren Schritte erheblich vereinfacht und es ist möglich, verschiedene Methoden des Text Minings unkompliziert auszuprobieren. Die Temi-Box bietet somit einen schnellen Zugang zum Text Mining für den

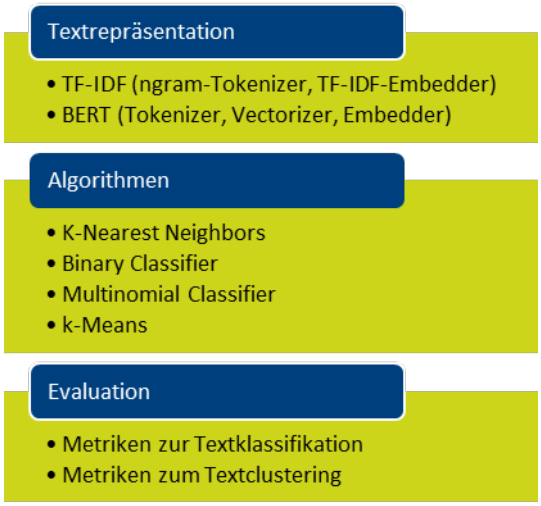
Einstieg und gewährleistet gleichzeitig die Erweiterbarkeit und Flexibilität für Expertinnen und Experten.

5 Methoden der Temi-Box

Die Temi-Box ermöglicht es Anwenderinnen und Anwendern, verschiedene Methoden und Techniken zum Text Mining einzusetzen und systematisch miteinander zu vergleichen. Für Datenvorbereitung, Modellierung und statistische Evaluation stellt sie verschiedene etablierte Methoden und Algorithmen aus den Bereichen Natural Language Processing (NLP), Klassifikation und Clustering zur Verfügung. Hierdurch können Nutzerinnen und Nutzer Einblicke in Stärken und Schwächen der verschiedenen Methoden gewinnen.

Abbildung 9 gibt einen Überblick über die wichtigsten implementierten Methoden in der Temi-Box.

Abbildung 9: Überblick über die wichtigsten implementierten Methoden in der Temi-Box



Quelle: Eigene Darstellung

Die Vorverarbeitung des Texts mit typischen Textbereinigungsschritten ist direkt in die Komponenten zur Textrepräsentation integriert, sofern es sich um TF-IDF- oder BERT-spezifische Schritte handelt. Die übrigen Schritte zur Textbereinigung sind spezifisch für den Anwendungsfall und werden daher in die Komponenten der Domäne integriert.

5.1 Einrichtung der Temi-Box

Die Temi-Box steht als Open-Source-Projekt auf GitHub zur Verfügung und kann dort kostenlos heruntergeladen und genutzt werden. Der Quellcode sowie eine umfassende Dokumentation mit leicht verständlichen Codebeispielen sind unter dem Link https://github.com/iab-de/temibox abrufbar.

5.2 Textrepräsentation

Zur Bildung einer numerischen Repräsentation des Texts stellt die Temi-Box zwei Verfahren zur Verfügung: TF-IDF als einfaches, auf Worthäufigkeiten basierendes Verfahren und BERT als Verfahren, das auch den Kontext eines Wortes berücksichtigt. Die Textrepräsentation durch TF-IDF wird als Vektor bezeichnet. BERT erzeugt einen wesentlich komplexeren Vektor, ein sog. Embedding. Zur Verbesserung der Lesbarkeit verwenden wir in diesem Bericht sowohl für die Textrepräsentation durch TF-IDF als auch durch BERT die Bezeichnung Embedding.

5.2.1 TF-IDF (Term Frequency – Inverse Document Frequency)

TF-IDF ist eine klassische Methode zur Gewichtung von Wörtern in einem Dokument (Sparck Jones 1972). TF-IDF wird als Produkt aus Term Frequency (TF), also der Häufigkeit eines Worts in einem Dokument, und der Inverse Document Frequency (IDF), also der inversen Häufigkeit des Worts in allen Dokumenten, berechnet. Je größer die Term Frequency, also je häufiger ein Wort in einem Dokument vorkommt, desto eher sagt es etwas über den Inhalt des Dokuments aus. Die Inverse Document Frequency gewichtet Wörter höher, die nur in wenigen Dokumenten vorkommen, weil sich Dokumente dadurch unterscheiden lassen. Die Inverse Document Frequency eines Worts w wird berechnet als

$$IDF_w = log_{10} \frac{N}{df_w}$$

wobei N die Gesamtzahl aller Dokumente und df_w die Zahl der Dokumente, in denen das Wort w vorkommt, ist (Jurafsky/Martin 2023). TF-IDF ist eine Form der Textrepräsentation, die leicht verständlich und berechenbar ist. Allerdings kann sie Position und Kontext eines Wortes in einem Satz nicht berücksichtigen (Subakti/Murfi/Hariadi 2022).

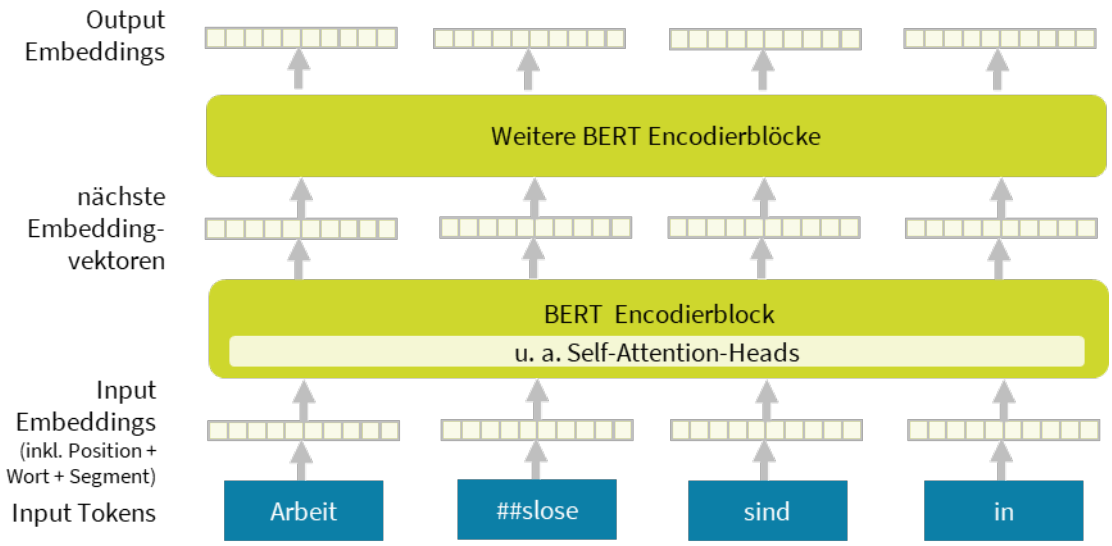
5.2.2 BERT (Bidirectional Encoder Representations from Transformers)

BERT, die Kurzform für Bidirectional Encoder Representations from Transformers, wurde von Devlin et al. (2019) vorgestellt und markiert einen Meilenstein in der Verarbeitung natürlicher Sprache (Li et al. 2022). BERT berücksichtigt bei der Erstellung der Textrepräsentationen den Kontext auf beiden Seiten des Tokens und erreicht so ein tieferes Verständnis von Wortbeziehungen.

Ein BERT-Modell wird in zwei Schritten erstellt: durch Pretraining und Finetuning. Während des Pretrainings wird das Modell auf einer umfangreichen, ungelabelten Datenmenge über verschiedene Pretrainingsaufgaben trainiert und lernt allgemeine syntaktische und semantische Eigenschaften der Sprache (Paaß/Giesselbach 2023). Im klassischen Ansatz wurde BERT von Devlin et al. (2019) mit zwei Aufgaben vortrainiert: In der maskierten Sprachmodellierung (MLM) wurde ein bestimmter Prozentsatz des Textes „maskiert“, also durch ein spezielles Symbol ersetzt. Die Aufgabe für BERT bestand darin, den fehlenden Text vorherzusagen. Das ermöglicht es dem Modell, eine Beziehung zwischen dem maskierten Text und seinem umgebenden Kontext links und rechts, also bidirektional, herzustellen. Die zweite Aufgabe war die Vorhersage des nächsten Satzes (NSP). Dabei wurden BERT jeweils zwei Sätze präsentiert und das Modell lernte vorherzusagen, ob der zweite Satz auf den ersten folgt. Diese Aufgabe ermöglichte es dem Modell, Beziehungen zwischen Sätzen herzustellen (Gasparetto et al. 2022). Nach dem Pretraining folgt zur Erstellung eines BERT-Modells im zweiten Schritt das Finetuning. Dazu wird das BERT-Modell mit den vortrainierten Parametern initialisiert und die Parameter werden

anhand von gelabelten Daten für konkrete Aufgaben feinabgestimmt. Dadurch entstehen für verschiedene nachgelagerte Aufgaben separate fein abgestimmte Modelle, auch wenn sie mit denselben vortrainierten Parametern initialisiert werden. Dieser Ansatz wird auch als Transferlernen bezeichnet, da das beim Vortraining erworbene Wissen auf eine verwandte Anwendung übertragen wird (Paaß/Giesselbach 2023).

Abbildung 10: Erstellung von Output Embeddings zur Textrepräsentation mit BERT



Quelle: Darstellung in Anlehnung an Paaß/Giesselbach (2023)

Die Grundarchitektur verschiedener BERT-Modelle ist immer einheitlich. Wie in Abbildung 10 dargestellt, verwendet BERT als Input Text, der in Tokens konvertiert wurde, häufig mit Hilfe des Wordpiece Tokenizer (Wu et al. 2016). Die Information zu jedem Token wird um Informationen zu Position und Text-Segment angereichert und als Input Embedding BERT zur Verfügung gestellt. In vielen BERT-Modellen ist die Dimension dieser Input-Embeddings auf 768 festgelegt. Diese Zahl bleibt über alle Schichten hinweg gleich und entspricht den Hidden States eines BERT-Modells. Die maximale Anzahl der Tokens, die das Modell gleichzeitig verarbeiten kann, wird als Kontextgröße bezeichnet und liegt für einfache Modelle wie BERTbase oder DistilBERT bei 512. Die Input-Embeddings durchlaufen mehrere Schichten von Transformer-Blöcken, die auf dem Ansatz von Vaswani et al. (2017) basieren. In diesen Blöcken werden die Embeddings unter anderem durch verschiedene Self-Attention-Heads verarbeitet, um die Beziehungen zwischen den Tokens zu erfassen. Am Ende dieses Prozesses werden Output-Embeddings erzeugt, also Repräsentationen zu Tokens, die Kontextinformationen beinhalten.

Verschiedene BERT-Modell-Varianten unterscheiden sich in der Anzahl der Schichten, der Hidden States und der Self-Attention-Heads. Aus diesen drei Größen ergibt sich die Gesamtzahl der Modellparameter. Sie ist ein Maß für die Komplexität des BERT-Modells. Je größer die Zahl der Parameter, desto besser ist in der Regel die Leistung der Modelle. Allerdings steigt dadurch auch der Speicherbedarf und der Ressourcenbedarf für das Training (Paaß/Giesselbach 2023). Um spezifische Anforderungen, wie eine bessere Performance oder Effizienz, zu erfüllen, wurden

zahlreiche Varianten vortrainierter Modelle auf Basis von BERT entwickelt. Beispiele sind RoBERTa, DistilBERT oder für deutschsprachige Texte GBERT.

Liu/Lapata (2019) optimierten BERT mit RoBERTa (Robustly Optimized BERT Approach), indem sie durch Anpassungen im Pre-Trainingsprozess eine verbesserte Leistung erzielten. Zu den wesentlichen Änderungen gehören die dynamische Maskierung von Tokens, die Entfernung der Next Sentence Prediction (NSP) Aufgabe, die Verwendung größerer Batch-Größen über eine größere Datenmenge und die Verlängerung der Trainingszeiten und Sequenzen. Mit diesen Änderungen konnte RoBERTa die Leistungen anderer Modelle bei verschiedenen Aufgaben erreichen oder sogar übertreffen.

DistilBERT (Sanh et al. 2020) ist eine komprimierte Version von BERT, die während des Pre-Trainings die Wissensdestillation nutzt. Dieses Vorgehen ermöglicht es, die Größe von BERT um etwa 40 Prozent zu reduzieren, die Trainingsdauern um etwa 60 Prozent zu verkürzen, jedoch 97 Prozent der Sprachverständnisfähigkeiten beizubehalten. DistilBERT ist somit eine effiziente und ressourcenschonende Alternative zu BERT.

GBERT (German BERT, Chan/Schweter/Möller (2020)) ist eine speziell für die deutsche Sprache angepasste Variante von BERT. Das Modell wurde auf einem umfangreichen deutschen Textkorpus trainiert, um die sprachspezifischen Nuancen und grammatikalischen Strukturen von deutschen Texten besser zu erfassen.

Neben RoBERTa, DistilBERT und GBERT gibt es zahlreiche weitere Modelle mit spezifischen Stärken und Schwächen für verschiedene Anwendungsfälle. Umfassende Dokumentationen und Benchmarks liefern dazu Plattformen wie Hugging Face. Die Temi-Box lässt sich mit den meisten BERT-Varianten kombinieren.

5.2.3 Vergleich TF-IDF und BERT

BERT erreicht in vielen Aufgaben zum Text Mining sehr gute Ergebnisse (Devlin et al. 2019). Beim Vergleich von BERT mit klassischen Ansätzen wie TF-IDF zeigt in der Regel BERT bessere Leistungen. Selbst bei kleinen Datenmengen erzielt BERT eine höhere Genauigkeit als das überwachte Training auf dem kleinen Datensatz, weil BERT mit seiner Fähigkeit zum Transferlernen Wissen über Sprache nutzen kann, das während des Vortrainings erworben wurde (Paaß/Giesselbach 2023). TF-IDF hingegen hat seine Stärken, wenn Modelle einfach und leicht interpretierbar sein sollen, wenn die verfügbare Rechenkapazität und der Speicher beschränkt sind oder wenn die Verarbeitungsgeschwindigkeit beispielsweise bei Echtzeitanwendungen entscheidend ist.

5.3 Algorithmen

Die Temi-Box stellt eine Auswahl gängiger Algorithmen für zwei zentrale Aufgabenstellungen im Text Mining bereit, die Klassifikation und das Clustern von Texten. Um ein möglichst breites Spektrum an Analysemöglichkeiten zu bieten, umfasst die Toolbox sowohl Algorithmen aus dem Bereich des überwachten Lernens (supervised learning) als auch des unüberwachten Lernens (unsupervised learning). Algorithmen des überwachten Lernens zeichnen sich dadurch aus, dass sie mit gelabelten Daten trainiert werden. Das bedeutet, dass für den Trainingsdatensatz der gewünschte Output bereits vorliegen muss und das Modell daraus spezifische Muster und Beziehungen lernt. Im Gegensatz dazu lernen Algorithmen des unüberwachten Lernens

Zusammenhänge ohne Vorgaben. Beispiele sind die Identifizierung von Ähnlichkeiten zwischen Begriffen oder das Gruppieren ähnlicher Dokumente.

