

BACHELOR THESIS
Kristoffer Schaaf

Entwicklung einer Software zur Erkennung von Fake News auf Nachrichtenportalen

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Engineering and Computer Science
Department Computer Science

Kristoffer Schaaf

Entwicklung einer Software zur Erkennung von Fake News auf Nachrichtenportalen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Angewandte Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Stefan Sarstedt
Zweitgutachter: Prof. Dr. Marina Tropmann-Frick

Eingereicht am: 01.04.2025

Kristoffer Schaaf

Thema der Arbeit

Entwicklung einer Software zur Erkennung von Fake News auf Nachrichtenportalen

Stichworte

Machinelles Lernen, Fake News, Nachrichtenportale,

Kurzzusammenfassung

Arthur Dents Reise in eine neue Zukunft ...

Kristoffer Schaaf

Title of Thesis

Development of a software for the detection of fake news on news portals

Keywords

Machine Learning, Fake News, Text Mining, Classification, NLP

Abstract

Arthur Dents travel to a new future ...

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
Tabellenverzeichnis	viii
1 Einleitung	1
1.1 Hintergrund: Die zunehmende Verbreitung von Fake News und deren gesellschaftliche Auswirkungen	1
1.1.1 Wann entstanden Fake News	1
1.1.2 Wie definieren sich Fake News und wie sind sie aufgebaut	1
1.1.3 Aus welcher Motivation entstehen Fake News	3
1.1.4 Warum verbreiten sich Fake News	3
1.1.5 Wer konsumiert Fake News	3
1.1.6 Probleme beim Erkennen von Fake News	4
1.1.7 Welche potenziellen Indikatoren zum Erkennen bei Fake News gibt es	4
1.2 Wahl der Nachrichtenportale	5
1.3 Zielsetzung: Entwicklung einer Software zur automatisierten Fake-News-Erkennung	5
1.4 Aufbau der Arbeit	5
2 Grundlagen und Begriffsdefinitionen	6
2.1 Definition „Fake News“: Merkmale, Ziele, Beispiele	6
2.1.1 Klassifizierungen	6
2.2 Kategorisierung der Fake News Detection-Ansätze	6
2.3 Warum der Fokus auf Machine Learning?	6
2.4 Überblick über relevante Plattformen und deren Rolle im Medienkonsum .	6

3	Maschinelles Lernen zur Fake News Erkennung	7
3.1	Grundlagen von ML	7
3.1.1	Überwachtes vs. unüberwachtes Lernen	7
3.1.2	Klassische ML-Modelle (Logistic Regression, Naive Bayes, SVM, Entscheidungsbäume)	7
3.2	Performance-Metriken: Accuracy, Precision, Recall, F1-Score	7
3.3	Herausforderungen im Kontext von Fake News	7
4	Natural Language Processing	8
4.1	Machine Learning	8
4.1.1	Textbereinigung und Vorverarbeitung	8
4.1.2	Merkmalextraktion	10
4.1.3	Machine Learning Modelle	14
4.2	Deep Learning	14
4.2.1	Word Embeddings	14
4.2.2	Deep Learning-Modelle	14
4.3	Hybride Modelle	14
4.3.1	convolutional neural network (CNN) along with bi-directional LSTM	14
5	Relevante Datensätze und Auswahlkriterien	15
5.1	Vorstellung verfügbarer deutscher Fake-News-Datensätze	15
5.2	Auswahl und Begründung des finalen Datensatzes	15
5.3	Herausforderungen bei deutschen Textdaten (Klassenverteilung, Quellen- vielfalt)	15
6	Konzeption der Softwarelösung	16
6.1	Hauptkomponente	16
7	Umsetzung des Prototyps	20
7.1	Implementierung der Chrome Erweiterung	20
8	Evaluation und Ergebnisse	23
9	Diskussion	24
10	Fazit und Ausblick	25
	Literaturverzeichnis	26

A Anhang	30
A.1 Verwendete Hilfsmittel	30
Selbstständigkeitserklärung	31

Abbildungsverzeichnis

4.1	Vergleich der Sparse Matrizen	10
4.2	Vergleich verschiedener Modelle mit BoW, TF-IDF und Hashing [3]	13
7.1	Sequenzdiagramm Hauptkomponente	21

Tabellenverzeichnis

4.1	Vergleich der Vor- und Nachteile von BoW und TF-IDF	13
6.1	Vergleich verschiedener technischer Umsetzungsansätze	19
A.1	Verwendete Hilfsmittel und Werkzeuge	30

1 Einleitung

1.1 Hintergrund: Die zunehmende Verbreitung von Fake News und deren gesellschaftliche Auswirkungen

1.1.1 Wann entstanden Fake News

Fake News sind ein allgegenwärtiges Problem, doch hatten Sie Ihren ersten Auftritt bereits 44BC im römischen Reich [4]. Auch während des amerikanischen Bürgerkriegs 1779 wurden Sie als politischer Schachzug von Benjamin Franklin genutzt. Dieser schickte einen Brief an Captain Samuel Gerrish und schrieb in diesem über Grausamkeiten der Briten und deren Verbündeten. Diese Informationen wurde so veranschaulicht, dass sie die öffentliche Meinung bewusst beeinflussen sollten [21].

Der eigentliche Begriff "Fake News" wurde erst viele Jahre später durch Donald Trump im amerikanischen Wahlkampf 2016 bekannt [2] und diente hierbei als politischer Kampfbegriff [6].

Unter anderem ist Fake News auch ein Teil von Propaganda [6], welche schon lange als Mittel zur Meinungsmanipulation eines Volkes genutzt wird.

Heute ist Fake News die größte Drohung zu unserer angeblich freien Presse [21].

1.1.2 Wie definieren sich Fake News und wie sind sie aufgebaut

Fake News sind bewusst erstellte Online-Falschmeldungen, die teilweise oder vollständig unwahre Inhalte verbreiten, um Leser*innen gezielt zu täuschen oder zu manipulieren. Sie imitieren klassische Nachrichtenformate, nutzen auffällige Titel, emotionale Bilder und strategisch gestaltete Inhalte, um Glaubwürdigkeit zu erzeugen und Aufmerksamkeit zu

gewinnen. Ziel ist es, durch das Verbreiten dieser Inhalte Klicks, Reichweite und damit finanzielle oder ideologische Vorteile zu erzielen [4].

