INSTITUT FÜR INFORMATIK
Datenbanken und Informationssysteme

Universitätsstr. 1 D–40225 Düsseldorf

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Automatisierte Vorhersage von Benutzerreaktionen auf Facebook

Roland Kahlert

Masterarbeit

Beginn der Arbeit:	02. Januar 2017
Abgabe der Arbeit:	27. Juni 2017
Gutachter:	Prof. Dr. Stefan Conrad
	Prof. Dr. Martin Lercher

Erklärung

Hiermit versichere ich, dass ich diese Masterarbeit selbstständig verfasst habe. Ich habe dazu keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Düsseldorf, den 27. Juni 2017

_____

Roland Kahlert

Zusammenfassung

Das soziale Netzwerk Facebook gilt als populärste Plattform und kann kontinuierlich wachsende Benutzerzahlen vermelden. Im Februar 2016 wurde eine neue Funktion eingeführt, mit der es möglich ist, auf Beiträge anderer Benutzer zu „reagieren", indem eine von sechs Emotionen ausgewählt wird. Populäre Facebook-Präsenzen, wie die von Nachrichtenseiten, werden von sehr vielen Benutzern gelesen. Die Beiträge auf diesen Präsenzen, die zu den Nachrichtenartikeln führen, werden dementsprechend in großer Anzahl mit Benutzerreaktionen versehen. Dadurch werden die Nachrichtenartikel mit Emotionen versehen und eignen sich als Datensatz für eine Emotionsanalyse.

Im Bereich des *Natural Language Processings* ist die Emotionsanalyse eine Spezialisierung der Sentimentanalyse. Anstatt einen Text als positiv oder negativ zu klassifizieren, soll genauer untersucht werden, welche Emotionen ein Text beim Leser auslösen kann. Bisher mangelte es an Datensätzen, die für diesen Anwendungszweck geeignet sind und umfangreich genug sind. Diese Arbeit zeigt, dass es möglich ist, aus den Beiträgen auf Facebook und den Artikeln auf Nachrichtenseiten einen beliebig großen Datensatz zu extrahieren.

Mit dieser Grundlage wird mittels *Maschinellen Lernens* ein Modell entwickelt, dass die Verteilung der Benutzerreaktionen auf Facebook eines Nachrichtenbeitrags vorhersagen kann. Zur Ermittlung des besten Modells werden verschiedene Regressionsverfahren gegenüber gestellt. Dabei weist die lineare Regression, sowie die ähnlich arbeitende Ridge Regression, den geringsten Fehler in der Vorhersage auf.

Weiterhin werden auch verschiedene Features vorgestellt und bezüglich ihres Vorhersagefehlers verglichen. Zur Anwendung kommen dabei unter anderem Textstatistiken, Word2Vec, Emotionslexika und der Bag-of-Words-Ansatz. Während der Evaluation erzielt Letzterer die besten Ergebnisse. Für dieses Ergebnis ist es nötig, als Textgrundlage die Nachrichtenartikel zu verwenden, eine zusätzliche Gewichtung durch TD-IDF anzuwenden und mittels LSI 1000 Konzepte zu identifizieren. Mit dem Word2Vec-Verfahren können die zweitbesten Vorhersagen beobachtet werden. Weiterhin werden Kombinationen der besten Features untersucht, um die Vorhersageleistung zusätzlich verbessern zu können. Letztlich kann ein Modell erstellt werden, dessen Vorhersagefehler um 20% geringer ausfällt als die der Baseline.

Inhaltsverzeichnis

1 Einleitung

Zunächst wird erläutert, was zur Themenwahl dieser Abschlussarbeit motivierte. Daraufhin wird die Zielsetzung vorgestellt, um ein Verständnis des Zwecks der Arbeit zu ermöglichen.

1.1 Motivation

In der täglichen Kommunikation spielen soziale Medien eine große Rolle. Insbesondere gilt Facebook[1] mit 1.23 Milliarden täglich aktiven Benutzern[2] als eine der am häufigsten genutzten Plattformen. Private Benutzer können dort ebenso wie gewerbliche Benutzer eigene Beiträge in Form von Texten, Bildern oder Videos erstellen und Beiträge Anderer kommentieren oder gefühlte Emotionen mitteilen.

Im Forschungsbereich der computergestützten Analyse natürlicher Sprache ist die Sentimentanalyse ein intensiv untersuchtes Teilgebiet. Dabei werden Texte automatisiert als positive, negative oder neutrale Aussage klassifiziert, so wie es beispielsweise bei der Bewertung von Kundenrezensionen [HL04] oder Filmkritiken [KI06] der Fall ist.

Darauf aufbauend kann durch die Emotionserkennung eine feingranulare Bewertung der Aussagen getroffen werden, um somit einen tieferen Einblick in die Gefühle der Autoren gewinnen zu können [KLC17]. Allerdings ist diese Disziplin von der Forschung noch relativ unbeachtet, da es an für Forschungszwecke nutzbaren Datensätzen mangelt. Buechel und Hahn analysierten [BH16] die Verfügbarkeiten von Datensätzen für Emotionsanalysen und kamen zu dem Schluss, dass es lediglich drei bekannte Sätze mit relativ geringem Umfang von unter 3 000 Einträgen gab. Daraufhin entwickelten sie selbst einen Textkorpus bestehend aus 10 000 Textsätzen [BH17].

Um einen Datensatz mit beliebig vielen Einträgen erzeugen zu können, ist Facebook als Datenquelle in der Emotionsanalyse interessant, da durch die Funktion der Benutzerreaktionen ein Beitrag mit Emotionen verknüpft werden kann. Dies ist für Methoden des maschinellen Lernens essentiell, um darauf aufbauend Vorhersagen treffen zu können.

1.2 Zielsetzung

In dieser Ausarbeitung soll mit Mitteln des maschinellen Lernens ein Modell entwickelt werden, das in der Lage ist, die Verteilung von Benutzerreaktionen eines Beitrags auf Facebook vorherzusagen. Diese Vorhersage soll auf Basis des im Beitrag vorhandenen Textes sowie etwaiger vom Beitrag verlinkten Artikel getroffen werden. Zur Erarbeitung des Modells werden Facebook-Beiträge und Nachrichtenartikel ausgewählter deutscher und englischer Zeitungen gesammelt.

Ein solches Modell könnte daraufhin als Basis eines Werkzeugs verwendet werden, das es beispielsweise Journalisten ermöglicht, die beim Leser hervorgerufenen Emotionen bereits vor Veröffentlichung des Artikels abzuschätzen. Darüber hinaus wäre es auch

[1] http://www.facebook.com
[2] http://newsroom.fb.com/company-info/

für Marketing-Zwecke interessant, die Reaktion von Menschen auf Texte vorhersagen zu können.

1.3 Gliederung der Arbeit

Um die Umsetzung näher beleuchten zu können, ist es zuvor notwendig, in Kapitel 2 die grundlegenden Begriffe zu erläutern. Daraufhin werden in Kapitel 3 die Bausteine beschrieben, aus denen das entwickelte Modell besteht. Im anschließenden Kapitel folgt eine Übersicht der verschiedenen Algorithmen, die in dieser Arbeit genutzt werden, um das maschinelle Lernen zu bewerkstelligen. In Kapitel 6 werden verwandte Arbeiten vorgestellt, die sich in der gleichen Domäne bewegen oder sich ebenfalls der Emotionsanalyse widmen. Darauffolgend wird in Kapitel 5 die Evaluation durchgeführt. Einleitend wird dabei der genutzte Datensatz vorgestellt und die eingesetzte Messmethode definiert. Danach werden die Ergebnisse visualisiert um darauf aufbauend eine Interpretation der Vorhersagequalität zu formulieren. Mit dem abschließenden Fazit wird die Arbeit rekapituliert und es wird ein Ausblick auf mögliche Weiterentwicklungen gewährt.