Als Input nutzen die Algorithmen Embeddings, die durch Techniken wie TF-IDF und BERT generiert werden. Bei der Verwendung von BERT entspricht die Textklassifikation der Feinabstimmung eines vortrainierten BERT-Modells. Dazu wird das BERT-Modell um eine oder mehrere zusätzliche Schichten erweitert, die die Klassifikatoren beinhalten und die Output-Embeddings als Eingaben verwenden (Paaß/Giesselbach 2023). Während der Feinabstimmung werden die Parameter dieser zusätzlichen Schicht(en) von Grund auf neu erlernt und in der Regel werden auch die Parameter des vortrainierten BERT-Modells leicht angepasst, um die spezifischen Anforderungen der Aufgabe zu erfüllen. Im Gegensatz dazu dient die TF-IDF-Textrepräsentation lediglich als Input, ohne dass sie selbst modifiziert wird. Beim Textclustering werden TF-IDF und BERT-Embeddings nicht verändert.

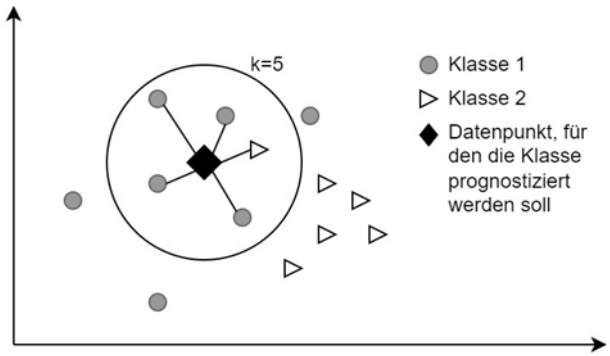
5.3.1 Textklassifikation

Zur Textklassifikation werden vordefinierte Kategorien einem Text zugeordnet. Als Algorithmen bietet die Temi-Box dafür K-Nearest Neighbors (KNN), binäre und multinomiale Klassifikatoren.

K-Nearest Neighbors

K-Nearest Neighbors (KNN) ist ein einfaches Verfahren, das in der Textklassifikation häufig eingesetzt wird (Cover/Hart 1967; Kowsari et al. 2019).

Abbildung 11: Grundidee des K-Nearest Neighbors Algorithmus



Anmerkung: Die Grundidee von KNN mit k = 5. Jeder Datenpunkt repräsentiert einen Text, Datenpunkte mit unterschiedlichen Formen repräsentieren verschiedene Klassen.
Quelle: Darstellung in Anlehnung an Kowsari et al. (2019) und Li et al. (2022)

Wie Abbildung 11 veranschaulicht, besteht die Grundidee darin, dass der KNN-Algorithmus die k ähnlichsten Nachbarn zu einem Datenpunkt identifiziert und aus den Häufigkeiten der Nachbarklassen die Klasse des Datenpunkts ableitet. Die Schritte dazu sind (Taha et al. 2024):

1. Auswahl einer Distanzmetrik zur Berechnung der Ähnlichkeit zwischen Textdokumenten, z. B. die Kosinus-Ähnlichkeit.
2. Finden der k nächsten Nachbarn unter den Trainingsbeispielen zu einem gegebenen Textdokument. k ist ein Parameter, der zur Verbesserung der Leistung angepasst werden kann. In der Temi-Box liegt k standardmäßig bei 5.

3. Bestimmen der Klasse des Textdokuments aus der Mehrheit der Nachbarklassen. Bei Unentschiedenheit werden Strategien wie die Wahl der Klasse des nächsten Nachbarn oder eine gewichtete Abstimmung angewandt.
4. Parameter Tuning: Die Leistung wird durch den Wert von k und die Distanzmetrik beeinflusst (Gasparetto et al. 2022; Kowsari et al. 2019). Die beiden Parameter können durch Kreuzvalidierung optimiert werden.
5. Wenn die Klassen sehr ungleichmäßig verteilt sind, tendiert KNN dazu, Klassen mit vielen Daten stärker zu berücksichtigen (Li et al. 2022). Deshalb können bei unbalancierten Klassen Techniken wie das Weighted KNN, bei dem die Nachbarn je nach ihrer Klasse unterschiedlich gewichtet werden, effektiver sein. Die Temi-Box lässt sich dazu erweitern.

KNN ist ein einfaches und häufig verwendetes Verfahren zur Textklassifikation, das sich auch zur Klassifikation mit mehreren Klassen gut eignet. An seine Grenzen kommt KNN bei großen Datenmengen und einer großen Anzahl an Klassen (Gasparetto et al. 2022).

Binärer Klassifikator

Die Textklassifikation mit BERT erfolgt typischerweise durch die Feinabstimmung eines vortrainierten BERT-Modells (Paaß/Giesselbach 2023). Dazu wird das BERT-Modell um eine oder mehrere zusätzliche Schichten erweitert, wobei beim binären Klassifikator die letzte Schicht aus einem logistischen Klassifikator besteht. Als Input verwenden die zusätzlichen Schichten die Output-Embeddings des vortrainierten BERT-Modells.

Ein binärer Klassifikator ist darauf ausgelegt, zwischen zwei möglichen Ausgängen zu unterscheiden. Dazu berechnet er die Wahrscheinlichkeit, dass ein Text zu einem der beiden Ausgänge gehört, und trifft eine Entscheidung basierend auf einem Schwellenwert (oft 0,5). Bei der Multi-Label-Klassifizierung gibt der Klassifikator eine Wahrscheinlichkeit für jede einzelne Klasse zurück, dass sie zutreffend ist, daher wird er auch als „one-vs-all“ oder „one-vs-rest“ bezeichnet. Diese Wahrscheinlichkeiten werden genutzt, um eine Rangfolge der Klassen zu erstellen und daraus die relevanten Klassen für einen bestimmten Text zu identifizieren.

Neben BERT-Embeddings ermöglicht die Temi-Box auch TF-IDF-Embeddings als Input zu verwenden.

Multinomialer Klassifikator

Die Textklassifikation mit einem multinomialen Klassifikator nutzt eine Architektur analog zum binären Klassifikator. Dabei wird das vortrainierte BERT-Modell um eine oder mehrere zusätzliche Schichten erweitert, wobei die letzte Schicht aus einem multinomialen Klassifikator besteht. Dieser Klassifikator ist darauf ausgelegt, zwischen mehr als zwei Klassen zu unterscheiden und wird eingesetzt, wenn das Textklassifikationsproblem mehrere mögliche Klassen umfasst. Der multinomiale Klassifikator berechnet die Wahrscheinlichkeit, mit der ein Text zu jeder der möglichen Klassen gehört, und weist dem Text die Klasse mit der höchsten Wahrscheinlichkeit zu.

Im Gegensatz zur binären Klassifikation trifft der multinomiale Klassifikator Entscheidungen für alle Klassen gleichzeitig. Dies führt zu einer erhöhten Komplexität des Modells und einer größeren Anzahl an trainierbaren Parametern. Wenn Klassen hinzugefügt oder entfernt werden,

ist in der Regel ein erneutes Training des Klassifikators erforderlich, da die Wahrscheinlichkeitsverteilung über alle Klassen neu berechnet werden muss.

Ein wesentlicher Vorteil des multinomialen Ansatzes ist die konsistente Behandlung der Klassenbeziehungen, da alle Klassen gleichzeitig optimiert werden. Dies kann zu einer höheren Gesamtgenauigkeit führen, insbesondere wenn die Klassen stabil sind und sich nicht häufig ändern. Die Entscheidung zwischen einem binären und einem multinomialen Klassifikator hängt daher stark von der Stabilität der Klassen und den spezifischen Anforderungen der Aufgabenstellung ab.

5.3.2 Textclustering

Textclustering kann verwendet werden, um Gruppen von Dokumenten mit ähnlichem Inhalt zu finden (Allahyari et al. 2017; Gaikwad/Chaugule/Patil 2014). Das Clustering von Texten kann auf verschiedenen Granularitätsebenen erfolgen, wobei die Cluster aus Dokumenten, Absätzen, Sätzen oder Begriffen bestehen können. Die Ähnlichkeit wird mit Hilfe einer Ähnlichkeitsfunktion berechnet. Je ähnlicher der Inhalt der Dokumente innerhalb eines Clusters und je unähnlicher der Inhalt zwischen den Clustern ist, desto besser ist die Qualität des Clustering.

Im Unterschied zur Textklassifikation braucht das Textclustering keine gelabelten Daten. Zum Clustern von Textdaten können eine Reihe von Algorithmen verwendet werden (Allahyari et al. 2017; Xu/Wunsch 2005). In der Temi-Box steht mit k-Means ein gängiger und effizienter Algorithmus zur Verfügung, der mit TF-IDF und BERT kombiniert werden kann (Petukhova/Matos-Carvalho/Fachada 2025; Subakti/Murfi/Hariadi 2022; Xu/Gu/Ji 2024).

k-Means

k-Means ist ein weit verbreiteter Clustering Algorithmus (Allahyari et al. 2017; Hotho/Nürnberger/Paaß 2005; Petukhova/Matos-Carvalho/Fachada 2025). Zum Clustern von Dokumenten gruppiert k-Means die Dokumente in k Cluster, indem die Summe der Quadrate innerhalb des Clusters, d. h. die Varianz, minimiert wird.

Der K-Means Algorithmus arbeitet in der Trainingsphase in mehreren Schritten: Zunächst werden k zufällige Datenpunkte als Clusterzentren, die Centroide, initialisiert. Anschließend werden die Dokumente den Centroiden mit der größten Ähnlichkeit zugewiesen. Danach werden die Cluster Centroiden neu berechnet. Dieser Prozess wird wiederholt, bis sich die Clusterzentren nicht mehr verändern oder eine vorher festgelegte Anzahl von Iterationen erreicht ist.

k-Means ist besonders für große Datensätze gut geeignet. Die Performance hängt allerdings stark von der Initialisierung der Centroiden und dem a priori festgelegten Wert von k ab.

5.4 Evaluationsmetriken

Evaluationsmetriken ermöglichen, die Qualität verschiedener Methoden oder die Auswirkungen unterschiedlicher Parametereinstellungen bei der Feinabstimmung zu vergleichen. Je nach Anwendungsfall können verschiedene Metriken geeignet sein, um die Leistung eines Textklassifikations- oder Textclusteringmodells zu bewerten. Daher stellt die Temi-Box eine Vielzahl von Evaluationsmetriken zur Verfügung, die im Folgenden näher beschrieben werden.

5.4.1 Textklassifikation

Welche Metriken zur Textklassifikation berechnet werden können, hängt davon ab, wie viele Labels einem Text zugeordnet werden (Li et al. 2022). Dementsprechend lassen sich Metriken für Single-Label und Multi-Label-Klassifikationen unterscheiden.

Wenn einem Text genau ein Label zugeordnet werden kann, können diese Single-Label-Metriken verwendet werden:

- Confusion Matrix: Die Confusion Matrix ist eine Tabelle, die die Anzahl der korrekten und fehlerhaften Klassifikationen für jede Klasse darstellt.

Tabelle 1: Confusion Matrix zur Evaluation eines binären Klassifikationsmodells

		Tatsächliche Labels		
		positive	negative	
Vorhergesagte Labels	Positive	True positive (tp)	False positive (fp)	$Precision = \frac{tp}{tp + fp}$
	negative	False nagative (fn)	True negative (tn)	
		$Recall(TPR) = \frac{tp}{tp + fn}$	$FPR = \frac{fp}{fp + tn}$	$Accuracy = \frac{tp + tn}{tp + fp + tn + fn}$

Anmerkung: Die Werte für Accuracy, Precision, Recall (True Positive Rate TPR) und False Positive Rate (FPR) lassen sich daraus ableiten.
Quelle: nach Jurafsky/Martin (2023)

- Accuracy: Der Anteil der korrekten Klassifikationen an der Gesamtanzahl der Klassifikationen wird als Accuracy bezeichnet. Die Accuracy ist ein leicht verständliches Maß. Sie kann allerdings nur dann sinnvoll interpretiert werden, wenn die Klassen ausgewogen sind, also ähnlich oft vorkommen (Jurafsky/Martin 2023).

Sind die Klassen nicht ausgewogen, sind die Metriken Precision, Recall, F1-Score und ROC-AUC-Score aussagekräftiger.

- Precision: Die Precision gibt den Anteil der korrekt positiven Klassifikationen an allen positiven Klassifikationen an. Die Precision ist wichtig, wenn die positive Vorhersage korrekt sein soll.
- Recall: Der Recall misst den Anteil der korrekt positiven Klassifikationen an allen tatsächlich positiven Klassifikationen. Der Recall ist von Bedeutung, wenn möglichst viele positive Labels identifiziert werden sollen. Der Recall wird deshalb auch als Sensitivität oder True Positive Rate (TPR) bezeichnet.
- F1-Score: Um Precision und Recall in einem Wert abbilden zu können und eine ausgewogene Klassifikationsmetrik zu erhalten, wurde der F1-Score entwickelt (Rijsbergen 1979). Der F1-Score ist ein harmonisches Mittel zwischen Precision und Recall:

$$F1\text{-}Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Wenn Precision und Recall gleichzeitig möglichst hoch sein sollen, ist der F1-Score wesentlich.

- ROC-AUC-Score: Der ROC-AUC-Score berechnet die Fläche unterhalb der Receiver Operation Characteristic (ROC)-Kurve (Fawcett 2006; Saito/Rehmsmeier 2015). Diese Kurve stellt die

True Positive Rate gegen die False Positive Rate für verschiedene Schwellenwerte dar. Ein Klassifikator mit zufälliger Leistung erzeugt eine diagonale Linie von (0, 0) nach (1, 1) auf der ROC-Kurve. Die Fläche unter der ROC-Kurve (AUC) beträgt 0,5 für einen zufälligen Klassifikator. Im Gegensatz dazu erreicht ein perfekter Klassifikator eine AUC von 1,0 (Hanley/McNeil 1982). Bei stark unausgeglichenen Klassenverteilungen ist der Einsatz des ROC-AUC-Score nicht empfehlenswert (Saito/Rehmsmeier 2015).

Wenn einem Text mehrere Labels gleichzeitig zugeordnet werden können, sind Multi-Label-Metriken besonders geeignet, um die Qualität der Textklassifikation zu bewerten (Tarekegn/Giacobini/Michalak 2021). In unserem Anwendungsfall, der sich mit der Zuordnung von Themen zu Publikationen befasst, erhalten die Nutzerinnen und Nutzer k Themenvorschläge, die als Top-k Prognosen oder Top-k Vorschläge bezeichnet werden. Diese Vorschläge können anschließend den entsprechenden Publikationen zugeordnet werden. Für die Evaluation ist es entscheidend, nicht die Qualität aller Prognosen insgesamt zu betrachten, sondern sich auf die Qualität der vorgeschlagenen Top-k-Prognosen zu konzentrieren (Schröder/Thiele/Lehner 2011). Aus diesem Grund bietet die Temi-Box spezifische Metriken zur Bewertung der Top-k Prognosen:

- Precision@k: Precision@k misst den Anteil der relevanten Treffer innerhalb der Top-k Prognosen an allen Top-k Prognosen, gemittelt über alle Instanzen. In unserem Anwendungsfall gibt Precision@10 an, wie viele der Top-10 Vorschläge tatsächlich relevante Themen sind.
- Recall@k: Der Recall@k misst den Anteil der Treffer innerhalb der Top-k Prognosen an allen tatsächlich zutreffenden Klassifikationen, gemittelt über alle Instanzen. In unserem Anwendungsfall gibt beispielsweise Recall@10 an, wie viele der tatsächlich relevanten Themen in den Top-10 Vorschlägen enthalten sind.