Fake News fallen in die Kategorien Satire, Clickbait, Gerüchte, Stance News, Propaganda und Large Scale Hoaxes [21].

- **Satire:** ist eine humorvolle oder übertriebene Darstellung gesellschaftlicher oder politischer Themen, die Kritik üben soll.
- **Clickbait:** bezeichnet reißerische Überschriften oder Vorschaubilder, die Neugier wecken und zum Anklicken eines Inhalts verleiten sollen, oft ohne den Erwartungen gerecht zu werden.
- **Gerüchte:** sind unbestätigte Informationen, die sich schnell verbreiten und oft falsch oder irreführend sind.
- **Stance News:** sind Nachrichten, die eine klare Meinung oder politische Haltung einnehmen, statt neutral zu berichten.
- **Propaganda:** ist die gezielte Verbreitung von Informationen oder Meinungen, um das Denken und Handeln von Menschen zu beeinflussen, meist im Interesse einer bestimmten Gruppe oder Ideologie.
- **Large Scale Hoaxes:** sind absichtlich erfundene Falschmeldungen oder Täuschungen, die weit verbreitet werden und viele Menschen täuschen sollen.

Die eigentliche Nachricht ist aufgebaut in folgende Teile:

- **Quelle:** gibt den Ersteller der Nachricht an.
- **Titel:** erzielt die Aufmerksamkeit der Lesenden.
- **Text:** enthält die eigentliche Information der Nachricht.
- **Medien:** in Form von Bildern oder Videos.

Fake News können die Form von Text, Fotos, Filmen oder Audio annehmen und sind dementsprechend auf jeder Plattform auffindbar, die die Verbreitung nicht unterbindet. Die momentan populärste Plattform zum Teilen der Fake News ist WhatsApp [2].

1.1.3 Aus welcher Motivation entstehen Fake News

Das Hauptinteresse der Ersteller der Fake News ist das Verdienen von Geld. Auf die Artikel wird Werbung geschaltet und anhand einer entsprechenden Reichweite ergibt sich der verdiente Betrag. Je mehr Reichweite, desto mehr Verdienst für die Ersteller [4].

1.1.4 Warum verbreiten sich Fake News

In sozialen Medien neigen Nutzer aufgrund von FOMO (Fear of Missing Out) dazu, Fake News zu teilen, um Anerkennung zu gewinnen und soziale Zugehörigkeit zu erfahren. Besonders häufig werden kontroverse, überraschende oder bizarre Inhalte verbreitet – insbesondere dann, wenn sie starke Emotionen wie Freude, Wut oder Aufregung hervorrufen. Das Teilen solcher Inhalte stärkt das eigene Ansehen, da es signalisiert, über neue und relevante Informationen zu verfügen. Fake News bestehen meist aus eindrucksvoll präsentierten Falschinformationen [4].

Ein Grund für die schnelle Verbreitung von Fake News liegt in ihrer Aufmachung: Häufig wird die zentrale Aussage bereits in der Überschrift formuliert, oft mit Bezug auf konkrete Personen oder Ereignisse. Dadurch überspringen viele Leser den Artikel selbst, was die Wirkung von Schlagzeilen verstärkt. Die Inhalte sind meist kurz, wiederholend und wenig informativ. Anders als bei seriösen Nachrichten, bei denen Argumente überzeugen sollen, wirken Fake News über einfache Denkabkürzungen (Heuristiken) und die Bestätigung bestehender Überzeugungen. Nutzer müssen sich also nicht mit komplexen Inhalten auseinandersetzen, sondern lassen sich durch intuitive Übereinstimmungen überzeugen. Besonders bei geringer kognitiver Anstrengung – etwa durch Müdigkeit oder Unaufmerksamkeit – steigt die Wahrscheinlichkeit, dass Fake News geglaubt und weiterverbreitet werden [15].

1.1.5 Wer konsumiert Fake News

Laut [15] sind folgende Gruppen die größten Konsumenten:

- **Geringe Bildung oder digitale Kompetenz:** Personen mit niedriger formaler Bildung oder unzureichenden digitalen Fähigkeiten sind anfälliger für Falschinformationen.

- **Aussagen oder Nähe zur Informationsquelle:** Informationen von Personen, denen man persönlich nahe steht oder vertraut, werden eher geglaubt – unabhängig vom Wahrheitsgehalt.
- **Parteizugehörigkeit oder politische Überzeugung:** Menschen neigen dazu, Fake News zu glauben und zu verbreiten, wenn diese mit ihrer ideologischen Einstellung übereinstimmen.
- **Misstrauen gegenüber den Medien:** Wer etablierten Medien nicht vertraut, ist eher bereit, alternative (oftmals falsche) Quellen zu konsumieren und zu verbreiten.
- **Geringere kognitive Fähigkeiten:** Personen mit niedrigerer kognitiver Verarbeitungskapazität sind anfälliger für einfache, irreführende Inhalte und hinterfragen diese seltener kritisch.

Außerdem scheinen konservative, rechtsgerichtete Menschen, ältere Personen und weniger gebildete Menschen eher dazu zu neigen, Fake News zu glauben und zu verbreiten [4].