2 Grundlagen

In diesem Kapitel werden die grundlegenden Prinzipien erläutert, die zum Verständnis der Arbeit notwendig sind. Es werden verschiedene Emotionsmodelle vorgestellt, sowie das Natural Language Processing und das Maschinelle Lernen erklärt.

2.1 Benutzerreaktionen auf Facebook

Im Februar 2016 hat Facebook, zusätzlich zu dem einfachen *Like*, weitere Reaktionsmöglichkeiten zu Beiträgen hinzugefügt[3]. Der Benutzer hat somit die Möglichkeit, nicht bloß sein „Gefallen" mitzuteilen, sondern kann seine Emotionen differenzierter durch eine der Kategorien *Like*, *Love*, *Haha*, *Wow*, *Sad* und *Angry* ausdrücken. Die Reaktionen werden dabei auf der Plattform überwiegend bildlich durch die in Abbildung 1 dargestellten Symbole repräsentiert.

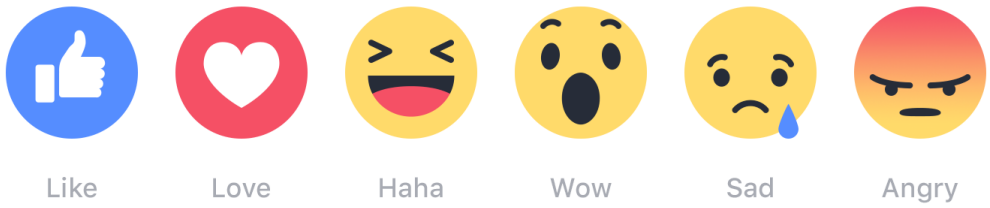

Abbildung 1: Bildliche Darstellung der Reaktionstypen

Darüber hinaus werden temporär die in Abbildung 2 gezeigten zusätzlichen Reaktionen durch Facebook freigeschaltet. Die Reaktion *Thankful* wird jedes Jahr für zwei Wochen zu Muttertag aktiviert[4]. Im Juni 2017 wurde außerdem die *Pride*-Reaktion hinzugefügt[5]. Dadurch wird es Benutzern ermöglicht, ihre Unterstützung für die LGBTQ-Community (Lesbian, Gay, Bisexual, Transgender und Queer) auszudrücken.

Abbildung 2: Temporäre Reaktionen *Thankful* und *Pride*

[3] http://newsroom.fb.com/news/2016/02/reactions-now-available-globally/
[4] http://fortune.com/2017/05/12/facebook-thankful-flower-reaction-back/
[5] https://www.buzzfeed.com/blakemontgomery/how-to-get-a-rainbow-flag-reaction-on-facebook

2.2 Emotionsmodelle

Neben den von Facebook eingeführten Emotionsklassen existieren bereits andere Modelle. Ekman leitete 1992 [Ekm92] aus menschlichen Gesichtsausdrücken die sechs **Basisemotionen** *Wut, Abscheu, Angst, Freude, Trauer* und *Überraschung* ab.

Eine ähnliche Ausprägung besitzt **Plutchiks Rad der Emotionen** [Plu80], welches Gefühlsausprägungen in eine der Klassen *Wut, Neugier, Abscheu, Angst, Freude, Trauer, Überraschung* und *Vertrauen* einordnet. Dabei ist jede Klasse in verschiedene Intensivitätsstufen unterteilt, die beispielsweise eine Steigerung der Gereiztheit über Verärgerung bis hin zu Wut abbildbar machen.

Einen anderen Ansatz verfolgt Russell und Mehrabians *Valence-Arousal-Dominance* (**VAD**) Modell [RM77]. Dabei handelt es sich um einen dreidimensionalen Vektorraum in dem eine Emotion eine unterschiedliche Ausprägung in jeder der Dimensionen besitzen kann. Die Achse *Valence* beschreibt wie viel Freude oder Unfreude gefühlt wird, *Arousel* drückt den Grad Anteilnahme aus und *Dominance* charakterisiert das Level an Kontrolle das verspürt wird.

2.3 Natural Language Processing

Unter dem Begriff *Natural Language Processing* (**NLP**) oder *Computerlinguistik* werden Ansätze zusammengefasst, die natürliche Sprache in Textform durch Computerunterstützung syntaktisch sowie semantisch verarbeiten. Dabei können verschiedene Ziele, wie etwa Stilanalyse oder Informationsextraktion, verfolgt werden. Die Aufgabenstellung ist entscheidend dafür, welche Teilmenge an Werkzeugen dabei zum Einsatz kommen. Im Folgenden werden die Mittel vorgestellt, die in Kapitel 3 zum Einsatz kommen.

2.3.1 Tokenisierung

Die Aufgabe der **Tokenisierung** besteht darin, ein Dokument in einzelne Bestandteile, die sogenannten Tokens, zu zerlegen. Ein Token kann aus einem Wort, Satzzeichen oder aus Ziffern bestehen. Um die Zerlegung des Dokuments vollziehen zu können, ist eine Definition des Begriffs „Wort" notwendig. Carstensen et al. [CEE+09] stellen heraus, dass in segmentierten Schriftsystemen, die unter anderem von der englischen und deutschen Sprache verwendet werden, ein Wort als triviale Verkettung alphanumerischer Zeichen definiert ist und von Leerzeichen oder Punktuation begrenzt werden. Beispielsweise würde der Satz „In einer Fußballmannschaft spielt ein Torwart, sowie 10 Feldspieler." wie folgt tokenisiert werden:

2.3.2 N-Gramme

Eine Erweiterung der Tokens stellen **N-Gramme** dar. Ein N-Gramm ist eine Sequenz von n Tokens, die aus einem Dokument extrahiert wurden [JM00]. Ein 2-Gramm (oder **Bi-**

gramm) für Wort-Tokens ist beispielsweise die Textpassage *„Er sagt"*. Ein 1-Gramm (oder **Unigramm**) ist hingegen nur ein einzelnes Wort und somit äquivalent zu einem Token.

2.3.3 Part-of-Speech Tagging

Der Prozess des *Part-of-Speech* (**POS**) Taggings beschreibt die Zuordnung eines Tokens zu einem POS-Tag. Dadurch wird die Wortart eines Tokens bestimmt. Dies ermöglicht zwischen Nomen, Adjektiven und Verben zu unterscheiden. Häufig wird dabei für englische Texte das von Santorini beschriebene *Penn Treebank Tagset* [San90] eingesetzt, welches eine feingranulare Unterscheidung von 36 Klassen für Wörter ermöglicht, indem anhand der Tempi des Numerus und weiterer Eigenschaften unterschieden wird.

2.3.4 Wortstammbildung

Lovins beschreibt [Lov68] die **Wortstammbildung** als regelbasierten Prozess, indem ein Wort auf den jeweiligen Wortstamm zurückgeführt wird, so dass unter anderem der Kasus oder die Flexion eliminiert werden. Beispielsweise teilen sich die beiden Wörter *„fahren"* und *„Fahrer"* den Wortstamm *„fahr"*.

Das Ziel dieser Transformation besteht darin, Wörter mit ähnlicher Semantik durch einen gemeinsamen Repräsentanten darzustellen. Dadurch ist es möglich, die in Kapitel 3 vorgestellten Verfahren zu verbessern. Diese können ansonsten keine Verbindung zwischen unterschiedlich gebildeten Ausprägungen eines Wortes herstellen.