5.4.2 Textclustering

Zur Evaluation von Textclustering gibt es mehrere Metriken, die verwendet werden können, um die Qualität und Effektivität der Cluster zu bewerten. Eine gängige Metrik ist der Davies-Bouldin Index (Davies/Bouldin 1979). Dieser Index gibt die durchschnittliche Ähnlichkeit zwischen den Clustern an, wobei die Ähnlichkeit ein Maß ist, das den Abstand zwischen den Clustern mit der Größe der Cluster selbst vergleicht. Ein niedrigerer Davies-Bouldin-Index weist auf ein Modell mit besserer Trennung zwischen den Clustern hin, wobei Null der niedrigste mögliche Wert ist.

6 Experimente

In unseren Experimenten verwenden wir den Datensatz „IABPub105“, der ursprünglich für die Franconian Data Science League im April 2023 erstellt wurde. Für diesen Hackathon zur Förderung und Vernetzung von Data Scientists in der fränkischen Region erstellten wir den vereinfachten Datensatz mit IAB-Publikationen und stellten die Themenzuordnung als zentrale Aufgabe. Eine detaillierte Beschreibung der Erstellung und Charakterisierung des Datensatzes findet sich in Abschnitt 6.1.

Für die Entwicklung der Temi-Box haben wir uns intensiv mit den Anwendungsfällen der Verschlagwortung von Publikationen und deren Themenzuordnung auf der IAB-Infoplattform auseinandergesetzt (siehe Abschnitt 2). Beide Szenarien stellen aus methodischer Sicht Multi-Label-Klassifikationsprobleme dar, deren Lösungsansätze methodisch analog verlaufen.

Für unsere Experimente haben wir uns aus mehreren Gründen entschieden, nur die Themenzuordnung darzustellen. Das basiert auf mehreren Überlegungen: Erstens bietet die Themenzuordnung eine geringere Anzahl an Labels (105 Themen im Vergleich zu 2.114 Schlagwörtern im Datensatz IABPub105) und eine niedrigere durchschnittliche Anzahl an Labels pro Publikation (etwa 3 Themen gegenüber 10 Schlagwörtern). Dies führt zu deutlich kürzeren Trainingszeiten. Zweitens war es für die Themenzuordnung ausreichend, Abstracts und Titel der Publikationen zu verwenden, während die Verschlagwortung den Zugriff auf die Volltexte vorausgesetzt hätte. Viele Schlagwörter lassen sich nur über Textbausteine aus dem Volltext zuordnen. Eine alleinige Nutzung der Abstracts würde deshalb zu Leistungseinbußen führen. Die Aufbereitung von Volltexten ist jedoch aufwendiger und bietet keinen zusätzlichen Mehrwert für unsere Experimente. Drittens ist die Komplexität der Themenzuordnung geringer, da Schlagwörter eine größere Bandbreite an Dimensionen abdecken. Diese umfassen beispielsweise Begriffe aus verschiedenen Wissenschaftsdisziplinen, Wirtschaft, Betrieb, Ausbildung und Beruf sowie unterschiedliche Personengruppen und geografische Schlagwörter. Diese Vielschichtigkeit der Schlagwörter macht eine präzise Zuordnung deutlich schwieriger.

Nach der detaillierten Beschreibung des Datensatzes IABPub105 stellen wir unsere Experimente dar, die sowohl die Klassifikation als auch das Clustering von Texten umfassen und speziell auf die Nutzung der Temi-Box ausgerichtet sind. Wir vergleichen Evaluationsmetriken hinsichtlich verschiedener Schwellenwerte, analysieren die Trainingszeiten unterschiedlicher Modelle und evaluieren die Performance verschiedener Algorithmen, um die besten Ansätze für die Themenzuordnung und das Clustering zu identifizieren.

6.1 Datensatz „IABPub105“

Für unsere Experimente nutzten wir einen vereinfachten Teil-Datensatz von IAB-Publikationen aus der IABinfoplattform, den wir als „IABPub105“ bezeichnen.

Aus der gesamten Datengrundlage der IABinfoplattform wählten wir für den Datensatz Publikationen mit IAB-(Mit-)Autorenschaft und verfügbaren Volltexten aus. Letztlich verzichteten wir jedoch auf die Nutzung der Volltexte, da der potenzielle Erkenntnisgewinn den erheblichen Mehraufwand für die Textaufbereitung und die verlängerten Trainingszeiten nicht rechtfertigte. Da die Trainingsdauer linear von der Datenmenge abhängt und Volltexte deutlich umfangreicher als Abstracts sind, hätte sich die Trainingszeit entsprechend verlängert.

Im nächsten Schritt vereinfachten wir den Datensatz, indem wir die Themen auf diejenigen beschränkten, die tatsächlich in den ausgewählten Publikationen präsent und somit lernbar waren. Darüber hinaus konzentrierten wir uns auf Themen, die den Themendossiers entsprachen, und schlossen damit die weitverzweigten Unterthemen aus.

Da die Publikationen häufig mit mehreren Themen-Labels versehen sind, handelt es sich um eine Multi-Label-Klassifikationsaufgabe. Der resultierende Datensatz umfasst insgesamt 1.932

Publikationen, die 105 verschiedenen Themen zugeordnet sind. Die Anzahl der Themenzuordnungen variiert erheblich, wie in den Auszählungen in Tabelle 2 deutlich wird.

Tabelle 2: Auszählungen zum Datensatz IABPub105

	Trainingsdatensatz	Testdatensatz	Gesamt
	Anzahl	Anzahl	Anzahl
Publikationen	1.743	189	1.932
Themen	105	82	105
Anzahl Themen pro Publikation			
Durchschnitt	1,8	2,5	1,9
Minimum - Maximum	1 - 13	1 - 21	1 - 21

Quelle: IABPub105, Datenauszug aus der IABinfoplattform vom Februar 2023. Eigene Berechnung.

Jedes Thema ist durch eine eindeutige ID, Titel und Themenbeschreibung gekennzeichnet, während jede Publikation eine eindeutige ID, Titel und Abstract besitzt.

Bei der Erstellung des Trainingsdatensatzes achteten wir darauf, dass alle Themen mindestens einmal vertreten sind. Zudem stellten wir sicher, dass die Publikationen des Testdatensatzes nicht im Trainingsdatensatz enthalten sind, um eine klare Trennung zu gewährleisten.

6.2 Anwendungsfall 1: Klassifikation

Im Anwendungsfall "Klassifikation" besteht das Ziel unserer Prognosen darin, eine Liste mit genau 10 Themen vorzuschlagen, aus der eine Person die passenden auswählen kann. Die Festlegung auf eine definierte Themenanzahl wurde in Abstimmung mit Expertinnen und Experten zur Themenzuordnung für die IAB-Infoplattform getroffen, um eine überschaubare und relevante Themenauswahl zu gewährleisten.

6.2.1 Methoden

Für die Experimente zur Klassifikation vergleichen wir verschiedene Methoden und Konfigurationsmöglichkeiten:

- Embedding[1]
  - TF-IDF
  - BERT mit den folgenden vortrainierten Modellen
    - DistilBERT
    - BERT-base-german-cased (im Folgenden „GermanBERT“ abgekürzt)
- Klassifikations-Algorithmus[2]
  - KNN
  - binärer Klassifikator und multinomialer Klassifikator jeweils mit unterschiedlichen Layer-Konfigurationen

[1] Für weiterführende Informationen zu den Embedding-Methoden siehe Abschnitt 5.2.

[2] Für eine detaillierte Darstellung der Klassifikations-Algorithmen zur Text Klassifikation siehe Abschnitt 5.3.1.

Insgesamt resultieren aus diesen Kombinationen 139 unterschiedliche Modelle.

Da wir im Anwendungsfall immer genau zehn Themen vorschlagen, vergleichen wir die Ergebnisse der Modelle anhand dieser Evaluationsmetriken[3]:

- Recall@10 und
- teilweise Precision@10 sowie F1-Score@10.

Für unseren Anwendungsfall ist der Recall-Wert entscheidend. Der Recall bezieht sich auf die Fähigkeit des Modells, tatsächlich relevante Themen korrekt aus der Gesamtheit der relevanten Themen zu identifizieren. Er setzt also die Anzahl der tatsächlich relevanten Themen in der Liste der zehn vorgeschlagenen Themen in Bezug zur Gesamtzahl der relevanten Themen. Ein hoher Recall-Wert bedeutet, dass das Modell in der Lage ist, einen großen Anteil der relevanten Themen in der Vorschlagsliste zu erfassen. Ein niedriger Recall-Wert würde darauf hindeuten, dass viele relevante Themen nicht in der Liste der vorgeschlagenen Themen enthalten sind.

Die Konsequenzen, die sich aus der Fokussierung auf genau zehn Vorschläge für Precision und F1-Score ergeben, werden im folgenden Abschnitt näher erläutert.

6.2.2 Verschiedene Schwellenwerte im Vergleich

In der Themenzuordnung von Publikationen spielt die Wahl des Schwellenwerts eine wichtige Rolle für die Evaluationsmetriken wie Precision, Recall und F1-Score. Ein Klassifikator weist jedem Thema für jede Publikation einen Prognosewert zwischen 0 und 1 zu. Ein hoher Prognosewert deutet darauf hin, dass das Thema der Publikation zugeordnet werden sollte, während ein niedriger Wert das Gegenteil signalisiert. Der Schwellenwert bestimmt, ab welchem Prognosewert ein Thema als relevant betrachtet wird.

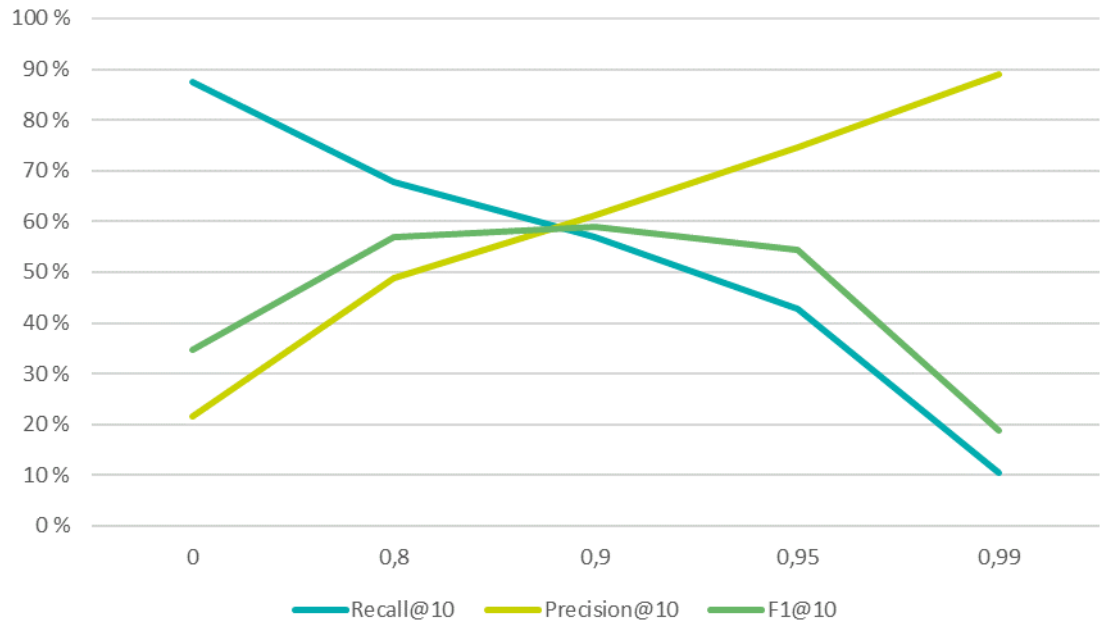
In unserem Anwendungsfall, bei dem genau zehn Themen vorgeschlagen werden sollen, wird der Schwellenwert auf 0 gesetzt. Dies bedeutet, dass die zehn Themen mit den höchsten Prognosewerten ausgewählt werden, unabhängig davon, wie niedrig diese Werte sind. Diese Herangehensweise maximiert den Recall, da alle relevanten Themen mit den höchsten Prognosewerten berücksichtigt werden. Allerdings führt dies zu einer geringeren Precision, da – wie Tabelle 2 zeigt – im Durchschnitt nur 2,5 Themen tatsächlich relevant sind und somit 7,5 falsche Prognosen entstehen.

Die Anpassung des Schwellenwerts hat gegenläufige Effekte auf die Evaluationsmetriken. Abbildung 12 illustriert den Effekt unterschiedlicher Prognoseschwellenwerte auf die Evaluationsmetriken@10.

[3] Für weiterführende Informationen zu den Evaluationsmetriken siehe Abschnitt 5.4.

Abbildung 12: Evaluationsmetriken für unterschiedliche Prognoseschwellenwerte

Prognoseschwellenwerte, Evaluationsmetriken@10



Anmerkung: Die Metriken wurden für das Modell mit GermanBERT-Embedding, binärem Klassifikator und zwei Layern mit jeweils 256 Neuronen ermittelt. Es wurden immer 10 Themen prognostiziert. Den höchsten Recall erzielt das Modell mit einem Schwellenwert von 0. Die höchste Precision erzielt es mit einem Schwellenwert maximal nahe an 1[4] – hier werden sehr wenige Themen zu Publikationen zugeordnet. Deshalb ist dieser Schwellenwert für unseren Anwendungsfall nicht zielführend.

Quelle: IABPub105 mit 3.557 Datenzeilen. Eigene Berechnung.

Mit einem höheren Schwellenwert werden insgesamt weniger Themen den Publikationen zugeordnet. Dies führt zu einem Rückgang des Recalls, da auch weniger korrekte Zuordnungen gemacht werden. Gleichzeitig steigt die Precision, da weniger falsche Zuordnungen beobachtet werden. Der F1-Score, der das harmonische Mittel von Precision und Recall darstellt, erreicht sein Maximum an dem Punkt, an dem Recall und Precision gleich sind.

Für die weiteren Experimente bleibt der Schwellenwert bei 0, um immer zehn Themenvorschläge zu einer Publikation zu gewährleisten, auch wenn dies bedeutet, dass die Präzision leidet.