1.1.6 Probleme beim Erkennen von Fake News

Fake News können erst erkannt werden, nachdem diese erstellt und im Internet verbreitet wurden. [21]

1.1.7 Welche potenziellen Indikatoren zum Erkennen bei Fake News gibt es

Gerade im Bereich der sozialen Medien gibt es relativ zuverlässige Indikatoren, die Fake News als solche enttarnen [14]:

- **Fortlaufende Großschreibung:** Beispiel: GROßSCHREIBUNG
- **Übermäßige Nutzung von Satzzeichen:** Beispiel: !!!
- **Falsche Zeichensetzung am Satzende:** Beispiel: !!1
- **Übermäßige Nutzung von Emoticons, besonders auffälliger Emoticons**
- **Nutzung des Standard-Profilbildes**
- **Fehlende Account-Verifizierung, besonders bei prominenten Personen**

1.2 Wahl der Nachrichtenportale

Im Paper der University of Applied Sciences Upper Austria wird die Qualität verschiedener deutscher Nachrichtenportalen mit „Machine Learning“-Modellen getestet. Das Ergebnis zeigt, dass Spiegel, Die Zeit und Süddeutsche die besten Portale sind und Express, BZ-Berlin und Bild die schlechtesten - also auch am meisten Fake News verbreiten [22].

1.3 Zielsetzung: Entwicklung einer Software zur automatisierten Fake-News-Erkennung

Motiviert durch die Arbeiten der University of Applied Sciences Upper Austria und der TU Darmstadt werde ich in dieser Arbeit die Entwicklung eines weiteren Tools dokumentieren. Dieses Tool soll wie auch das Browser Plugin TrustyTweet eine Unterstützung zum Erkennen von Fake News anbieten. Ob dieses Tool als Browser Plugin oder als eine andere Form der Software implementiert wird, steht zum jetzigen Zeitpunkt noch nicht fest. Auch ob eine Black Box oder White Box - das heißt, kann der User zum Beispiel sehen, warum der Artikel als Fake News deklariert wird - Architektur genutzt wird, wird im Laufe der Arbeit entschieden. Ziel ist es, dass das Tool nicht wie TrustyTweet auf Twitter eingesetzt wird, sondern auf verschiedenen Nachrichtenportalen. Welche Portale hierfür genutzt werden, hängt von der noch ausstehenden Entscheidung der genutzten Datensätze ab.

1.4 Aufbau der Arbeit

2 Grundlagen und Begriffsdefinitionen

2.1 Definition „Fake News“: Merkmale, Ziele, Beispiele

2.1.1 Klassifizierungen

Linguistische Features werden von Textmaterial auf verschiedenen Leveln gesammelt, z.B. Buchstaben, Wörter, Sätze und Features auf dem Satzlevel (Häufigkeit von Funktionswörtern? und Sätzen) [26]

Text Tokenisierung [25]

Aufteilung der Features in Syntactic, Semantic, Sentiment, Lexical, Style-based [21] p7

2.2 Kategorisierung der Fake News Detection-Ansätze

2.3 Warum der Fokus auf Machine Learning?

2.4 Überblick über relevante Plattformen und deren Rolle im Medienkonsum

3 Maschinelles Lernen zur Fake News Erkennung

3.1 Grundlagen von ML

3.1.1 Überwachtes vs. unüberwachtes Lernen

3.1.2 Klassische ML-Modelle (Logistic Regression, Naive Bayes, SVM, Entscheidungsbäume)

3.2 Performance-Metriken: Accuracy, Precision, Recall, F1-Score

3.3 Herausforderungen im Kontext von Fake News

4 Natural Language Processing

4.1 Machine Learning

4.1.1 Textbereinigung und Vorverarbeitung

- **Titel und Inhalt der Artikel zusammenfügen [5]:** Damit keine wichtigen Informationen verloren gehen, werden Titel und Inhalt des Artikels zusammengefasst. Gerade der Titel kann durch z.B. Clickbait (siehe 1.1.2) schnell Hinweise auf eventuelle Fake News geben.
- **Akzente und Sonderzeichen entfernen [5] [19] [3]:** Akzente führen dazu, dass Wörter wie „café“ und „cafe“ unterschiedlich behandelt werden, obwohl sie semantisch gleich sind. Das Entfernen dieser erhöht die Generalisierung. Sonderzeichen stören einfache Tokenizer (z. B. bei Bag-of-Words), führen zu vielen seltenen Tokens und zu überdimensionierten Vektoren (siehe 4.1.2).
- **Alle Buchstaben zu Kleinbuchstaben konvertieren [19] [23] [3]:** Ähnlich wie zum vorherigen Punkt erhöht die durchgehende Kleinschreibung aller Buchstaben die Generalisierung und verhindert somit unnötige Duplikate im Vokabular.
- **Leere Spalten entfernen [23]:** Leere Spalten enthalten keine Information. Sie können bei der Vektorisierung oder Modellerstellung Fehler verursachen und werden als einfache Datenbereinigungsmaßnahme entfernt.
- **Kontraktionen auflösen (ans -> an das) [5]:** Im deutschen sind Kontraktionen zwar nicht so häufig wie im englischen, sie kommen aber trotzdem vor und sollten aufgelöst werden. Dies vermeidet fragmentierte Token und verbessert die Semantik und Trennbarkeit im Modell.

- **Stoppwörter entfernen [5] [19] [3]:** Wörter wie „der“, „ist“, „und“ tragen wenig zur inhaltlichen Differenzierung bei. Das Entfernen dieser verbessert die semantische Gewichtung relevanter Begriffe [20].
- **Rechtschreibfehler korrigieren [19]:** Tippfehler führen zu seltenen Tokens und stören die Generalisierung. In offiziellen Artikeln sind zwar selten Rechtschreibfehler zu finden, aber falls vorhanden, hilft die Korrektur zur Verbesserung der Modellqualität.
- **Lemmatisieren [5] [19] [3]:** Bei der Lemmatisierung werden verschiedene Wortformen auf die Grundform zurückgeführt („läuft“, „lief“, „laufen“ wird zu „laufen“). So erkennt das Modell gleiche Bedeutungen trotz grammatischer Variation.
- **Tokenisierung [19]:** In der Tokenisierung werden die Texte in einzelne Wörter oder Einheiten (Tokens) zerlegt, die für Modelle verarbeitbar sind. Dies ist eine Grundvoraussetzung für alle weiteren NLP-Schritte wie TF-IDF oder Word Embeddings.