2.3.5 Lemmatisierung

Die deutsche Sprache gilt als sehr stark flektierende Sprache. Das bedeutet, dass häufig starke Verben verwendet werden, die bei der Beugung vom Wortstamm abweichen, wie es beispielsweise beim Wort *„singen"*, im Präteritum *„sang"*, erkennbar ist. Die **Lemmatisierung** bezeichnet die Überführung eines Wortes auf das zugehörige Lexem [CEE+09]. Ein Lexem bildet dabei im Gegensatz zum Wortstamm immer ein valides Wort. Beim vorherigen Beispiel lautet das Lexem wiederum *„singen"*. Zur Überführung kommen Wörterbücher wie *WordNet* [Mil95] für englische Texte oder *IWNLP* [LC15] für deutsche Texte, zum Einsatz.

2.4 Maschinelles Lernen

Die Motivation, das Maschinelle Lernen (**ML**) zu Nutzen, besteht darin, Wissen aus einer Menge an Daten zu extrahieren, wenn die Datenmenge zu groß oder komplex ist, um manuell Strukturen erkennen zu können. ML-Algorithmen verfolgen das Ziel, aus bekannten Daten ein Modell zu erlernen, das in der Lage ist, Vorhersagen für noch unbekannte Daten zu treffen [KP98].

Abhängig vom Einsatzgebiet und der Datenstruktur eignen sich bestimmte Algorithmen besser dazu, den Anwendungszweck zu realisieren als andere. Dabei lassen sich die Verfahren in eine der Klassen *unüberwachtes* und *überwachtes* Lernen einteilen.

2.4.1 Unüberwachtes Lernen

Zur Realisierung des unüberwachten Lernens haben die Daten in folgender Struktur vorzuliegen:

$$(x_{1,i}, ..., x_{m,i}) \quad \forall 1 \leq i \leq n$$

Wobei n die Anzahl der Datenfälle bezeichnet und jeder Datenfall als numerischer Vektor der Länge m ausgeprägt ist. Dabei spricht man auch von m-dimensionalen Featurevektoren.

Ein Anwendungsfall des unüberwachten Lernens ist das **Clustering**, bei dem die Datenfälle in Cluster gruppiert werden. In der Regel wird ein Distanzmaß, wie etwa die euklidische Distanz, zwischen den Datenfällen definiert. Daraufhin wird jeder Datenfall zu dem Cluster zugeordnet, in dem eine maximale Ähnlichkeit zu Datenfällen in diesem Cluster und eine maximale Unähnlichkeit zu Fällen in anderen Clustern bestehen [ELLS11].

2.4.2 Überwachtes Lernen

Falls für jeden Datenfall zusätzlich ein Zielwert y_i – auch als *Label* bezeichnet – wie folgt bekannt ist:

$$(x_{1,i}, ..., x_{m,i}, y_i) \quad \forall 1 \leq i \leq n$$

ist vom überwachten Lernen die Rede. Der mit Labels versehene Datensatz wird verwendet, um ein Modell zu erlernen, das zu einem Datenvektor $\vec{x_i}$ das Label $f(\vec{x_i}) = \hat{y}_i$ vorhersagt. Die Güte eines überwachten Lernalgorithmus wird dahingehend optimiert, dass die Vorhersage $\hat{y}_i$ möglichst dem realen Wert y_i entspricht. Um dies formalisieren zu können, ist eine Unterscheidung anhand des Datentyps des Labels nötig.

- Falls $y_i \in \mathbb{N}$, so spricht man von einer **Klassifizierung**. Bei dieser Variante gilt es, die Anzahl an korrekten Zuordnungen von Datenfällen zur jeweiligen Klasse zu maximieren.
- Ist hingegen $y_i \in \mathbb{R}$, so handelt es sich um eine **Regression**. Hier kann beispielsweise die absolute Abweichung über alle Datenfälle durch die Summe $\sum_{i=1}^{n} |y_i - \hat{y}_i|$ beschrieben werden, die es zu minimieren gilt.

2.4.3 Multi-Label und Multi-Output

Ein Spezialfall des überwachten Lernens liegt vor, wenn die Zielwerte der Datenfälle mehrere Dimensionen annehmen können.

$$(x_{1,i}, ..., x_{m,i}, y_{1,i}, ..., y_{l,i}) \quad \forall 1 \leq i \leq n$$

Dies wird auch als **Multi-Label**-Datensatz mit l Labels bezeichnet.

Im Falle der *Klassifizierung* ist l gleich der Klassenanzahl. Die Variable $y_{k,i} \in \{0, 1\}$ drückt dabei die Zugehörigkeit des Datenfalls zur Klasse k aus. Somit kann jeder Fall zu beliebig vielen Klassen zugeordnet werden. Bei einer *Regression* kann weiterhin $y_{k,i} \in \mathbb{R}$ gelten. Die Labels beschreiben dann jeweils unterschiedliche Eigenschaften des Datenfalls.

Ein **Multi-Output**-Lernalgorithmus ist in der Lage, eine Vorhersage in der Form $f(\vec{x_i}) = (\hat{y}_{1,i}, ..., \hat{y}_{l,i})$ zu liefern. Unter der Voraussetzung, dass ein Multi-Label-Datensatz vorliegt, können also alle Labels mit einem Modell vorhergesagt werden.

3 Feature Engineering

Im Abschnitt 2.4.1 wurde bereits erläutert, dass Datenfälle für die Anwendung im maschinellen Lernen als Featurevektoren dargestellt werden. Die Ursprungsdaten, auf die im Unterkapitel 5.1 noch näher eingegangen wird, liegen in natürlicher Sprache in Textform vor. Die manuelle Überführung der Ursprungsdaten in eine Vektordarstellung wird als **Feature Engineering** bezeichnet. Idealerweise sollten die Features einen semantischen Bezug zum Thema aufweisen und dadurch die Aussicht auf ein akkurates ML-Modell erhöhen.

3.1 Textstatistiken

Eine triviale Möglichkeit, Text in einer numerischen Form abzubilden, ist es Statistiken aus den Dokumenten zu erheben. Die nachstehende Auflistung nennt die ermittelten Werte.

- Anzahl Sätze n_{sent}
- Anzahl Wörter n_{word}
- Anzahl eindeutiger Wörter n_{uniq}
- Anzahl Zeichen n_{char}
- Anzahl Wörter mit mehr als zwei Silben n_{cplx}
- Anzahl einsilbiger Wörter n_{sing}
- Anzahl Wörter mit mehr als sechs Buchstaben n_{long}

Darüber hinaus werden Metriken berechnet, welche die Lesbarkeit eines Textes quantifizieren sollen. Die Motivation dieses Features zu wählen, besteht darin, dass die Art wie komplex ein Text geschrieben ist, Einfluss auf die Emotionen des Lesers hat. Für englischsprachigen Text wird der **Gunning-Fog-Index** [Gun52] eingesetzt, der versucht, die Lesbarkeit auf das Intervall $[6, 17]$ entsprechend der US-Schuljahre abzubilden. Dieser ist wie folgt definiert:

$$\mathrm{GFI} = 0.4 \cdot \left(\frac{n_{word}}{n_{sent}} + 100 \frac{n_{cplx}}{n_{word}} \right)$$

Die **Wiener Sachtextformel** [BV84] verfolgt den selben Ansatz für deutschsprachige Texte und bildet die Lesbarkeit auf das Intervall $[4, 15]$ ab. Es existieren vier verschiedene Ausprägungen dieser Formel, die sich lediglich in Details unterscheiden, weshalb im Folgenden nur die erste Formel betrachtet wird.