6.2.3 Trainingsdauer verschiedener Methoden im Vergleich

Für das Training verwenden wir einen Server mit einer NVIDIA RTX 6000 GPU mit 24 GB. Um den zeitlichen Aufwand verschiedener Experimente vergleichen zu können, haben wir in Tabelle 3 eine Übersicht der ungefähren Trainingsdauern für verschiedene Konfigurationen zusammengestellt.

[4] Damit überhaupt eine positive Zuordnung durch das Modell erfolgt, muss der Schwellenwert unterhalb von 1 liegen. Der maximale Score, den der Klassifikator berechnet, liegt bei 99,99%.

Tabelle 3: Vergleich der Trainingsdauer verschiedener Experiment-Konfigurationen

Embedding	Modell	Klassifikations-Algorithmus	Trainingsdauer Ungefähr in Min.
TF-IDF	-	KNN	10
TF-IDF	-	binary	60
TF-IDF	-	multinomial	25
BERT	GermanBERT	binary	70
BERT	GermanBERT	multinomial	25
BERT	DistilBERT	binary	60
BERT	DistilBERT	multinomial	20

Quelle: Eigene Recherche.

Die kürzeste Trainingsdauer wurde mit der Kombination aus TF-IDF und einem KNN-Klassifikator erreicht, da der KNN-Algorithmus weniger komplexe Berechnungen erfordert und somit besonders schnell ist.

Im Gegensatz dazu benötigen die Experimente mit binären Klassifikatoren die längste Trainingszeit. Dies liegt daran, dass den binären Klassifikatoren im Trainingsmodus zu jeder Publikation sowohl k richtige als auch k falsche Themen präsentiert werden, was mehr Informationen und somit mehr Verarbeitungszeit und Speicher erfordert. Da die Auswahl der positiven und negativen Themen zufällig erfolgt, werden binäre Klassifikatoren üblicherweise länger trainiert, um dem Modell eine größere Vielfalt an nützlichen Beispielen zu bieten.

Experimente mit dem multinomialen Klassifikator liegen zeitlich zwischen KNN und den binären Klassifikatoren. Der multinomiale Klassifikator berechnet ein einziges Modell, das die Wahrscheinlichkeiten für alle Themen gleichzeitig abbildet. Obwohl dieses Modell komplexer ist, wurde es in unseren Experimenten schneller berechnet als die binären Klassifikatoren, da es nur anhand der Publikationen mit seinen k richtigen Themen trainiert wird.

Es ist wichtig zu beachten, dass die Laufzeiten innerhalb der Konfigurationen auch von der Anzahl der Layer und Neuronen abhängen. Eine höhere Anzahl an Layern oder Neuronen kann die Komplexität des Modells erhöhen, was wiederum die Trainingsdauer verlängert. Darüber hinaus spielt die Hardware-Ausstattung eine entscheidende Rolle, insbesondere leistungsfähigere Prozessoren und größere Arbeitsspeicher können die Berechnungen beschleunigen.

Trainingsdauern lassen keine direkten Rückschlüsse auf die Zeit zu, die für die Vorhersage benötigt wird. In der Inferenzphase, in der das Modell Vorhersagen auf Basis neuer Daten trifft, ist beispielsweise zu erwarten, dass binärer und multinomialer Klassifikator ähnlich schnell arbeiten. Die Geschwindigkeit dieser Phase hängt davon ab, wie effizient das Modell die Eingabedaten verarbeiten kann, nicht davon, wie lange das Training gedauert hat.

6.2.4 Verschiedene Embeddings und Klassifikatoren im Vergleich

Um die Leistungsfähigkeit verschiedener Embeddings und Klassifikatoren zu vergleichen, ziehen wir nicht alle 139 möglichen Konfigurationen heran, sondern konzentrieren uns auf die jeweils

besten Modelle[5]. In unserer Analyse betrachten wir also zwei BERT-Embeddings und ein TF-IDF-Embedding jeweils mit einem binären und multinomialen Klassifikator sowie ein TF-IDF-Embedding mit dem KNN- Klassifikator. Tabelle 4 gibt einen Überblick über die analysierten Konfigurationen, ihre jeweils beste Netzwerkarchitektur (sofern möglich) und ihre Leistung.

Tabelle 4: Konfigurationen der für den Vergleich verschiedener Embeddings und Klassifikatoren

Beste Konfigurationen verschiedener Klassifikatoren und Embeddings anhand des Recall@10

Embedding	Modell	Klassifikations-Algorithmus	Anzahl Layer	Beste Layer-Konfiguration	Recall@10
BERT	GermanBERT	binary	1	4096	87 %
BERT	DistilBERT	binary	1	32	86 %
BERT	GermanBERT	multinomial	1	4096	75 %
BERT	DistilBERT	multinomial	1	4096	73 %
TF-IDF	-	multinomial	2	4096, 4096	70 %
TF-IDF	-	binary	2	4096, 4096	63 %
TF-IDF	-	KNN	-	-	51 %

Anmerkung: In der Tabelle sind die besten Konfigurationen für den Recall@10 aufgeführt. Demnach erzielt beispielsweise das GermanBERT-Embedding mit binärem Klassifikator und einem Layer mit 256 Neuronen schlechtere Ergebnisse als der hier aufgeführte mit zwei Layern mit je 256 Neuronen.

Quelle: IABPub105 mit 3.557 Datenzeilen. Eigene Berechnung.

Der Vergleich verschiedener Konfigurationen anhand des Recall@10 in Abbildung 13 verdeutlicht die Überlegenheit von BERT-basierten Modellen. BERT-Embeddings - sowohl GermanBERT als auch DistilBERT - zeigen insgesamt höhere Recall@10-Werte im Vergleich zu den TF-IDF-basierten Ansätzen. Dies deutet darauf hin, dass die kontextuelle Einbettung von BERT-Modellen effektiver der TF-IDF-basierte Ansatz ist, um relevante Themen in den Top-10-Ergebnissen zu identifizieren. Zudem zeigt sich der Einfluss Klassifikationsart. Die binäre Klassifikation erzielt bei BERT-Modellen höhere Recall-Werte als die multinomiale Klassifikation. Die Auswahl des passenden Embeddings und Klassifikators ist somit von zentraler Bedeutung für die Leistungsfähigkeit der Modelle.

6.2.5 Verschiedene Netzwerkarchitekturen im Vergleich

Um den Einfluss der Netzwerkarchitektur des Klassifikators zu untersuchen, haben wir die Performance von BERT- und TF-IDF-Modellen für verschiedenen Layer- und Neuronenzahlen analysiert. Wir haben mit diesen Varianten experimentiert:

- 1 oder 2 Layer mit einer Auswahl von 16, 32, 256, 1024 oder 4096 Neuronen
- 3 Layer testweise mit folgenden Anzahlen an Neuronen
  - 32, 32, 32
  - 16, 1024, 4096
  - 4096, 1024, 16
  - 4096, 16, 1024
  - 1024, 4096, 16

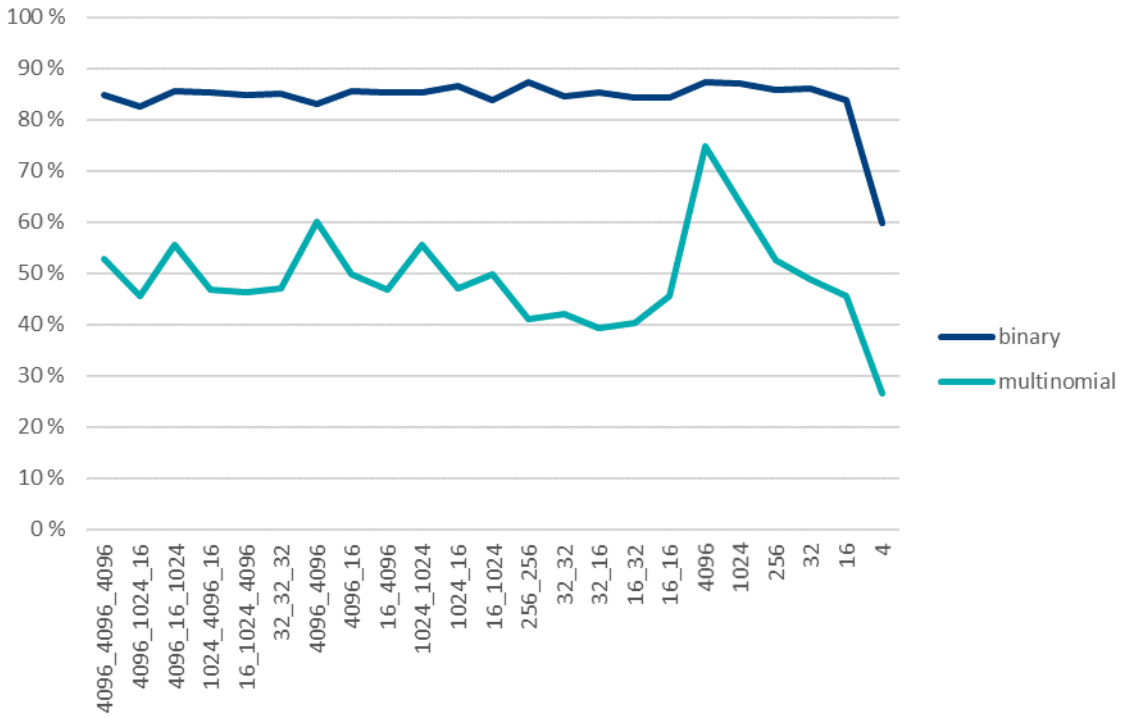
[5] Ein Überblick über die Qualität aller 139 möglichen Konfigurationen ist in „Anhang 1: Übersicht über die Metriken aller Klassifikationsexperimente“ zu finden.

– 4096, 4096, 4096

Abbildung 13 illustriert die Ergebnisse für den Recall@10 unter Verwendung des GermanBERT-Embeddings mit sowohl binären als auch multinomialen Klassifikatoren.

Abbildung 13: Performance verschiedener Netzwerkarchitekturen zur Feinabstimmung von GermanBERT

Netzwerkarchitektur jeweils mit Zahl der Neuronen pro Schicht, Recall@10



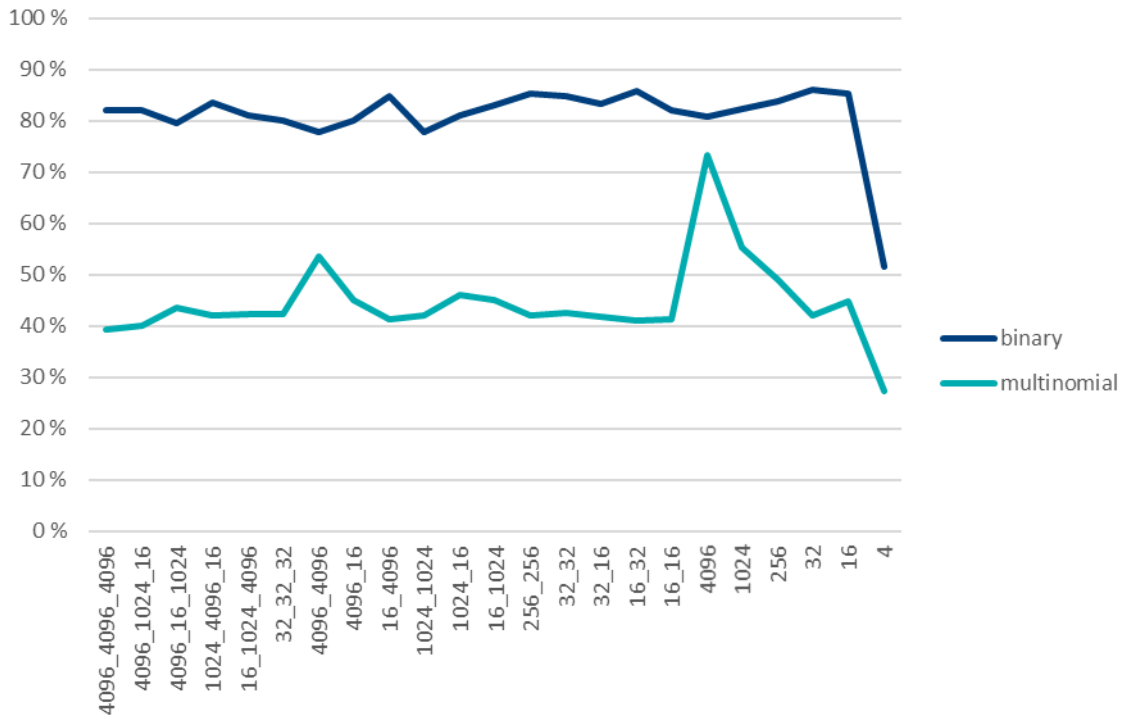
Quelle: IABPub105 mit 3.557 Datenzeilen. Eigene Berechnung.

Die Analyse zeigt, dass mehr als ein Layer für beide Klassifikatoren keine signifikanten Verbesserungen bringt, da das vortrainierte BERT-Modell offenbar bereits die wesentlichen Informationen enthält. Die Ergebnisse des multinomialen Klassifikators liegen deutlich unter denen des binären Klassifikators. Innerhalb der Konfigurationen mit binären Klassifikatoren sind die Unterschiede im Recall@10 bei variierender Anzahl von Neuronen und Layern gering – lediglich die Konfiguration mit einem Layer und vier Neuronen erweist sich als unzureichend.

Der multinomiale Klassifikator hingegen braucht ein komplexeres Modell für eine gute Performance. Dies spiegelt sich in den Ergebnissen wider: Bei einem Layer verbessert sich die Leistung mit der Anzahl der Neuronen, sodass die Konfiguration mit einem Layer und 4096 Neuronen den anderen deutlich überlegen ist.

Abbildung 14 zeigt die Ergebnisse für den Recall@10 unter Verwendung des DistilBERT-Embeddings mit sowohl binären als auch multinomialen Klassifikatoren.