Nutzung einer dualen Feature-Pipeline

Ein Problem welches das Entfernen der Akzente und Sonderzeichen und das Konvertieren aller Buchstaben zu Kleinbuchstaben mit sich bringt ist, dass viele wichtige Hinweise zum Erkennen von Fake News verloren gehen. Wie in Kapitel 1.1.7 beschrieben, sind fortlaufende Großschreibung, übermäßige Nutzung von Satzzeichen und falsche Zeichensetzung am Satzende potenzielle Indikatoren für Fake News.

Eine duale Feature-Pipeline kann dieses Problem lösen. Implementiert wird eine „cleaned“ Version (z.B. für inhaltliche Bedeutung) mit standardisierten, inhaltlichen Features und eine „rohe“ Version (z.B. für Stilmerkmale) mit stilistischen, rohen Textfeatures.

So werden semantische und stilistische Hinweise genau so genutzt wie ein Mensch es beim Lesen macht.

Die Notwendigkeit der Stilmerkmale ist aber diskutierbar. Die Datensätze werden ausschließlich aus Artikeln von offiziellen Nachrichtenportalen zusammengesetzt. Diese schreiben meist sauber, ohne Caps-Lock oder auffällige Sonderzeichen. Stilistische Merkmale wie viele Ausrufezeichen, Emojis oder absichtliche Rechtschreibfehler kommen dort nicht vor – also sind sie in diesem Fall auch keine verlässlichen Fake-News-Signale.

4.1.2 Merkmalsextraktion

Bag-of-words

Das Bag-of-Words-Modell ist ein einfaches Verfahren zur Textrepräsentation, bei dem ein Dokument als Vektor der Häufigkeiten einzelner Wörter dargestellt wird – unabhängig von deren Reihenfolge oder Kontext. Es zählt lediglich das Vorkommen jedes Wortes aus einem festen Vokabular [7].

TF-IDF

TF-IDF ist ein gewichtetes Modell zur Textdarstellung, das berücksichtigt, wie häufig ein Wort in einem Dokument vorkommt (TF) und wie selten es im gesamten Kontext ist (IDF). Es dient dazu, häufige, aber wenig informative Wörter zu reduzieren und aussagekräftige Begriffe zu betonen [13].

Sparse Matrizen werden sowohl von Bag-of-Words als auch von TF-IDF genutzt. Eine Matrix wird als sparse bezeichnet, wenn der Anteil der Nicht-Null-Werte im Verhältnis zur Gesamtanzahl der Dokumente sehr klein ist. Pro hinzugefügtem Dokument wird eine Zeile erstellt und pro Wort im Vokabular eine Spalte. Da jedes Dokument nur einen Bruchteil der Wörter des Gesamtvokabulars enthält, bestehen der Großteil einer solchen Matrix aus Nullen.

	Doc 1	Doc 2	Doc 3
baking	0	1	1
cake	1	1	1
chocolate	1	0	0
he	0	0	1
her	0	0	1
is	0	1	1
loves	1	0	0
she	1	1	0
surprise	0	0	1
to	0	0	1

(a) Bag-of-words Sparse Matrix [5]

	Doc 1	Doc 2	Doc 3
baking	0	0.52	0.32
cake	0.34	0.40	0.25
chocolate	0.58	0	0
he	0	0	0.42
her	0	0	0.42
is	0	0.52	0.52
loves	0.55	0	0
she	0.44	0.52	0
surprise	0	0	0.42
to	0	0	0.42

(b) TF-IDF Sparse Matrix [5]

Abbildung 4.1: Vergleich der Sparse Matrizen

In Abbildung 4.1 wurden den beiden Matrizen jeweils die drei Dokumente:

- Doc1 - She loves chocolate cake
- Doc2 - She is baking a cake
- Doc3 - He is baking a cake to surprise her

hinzugefügt. In der Matrix 4.1a werden in jeder Zelle in welcher das Dokument das entsprechende Wort beinhaltet eine 1 gesetzt. In 4.1b wird statt einer 1 eine Gewichtung über die Häufigkeit der Wörter in allen Dokumenten hinweg erstellt und eingetragen. Sie bewertet die Wichtigkeit eines Wortes in einem Dokument relativ zur gesamten Sammlung von Dokumenten. Dabei wird die Termfrequenz (TF) mit der invertierten Dokumentfrequenz (IDF) multipliziert. Je höher der resultierende Wert, desto relevanter ist das Wort für das jeweilige Dokument. Die Formel lautet:

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D) \quad (4.1)$$

Dabei ist t das Wort, d das Dokument und D die gesamte Dokumentensammlung [3].

Die TF misst, wie häufig ein bestimmter Begriff t in einem Dokument d vorkommt. Sie beschreibt die lokale Bedeutung eines Wortes innerhalb des Dokuments.

$$\text{tf}(t, d) = \frac{\text{Anzahl der Vorkommen von } t \text{ in } d}{\text{Gesamtanzahl der Wörter in } d} \quad (4.2)$$

Die IDF bewertet, wie selten ein Begriff t in der gesamten Dokumentensammlung D ist. Je seltener ein Begriff in vielen Dokumenten vorkommt, desto höher ist sein IDF-Wert.

$$\text{idf}(t, D) = \log \left(\frac{N}{\text{df}(t)} \right) \quad (4.3)$$

Dabei ist N die Gesamtanzahl der Dokumente in der Matrix und $\text{df}(t)$ die Anzahl der Dokumente, in denen der Begriff t vorkommt [18].

Die in [12] beschriebene Relevance Frequency (RF) ist eine überwachte Gewichtungsform der IDF-Komponente im TF-IDF, die nicht nur zählt, in wie vielen Dokumenten ein Begriff vorkommt, sondern berücksichtigt, in welchen Klassen der Begriff besonders häufig oder exklusiv ist. Die Formel lautet:

$$\text{rf}(t) = \log \left(2 + \frac{P(t)}{\max(1, N(t))} \right) \quad (4.4)$$

Mit $P(t)$ für die Anzahl der relevanten Dokumente (z. B. positive Klasse), in denen der Term t und $N(t)$ für die Anzahl der irrelevanten Dokumente (z. B. negative Klasse), in denen der Term t vorkommt.