$$\mathrm{WSTF}_1 = 0.1935 \cdot 100 \frac{n_{cplx}}{n_{word}} + 0.1672 \cdot \frac{n_{word}}{n_{sent}} + 0.1297 \cdot 100 \frac{n_{long}}{n_{word}} - 0.0327 \cdot 100 \frac{n_{sing}}{n_{word}} - 0,875$$

3.2 Part-of-Speech Verteilung

Wie bereits in Abschnitt 2.3.3 erläutert ist es das Ziel des POS-Taggings, die Wortart eines Wortes zu bestimmen. Zur Nutzung als Feature wird an dieser Stelle das *Universal POS Tagset* [PDM12] verwendet, da es eine gröbere Einteilung als das *Penn Treebank Tagset* vornimmt und dadurch besser auf verschiedene Sprachen anwendbar ist. Dabei werden 12 Tags für Verben, Nomen, Pronomen, Adjektive, Adverben, Adpositionen, Konjunktionen, Bestimmungswörter, Ziffern, Partikel/Funktionswörter und restliche Wörter definiert. Der Anteil eines POS-Tags t ist über die Gesamtanzahl der diesem Tag zugeordneten Wörter n_t definiert.

$$f_{ratio}(t) = \frac{n_t}{n_{word}}$$

Zur Ermittlung der POS-Tags ist es als Vorverarbeitung nötig, den Text zu Tokenisieren. Beide Aufgaben werden mit dem von Honnibal und Johnson entwickelten Modell [HJ15] bewerkstelligt, welches eine Genauigkeit von 97.2% aufweist.

3.3 Bag-of-Words

Der *Bag-of-Words*-Ansatz (**BOW**) transformiert ein Dokument auf eine Darstellung, die die Anzahl der Vorkommen eines jeden Wortes erfasst. Die Reihenfolge der Wörter geht dabei verloren, wobei Manning et al. [MRS08] die Behauptung aufstellt, dass die beiden Sätze mit identischem Wortschatz „Paul mag Julia" und „Julia mag Paul" einen ähnlichen Inhalt wiedergeben. Zur Überführung in einen Featurevektor wird ein Wörterbuch erstellt, dass alle Wörter kennt, die in der Gesamtheit der Dokumente, dem sogenannten Dokumentenkorpus, vorkommen. Auf Basis eines Wörterbuchs mit m Einträgen und fixierter Wortreihenfolge kann der Vektor für alle n Dokumente bestimmt werden, wobei $tf_{t,d}$ die Anzahl der Vorkommen des Wortes t im Dokument d beschreibt, was als **Termfrequenz** bezeichnet wird.

$$(tf_{1,d}, ..., tf_{t,d}, ..., tf_{m,d}) \quad \forall 1 \leq d \leq n \tag{1}$$

Dabei kann beobachtet werden, dass unter den häufigsten vorkommenden Wörtern sehr viele Füllwörter sind. Solche auch als **Stoppwörter** bezeichneten Terme sind etwa Artikel (*der, die, das*) oder Konjunktionen (*und, oder*). Da Stoppwörter in fast jedem Dokument auftreten eignen sie sich nicht zur Differenzierung der Dokumente. Darüber hinaus besitzen sie nur einen geringen semantischen Wert, weshalb es möglich ist sie ohne nennenswerten Verlust an Informationen aus der BOW auszuschließen. Damit kann die Größe des Alphabets, welches es zu verwalten gilt, reduziert werden. Um dies zu realisieren, kann manuell eine Liste aller Stoppwörter der jeweiligen Sprache gepflegt und zur Filterung eingesetzt werden.

Das BOW-Verfahren kann auch mit der in Abschnitt 2.3.4 dargestellten *Wortstammbildung* kombiniert werden. Dadurch können Wörter mit gleichem Wortstamm auf den selben Eintrag im Wörterbuch abgebildet werden. Das erhöht die Wahrscheinlichkeit einer

akkuraten Wiedergabe des Dokumenteninhalts im Featurevektor. Zur Bildung des Wortstammes wird die von Porter mitentwickelte Plattform *Snowball* [Por01] verwendet, da dort Algorithmen für verschiedene Sprachen (unter anderem Deutsch und Englisch) implementiert sind. Dabei handelt es sich um etablierte Verfahren; sie werden auch von der verwendeten Software-Bibliothek *NLTK*[6] als präferierte Implementierung propagiert. Alternativ kann die bereits in Abschnitt 2.3.5 beschriebene *Lemmatisierung* angewendet werden.

Anstatt lediglich einzelne Wörter zu betrachten, ist es auch möglich die in Abschnitt 2.3.2 ausgeführten *N-Gramme* als Tokens in der BOW einzusetzen. Die Verwendung von Bigrammen bieten der Vorteil, dass Kombinationen aus zwei Wörtern verarbeitet werden. Dadurch kann möglicherweise mehr Semantik repräsentiert werden als durch einzelne Wörter und somit können Bigramme für die Emotionsanalyse eine höhere Relevanz haben.

Der bisherige BOW-Ansatz hat allerdings den Nachteil, dass alle Wörter gleichwertig im Wörterbuch vertreten sind. Im Umfeld des *Information Retrieval* ist es jedoch wünschenswert, dass ein im Dokumentenkorpus relativ seltenes Wort stärker gewichtet wird als ein geläufiges Wort, da somit spezifische Dokumente besser gefunden werden können. Das **TF-IDF**-Verfahren realisiert diese Idee durch Multiplikation der Termfrequenz (TF) mit der inversen Dokumentfrequenz (IDF). Die Termfrequenz $tf_{t,d}$ ist aus (1) bekannt und die inverse Dokumentfrequenz skaliert auf einer logarithmischen Skala, in wie vielen Dokumenten der Term t auftritt (df_t).

$$idf_t = log\frac{n}{df_t} \tag{2}$$

Entsprechend ist das TF-IDF-Schema in (3) definiert, in dem die Termfrequenzen mit (2) gewichtet wird.

$$tf\text{-}idf_{t,d} = tf_{t,d} \cdot idf_t \tag{3}$$

Somit bildet sich der angepasste Featurevektor für (1) durch Einführung von TF-IDF.

$$(tf\text{-}idf_{1,d}, ..., tf\text{-}idf_{t,d}, ..., tf\text{-}idf_{m,d}) \quad \forall 1 \leq d \leq n \tag{4}$$

3.4 Word2Vec

Unter dem Begriff **Word2Vec** werden Algorithmen beschrieben, die Wörter in eine Vektordarstellung überführen und erstmals von Mikolov et al. [MSC+13] beschrieben wurden. Ein Vertreter dieser Algorithmen ist das unüberwachte Lernverfahren *Global Vectors for Word Representation* (**GloVe**) [PSM14]. GloVe stellt bereits trainierte Modelle zur Verfügung, die sich sofort einsetzen lassen und einen Vektorraum mit 300 Dimensionen aufspannen. Diese Modelle wurden für mehrere Sprachen auf Texten von Webseiten trainiert, die der Dienst *Common Crawl*[7] bereitstellt.

[6] http://www.nltk.org
[7] http://commoncrawl.org

Das Ziel der Abbildung von Wörtern in diesen Vektorraum besteht darin, dass der linguistische Kontext eines Wortes erfasst wird und die euklidische Distanz des Vektors zu semantisch ähnlichen Wörtern gering ist. Beispielsweise liefert eine Suche nach den ähnlichsten Wörtern zum englischen Begriff „frog" die Ergebnisse „frogs, toad, litoria, leptodactylidae, rana", wobei die letzten drei Begriffe Froscharten repräsentieren.