Abbildung 14: Performance verschiedener Netzwerkarchitekturen zur Feinabstimmung von DistilBERT
Netzwerkarchitektur jeweils mit Zahl der Neuronen pro Schicht, Recall@10

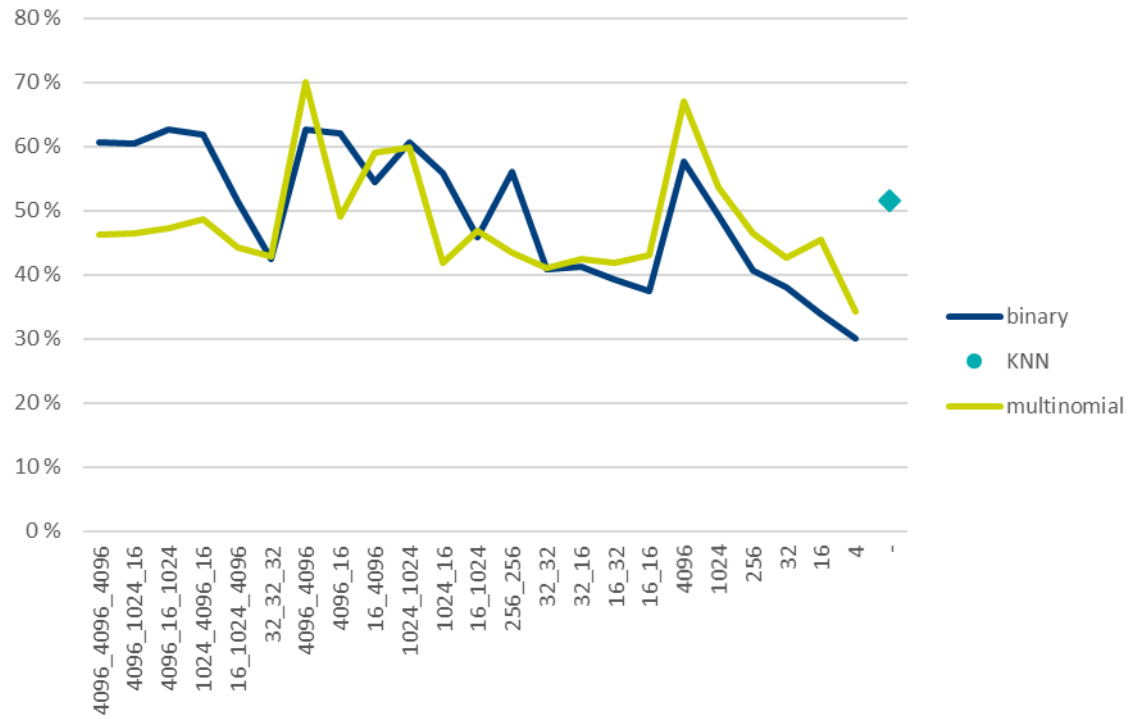


Quelle: IABPub105 mit 3.557 Datenzeilen. Eigene Berechnung.

Hier ist ein ähnliches Muster wie beim GermanBERT-Embedding erkennbar. Auch hier bringt die Konfiguration mit mehr als einem Layer keinen signifikanten Vorteil. Die Leistung des multinomialen Klassifikators bleibt ebenfalls hinter der des binären Klassifikators zurück. Bei einem Layer verbessert sich die Performance des multinomialen Klassifikators mit zunehmender Anzahl von Neuronen, also höherer Modellkomplexität. Im Gegensatz dazu erweist sich für den binären Klassifikator bereits eine Konfiguration mit 16 Neuronen als ausreichend, um gute Ergebnisse zu erzielen.

Abbildung 15 zeigt die Ergebnisse für den Recall@10 unter Verwendung des TF-IDF-Embeddings mit binären, KNN- und multinomialen Klassifikatoren.

Abbildung 15: Performance verschiedener Netzwerkarchitekturen auf Basis des TF-IDF-Embeddings
Netzwerkarchitektur jeweils mit Zahl der Neuronen pro Schicht, Recall@10



Anmerkung: KNN nutzt keine Netzwerkarchitektur. Daher ist die Architektur für KNN nicht beschrieben.
Quelle: IABPub105 mit 3.557 Datenzeilen. Eigene Berechnung.

Im Gegensatz zu den BERT-Embeddings zeigt sich hier ein deutlich anderes Bild. Insgesamt erzielen die multinomialen Klassifikatoren die besten Ergebnisse. Sowohl für binäre als auch für multinomiale Klassifikatoren gilt, dass eine hohe Anzahl von Neuronen zu besseren Ergebnissen führt. Diese Tendenz ist besonders beim binären Klassifikator ausgeprägt. Für Konfigurationen mit drei, zwei oder einem Layer liefert die Variante mit 4096 Neuronen jeweils die besten Ergebnisse, während die Leistung mit abnehmender Neuronenzahl sinkt. Beim multinomialen Klassifikator ist zusätzlich zu beobachten, dass eine Konfiguration mit drei Layern sehr schlechte Ergebnisse liefert. Dies deutet darauf hin, dass eine übermäßige Komplexität in der Netzwerkarchitektur nicht immer vorteilhaft ist und möglicherweise zu einer Überanpassung oder ineffizienten Nutzung der Modellressourcen führt. Der KNN-Klassifikator, der keine Netzwerkarchitektur nutzt, kann mit den leistungsstärksten Konfigurationen nicht mithalten, übertrifft jedoch die Konfigurationen mit nur wenigen Neuronen, was seine relative Stärke in einfacheren Szenarien zeigt.

Insgesamt zeigt sich, dass über alle Experimente hinweg die Konfiguration mit dem GermanBERT-Embedding, einem binären Klassifikator und einem Layer mit 4096 Neuronen die besten Ergebnisse erzielt. Diese Erkenntnisse unterstreichen die Bedeutung der sorgfältigen Auswahl und Anpassung der Netzwerkarchitektur, um die Leistungsfähigkeit der Modelle zu maximieren.

6.2.6 Interpretation

Die Experimente zur Klassifikation zeigen, wie die Wahl der Methoden und Konfigurationen die Modellleistung beeinflussen kann. Insgesamt haben wir 139 Modelle mit TF-IDF und BERT-Embeddings (GermanBERT und DistilBERT) sowie KNN, binären und multinomialen Klassifikatoren verglichen.

Um für unseren spezifischen Anwendungsfall immer zehn Vorschläge zu gewährleisten und den Recall zu maximieren, haben wir einen Schwellenwert von 0 gewählt. In unseren Experimenten konnten wir zeigen, dass die Wahl des Schwellenwerts entscheidend die Evaluationsmetriken beeinflusst. Ein Schwellenwert von 0 führt zu einer geringeren Precision, was für unseren Anwendungsfall akzeptabel war.

In Bezug auf die Trainingsdauer erwies sich die Kombination aus TF-IDF und KNN als am schnellsten, da KNN weniger komplexe Berechnungen erfordert. Binäre Klassifikatoren benötigten die längste Trainingszeit, da der Trainingsaufwand am höchsten ist.

Die Analyse der Klassifikatoren zeigt, dass die BERT-Embeddings GermanBERT und DistilBERT die besten Recall@10-Werte erzielen. Dies unterstreicht ihre Fähigkeit, kontextuelle Informationen effektiv zu nutzen. Dass GermanBERT leicht bessere Ergebnisse als das DistilBERT-Modell erzielt, lässt sich möglicherweise damit erklären, dass die Publikationen unseres Testdatensatzes ausschließlich deutsche Texte beinhalten. Bei mehrsprachigen Testdaten wäre eine bessere Leistung des DistilBERT-Modells zu erwarten.

Binäre Klassifikatoren schnitten bei BERT-Modellen besser ab als multinomiale. Dies ist darauf zurückzuführen, dass binäre Klassifikatoren sowohl den Text des Themas als auch den Text des Abstracts als Input erhalten und darauf basierend entscheiden, ob eine hohe Übereinstimmung vorliegt. Im Gegensatz dazu ordnen multinomiale Klassifikatoren die Themen den Texten zu, ohne den Text des Themas direkt zu berücksichtigen. Diese Vorgehensweise kann bei einer geringen Anzahl von Beobachtungen pro Thema, wie im aktuellen Datensatz, zu suboptimalen Ergebnissen führen. Es ist jedoch zu erwarten, dass sich die Leistung des multinomialen Klassifikators mit einer größeren Datengrundlage deutlich verbessern würde, da mehr Daten eine robustere Schätzung der Wahrscheinlichkeitsverteilungen über die Klassen ermöglichen.

Bei der Verwendung von TF-IDF-Embeddings erzielen ideal konfigurierte multinomiale oder binäre Klassifikatoren die besten Ergebnisse. Allerdings kann eine nicht ideale Konfiguration zu einer Leistungsminderung führen, die dazu führt, dass die Ergebnisse schlechter ausfallen als bei einem KNN-Klassifikator. Der KNN-Klassifikator, der auf einer einfacheren Methodik basiert, zeigt in solchen Fällen eine robustere Performance, da er weniger empfindlich auf suboptimale Parametereinstellungen reagiert.

Unabhängig von der Art des Embeddings besteht ein wesentlicher Vorteil des binären Klassifikators darin, dass die Liste der Labels flexibel und nicht statisch ist. Bei der Einführung eines neuen Labels kann das bestehende Modell für jedes vorhandene Dokument prüfen, ob dieses neue Label zutreffend ist. Dies ermöglicht eine dynamische Anpassung an sich ändernde Klassifikationsanforderungen, ohne dass das gesamte Modell neu trainiert werden muss. Im Gegensatz dazu sind KNN- und multinomiale Klassifikatoren auf die Labels beschränkt, die während des Trainingsprozesses vorhanden waren. Bei der Einführung eines neuen Labels ist es erforderlich, den gesamten Trainingsvorgang zu wiederholen, da die bestehenden Modelle nicht

für zusätzliche Labels verwendet werden können. Weiterführende Details zu den Klassifikatoren sind in Abschnitt 5.3.1 zu finden.

Die Untersuchung der Netzwerkarchitekturen verdeutlicht, wie die Anzahl der Layer und Neuronen die Performance der Modelle beeinflussen kann. Bei BERT-Embeddings, wie GermanBERT und DistilBERT, zeigt sich, dass – mit Ausnahme des kleinsten Netzes – die Verwendung von mehr als einem Layer keine signifikanten Verbesserungen bringt, da das vortrainierte Modell bereits wesentliche Informationen enthält. Im Gegensatz dazu profitiert das TF-IDF-Embedding von einem tieferen Netzwerk und einer größeren Anzahl an Neuronen, was zu besseren Ergebnissen führt.

6.3 Anwendungsfall 2: Clustering

6.3.1 Methoden

Für die Experimente stellen wir verschiedene Methoden und Konfigurationsmöglichkeiten gegenüber:

- Embedding[6]
  - TF-IDF
  - BERT mit den folgenden vortrainierten Modellen:
    - DistilBERT
    - BERT-base-german-cased (im Folgenden „GermanBERT“ abgekürzt)
    - BERT-base-german-cased finetuned mit Klassifizierungsproblem aus Abschnitt 6.2 (im Folgenden „GermanBERT-tuned“ abgekürzt)[7]
- Clustering-Algorithmus[8]
  - K-Means

Wir erhalten schließlich vier unterschiedliche Modelle, deren Ergebnisse wir quantitativ, qualitativ und grafisch vergleichen.

In unserem Anwendungsfall weist eine Publikation durchschnittlich 1,9 Themen auf (siehe Tabelle 2) und es ist zu erwarten, dass viele inhaltliche Überschneidungen vorhanden sind. Üblicherweise wird beim Clustering eine Einteilung in überschneidungsfreie Cluster vorgenommen. Dennoch ermöglicht das Gruppieren der Publikationen in eine geringe Anzahl von Clustern die Identifikation von Zusammenhängen auf einem hohen Abstraktionsniveau. Das Ziel des Clusterings besteht daher darin, die Themenschwerpunkte der Publikationen zu identifizieren. Für alle vier Modelle haben wir einheitlich vorgegeben, die ideale Anzahl an Clustern im Bereich von mindestens 3 bis maximal 10 Clustern zu bestimmen.

[6] Für weiterführende Informationen zu den Embedding-Methoden siehe Abschnitt 5.2.

[7] Im Training des Klassifizierungsproblems wird nicht nur der Klassifikator, sondern auch das Embedding trainiert. Während das GermanBERT-Embedding auf Basis der Zusammenhänge deutscher Sprache Vektoren erstellt, ist das GermanBERT-tuned Embedding noch speziell auf den Wortschatz der vorliegenden Dokumente und den Anwendungsfall Themenzuordnung angepasst.

[8] Für eine detaillierte Darstellung des Clustering-Algorithmus siehe Abschnitt 5.3.2.

6.3.2 Quantitativer Vergleich der Clustering-Methoden

Den quantitativen Vergleich der Clustering-Methoden erstellen wir anhand des Davies-Bouldin-Werts[9], der die durchschnittliche Ähnlichkeit zwischen Clustern bewertet. Ein niedrigerer Wert deutet auf eine bessere Trennung der Cluster hin. Null ist der niedrigste Wert, der möglich ist.

Die Ergebnisse der Experimente zeigen deutliche Unterschiede zwischen den verwendeten Embeddings (siehe Tabelle 5). Das feinabgestimmte GermanBERT-Modell („GermanBERT-tuned“) erzielte mit einem Davies-Bouldin-Wert von 2,65 die beste Leistung, was auf eine überlegene Clustertrennung hinweist. TF-IDF schnitt mit einem Davies-Bouldin-Wert von 3,96 am schlechtesten ab, was auf eine weniger effektive Clusterbildung im Vergleich zu den BERT-basierten Modellen hindeutet.

Tabelle 5: Vergleich des Clustering für verschiedene Embeddings anhand des Davies-Bouldin-Werts

Embedding	Davies-Bouldin-Wert
TF-IDF	3,96
DistilBERT	3,01
GermanBERT	3,22
GermanBERT-tuned	2,65

Quelle: IABPub105 mit 3.557 Datenzeilen. Eigene Berechnung.

6.3.3 Qualitativer Vergleich der Clustering-Methoden

Der qualitative Vergleich der Clustering-Methoden basiert auf der Analyse der häufigsten Themen innerhalb der gebildeten Cluster. Bei einer guten Clusterbildung erwarten wir, dass die Cluster unterschiedliche Themen aufweisen und die Themen innerhalb eines Clusters in einem inhaltlichen Zusammenhang stehen.

Die Ergebnisse in Tabelle 6 zeigen, wie die verschiedenen Embeddings die thematische Struktur der Publikationen erfassen.

Anhand der häufigsten Themen für jedes Cluster sehen wir, dass – wie erwartet - die Abgrenzung der Cluster mit TF-IDF-, DistilBERT- und GermanBERT-Embeddings schlechter funktioniert als mit dem GermanBERT-tuned-Embedding. Cluster 1 und 3 des TF-IDF-Embeddings haben Überschneidungen der gemeinsamen Top3-Themen mit Cluster 2. Das Thema „SGB II- …“ ist unter den Top3-Themen für jedes Cluster des DistilBERT-Embeddings und „Migration und Integration“ für jedes Cluster des GermanBERT-Embeddings, während in den 8 Clustern des GermanBERT-tuned-Embeddings nur ein Thema („Transformationsprozess in Ostdeutschland – Wirtschaft, Arbeitsmarkt und Gesellschaft“) zwei Mal in den Top3-Themen enthalten ist.

Ein weiteres Indiz für eine hohe Qualität ist eine gleichmäßige Verteilung der Publikationen in Cluster. Auch hier wird eine schlechtere Qualität durch das Clustering mit DistilBERT- und GermanBERT-Embedding ersichtlich, da nur 1 Prozent der Publikationen einem Cluster bzw. 65

[9] Für weiterführende Informationen zu den Evaluationsmetriken siehe Abschnitt 5.4.2.

Prozent aller Publikationen einem Cluster zugeordnet wurden. Beim Clustering mit dem TF-IDF-Embedding sind die Verteilungen nicht ganz so stark verzerrt, aber dennoch eindeutig verzerrt, dadurch dass Cluster 1 und 2 45 Prozent bzw. 47 Prozent der Publikationen zugeordnet sind, Cluster 3 nur 8 Prozent. Zusätzlich lassen sich beim GermanBERT-tuned-Embedding für viele Cluster ein Zusammenhang anhand der Top3-Themen erkennen, besonders für Cluster 2 „Migration“ und Cluster 6 „Lohngestaltung“.