Während klassisches IDF ein Wort umso höher gewichtet, je seltener es allgemein in der Gesamtmatrix ist, gewichtet RF hingegen ein Wort umso höher, je stärker es mit einer bestimmten Zielklasse assoziiert ist. Dadurch hebt RF Begriffe hervor, die klassenunterscheidend sind was beim Arbeiten mit überwachten Modellen relevant ist.

In der IF-IDF wird für IDF wird nun also RF eingesetzt und es ergibt sich folgende Formel:

$$\text{tfidf}(t, d) = \frac{\text{Anzahl der Vorkommen von } t \text{ in } d}{\text{Gesamtanzahl der Wörter in } d} \cdot \log \left(2 + \frac{P(t)}{\max(1, N(t))} \right) \quad (4.5)$$

- Mit dem Wort t und dem Dokument d
- $P(t)$ für die Anzahl der relevanten Dokumente (z. B. positive Klasse), in denen der Term t vorkommt
- $N(t)$ für die Anzahl der irrelevanten Dokumente (z. B. negative Klasse), in denen der Term t vorkommt

Vergleich Bag-of-words und TF-IDF

Aus 4.1 zu erkennen ist, dass TF-IDF in vielen Anwendungen leistungsfähiger ist als BoW. Insbesondere bei Texten mit hohem Vokabularumfang.

Hashing Vectorizer

Ein Hashing Vectorizer ist eine Methode zur Umwandlung von Text in numerische Merkmalsvektoren, ohne dass ein Vokabular explizit erstellt oder gespeichert wird. Stattdessen wird eine Hash-Funktion verwendet, um jedes Wort auf einen Index im Feature-Vektor abzubilden [5].

Bag-of-Words (BoW)	TF-IDF
Einfache Implementierung [7]	Berücksichtigt Wortwichtigkeit in gesamter Matrix [13]
Keine Gewichtung — häufige Wörter dominieren	Seltener, aber informativer Inhalt wird stärker gewichtet [8]
Hohe Dimensionalität (Sparse Matrix)	Gleiches Problem, aber mit informativeren Werten [1]
Ignoriert Wortreihenfolge und Kontext [24]	Gleiches Grundproblem, aber geringfügig bessere Performance [17]
Nützlich für einfache Klassifikatoren	Bessere Klassifikationsergebnisse in Kombination mit SVM oder Logistic Regression [16]

Tabelle 4.1: Vergleich der Vor- und Nachteile von BoW und TF-IDF

In Abbildung 4.2 wird der Vergleich zwischen Machine-Learning-Modellen unter Verwendung von BoW-, TF-IDF- und Hashing-Features gezeigt. Die y-Achse präsentiert den F1-Score, also die Precision und Recall-Werte der jeweiligen Modelle. Das Random-Forest-Modell (RF) zeigt eine schwache Leistung bei der Verwendung von Hashing, während die linearen Modelle (z.B. SVM (Support Vector Machine) und LR (Logistische Regression)) ihre F1-Werte mit Hashing-Features verbessern konnten.

Die Verbesserung ist aber nur minimal. Der F1-Score bei SVM ist ohne Hashing bei 0.89 und nach bei 0.90. Bei LR steigt der Wert auch von 0.87 und 0.88.

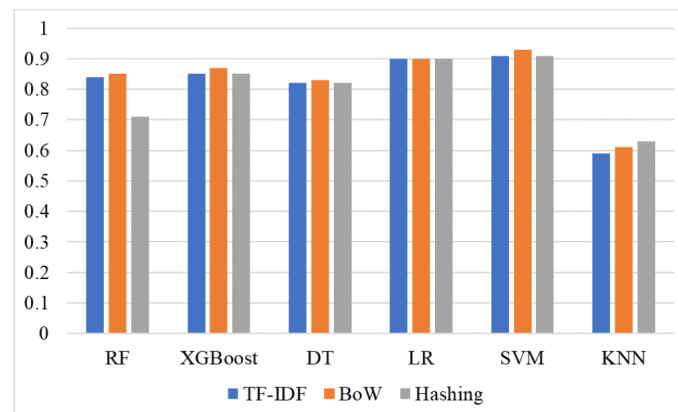


Abbildung 4.2: Vergleich verschiedener Modelle mit BoW, TF-IDF und Hashing [3]

4.1.3 Machine Learning Modelle

Naive Bayes

Support Vector Machines

Logistische Regression

4.2 Deep Learning

4.2.1 Word Embeddings

Word2Vec

GloVe

BERT-Tokenisierung

4.2.2 Deep Learning-Modelle

CNN

LSTM

Transformer: BERT

4.3 Hybride Modelle

4.3.1 convolutional neural network (CNN) along with bi-directional LSTM

[5]

5 Relevante Datensätze und Auswahlkriterien

5.1 Vorstellung verfügbarer deutscher Fake-News-Datensätze

5.2 Auswahl und Begründung des finalen Datensatzes

5.3 Herausforderungen bei deutschen Textdaten (Klassenverteilung, Quellenvielfalt)

6 Konzeption der Softwarelösung

6.1 Hauptkomponente

Die Hauptkomponente hat die Aufgabe die Artikel auf den verschiedenen Nachrichtentportalen zu lesen und zu ergänzen. Hierfür muss erkannt werden auf welcher Seite sich der Nutzer befindet. Außerdem muss das html dieser Seite ausgelesen und analysiert werden können.

Als Beispiel die Seite Bild.de: Je nach Fenstergröße hat die Seite entweder die Domäne *https://www.bild.de/* oder *https://m.bild.de/*.