3.5 Emotionslexika

Emotionslexika assoziieren eine gewählte Menge an Wörtern mit Emotionen. Eine Ausprägung dessen stellt **EmoLex** [MT13] dar, welches 14 182 englische Wörter umfasst. Jedes Wort wird dabei einer oder mehreren der in Abschnitt 2.2 erwähnten acht Emotionsklassen von *Plutchiks Rad der Emotionen* binär zugeordnet. Die Zuordnungen wurden durch *Crowdsourcing*, also durch Delegierung der Arbeit Gruppen bezahlter Arbeitskräfte, ermittelt. Es wurden Übersetzungen unter Zuhilfenahme von *Google Translate* in über 20 Sprachen erstellt, wobei die deutsche Version des Lexikons 11 812 Wörter enthält.

Eine weiterer Vertreter von Emotionslexika ist **DepecheMood** [SG14], welches mit 37 771 englischen Wörtern noch umfangreicher ist. Die Grundlage bot hier eine Internetnachrichtenseite, auf der Benutzer nach dem Lesen eines Artikels ihre Emotion mitteilen konnten. Dies ähnelt den in Absatz 2.1 beschriebenen Benutzerreaktionen auf Facebook. Allerdings entsprechen die Emotionsklassen (*Besorgt, Amüsiert, Wütend, Gelangweilt, Gleichgültig, Fröhlich, Inspiriert* und *Traurig*) keinem der verbreiteten Modelle; sie wurden durch die externe Datenquelle vorgegeben. Die Zuordnungen wurden daraufhin aus den mit Emotionen bewerteten Artikeln berechnet, in dem die Häufigkeiten der Wörter mit den Emotionsausprägungen am Artikel assoziiert wurden. Dabei wurde pro Wort und Emotionsklasse ein Gewicht im Intervall $[0, 1]$ vermerkt. Übersetzungen des Lexikons in andere Sprachen, neben Englisch, existieren nicht.

3.6 Dimensionsreduktion

Als Ausgabe liefert das BOW-Verfahren einen Vektor pro Datenfall, wobei die Länge dieser Vektoren der Größe des verwendeten Wörterbuchs entspricht. Im Extremfall würde der Dokumentkorpus jedes Wort der jeweiligen Sprache mindestens einmal verwenden. Für die englische Sprache listet das *Oxford English Dictionary* 171 476 Wörter[8], so dass auch die Dimensionalität des BOW-Wörterbuchs auf diesen Umfang anwachsen würde.

In Kapitel 4 werden Regressionsalgorithmen vorgestellt, die sich in der Funktionsweise und dadurch auch im Laufzeitverhalten unterscheiden. Dennoch kann allgemein angenommen werden, dass die Rechenlaufzeit und der Speicherverbrauch einerseits mit der Anzahl sowie mit der Dimensionalität der Datenfälle skaliert. Aus diesem Grund ist es nötig, eine **Dimensionsreduktion** vorzunehmen, um ML-Modelle in realistischer Zeit trainieren zu können.

Die zuvor beschriebenen Vektoren lassen sich in ihrer Gesamtheit als Matrix C beschreiben. Manning et al. dokumentieren in [MRS08] das *Latent Semantic Indexing* (**LSI**) als

[8] https://en.oxforddictionaries.com/explore/how-many-words-are-there-in-the-english-language (Abgerufen am 12.04.2017)

Möglichkeit, die Matrix C durch C_k zu ersetzen. Dies wird durch eine *Singulärwertzerlegung* erreicht, die Zusammenhänge zwischen Wörtern in den Dokumenten untersucht und anschließend eng verwandte Ausdrücke als sogenannte „Konzepte" zusammenfasst. Die Spaltenanzahl k der angenäherten Matrix kann beliebig gewählt werden, wobei die Autoren mit einer Größe im unteren dreistelligen Bereich die besten Ergebnisse erzielen konnten.

4 Regressionsalgorithmen

Im Abschnitt 2.4.2 wurden bereits die Grundlagen des überwachten maschinellen Lernens erläutert. Nun werden verschiedene Verfahren vorgestellt, die jeweils dem selben Muster unterliegen: In einer Trainingsphase wird durch Eingabe der Trainingsdaten ein Modell aufgebaut. Dieses Modell kann dazu verwendet werden, Vorhersagen für ungesehene - also nicht im Training verwendete - Daten zu liefern. Der Fokus liegt dabei auf Verfahren, die für eine Regression geeignet sind und dabei mehrere Label gleichzeitig vorhersagen können (Multi-Output).

4.1 k-Nearest-Neighbour

Der Algorithmus *k-Nearest-Neighbour* (**KNN**) ist insbesondere durch Anwendungen in der Klassifizierung bekannt. In der folgenden Abbildung 3 sind zwei Kategorien durch Quadrate und Dreiecke dargestellt. Für einen unklassifizierten Datenfall (hier als Kreis repräsentiert) wird die Klassenzugehörigkeit ermittelt, in dem die anhand der euklidischen Distanz ermittelten nächsten k Nachbarn betrachtet werden. Im Beispiel wurde für den Parameter k der Wert 3 gewählt und somit fallen alle Nachbarn innerhalb des durchgezogenen Kreises in die Auswahl, wodurch sich die Vorhersage zugunsten der Dreiecke ergibt. Würde hingegen $k = 5$ gewählt, so würde die Kategorisierung zugunsten der Quadrate ausfallen. Ein kleines k ist somit besonders abhängig von den Daten in der unmittelbaren Entfernung, während ein größerer Parameter das Verfahren unempfindlicher gegenüber Ausreißern macht.

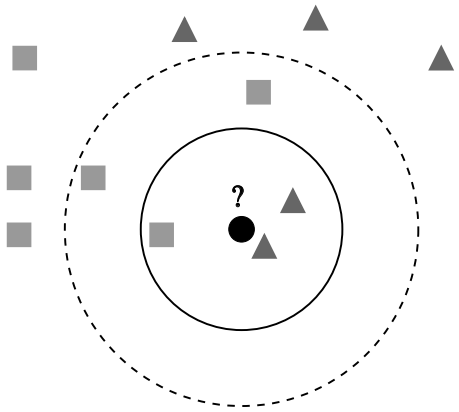

Abbildung 3: Visualisierung KNN-Klassifizierung mit zweidimensionalen Features
CC 3.0 BY-SA von Antti Ajanki[9]

Für den Fall, dass keine diskreten Kategorien vorliegen, sondern die Labels einen kontinuierlichen Wertebereich annehmen, kann der Algorithmus auch zur Regression eingesetzt werden [Alt92]. Zur Realisierung dessen werden weiterhin die nächsten k Nachbarn betrachtet, aber als Zielwert wird das arithmetische Mittel der Nachbarwerte berech-

[9] https://commons.wikimedia.org/wiki/File:KnnClassification.svg

net. Beispielsweise wird für die im folgenden Kapitel 5 durchgeführten Untersuchungen $k = 5$ gewählt, da dies ein in der Literatur häufig eingesetzter Wert ist.

KNN wird auch zu den sogenannten *lazy learning*-Verfahren gezählt, da in der Trainingsphase die Daten lediglich im Modell gespeichert werden müssen. Erst bei der Vorhersage von ungesehenen Daten sind Berechnungen notwendig, da die nächsten k Nachbarn gesucht werden müssen.

4.2 Lineare Regression

In der Statistik wird häufig die **Einfache Lineare Regression** angewendet, um einen Zielwert in Abhängigkeit von einer oder mehreren Variablen vorherzusagen [BEPW08]. Zu diesem Zweck, kann wie in Abbildung 4 dargestellt, eine lineare Funktion definiert werden, dessen Koeffizienten daraufhin aus den Daten abgeleitet werden.