Tabelle 6: Überblick über die häufigsten Themen der Cluster

Embedding	Cluster	Anteil zugeordneter Publikationen	Häufigstes Thema	Zweithäufigstes Thema	Dritthäufigstes Thema
TF-IDF	1	45 %	Migration und Integration	SGB II – Grundsicherung für Arbeitsuchende	Evaluation der Arbeitsmarktpolitik
	2	47 %	Transformationsprozess in Ostdeutschland – Wirtschaft, Arbeitsmarkt und Gesellschaft	Evaluation der Arbeitsmarktpolitik	SGB II – Grundsicherung für Arbeitsuchende
	3	8 %	Regionale Arbeitsmärkte in Deutschland	Transformationsprozess in Ostdeutschland – Wirtschaft, Arbeitsmarkt und Gesellschaft	Gender und Arbeitsmarkt
DistilBERT	1	1 %	SGB II – Grundsicherung für Arbeitsuchende	Regionale Arbeitsmärkte in Deutschland	-
	2	22 %	Transformationsprozess in Ostdeutschland – Wirtschaft, Arbeitsmarkt und Gesellschaft	SGB II – Grundsicherung für Arbeitsuchende	Evaluation der Arbeitsmarktpolitik
	3	45 %	Evaluation der Arbeitsmarktpolitik	SGB II – Grundsicherung für Arbeitsuchende	Migration und Integration
	4	32 %	Migration und Integration	Transformationsprozess in Ostdeutschland – Wirtschaft, Arbeitsmarkt und Gesellschaft	SGB II – Grundsicherung für Arbeitsuchende
GermanBERT	1	16 %	Migration und Integration	SGB II – Grundsicherung für Arbeitsuchende	Evaluation der Arbeitsmarktpolitik
	2	19 %	Transformationsprozess in Ostdeutschland – Wirtschaft, Arbeitsmarkt und Gesellschaft	Migration und Integration	Atypische Beschäftigung
	3	65 %	SGB II – Grundsicherung für Arbeitsuchende	Evaluation der Arbeitsmarktpolitik	Migration und Integration
GermanBERT-tuned	1	14 %	Der Arbeitsmarkt für Akademikerinnen und Akademiker	Erwachsenenbildung und Arbeitsmarkt	Berufliche Mobilität
	2	11 %	Migration und Integration	Fluchtmigrantinnen und -migranten – Bildung und Arbeitsmarkt	Auswirkungen des Krieges gegen die Ukraine auf Wirtschaft, Arbeitsmarkt und Fluchtmigration in Deutschland
	3	19 %	Regionale Arbeitsmärkte in Deutschland	Transformationsprozess in Ostdeutschland –	Auswirkungen der Covid-19-Pandemie auf Wirtschaft und

			Wirtschaft, Arbeitsmarkt und Gesellschaft	Arbeitsmarkt in Deutschland
4	9 %	Arbeitszeit: Verlängern? Verkürzen? Flexibilisieren?	Gender und Arbeitsmarkt	Arbeitszeitpräferenzen der Beschäftigten
5	20 %	SGB II – Grundsicherung für Arbeitsuchende	Evaluation der Arbeitsmarktpolitik	Langzeitarbeitslosigkeit und Langzeitleistungsbezug in Deutschland – Ursachen, Konsequenzen, Maßnahmen
6	7 %	Mindestlohn	Niedriglohnarbeitsmarkt	Tarifbindung in Deutschland
7	8 %	Transformationsprozess in Ostdeutschland – Wirtschaft, Arbeitsmarkt und Gesellschaft	Ältere im Betrieb	Menschen mit Behinderungen in Ausbildung und Beruf
8	10 %	Atypische Beschäftigung	matching – Suchprozesse am Arbeitsmarkt	Digitale Arbeitswelt – Chancen und Herausforderungen für Beschäftigte und Arbeitsmarkt

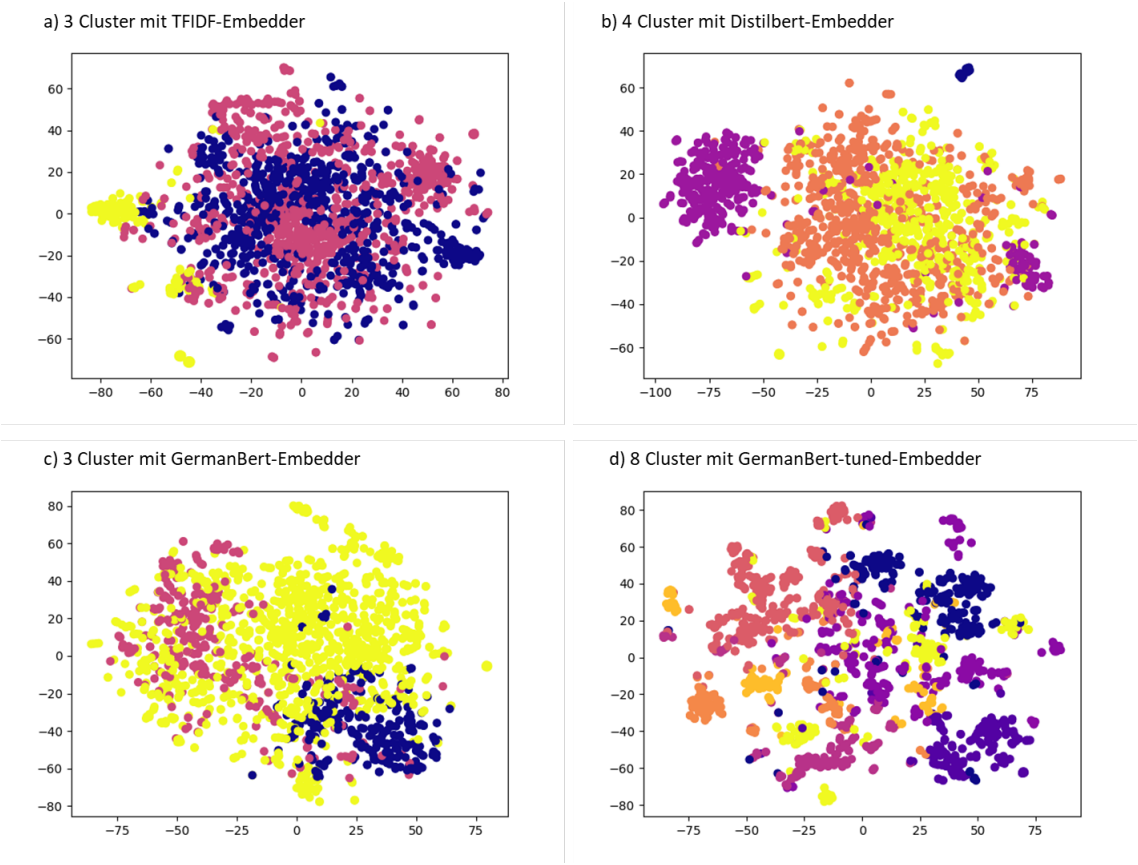
Anmerkung: Die Tabelle zeigt die drei Themen, die den Publikationen des jeweiligen Clusters am häufigsten zugeordnet sind.
Quelle: Eigene Berechnungen.

6.3.4 Grafischer Vergleich der Clustering-Methoden

Für die grafische Darstellung der Cluster in Abbildung 16 werden die 786-dimensionalen Embeddings auf eine zweidimensionale Ebene reduziert[10]. Jeder Punkt repräsentiert ein Embedding einer Publikation und ist entsprechend dem zugeordneten Cluster eingefärbt. Diese Visualisierung ermöglicht es, die räumliche Verteilung der Embeddings zu analysieren und zu überprüfen, ob eine Häufung von Embeddings auf ein bestimmtes Cluster hindeutet. Es ist jedoch wichtig zu beachten, dass diese Interpretation nur für den zweidimensionalen Raum gilt und im hochdimensionalen Raum andere Strukturen bestehen können.

[10] Die Reduktion der Dimensionen erfolgt über die Methode „T-distributed Stochastic Neighbor Embedding“ aus dem sklearn-Paket (scikit-learn developers 2024).

Abbildung 16: Cluster-Darstellung auf Basis verschiedener Embeddings



Quelle: Eigene Berechnung

In Teilabbildung a, die die Clusterbildung mit TF-IDF-Embedding zeigt, ist zu erkennen, dass zwei Cluster räumlich aufeinanderliegen, was die Visualisierung einer klaren Trennung erschwert, die im hochdimensionalen Raum allerdings bestehen kann. Ein weiteres Cluster, das gelb gefärbt ist, deutet jedoch auf eine tatsächliche Häufung der Embeddings hin.

Ähnlich verhält es sich in Teilabbildung b, die die Clusterbildung mit DistilBERT-Embeddings darstellt. Hier sind zwei Cluster (orange und gelb gefärbt) ebenfalls nicht klar getrennt, während die lila und blau gefärbten Cluster eine deutliche Häufung der Embeddings aufweisen.

In Teilabbildung c zur Clusterbildung mit GermanBERT-Embeddings könnten die blau und rosa gefärbten Cluster eine tatsächliche Häufung der Embeddings aufweisen und damit eine gewisse thematische Kohärenz besitzen. Im Gegensatz dazu ist das gelbe Cluster über den gesamten Raum verteilt, was auf eine weniger klare thematische Abgrenzung hinweist.

Die Clusterbildung mit GermanBERT-tuned-Embedding in Teilabbildung d zeigt deutliche Cluster mit einer erkennbaren Häufung der Embeddings. Dies deutet darauf hin, dass das feinabgestimmte Modell eine verbesserte Fähigkeit zur Erkennung thematischer Zusammenhänge besitzt. Dennoch ist auch hier keine klare Abgrenzung der Cluster mit definierten Zentren und ausreichender Distanz zum nächsten Cluster gegeben. Dies zeigt, dass trotz der verbesserten Clusterbildung durch das GermanBERT-tuned-Embedding die Herausforderung der klaren Trennung und Abgrenzung der Cluster weiterhin besteht.

6.3.5 Interpretation

Die Clustering-Experimente verdeutlichen, dass die Wahl der Embedding-Methode einen erheblichen Einfluss auf die Qualität der Clusterbildung hat. Quantitativ, anhand des Davies-Bouldin-Werts, qualitativ durch die inhaltliche Abgrenzung und Kohärenz der Themen sowie über die grafische Darstellung der Ergebnisse zeigte sich, dass die BERT-basierten Modelle, insbesondere das GermanBERT-tuned-Modell, die beste Clusterbildung erreichten. Im Gegensatz dazu wies TF-IDF die schwächste Leistung auf, was die hohen Fähigkeiten feinabgestimmter BERT-Modelle zur Clusterbildung unterstreicht.

Die Unterschiede zwischen GermanBERT und DistilBERT könnten darauf zurückzuführen sein, dass alle Publikationen in unserer Datengrundlage deutsche Abstracts enthalten. Bei mehrsprachigen Texten könnte DistilBERT möglicherweise eine bessere Leistung erzielen.

Insgesamt zeigen die Ergebnisse der Cluster-Experimente, dass die gewählten Methoden in der Lage sind, einige thematische Schwerpunkte in Form von Clustern zu identifizieren. Dennoch bleibt die inhaltlich klare Abgrenzung der Cluster eine Herausforderung, möglicherweise auch aufgrund der inhaltlichen Überschneidungen in den Publikationen.

7 Anwendungsfelder

Die Anwendungsfelder für die Temi-Box sind äußerst vielfältig. Initial wurde die Temi-Box für die Themenzuordnung zu Publikationen entwickelt. Darüber hinaus lässt sich die Klassifikation von Texten unmittelbar in verschiedenen weiteren Bereichen der Arbeitsmarkt- und Berufsforschung einsetzen, darunter

- die Zuordnung von Schlagwörtern aus einem Thesaurus zu Publikationen,
- die Analyse von Stellenanzeigen und Berufsbeschreibungen beispielsweise im Hinblick auf bestimmte Kompetenzen,
- die Beschreibung von Unternehmen sowie
- die Auswertung von Interviews.

Neben der klassischen Zuordnung von Labels zu Texten gibt es auch praktische Anwendungen, wie Frage-Antwort-Systeme oder Textzusammenfassungen, die nicht mehr unmittelbar als Textklassifikation erkennbar sind. In solchen Fällen wird der intuitive Begriff „Label“ durch eine Auswahl an Sätzen ersetzt, für die entschieden wird, ob sie in die Antwort mit aufgenommen werden (Gasparetto et al. 2022). Zentrale Anwendungsfelder für Textklassifikation und potenzielle Einsatzgebiete der Temi-Box im Bereich der Arbeitsmarkt- und Berufsforschung umfassen unter anderem (Li et al. 2022; Minaee et al. 2021)

- die Stimmungsanalyse, bei der emotionale Zustände in einem Text kategorisiert werden, wie beispielsweise Stimmungen in Interviews, Stellenanzeigen oder die öffentliche Wahrnehmung von Arbeitgebern in Bewertungsportalen,
- das Topic Labeling, das das Erkennen von Themen in einem Text ermöglicht, etwa Schlüsselthemen und Trends in Stellenanzeigen, Berufsbeschreibungen oder Arbeitsmarktberichten,

- die Nachrichtenklassifizierung, bei der Nachrichten automatisch Kategorien zugeordnet werden, um Trends, Entwicklungen oder Arbeitsmarktereignisse wie Fusionen, Übernahmen, Entlassungen oder Neueinstellungen in Nachrichten oder Pressemitteilungen von Unternehmen zu erkennen.

Darüber hinaus bietet die Temi-Box Potenzial für Weiterentwicklungen, wie die Beantwortung von Fragen, die Erkennung benannter Entitäten (Named Entity Recognition), die Zusammenfassung von Texten und das Ermitteln ähnlicher Dokumente. Diese vielfältigen Anwendungsfelder unterstreichen die Flexibilität und Leistungsfähigkeit der Temi-Box im Bereich der Textanalyse.

8 Zusammenfassung und Herausforderungen

Textklassifikation und -clustering sind zentrale Aufgaben im Bereich des Text Mining, die es ermöglichen, aus unstrukturierten Texten strukturierte Daten zu extrahieren. Die Text Mining Toolbox Temi-Box bietet mit ihrem entwickelten und veröffentlichten Code eine effiziente Möglichkeit, Texte zu klassifizieren oder in Cluster zu unterteilen. Die enthaltenen und erweiterbaren Methoden können anhand etablierter Messgrößen bewertet werden. Für die Klassifikation und das Clustern von Texten existieren zahlreiche etablierte Methoden, deren Kombination jedoch häufig einen erheblichen Programmieraufwand erfordert, den viele Forscherinnen und Forscher scheuen. Der bereitgestellte Code der Temi-Box ermöglicht es, gezielt Erfahrungen im Einsatz unterschiedlicher, etablierter Klassifikations- und Clustering-Algorithmen zu sammeln. Projekte können auf diesem Code aufbauen, um in einem frühen Stadium den potenziellen Einsatz zu testen, wobei die erforderlichen Programmierkenntnisse auf ein Minimum beschränkt sind.