Die Startseite ist wie folgt aufgebaut:

```
<!DOCTYPE html>
<html>
  <head>
    ...
  </head>
  <body>
    <div id="app">
      ...
      <div id="page-content">
        <header/>
        <main>
          <!-- Es gibt auf der Startseite
               ueber 50 dieser section-Elemente -->
          <section>
            <article/>
          </section>
```



```
        </main>
      <footer />
    </div>
  </div>
</body>
</html>
```

Wenn ein Artikel geöffnet ist, ist der DOM dem der Startseite sehr ähnlich. Der einzige wesentliche Unterschied ist, dass im *main*-Element nur noch ein *article*-Element ist und nicht beliebig viele *section*-Elemente. Ob ein Artikel geöffnet ist, kann also anhand der Anzahl der *article*-Elemente bestimmt werden.

```
<article>
  <h2 class="document-title_document-title-article">
    <span class="kicker">Kicker des Artikels</span>
    <span class="headline">Titel des Artikels</span>
  </h2>
</article>
<div class="article-body">
  <!-- Pro Artikel gibt es ca. 10 p-Elemente -->
  <p>Inhalt des Artikels</p>
</div>
```

Der Titel und Inhalt des Artikels kann den entsprechenden html-Elementen entnommen werden. Diese werden dann an die API gesendet und dort verarbeitet. Der Rückgabewert der API enthält dann die Info ob der Artikel falsch oder echt ist. Diese wird in einem von der Hauptkomponente erzeugten *div*-Container über dem Artikel eingefügt.

Zur Bestimmung des geeignetsten Tools um diese Anforderungen umzusetzen wurden verschiedene Umsetzungsansätze verglichen (siehe 6.1). Aufgrund des begrenzten Zugriffs auf die zu analysierende Seiten, bieten sich die beiden Client-seitigen Umsetzungen eine Chrome Extension zu implementieren oder über Tampermonkey Userscripts auszuführen am ehesten an.

Im Vergleich zu Userscripts unterstützt die Extension mehrere Komponenten (Content Scripts, Background Scripts, Popup, Optionsseite). Anhand dieser können der DOM beobachtet, ein persistenter Speicher genutzt, Kontextmenüs erstellt und auf Browseraktionen reagiert werden (z.B. Tabwechsel, Navigation). Ein Userscript hingegen ist ein

einfaches Script, das nur beim Laden einer Seite aktiv ist und dementsprechend keine Hintergrundverarbeitung und keine erweiterten UI-Komponenten zur Verfügung stellt.

Zur Implementierung der Hauptkomponente wird also eine Chrome Extension genutzt.

Kriterium	Chrome Extension	Userscript (Tamper-monkey)	Proxy-Server	Scraper Plattform +
DOM-Zugriff beim Nutzer	Ja	Ja	Nein	Nein
Einbindung auf bild.de direkt	Ja	Ja	Ja (indirekt)	Nein
Installation durch Nutzer	Mittel	Einfach	Nicht erforderlich	Nicht erforderlich
Komplexität der Umsetzung	Mittel	Gering	Hoch	Mittel
Wartbarkeit & Updates	Gut	Gut	Aufwändig	Mittel
Performance beim Nutzer	Hoch	Hoch	Hoch	Hoch
Skalierbarkeit	Hoch	Eingeschränkt	Mittel	Hoch
Für öffentliche Verbreitung geeignet	Ja	Eingeschränkt	Eingeschränkt	Ja
API-Nutzung zur Fake-Erkennung	Ja	Ja	Ja	Ja
Entwicklerkontrolle über UI	Hoch	Mittel	Hoch	Mittel

Tabelle 6.1: Vergleich verschiedener technischer Umsetzungsansätze

7 Umsetzung des Prototyps

7.1 Implementierung der Chrome Erweiterung

Die Chrome Erweiterung wurde mit der Version Manifest 3 implementiert. Genutzt wurden ein *Service Worker* ein *Content Script* pro Nachrichtenportal und ein *Popup*.

Service Worker steuern eine Seite genau dann, wenn ein Service Worker auf dieser Netzwerkanfragen in seinem Namen abfangen kann. Der Service Worker kann dann Aufgaben für die Seite innerhalb eines bestimmten Scopes ausführen

Der Lifecycle eines Service Workers ist in folgende Events unterteilt: installing, installed, activating, activated.

Nach Abschluss der Aktivierung steuert der Service Worker die Seite standardmäßig erst bei der nächsten Navigation oder Seitenaktualisierung [11].

Content Scripts sind Dateien, die im Kontext von Webseiten ausgeführt werden. Mit dem standardmäßigen Document Object Model (DOM) können sie Details der Webseiten lesen, die der Browser besucht, Änderungen daran vornehmen und Informationen an die übergeordnete Erweiterung weitergeben [10].

Die Kommunikation mit den Service Worker erfolgt über die Extension-API *runtime*.

Pop-ups sind Aktionen, bei denen ein Fenster angezeigt wird, über das Nutzer mehrere Erweiterungsfunktionen aufrufen kann. Sie werden durch ein Tastenkürzel, durch Klicken auf das Aktionssymbol der Erweiterung oder durch Drücken von `chrome.action.openPopup()` ausgelöst. Pop-ups werden automatisch geschlossen, wenn der Nutzer sich auf einen Bereich des Browsers außerhalb des Pop-ups konzentriert [9].