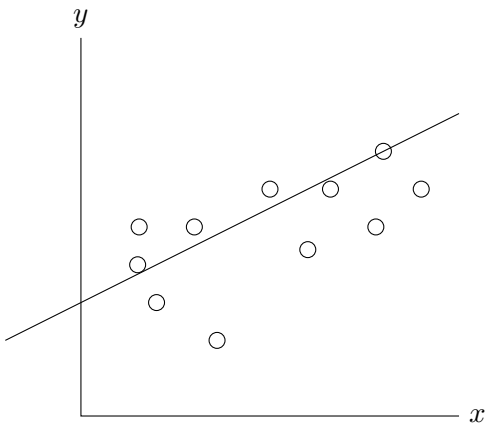

Abbildung 4: Einfache Lineare Regression

Damit die Funktion eine möglichst gute Vorhersage liefern kann, werden die Koeffizienten nach der *Methode der kleinsten Quadrate* ermittelt, die folgendes Problem löst:

$$\min_w ||Xw - y||_2^2$$

Hierbei bezeichnet X die bekannten Datenfälle und y deren Labels.

Allerdings besteht die Möglichkeit, dass keine eindeutige Lösung des Problems existiert. In diesem Fall neigt die Methode der kleinsten Quadrate dazu, eine überspezialiserte Lösung zu finden. Daraufhin hat Tikhonov eine „Bestrafung" der Koeffizientengröße eingeführt [Tik43], die mit dem Parameter α gesteuert werden kann, wobei $\alpha > 0$.

$$\min_w ||Xw - y||_2^2 + \alpha||w||_2^2$$

Dieses Verfahren ist als *Tikhonov Regulierung* bekannt und wird in der Statistik auch als **Ridge Regression** beschrieben.

4.3 Support Vector Regression

Mit den *Support Vector Machines* existiert ein Ansatz, der ebenfalls oft im Kontext der Klassifizierung eingesetzt wird und deswegen häufig auch als *Support Vector Classification* bezeichnet wird. Die Abwandlung *Support Vector Regression* (**SVR**) greift die zugrundeliegende Idee der Stützvektoren auf, weist aber eine andersartige Verwendung auf, weshalb im Folgenden direkt auf die Regressionsvariante eingegangen wird.

In [Vap95] beschreibt Vapnik, die ϵ-SVR als Versuch, eine Funktion $f(x)$ zu formulieren, die in der Lage ist alle Trainingsdaten mit einem Fehler geringer als ϵ abzubilden. Die Funktion kann dabei eine lineare Form annehmen.

$$f(x) = wx + b \quad w \in X, b \in \mathbb{R}$$

Hierbei bezeichnet X die Menge der Trainingsdaten. Die Herausforderung besteht nun darin, ein besonders kleines w zu finden, also beispielsweise die Norm $\frac{1}{2}||w||^2$ zu minimieren. Dabei sind die Einschränkungen zu betrachten, damit die festgelegten Fehlergrenzen nicht überschritten werden.

$$y_i - wx_i - b \leq \epsilon$$
$$wx_i + b - y_i \leq \epsilon$$

Abhängig von der Datenverteilung der Labels y_i und dem gewählten ϵ ist es mitunter nicht möglich eine Lösung zu finden, also alle Datenfälle in einen durch die Funktion und ϵ definierten Korridor zu legen. Aus diesem Grund werden die Variablen ξ_i, ξ_i^* eingeführt, um Punkte außerhalb der Fehlertoleranz miteinzubeziehen, was in Abbildung 5 dargestellt ist.

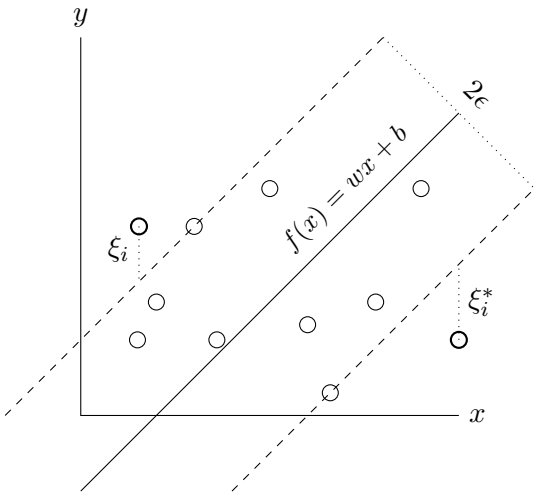

Abbildung 5: Lineare SVR mit Fehlertoleranz

Dadurch ergibt sich die angepasste Formulierung als Optimierungsaufgabe mit der Konstante $C > 0$, die beeinflusst, wie stark Abweichungen von der Fehlertoleranz erlaubt sein sollen.

$$\begin{aligned} &\text{minimiere} && \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ &\text{unter Einbehaltung von} && \begin{cases} y_i - wx_i - b \leq \epsilon + \xi_i \\ wx_i + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned}$$

Neben der linearen Form der Funktion $f(x)$ sind auch weitere Ausprägungen möglich. In der Literatur wird dies auch als *Kernel* bezeichnet. Im Kontext dieser Arbeit wird die SVR mit einer Linear-, Polynom-, Sigmoid- sowie einer radialen Basisfunktion (RBF) durchgeführt, um somit den Einfluss der unterschiedlichen Kernel vergleichen zu können.

Die SVR bietet nicht direkt die Option, eine Multi-Output-Vorhersage zu liefern. Allerdings ist es möglich, jedes Label separat als Single-Output-Regression zu definieren und jeweils ein eigenes Modell dafür zu trainieren. Dadurch ist es den unterschiedlichen Modellen aber nicht möglich, eventuelle Abhängigkeiten zwischen den Labels erkennen und nutzen zu können.

4.4 Entscheidungsbäume

Entscheidungsbäume oder auch **Decision Trees** sind Strukturen, die während der Trainingsphase einfache Regeln erlernen. Sie wurden anfangs von Breiman et al. [BFOS84] beschrieben. Zu gegebenen Eingabedaten können diese Regeln entscheiden, wie der Datenfall klassifiziert oder mit welchem Zielwert er bei einer Regression belegt werden soll. Die Regeln werden dabei so formuliert, dass sie binär entschieden werden können und sind üblicherweise in mehreren Ebenen verschachtelt.

In Abbildung 6 ist ein trainierter Entscheidungsbaum dargestellt. Die Features X sind eindimensional, während mittels $value$ zwei verschiedene Labels vorhergesagt werden. Der Schlüssel $samples$ gibt an, wie viele Datenfälle unter den jeweiligen Knoten fallen und mse ist die Abkürzung für *Mean Squared Error*. Dabei handelt es sich um ein Maß zur Quantifizierung der Abweichung zwischen der Vorhersage und den tatsächlichen Daten, welches in Abschnitt 5.2 in ähnlicher Form verwendet wird. Um eine Vorhersage treffen zu können, muss also ein Pfad von der Wurzel bis zu einem Blatt entsprechend der an den Knoten annotierten Bedingungen verfolgt werden. Die Label-Werte am Blatt entsprechen daraufhin der zurückgelieferten Schätzung.