Der vorliegende Forschungsbericht dokumentiert umfassend das methodische Vorgehen bei der Entwicklung und Nutzung der Temi-Box. Die Entwicklung erfolgte anhand der automatisierten Verschlagwortung von Publikationen sowie der Zuordnung von Publikationen zu Themen für die IAB-Infoplattform als reale Anwendungsfälle, um die Funktionalitäten der Temi-Box für den praktischen Einsatz zu optimieren. Nach einer Einführung in das Text Mining, insbesondere in die Klassifikation und das Clustern von Texten, wurden Anforderungen und technische Grundlagen der Temi-Box beschrieben, um Einblicke in Architektur und Funktionsweise des Systems zu geben. Die detaillierte Darstellung der implementierten Methoden und der Ergebnisse der Text Mining-Experimente zur automatisierten Themenzuordnung und zum Clustering soll Anwenderinnen und Anwendern wertvolle Hinweise zum Einsatz der Temi-Box und zur Interpretation der Ergebnisse bieten. Abschließend wurden mögliche Anwendungsfelder im Bereich der Klassifikation und Optionen zur Weiterentwicklung aufgezeigt. Insgesamt bietet dieser Bericht eine fundierte Grundlage für die Nutzung und Weiterentwicklung der Temi-Box und ihre Anwendung in verschiedenen Projekten mit Bezug zum Text Mining.

Trotz vielversprechender Fortschritte in der Textklassifikation bleiben Herausforderungen bestehen, die auch die Nutzung der Temi-Box betreffen. Li et al. (2022) identifizieren diese

Herausforderungen in den Bereichen Daten, Modellierung und Leistung. Die Qualität der Daten ist entscheidend für die Leistung von Text Mining-Modellen. Probleme können durch fehlende Labels, fachspezifische Terminologien wie Slang oder Abkürzungen sowie semantische Abhängigkeiten bei der Multi-Label-Klassifikation entstehen. In der Modellierung besteht Potenzial zur Optimierung von Algorithmen und Trainingszeiten, etwa durch Anpassung der Netzwerktiefe oder Lernrate. Ein weiteres Problem ist die semantische Robustheit, da wenige ungünstige Stichproben die Modellleistung stark beeinträchtigen können. Zudem sind Deep-Learning-Modelle in der Regel schwer interpretierbar, was die Nachvollziehbarkeit von Klassifizierungen erschwert. Ein vertieftes Verständnis der theoretischen Grundlagen und die Erforschung der Modelle in verschiedenen Anwendungsbereichen sind notwendig, um diese Lücken zu schließen. Tools wie die Temi-Box können durch ihren einfachen Zugang einen wichtigen Beitrag zur Erforschung und Optimierung von Text Mining-Methoden leisten und so zu einem besseren Verständnis beitragen.

Literatur

Allahyari, Mehdi; Pouriyeh, Seyedamin; Assefi, Mehdi; Safaei, Saied; Trippe, Elizabeth D.; Gutierrez, Juan B.; Kochut, Krys (2017): A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques, ArXiv.

Birunda, S. Selva; Devi, R. Kanniga (2020): A Review on Word Embedding Techniques for Text Classification.

Breiman, L. (2001): Random Forests. In: Machine Learning 45, S. 5–32..

Chan, Branden; Schweter, Stefan; Möller, Timo (2020): German's Next Language Model. Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain (Online), International Committee on Computational Linguistics.

Chen, Tianqi; Guestrin, Carlos (2016 of Conference): XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA.

Cover, Thomas M.; Hart, Peter E. (1967): Nearest Neighbor Pattern Classification. In: IEEE Transactions on Information Theory 13 (1), S. 21–27.

Cox, D. R. (1958): The Regression Analysis of Binary Sequences. In: Journal of the Royal Statistical Society Series B (Statistical Methodology) 20, S. 215–232.

Davies, David L.; Bouldin, Donald W. (1979): A Cluster Separation Measure. In: IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1, S. 224–227.

Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina (2019): BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota, Association for Computational Linguistics.

Fawcett, Tom (2006): An introduction to ROC analysis. In: Pattern Recognition Letters 27 (8), S. 861–874.

Gaikwad, Sonali Vijay; Chaugule, Archana; Patil, Pramod (2014): Text Mining Methods and Techniques. In: International Journal of Computer Applications 85 (17), S. 42–45.

Gasparetto, Andrea; Marcuzzo, Matteo; Zangari, Alessandro; Albarelli, Andrea (2022): A Survey on Text Classification Algorithms: From Text to Predictions. In: Information 13 (2).

Hanley, James A.; McNeil, Barbara J. (1982): The meaning and use of the area under a receiver operating characteristic (ROC) curve. In: Radiology 143 1, S. 29–36.

Hippner, Hajo; Rentzmann, René (2006): Text Mining. In: Informatik-Spektrum 29 (4), S. 287–290.

Honnibal, Matthew; Montani, Ines; Van Landeghem, Sofie; Boyd, Adriane (2020): spaCy: Industrial-strength Natural Language Processing in Python.

Hotho, Andreas; Nürnberger, Andreas; Paaß, Gerhard (2005): A Brief Survey of Text Mining. In: Journal for Language Technology and Computational Linguistics 20 (1), S. 19–62.

IAB (2017): IAB-Infoplattform. IAB-Forum. Abrufdatum: 12.08.2024.

Joachims, Thorsten (1998): Text categorization with Support Vector Machines: Learning with many relevant features. European Conference on Machine Learning (ECML' 98), Berlin, Heidelberg, Springer Berlin Heidelberg.

Jurafsky, Daniel; Martin, James H. (2023): Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.

Ke, Guolin; Meng, Qi; Finley, Thomas; Wang, Taifeng; Chen, Wie; Ma, Weidong; Ye, Qiwei; Liu, Tie-Yan (2017): LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Neural Information Processing Systems.

Kowsari, Kamran; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D. (2019): Text Classification Algorithms: A Survey. In: Information 10 (4).

Li, Qian; Peng, Hao; Li, Jianxin; Xia, Congying; Yang, Renyu; Sun, Lichao; Yu, Philip S.; He, Lifang (2022): A Survey on Text Classification: From Traditional to Deep Learning. In: ACM Transactions on Intelligent Systems and Technology 13 (2), S. 1–41.

Liu, Yang; Lapata, Mirella (2019): Text Summarization with Pretrained Encoders, Hong Kong, China.

Maron, M. E. (1961): Automatic Indexing: An Experimental Inquiryv. In: Journal of the Association for Computing Machinery 8 (3), S. 404–417.

Minaee, Shervin; Kalchbrenner, Nal; Cambria, Erik; Nikzad, Narjes; Chenaghlu, Meysam; Gao, Jiangeng (2021): Deep Learning--based Text Classification. In: ACM Computing Surveys 54 (3), S. 1–40.

Oyen, Renate; Paulsen, Jörg (2009): Schlagwortliste Arbeitsmarkt, Beruf und Berufsbildung. Band 1: Alphabetischer Teil. Gemeinsamer Thesaurus für die Literaturdatenbanken des Bundesinstituts für Berufsbildung (BIBB) und des Instituts für Arbeitsmarkt- und Berufsforschung (IAB), Nürnberg.

Paaß, Gerhard; Giesselbach, Sven (2023): Foundation Models for Natural Language Processing: Springer International Publishing.

Paszke, Adam; Gross, Sam; Massa, Francisco; Lerer, Adam; Bradbury, James; Chanan, Gregory; Killeen, Trevor; Lin, Zeming; Gimelshein, Natalia; Antiga, Luca; Desmaison, Alban; Köpf, Andreas; Yang, Edward; DeVito, Zach; Raison, Martin; Tejani, Alykhan; Chilamkurthy, Sasank; Steiner, Benoit; Fang, Lu; Bai, Junjie; Chintala, Soumith (2019): Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems.

Pedregosa, Fabian; Varoquaux, Gaël; Gramfort, Alexandre; Michel, Vincent; Thirion, Bertrand; Grisel, Olivier; Blondel, Mathieu; Prettenhofer, Peter; Weiss, Ron; Dubourg, Vincent; Vanderplas, Jake; Passos, Alexandre; Cournapeau, David; Brucher, Matthieu; Perrot, Matthieu; Duchesnay, Édouard (2011): Scikit-learn: Machine Learning in Python. In: Journal of Machine Learning Research 12, 2825-2830.

Petukhova, Alina Vladimirovna; Matos-Carvalho, João Pedro; Fachada, Nuno (2025): Text Clustering with Large Language Model Embeddings. In: International Journal of Cognitive Computing in Engineering 6, 100-108..

Rijsbergen, Cornelis J. van (1979): Information retrieval: London: Butterworths.

Saito, T.; Rehmsmeier, M. (2015): The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. In: PLoS ONE 10 (3), e0118432.

Sanh, Victor; Debut, Lysandre; Chaumond, Julien; Wolf, Thomas (2020): DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, ArXiv.

Schröder, Gunnar; Thiele, Maik; Lehner, Wolfgang (2011): Setting Goals and Choosing Metrics for Recommender System Evaluations. UCERSTI2 workshop at the 5th ACM conference on recommender systems, Chicago, USA.

scikit-learn developers (2024): sklearn.manifold.TSNE. scikit-learn 1.5 documentation, Abrufdatum: 1.12.2024.

Sparck Jones, Karen (1972): A statistical interpretation of term specificity and its application in retrieval. In: Journal of documentation 28 (1), S. 11–21.

Subakti, Alvin; Murfi, Hendri; Hariadi, Nora (2022): The performance of BERT as data representation of text clustering. In: Journal of Big Data 9 (1), S. 15.

Taha, Kamal; Yoo, Paul D.; Yeun, Chan; Taha, Aya (2024): A Comprehensive Survey of Text Classification Techniques and Their Research Applications: Observational and Experimental Insights, ArXiv.

Tarekegn, Adane Nega; Giacobini, Mario; Michalak, Krzysztof (2021): A review of methods for imbalanced multi-label classification. In: Pattern Recognition 118.

Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N.; Kaiser, Lukasz; Poloshukhin, Illia (2017): Attention Is All You Need. In: Advances in Neural Information Processing Systems, S. 6000–6010.

Wirth, Rüdiger; Hipp, Jochen (2000): CRISP-DM: Towards a standard process model for data mining. Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining, Manchester.

Wolf, Thomas; Debut, Lysandre; Sanh, Victor; Chaumond, Julien; Delangue, Clement; Moi, Anthony; Cistac, Pierric; Rault, Tim; Louf, Remi; Funtowicz, Morgan; Davison, Joe; Shleifer, Sam; von Platen, Patrick; Ma, Clara; Jernite, Yacine; Plu, Julien; Xu, Canwen; Le Scao, Teven; Gugger, Sylvain; Drame, Mariama; Lhoest, Quentin; Rush, Alexander (2020): Transformers: State-of-the-Art Natural Language Processing. Conference on Empirical Methods in Natural Language Processing.

Wu, Yonghui; Schuster, Mike; Chen, Zhifeng; Le, Quoc V.; Norouzi, Mohammad; Macherey, Wolfgang; Krikun, Maxim; Cao, Yuan; Gao, Qin; Macherey, Klaus; Klingner, Jeff; Shah, Apurva; Johnson, Melvin; Liu, ; Kaiser, Łukasz; Gouws, Stephan; Kato, Yoshikiyo; Kudo, Taku; Kazawa, Hideto; Stevens, Keith; Kurian, George; Patil, Nishant; Wang, Wei; Young, Cliff; Smith, Jason; Riesa, Jason; Rudnick, Alex; Vinyals, Oriol; Corrado, Greg; Hughes, Macduff; Dean, Jeffrey (2016): Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, ArXiv.

Xu, Qiang; Gu, Hao; Ji, ShengWei (2024): Text clustering based on pre-trained models and autoencoders. In: Frontiers in Computational Neuroscience 17.

Xu, Rui; Wunsch, Donald (2005): Survey of Clustering Algorithms. In: IEEE TRANSACTIONS ON NEURAL NETWORKS 16 (3).

Anhang

Anhang 1: Übersicht über die Metriken aller Klassifikationsexperimente

Tabelle 7: Metriken aller Klassifikations-Experimente im Vergleich
Insgesamt 139 Experimente, Schwellenwert von 0, absteigend sortiert nach Recall@10 und anschließend nach Recall@5

Modell	Klassifikator	Layer-Dimension	Recall@10	Precision @10	F1@10	Recall@5	Precision @5	F1@5
GermanBERT	binary	256_256	87,37%	21,59%	34,62%	76,87%	37,99%	50,85%
GermanBERT	binary	4096	87,37%	21,59%	34,62%	76,23%	37,67%	50,42%
GermanBERT	binary	1024	87,15%	21,53%	34,54%	77,30%	38,20%	51,13%
GermanBERT	binary	1024_16	86,51%	21,38%	34,28%	76,45%	37,78%	50,57%
DistilBERT	binary	32	86,08%	21,27%	34,11%	74,73%	36,93%	49,43%
GermanBERT	binary	32	86,08%	21,27%	34,11%	73,45%	36,30%	48,58%
GermanBERT	binary	256	85,87%	21,22%	34,03%	74,30%	36,72%	49,15%
DistilBERT	binary	16_32	85,87%	21,22%	34,03%	74,30%	36,72%	49,15%
GermanBERT	binary	4096_16_1024	85,65%	21,16%	33,94%	77,30%	38,20%	51,13%
GermanBERT	binary	4096_16	85,65%	21,16%	33,94%	73,88%	36,51%	48,87%
GermanBERT	binary	16_4096	85,44%	21,11%	33,86%	75,59%	37,35%	50,00%
GermanBERT	binary	32_16	85,44%	21,11%	33,86%	70,24%	34,71%	46,46%
GermanBERT	binary	1024_4096_16	85,22%	21,06%	33,77%	75,80%	37,46%	50,14%
GermanBERT	binary	1024_1024	85,22%	21,06%	33,77%	71,09%	35,13%	47,03%
DistilBERT	binary	256_256	85,22%	21,06%	33,77%	73,45%	36,30%	48,58%
DistilBERT	binary	16	85,22%	21,06%	33,77%	71,52%	35,34%	47,31%
GermanBERT	binary	32_32_32	85,01%	21,01%	33,69%	77,09%	38,10%	50,99%
GermanBERT	binary	16_1024_4096	84,80%	20,95%	33,60%	74,52%	36,83%	49,29%
GermanBERT	binary	4096_4096_4096	84,80%	20,95%	33,60%	71,31%	35,24%	47,17%
DistilBERT	binary	32_32	84,80%	20,95%	33,60%	75,59%	37,35%	50,00%
DistilBERT	binary	16_4096	84,80%	20,95%	33,60%	71,95%	35,56%	47,59%
GermanBERT	binary	32_32	84,58%	20,90%	33,52%	75,16%	37,14%	49,72%