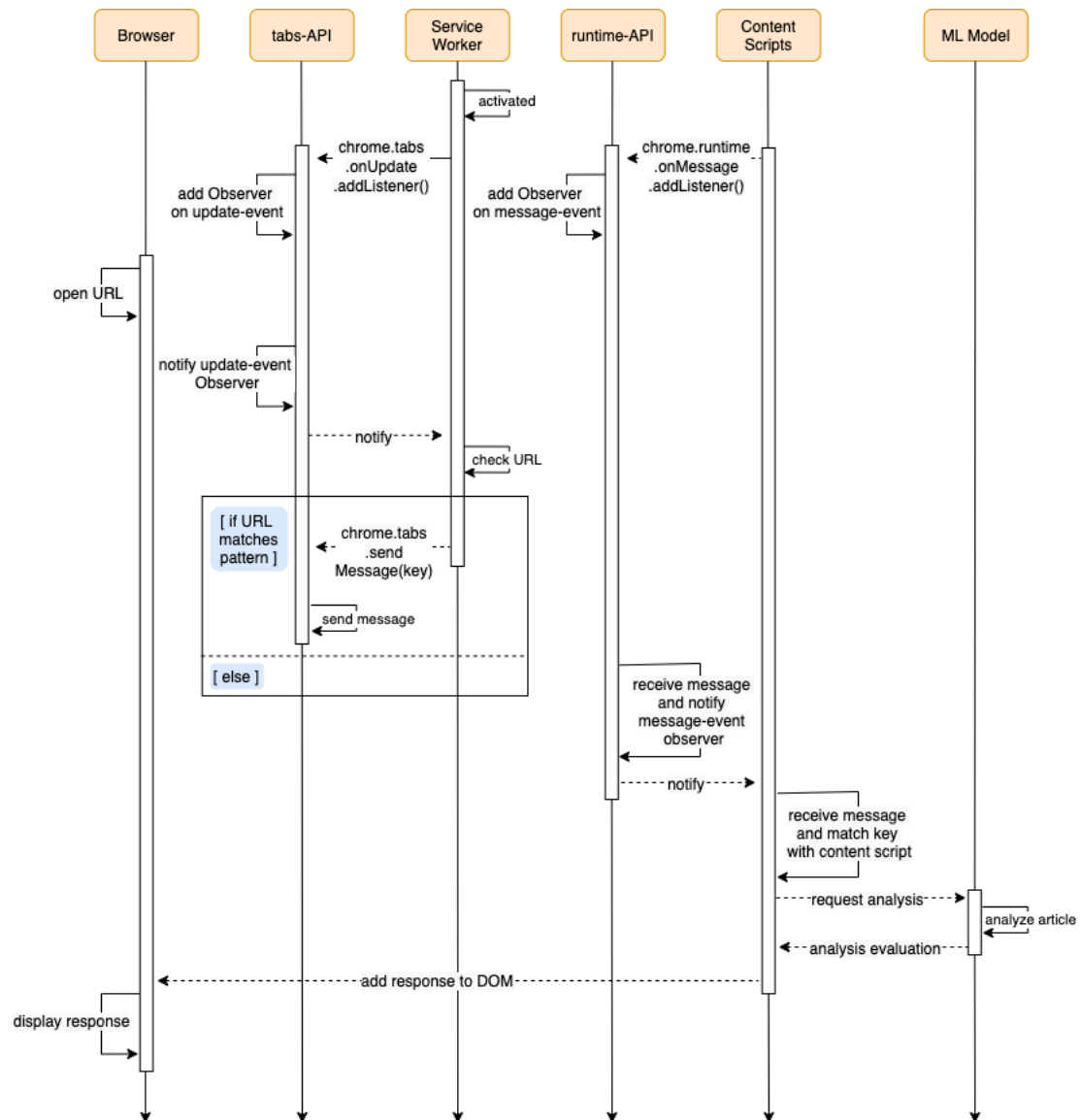


Abbildung 7.1: Sequenzdiagramm Hauptkomponente

In Abbildung 7.1 zu sehen ist das Sequenzdiagramm der Hauptkomponente. Wie in Kapitel 6.1 beschrieben wird zuerst die URL geprüft. Erfüllt diese die vorgegebenen Bedingungen wird der geöffnete Artikel gelesen und von einer weiteren Anwendung analysiert. Anschließend wird das Ergebnis der Analyse in einem *div*-Container über dem Artikel eingefügt.

Um die Veränderungen im Browser zu überwachen wurde die *tabs*-API von Chrome genutzt. Anhand dieser kann das Tab-System eines Browsers überwacht und zum Beispiel auch auf jede Veränderung der URL reagiert werden. Außerdem ermöglicht die API das Versenden von Nachrichten an alle aktiven Content Scripts. Diese werden dann im jeweiligen Content Script über die *runtime*-API empfangen und ausgelesen.

8 Evaluation und Ergebnisse

9 Diskussion

10 Fazit und Ausblick

Literaturverzeichnis

- [1] ALZAMI, F. ; UDAYANTI, E. D. ; PRABOWO, D. P.: Document Preprocessing with TF-IDF to Improve the Polarity Classification Performance. In: *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control* (2020). – URL <https://scholar.archive.org/work/pfclh2k6tffsdeh5ezp3qtdkoy/access/wayback/http://202.52.52.28/index.php/kinetik/article/download/1066/pdf>
- [2] ASHISH ; SONIA ; ARORA, Monika ; HEMRAJ ; RANA, Anurag ; GUPTA, Gaurav: An Analysis and Identification of Fake News using Machine Learning Techniques. In: *2024 11th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2024, S. 634–638
- [3] ASLAM, Naila ; XIA, Kewen ; RUSTAM, Furqan ; HAMEED, Afifa ; ASHRAF, Imran: Using Aspect-Level Sentiments for Calling App Recommendation with Hybrid Deep-Learning Models. In: *Applied Sciences* 12 (2022), 08, S. 8522
- [4] BAPTISTA, Joao P. ; GRADIM, Anabela: Understanding Fake News Consumption: A Review. In: *Social Sciences* 9 (2020), Nr. 10. – URL <https://www.mdpi.com/2076-0760/9/10/185>. – ISSN 2076-0760
- [5] BUDDHADEV, Manan ; PAREKH, Virtee: Fake News Detection: Benchmarking Machine Learning and Deep Learning Approaches. In: *ESP Journal of Engineering and Technology Advancements* 5 (2025), 04, S. 39–46
- [6] BÜRKER, Michael: *Fake-News, Propaganda & Co: Wie behalte ich den Überblick?* 2022. – URL <https://www.haw-landshut.de/aktuelles/beitrag/fake-news-propaganda-co-wie-behalte-ich-den-ueberblick>. – Zugriffsdatum: 2025-04-23. – Interview geführt von EINFALLSreich, Hochschule Landshut