Zur Konstruktion der Entscheidungsbäume wird der von Quinlan entwickelte *ID3*-Algorithmus [Qui86] verwendet. Dieser verfolgt einen Top-Down-Ansatz, bei dem beginnend an der Wurzel die Datenbasis rekursiv geteilt wird. Die Teilung wird dabei anhand des Features durchgeführt, welches die größte Reduktion der Standardabweichung bei den Label-Werten hervorruft. Dies wird solange wiederholt, bis eine vorher

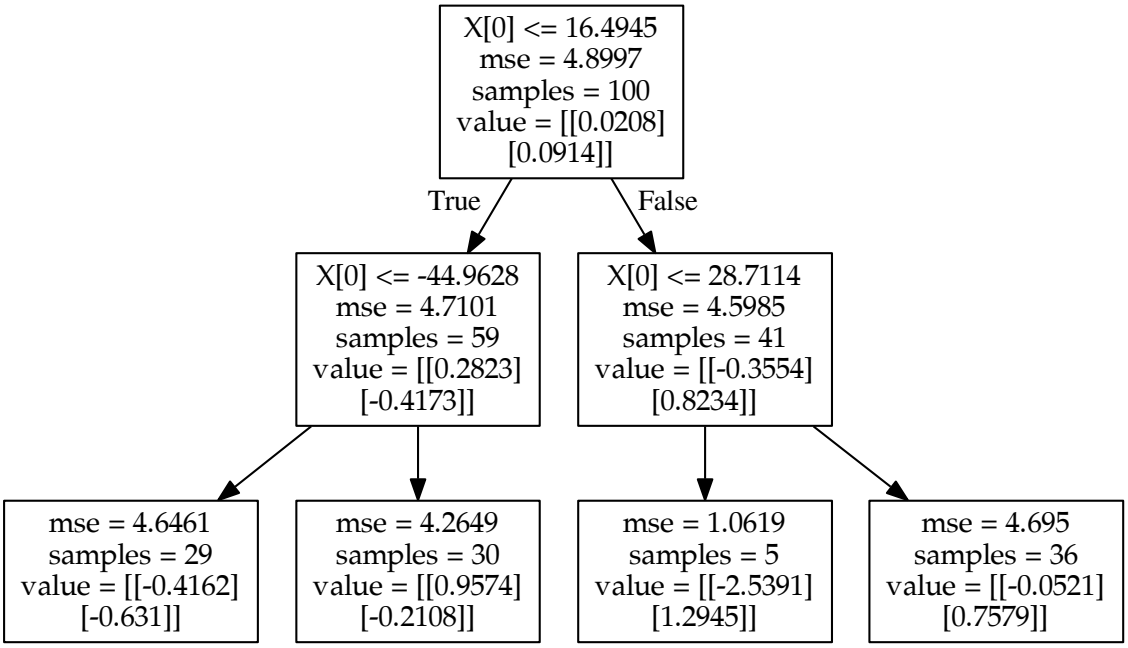

Abbildung 6: Darstellung eines Entscheidungsbaums zur Regression

festgelegte maximale Tiefe des Baumes erreicht wurde oder die Reduktion der Standardabweichung eine konfigurierte Schwelle nicht mehr überschreiten kann. Schlussendlich wird für das Attribut $value$ das arithmetische Mittel der Label-Werte aller Datenfälle, die unter den jeweiligen Knoten fallen, berechnet.

Ein Nachteil von Entscheidungsbäumen ist, dass sie zum *Overfitting* neigen. Das bedeutet, dass sich das Modell zu sehr auf die gewählte Trainingsmenge spezialisiert hat und bei ungesehenen Daten schlechter abschneidet. Um dem entgegenzuwirken, hat Breiman das Verfahren weiterentwickelt [Bre01] und dieses als **Random Forests** bezeichnet. Dabei handelt es sich um eine *Ensemble*-Methode, bei der aus den Trainingsdaten zufällige Stichproben extrahiert werden, mit denen jeweils separate Entscheidungsbäume gelernt werden. In der Vorhersagephase werden alle Bäume einbezogen und dann das Mittel aller Resultate zurückgeliefert.

Einen Schritt weiter gehen **Extremely Randomized Trees** (*ExtraTrees*) [GEW06], bei denen ebenfalls mehrere Bäume aus zufälligen Stichproben erzeugt werden. Darüber hinaus wird auch die Konstruktion der Bäume durch eine Zufallskomponente beeinflusst, in dem die Teilung nicht anhand der größten Reduktion der Standardabweichung entschieden wird, sondern an zufälligen Stellen durchgeführt wird.

Einen alternativen Ansatz, aus Entscheidungsbäumen ein besseres Lernverfahren zu konstruieren, verfolgte Friedman [Fri00] beim **Gradient Boosting**. Die Grundlage bilden „schwache Lerner", bei denen es sich ebenfalls um multiple Bäume handelt. Allerdings

wird dessen Vorhersagequalität bewusst schlecht gehalten, in dem die maximale Tiefe begrenzt wird. Dies liegt in der Idee begründet, dass die Entscheidungsbäume sich nicht auf die Fehlerminimierung der einfachen Fälle konzentrieren, sondern die komplexen Situationen in den Fokus nehmen sollen. Die einzelnen Bäume werden anhand eines additiven Modells hinzugefügt. Das bedeutet, dass in jedem Schritt jeweils ein neuer Baum ermittelt wird und die bestehenden unverändert bleiben. Durch Anwendung einer numerisch optimierbaren *Gradient Descent*-Prozedur werden die Parameter des neuen Baumes so bestimmt, dass der Vorhersagefehler des Gesamtmodelles sinkt. Wie bereits bei den vorherigen Ansätzen wird die Vorhersage dabei aus allen Bäumen berechnet. Das kann aber durch das Lernen mehrerer Modelle kompensiert werden.

5 Evaluation

In diesem Kapitel sollen nun die zuvor erläuterten Verfahren eingesetzt und bewertet werden. Dazu ist es zum einen notwendig die Quelle und Struktur des Datensatzes darzulegen, mit denen das maschinelle Lernen durchgeführt wird. Zum anderen muss die Art der Bewertung definiert werden, bevor schließlich die Ergebnisse diskutiert werden können.

5.1 Erhebung des Datensatzes

Für den Anwendungsfall des überwachten maschinellen Lernens ist die Verfügbarkeit eines annotierten Datensatzes unerlässlich. Nur so kann ein Modell trainiert werden. Die Annotation bezeichnet an dieser Stelle, wie in Kapitel 3 beschrieben, dass jeder Datenfall mit dem Label versehen ist, das das Modell im Nachhinein für weitere Daten vorhersagen soll. Als Datengrundlage für die Evaluation dienen Facebook-Beiträge von Nachrichtenseiten, sowie die vom Beitrag verlinkten Nachrichtenartikel. Die Benutzerreaktionen auf Facebook, die zu den Beiträgen abgegeben werden, spiegeln die Labels wieder. Dabei werden die temporären Reaktionen *Thankful* und *Pride* nicht betrachtet, da sie nur für einen kurzen Zeitraum durch Facebook freigeschaltet werden.

5.1.1 Struktur der Daten

Im ersten Schritt des Crawlings werden die Facebook-Präsenzen ausgewählter Nachrichtenseiten abgerufen. Die Redakteure erstellen für eine Teilmenge der auf der Nachrichtenseite veröffentlichten Artikel jeweils einen Beitrag innerhalb der Facebook-Präsenz. In diesem Beitrag wird zum ursprünglichen Artikel auf der Nachrichtenseite verlinkt und eventuell ein Anrisstext verfasst. Facebook-Benutzer haben daraufhin die Möglichkeit, Benutzerreaktionen zu den Beiträgen abzugeben.

Dieses Vorgehen hat mehrere Vorteile: Zum Einen sind die Beiträge auf den Facebook-Präsenzen der Nachrichtenseiten öffentlich und können somit ungehindert abgerufen werden. Dahingegen sind Beiträge individueller Benutzer standardmäßig in ihrer Sichtbarkeit eingeschränkt und können nur von befreundeten Benutzern gelesen werden. Weiterhin sind private Nachrichten oft von geringem Textumfang oder enthalten sogar nur ein Bild oder ein Video. Aus diesem Grund ist es sinnvoller nur die Beiträge von Nachrichtenseiten abzurufen.