Modell	Klassifikator	Layer-Dimension	Recall@10	Precision @10	F1@10	Recall@5	Precision @5	F1@5
GermanBERT	binary	16_16	84,37%	20,85%	33,43%	77,73%	38,41%	51,42%
GermanBERT	binary	16_32	84,37%	20,85%	33,43%	75,59%	37,35%	50,00%
GermanBERT	binary	16	83,94%	20,74%	33,26%	70,88%	35,03%	46,88%
GermanBERT	binary	16_1024	83,94%	20,74%	33,26%	70,24%	34,71%	46,46%
DistilBERT	binary	256	83,94%	20,74%	33,26%	73,45%	36,30%	48,58%
DistilBERT	binary	1024_4096_16	83,51%	20,63%	33,09%	70,66%	34,92%	46,74%
DistilBERT	binary	32_16	83,30%	20,58%	33,01%	72,81%	35,98%	48,16%
GermanBERT	binary	4096_4096	83,08%	20,53%	32,92%	72,16%	35,66%	47,73%
DistilBERT	binary	16_1024	83,08%	20,53%	32,92%	70,66%	34,92%	46,74%
GermanBERT	binary	4096_1024_16	82,66%	20,42%	32,75%	64,88%	32,06%	42,92%
DistilBERT	binary	1024	82,44%	20,37%	32,67%	67,24%	33,23%	44,48%
DistilBERT	binary	16_16	82,01%	20,26%	32,50%	71,09%	35,13%	47,03%
DistilBERT	binary	4096_1024_16	82,01%	20,26%	32,50%	69,81%	34,50%	46,18%
DistilBERT	binary	4096_4096_4096	82,01%	20,26%	32,50%	69,59%	34,39%	46,03%
DistilBERT	binary	16_1024_4096	81,16%	20,05%	32,16%	70,02%	34,60%	46,32%
DistilBERT	binary	1024_16	81,16%	20,05%	32,16%	68,74%	33,97%	45,47%
DistilBERT	binary	4096	80,94%	20,00%	32,07%	68,95%	34,07%	45,61%
DistilBERT	binary	32_32_32	80,09%	19,79%	31,74%	69,81%	34,50%	46,18%
DistilBERT	binary	4096_16	80,09%	19,79%	31,74%	68,95%	34,07%	45,61%
DistilBERT	binary	4096_16_1024	79,66%	19,68%	31,57%	69,59%	34,39%	46,03%
DistilBERT	binary	1024_1024	77,94%	19,26%	30,89%	66,81%	33,02%	44,19%
DistilBERT	binary	4096_4096	77,73%	19,21%	30,80%	64,24%	31,75%	42,49%
GermanBERT	multinomial	4096	74,73%	18,47%	29,61%	64,67%	31,96%	42,78%
DistilBERT	multinomial	4096	73,23%	18,10%	29,02%	57,39%	28,36%	37,96%
TF-IDF	multinomial	4096_4096	70,02%	17,30%	27,75%	54,60%	26,98%	36,12%
TF-IDF	multinomial	4096	67,02%	16,56%	26,56%	55,25%	27,30%	36,54%
GermanBERT	multinomial	1024	63,81%	15,77%	25,29%	54,60%	26,98%	36,12%

Modell	Klassifikator	Layer-Dimension	Recall@10	Precision @10	F1@10	Recall@5	Precision @5	F1@5
TF-IDF	binary	4096_4096	62,74%	15,50%	24,86%	47,75%	23,60%	31,59%
TF-IDF	binary	4096_16_1024	62,74%	15,50%	24,86%	47,75%	23,60%	31,59%
TF-IDF	binary	4096_16	62,10%	15,34%	24,61%	44,33%	21,90%	29,32%
TF-IDF	binary	1024_4096_16	61,88%	15,29%	24,52%	43,04%	21,27%	28,47%
TF-IDF	binary	1024_1024	60,60%	14,97%	24,01%	46,68%	23,07%	30,88%
TF-IDF	binary	4096_4096_4096	60,60%	14,97%	24,01%	44,97%	22,22%	29,75%
TF-IDF	binary	4096_1024_16	60,39%	14,92%	23,93%	46,68%	23,07%	30,88%
GermanBERT	multinomial	4096_4096	60,17%	14,87%	23,84%	48,82%	24,13%	32,29%
TF-IDF	multinomial	1024_1024	59,96%	14,81%	23,76%	45,18%	22,33%	29,89%
GermanBERT	binary	4	59,74%	14,76%	23,67%	37,90%	18,73%	25,07%
TF-IDF	multinomial	16_4096	59,10%	14,60%	23,42%	43,68%	21,59%	28,90%
TF-IDF	binary	4096	57,60%	14,23%	22,83%	38,54%	19,05%	25,50%
TF-IDF	binary	256_256	56,10%	13,86%	22,23%	40,04%	19,79%	26,49%
TF-IDF	binary	1024_16	55,89%	13,81%	22,15%	39,83%	19,68%	26,35%
GermanBERT	multinomial	1024_1024	55,67%	13,76%	22,06%	46,68%	23,07%	30,88%
GermanBERT	multinomial	4096_16_1024	55,67%	13,76%	22,06%	40,69%	20,11%	26,91%
DistilBERT	multinomial	1024	55,25%	13,65%	21,89%	44,33%	21,90%	29,32%
TF-IDF	binary	16_4096	54,39%	13,44%	21,55%	37,90%	18,73%	25,07%
TF-IDF	multinomial	1024	53,75%	13,28%	21,30%	38,97%	19,26%	25,78%
DistilBERT	multinomial	4096_4096	53,53%	13,23%	21,21%	40,26%	19,89%	26,63%
GermanBERT	multinomial	4096_4096_4096	52,89%	13,07%	20,96%	33,40%	16,51%	22,10%
GermanBERT	multinomial	256	52,68%	13,02%	20,87%	33,83%	16,72%	22,38%
DistilBERT	binary	4	51,61%	12,75%	20,45%	30,84%	15,24%	20,40%
TF-IDF	binary	16_1024_4096	51,39%	12,70%	20,36%	35,76%	17,67%	23,65%
TF-IDF	KNN	-	51,39%	25,13%	33,76%	47,75%	26,42%	34,02%
GermanBERT	multinomial	4096_16	49,89%	12,33%	19,77%	35,76%	17,67%	23,65%

Modell	Klassifikator	Layer-Dimension	Recall@10	Precision @10	F1@10	Recall@5	Precision @5	F1@5
GermanBERT	multinomial	16_1024	49,89%	12,33%	19,77%	31,69%	15,66%	20,96%
TF-IDF	binary	1024	49,25%	12,17%	19,52%	33,83%	16,72%	22,38%
DistilBERT	multinomial	256	49,04%	12,12%	19,43%	27,62%	13,65%	18,27%
TF-IDF	multinomial	4096_16	49,04%	12,12%	19,43%	34,26%	16,93%	22,66%
GermanBERT	multinomial	32	48,82%	12,06%	19,35%	29,98%	14,81%	19,83%
TF-IDF	multinomial	1024_4096_16	48,61%	12,01%	19,26%	32,55%	16,08%	21,53%
TF-IDF	multinomial	4096_16_1024	47,32%	11,69%	18,75%	29,12%	14,39%	19,26%
GermanBERT	multinomial	1024_16	47,11%	11,64%	18,67%	30,19%	14,92%	19,97%
GermanBERT	multinomial	32_32_32	47,11%	11,64%	18,67%	30,19%	14,92%	19,97%
GermanBERT	multinomial	1024_4096_16	46,90%	11,59%	18,58%	27,62%	13,65%	18,27%
GermanBERT	multinomial	16_4096	46,90%	11,59%	18,58%	27,41%	13,54%	18,13%
TF-IDF	multinomial	16_1024	46,90%	11,59%	18,58%	27,84%	13,76%	18,41%
TF-IDF	multinomial	4096_1024_16	46,47%	11,48%	18,41%	32,76%	16,19%	21,67%
TF-IDF	multinomial	256	46,47%	11,48%	18,41%	27,41%	13,54%	18,13%
GermanBERT	multinomial	16_1024_4096	46,25%	11,43%	18,33%	32,98%	16,30%	21,81%
TF-IDF	multinomial	4096_4096_4096	46,25%	11,43%	18,33%	31,48%	15,56%	20,82%
DistilBERT	multinomial	1024_16	46,04%	11,38%	18,24%	29,76%	14,71%	19,69%
TF-IDF	binary	16_1024	45,82%	11,32%	18,16%	33,83%	16,72%	22,38%
GermanBERT	multinomial	16	45,61%	11,27%	18,07%	28,48%	14,07%	18,84%
GermanBERT	multinomial	16_16	45,61%	11,27%	18,07%	27,41%	13,54%	18,13%
GermanBERT	multinomial	4096_1024_16	45,61%	11,27%	18,07%	27,41%	13,54%	18,13%
TF-IDF	multinomial	16	45,40%	11,22%	17,99%	27,41%	13,54%	18,13%
DistilBERT	multinomial	4096_16	44,97%	11,11%	17,82%	32,76%	16,19%	21,67%
DistilBERT	multinomial	16_1024	44,97%	11,11%	17,82%	27,41%	13,54%	18,13%
DistilBERT	multinomial	16	44,75%	11,06%	17,73%	27,41%	13,54%	18,13%

Modell	Klassifikator	Layer-Dimension	Recall@10	Precision @10	F1@10	Recall@5	Precision @5	F1@5
TF-IDF	multinomial	16_1024_4096	44,33%	10,95%	17,56%	27,41%	13,54%	18,13%
DistilBERT	multinomial	4096_16_1024	43,68%	10,79%	17,31%	28,27%	13,97%	18,70%
TF-IDF	multinomial	256_256	43,47%	10,74%	17,23%	27,41%	13,54%	18,13%
TF-IDF	multinomial	16_16	43,04%	10,63%	17,06%	27,41%	13,54%	18,13%
TF-IDF	multinomial	32_32_32	42,83%	10,58%	16,97%	27,41%	13,54%	18,13%
DistilBERT	multinomial	32_32	42,61%	10,53%	16,89%	27,41%	13,54%	18,13%
TF-IDF	multinomial	32	42,61%	10,53%	16,89%	27,41%	13,54%	18,13%
DistilBERT	multinomial	16_1024_4096	42,40%	10,48%	16,80%	27,41%	13,54%	18,13%
DistilBERT	multinomial	32_32_32	42,40%	10,48%	16,80%	27,19%	13,44%	17,99%
TF-IDF	binary	32_32_32	42,40%	10,48%	16,80%	27,41%	13,54%	18,13%
TF-IDF	multinomial	32_16	42,40%	10,48%	16,80%	27,41%	13,54%	18,13%
GermanBERT	multinomial	32_32	42,18%	10,42%	16,72%	27,41%	13,54%	18,13%
DistilBERT	multinomial	1024_4096_16	42,18%	10,42%	16,72%	27,41%	13,54%	18,13%
DistilBERT	multinomial	256_256	42,18%	10,42%	16,72%	27,41%	13,54%	18,13%
DistilBERT	multinomial	32	41,97%	10,37%	16,63%	27,41%	13,54%	18,13%
DistilBERT	multinomial	1024_1024	41,97%	10,37%	16,63%	25,91%	12,80%	17,14%
DistilBERT	multinomial	32_16	41,76%	10,32%	16,55%	27,41%	13,54%	18,13%
TF-IDF	multinomial	1024_16	41,76%	10,32%	16,55%	27,62%	13,65%	18,27%
TF-IDF	multinomial	16_32	41,76%	10,32%	16,55%	27,41%	13,54%	18,13%
DistilBERT	multinomial	16_16	41,33%	10,21%	16,38%	29,34%	14,50%	19,41%
DistilBERT	multinomial	16_4096	41,33%	10,21%	16,38%	27,41%	13,54%	18,13%
TF-IDF	binary	32_16	41,33%	10,21%	16,38%	28,27%	13,97%	18,70%
GermanBERT	multinomial	256_256	41,11%	10,16%	16,29%	27,41%	13,54%	18,13%
DistilBERT	multinomial	16_32	41,11%	10,16%	16,29%	29,98%	14,81%	19,83%
TF-IDF	multinomial	32_32	41,11%	10,16%	16,29%	30,19%	14,92%	19,97%
TF-IDF	binary	32_32	40,90%	10,11%	16,21%	27,84%	13,76%	18,41%
TF-IDF	binary	256	40,69%	10,05%	16,12%	28,05%	13,86%	18,56%
GermanBERT	multinomial	16_32	40,26%	9,95%	15,95%	27,19%	13,44%	17,99%

Modell	Klassifikator	Layer-Dimension	Recall@10	Precision @10	F1@10	Recall@5	Precision @5	F1@5
DistilBERT	multinomial	4096_1024_16	40,04%	9,89%	15,87%	27,41%	13,54%	18,13%
GermanBERT	multinomial	32_16	39,40%	9,74%	15,61%	28,05%	13,86%	18,56%
DistilBERT	multinomial	4096_4096_4096	39,40%	9,74%	15,61%	27,41%	13,54%	18,13%
TF-IDF	binary	16_32	39,19%	9,68%	15,53%	25,91%	12,80%	17,14%
TF-IDF	binary	32	38,12%	9,42%	15,10%	27,19%	13,44%	17,99%
TF-IDF	binary	16_16	37,47%	9,26%	14,85%	25,48%	12,59%	16,86%
TF-IDF	multinomial	4	34,26%	8,47%	13,58%	26,34%	13,02%	17,42%
TF-IDF	binary	16	33,83%	8,36%	13,41%	23,13%	11,43%	15,30%
TF-IDF	binary	4	29,98%	7,41%	11,88%	19,27%	9,52%	12,75%
DistilBERT	multinomial	4	27,41%	6,77%	10,86%	21,84%	10,79%	14,45%
GermanBERT	multinomial	4	26,55%	6,56%	10,52%	8,78%	4,34%	5,81%

Quelle: IABPub105 mit 3.557 Datenzeilen. Eigene Berechnung.

Abbildungsverzeichnis

Tabellenverzeichnis

Impressum

IAB-Forschungsbericht 13|2025

Veröffentlichungsdatum
8. Mai 2025

Herausgeber
Institut für Arbeitsmarkt- und Berufsforschung
der Bundesagentur für Arbeit
Regensburger Straße 104
90478 Nürnberg

Bezugsmöglichkeit dieses Dokuments
https://doku.iab.de/forschungsbericht/2025/fb1325.pdf

Bezugsmöglichkeit aller Veröffentlichungen der Reihe „IAB-Forschungsbericht“
https://iab.de/publikationen/iab-publikationsreihen/iab-forschungsbericht/

Website
https://iab.de

ISSN
2195-2655

DOI
10.48720/IAB.FB.2513

Rückfragen zum Inhalt
Christine Hirmer
Telefon: 0911 179-4583
E-Mail: Christine.Hirmer@iab.de

Lina Metzger
Telefon: 0911 179-6711
E-Mail: Lina.Metzger@iab.de