- [7] CICHOSZ, Piotr: A Case Study in Text Mining of Discussion Forum Posts: Classification with Bag of Words and Global Vectors. In: *International Journal of Applied Mathematics and Computer Science* (2018). – URL <https://sciendo.com/pdf/10.2478/amcs-2018-0060>
- [8] DAS, M. ; ALPHONSE, P. J. A.: A Comparative Study on TF-IDF Feature Weighting Method and Its Analysis Using Unstructured Dataset. In: *arXiv preprint arXiv:2308.04037* (2023). – URL <https://arxiv.org/pdf/2308.04037>
- [9] DEVELOPERS, Chrome for: *Add Popup*. <https://developer.chrome.com/docs/extensions/develop/ui/add-popup>. 2025. – Zugriff am 11. Mai 2025
- [10] DEVELOPERS, Chrome for: *Content scripts*. <https://developer.chrome.com/docs/extensions/develop/concepts/content-scripts>. 2025. – Zugriff am 11. Mai 2025
- [11] DEVELOPERS, Chrome for: *A service worker's life*. <https://developer.chrome.com/docs/workbox/service-worker-lifecycle>. 2025. – Zugriff am 11. Mai 2025
- [12] DOMENICONI, Giacomo ; MORO, Gianluca ; PASOLINI, Roberto ; SARTORI, Claudio: A Comparison of Term Weighting Schemes for Text Classification and Sentiment Analysis with a Supervised Variant of tf.idf, 02 2016, S. 39–. – ISBN 978-3-319-30162-4
- [13] ELOV, B. B. ; KHAMROEVA, S. M. ; ALAYEV, R. H.: Methods of Processing the Uzbek Language Corpus Texts. In: *Journal of Open Innovations* (2023). – URL <https://cyberleninka.ru/article/n/methods-of-processing-the-uzbek-language-corpus-texts>
- [14] HARTWIG, Katrin ; REUTER, Christian: *Fake News technisch begegnen – Detektions- und Behandlungsansätze zur Unterstützung von NutzerInnen*. S. 133–149. In: KLIMCZAK, Peter (Hrsg.) ; ZOGLAUER, Thomas (Hrsg.): *Wahrheit und Fake im postfaktisch-digitalen Zeitalter: Distinktionen in den Geistes- und IT-Wissenschaften*. Wiesbaden : Springer Fachmedien Wiesbaden, 2021. – URL https://doi.org/10.1007/978-3-658-32957-0_7. – ISBN 978-3-658-32957-0
- [15] HORNE, Benjamin ; ADALI, Sibel: This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real

- News. In: *Proceedings of the International AAAI Conference on Web and Social Media* 11 (2017), 03
- [16] IYER, Vivek: A Comparative Analysis of Sentiment Classification Models for Improved Performance Optimization. In: *NHSJS (National High School Journal of Science)* (2024). – URL <https://nhsjs.com/wp-content/uploads/2024/05/A-Comparative-Analysis-of-Sentiment-Classification-Models-for-Improved-Performance-Optimization.pdf>
- [17] PARMAR, M. ; TIWARI, A.: Enhancing Text Classification Performance using Stacking Ensemble Method with TF-IDF Feature Extraction. In: *5th International Conference on Artificial Intelligence and Data Science*, URL <https://ieeexplore.ieee.org/abstract/document/10493890/>, 2024
- [18] QAISER, Shahzad ; ALI, Ramsha: Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. In: *International Journal of Computer Applications* 181 (2018), 07
- [19] SABIR, Muhammad ; KHAN, Talha ; AZAM, Muhammad: A Comparative Study of Traditional and Hybrid Models for Text Classification. (2025), 03
- [20] SARKAR, Dipanjan: *A Practitioner's Guide to Natural Language Processing (Part I) – Processing & Understanding Text*. <https://towardsdatascience.com/a-practitioners-guide-to-natural-language-processing-part-i-processing-understanding-text-9f4abfd13e72/>. 2018. – Zugriff am 12. Mai 2025
- [21] SHARMA, Upasna ; SINGH, Jaswinder: A comprehensive overview of fake news detection on social networks. In: *Social Network Analysis and Mining* 14 (2024), Nr. 1, S. 120. – URL <https://doi.org/10.1007/s13278-024-01280-3>. ISBN 1869-5469
- [22] SIMONE, Sandler ; OLIVER, Krauss ; CLARA, Diesenreiter ; ANDREAS, Stöckl: Detecting Fake News and Performing Quality Ranking of German Newspapers Using Machine Learning. In: *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 2022, S. 1–5
- [23] SUDHAKAR, M. ; KALIYAMURTHIE, K.P.: Detection of fake news from social media using support vector machine learning algorithms. In: *Measurement: Sensors* 32

- (2024), S. 101028. – URL <https://www.sciencedirect.com/science/article/pii/S2665917424000047>. – ISSN 2665-9174
- [24] UMAR, M. ; ABUBAKAR, H. D. ; BAKALE, M. A.: Sentiment Classification: Review of Text Vectorization Methods: Bag of Words, TF-IDF, Word2vec and Doc2vec. In: *SLU Journal of Science and Technology* (2022). – URL https://www.academia.edu/download/107531976/Mahmood_and_Haisal_pub2.pdf
- [25] WAGNER, Wiebke: Steven Bird, Ewan Klein and Edward Loper: Natural Language Processing with Python, Analyzing Text with the Natural Language Toolkit. In: *Language Resources and Evaluation* 44 (2010), Nr. 4, S. 421–424. – URL <https://doi.org/10.1007/s10579-010-9124-x>. ISBN 1574-0218
- [26] ZHANG, Xichen ; HABIBI LASHKARI, Arash ; A. GHORBANI, Ali: A Lightweight Online Advertising Classification System using Lexical-based Features. In: *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017) - SECRIPT INSTICC* (Veranst.), SciTePress, 2017, S. 486–494. – ISBN 978-989-758-259-2

A Anhang

A.1 Verwendete Hilfsmittel

In der Tabelle A.1 sind die im Rahmen der Bearbeitung des Themas der Bachelorarbeit verwendeten Werkzeuge und Hilfsmittel aufgelistet.

Tabelle A.1: Verwendete Hilfsmittel und Werkzeuge

Tool	Verwendung
L ^A T _E X	Textsatz- und Layout-Werkzeug verwendet zur Erstellung dieses Dokuments

Erklärung zur selbständigen Bearbeitung

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

Ort

Datum

Unterschrift im Original