Zusätzlich kann auf die Artikel der Nachrichtenseiten zugegriffen werden. Diese erstrecken sich in der Regel über mehrere Absätze. Der daraus resultierende größere Textkorpus kann sich positiv auf die Funktion einiger in Kapitel 3 beschriebenen Features auswirken. Beispielsweise müssen für die auf Emotionslexika basierenden Ansätze im Text gewisse Wörter enthalten sein, die auch im Emotionslexikon vermerkt sind. Bei kurzen Texten ist dies dementsprechend unwahrscheinlich.

Letztendlich werden Beiträge der Nachrichtenseiten aufgrund der öffentlichen Sichtbarkeit und des größeren Interesses häufiger gelesen. Dadurch drücken auch Benutzer ihre

Emotionen als Benutzerreaktion häufiger aus. Das Verhältnis der Reaktionen ist unter diesen Beiträgen somit repräsentativ.

Die Auswahl der Nachrichtenseiten fiel auf drei englischsprachige (*Fox News, New York Times, The Guardian*) und drei deutschsprachige Portale (*Bild, Spiegel Online, Süddeutsche Zeitung*), wobei jeweils ein Boulevard-Magazin enthalten ist. Als Kriterium diente die Reichweite der Präsenzen auf Facebook, also wie oft Benutzerreaktionen zu den Beiträgen hinterlassen werden. Um repräsentative Daten zu erhalten, sollte die Gesamtanzahl der Reaktionen pro Post dabei mindestens im zweistelligen Bereich liegen.

Der zweite Schritt besteht darin, iterativ die Beiträge der Facebook-Präsenz zu verarbeiten und die in Tabelle 1 unter der Quelle „Facebook" vermerkten Felder zu speichern. Dabei werden Beiträge im Zeitraum von März 2016 (kurz nach Einführung der Benutzerreaktionen-Funktion) bis Dezember 2016 betrachtet.

Feldname	Beschreibung	Quelle
Id	Eindeutige Nummer des Beitrags auf Facebook	Facebook
Message	Verfasster Text des Beitrags	
Link	Hinterlegter Link zu Facebook-externen Inhalt	
Datum	Zeitstempel der Veröffentlichung	
Like	Anzahl Reaktionen vom Typ *Like*	
Love	Anzahl Reaktionen vom Typ *Love*	
Haha	Anzahl Reaktionen vom Typ *Haha*	
Wow	Anzahl Reaktionen vom Typ *Wow*	
Sad	Anzahl Reaktionen vom Typ *Sad*	
Angry	Anzahl Reaktionen vom Typ *Angry*	
Titel (Title)	Überschrift des Nachrichtenartikels	Nachrichtenseite
Artikeltext (Text)	Inhalt des Nachrichtenartikels	

Tabelle 1: Struktur der gesammelten Daten

Abschließend wird der von jedem Beitrag hinterlegte Link aufgerufen um den Artikeltext und Titel von der Nachrichtenseite zu extrahieren. Tabelle 2 ist zu entnehmen, wie viele Beiträge pro Nachrichtenseite gesammelt wurden. Außerdem ist der mittlere Artikelumfang sowie die mittlere Anzahl Benutzerreaktionen auf Facebook gelistet. Die Spalte „Anzahl Beiträge mit zu wenigen Reaktionen" bezeichnet, wie viele Beiträge in der Summe weniger als 50 Reaktionen haben. Diese individuellen Beiträge werden bei den folgenden Aufgaben ausgeklammert, weil, wie bereits zuvor erwähnt, die Repräsentativität nicht gegeben ist. Dabei wird der empirisch ermittelte Wert 50 gewählt, da dieser zur Verbesserung der Ergebnisse führt.

Die Reichweite der Beiträge und somit die Summe aller Reaktionen ist wesentlich von der verfassenden Nachrichtenseite abhängig. So hat die Facebook-Präsenz des Anbieters *Fox News* zum Zeitpunkt März 2017 über 15 Mio. Abonnenten, während die Präsenz der *Bild* nur 2 Mio. interessierte Benutzer verzeichnet. Allerdings gibt es auch zwischen den einzelnen Beiträgen eines Anbieters Schwankungen der Reaktionszahlen. Die Benutzer haben die Möglichkeit einen Beitrag mit ihren Freunden zu teilen, die dann wiederum

Nachrichtenseite	Bezeichnung auf Facebook	Anzahl Beiträge	Mittlere Wortlänge der Artikel	Mittlere Summe der Reaktionen	Anzahl Beiträge mit zu wenigen Reaktionen
Fox News	foxnews	4181	439.9	14305.7	1
New York Times	nytimes	3030	1312.8	3450.8	189
The Guardian	theguardian	7455	1162.7	1611.1	35
Bild	bild	8452	341.0	1570.1	65
Spiegel Online	spiegelonline	7174	566.4	593.3	854
Süddeutsche Zeitung	ihre.sz	4396	224.3	434.4	636

Tabelle 2: Umfang der gesammelten Daten

auch ihre Aufmerksamkeit auf den Beitrag richten. Wenn eine Nachricht besonders interessant ist, so wird der zugehörige Beitrag von vielen Benutzern geteilt, was die Reichweite vervielfachen kann. Dementsprechend ist die Summe der Reaktionen bei einigen Beiträgen deutlich größer als bei anderen. Dies steht allerdings nicht zwangsläufig im Zusammenhang mit dem Textinhalt. So hängt es von weiteren Parametern, wie etwa dem Zeitpunkt ab, ob der Beitrag in einer viralen Weise häufig geteilt wird oder ob der Beitrag unbeachtet bleibt.

Damit diese Dynamik die Untersuchungen nicht beeinflusst, werden die absoluten Reaktionszahlen normalisiert. Um dies zu erreichen wird die relative Verteilung der jeweiligen Reaktionstypen pro Beitrag berechnet.

5.1.2 Analyse der Daten

Zur besseren Einordnung der im weiteren Verlauf vorgestellten Vorhersageergebnisse wird die zuvor erwähnte Verteilung der Reaktionstypen, welche die Labels repräsentieren, deskriptiv analysiert. In Abbildung 7 ist deutlich zu erkennen, dass die Reaktion *Like* unabhängig von der Nachrichtenseite im Durchschnitt aller Beiträge den größten Anteil einnimmt. Dies ist darauf zurückzuführen, dass diese spezifische Reaktion bereits deutlich länger als alle anderen Typen existiert und somit aufgrund der Benutzergewohnheit vorzugsweise angewendet wird. Außerdem ist der *Like* in der Benutzerschnittstelle mit nur einem Klick zu tätigen, während für die restlichen Reaktionstypen ein weiterer Interaktionsschritt nötig ist.

Weiterhin ist zu sehen, dass alle anderen Reaktionen auf einem ähnlichen Niveau liegen, lediglich die Kategorie *Wow* wird über alle Nachrichtenportale hinweg besonders selten genutzt. Unterschiede ergeben sich hingegen beim Label *Haha*, welches bei *Spiegel Online* mehr als doppelt so häufig zum Einsatz kommt wie bei der *New York Times*.

Abschließend sollen noch die Korrelationen zwischen allen Reaktionen analysiert werden. Falls ein starker linearer Zusammenhang zwischen den Kategorien existieren würde, so wäre eine individuelle Betrachtung der Reaktionen nicht notwendig, da von einem Wert auf alle weiteren korrelierenden Werte geschlossen werden könnte. In Tabelle 3 sind diese nach dem Korrelationskoeffizienten von Pearson berechnet worden. Zwi-

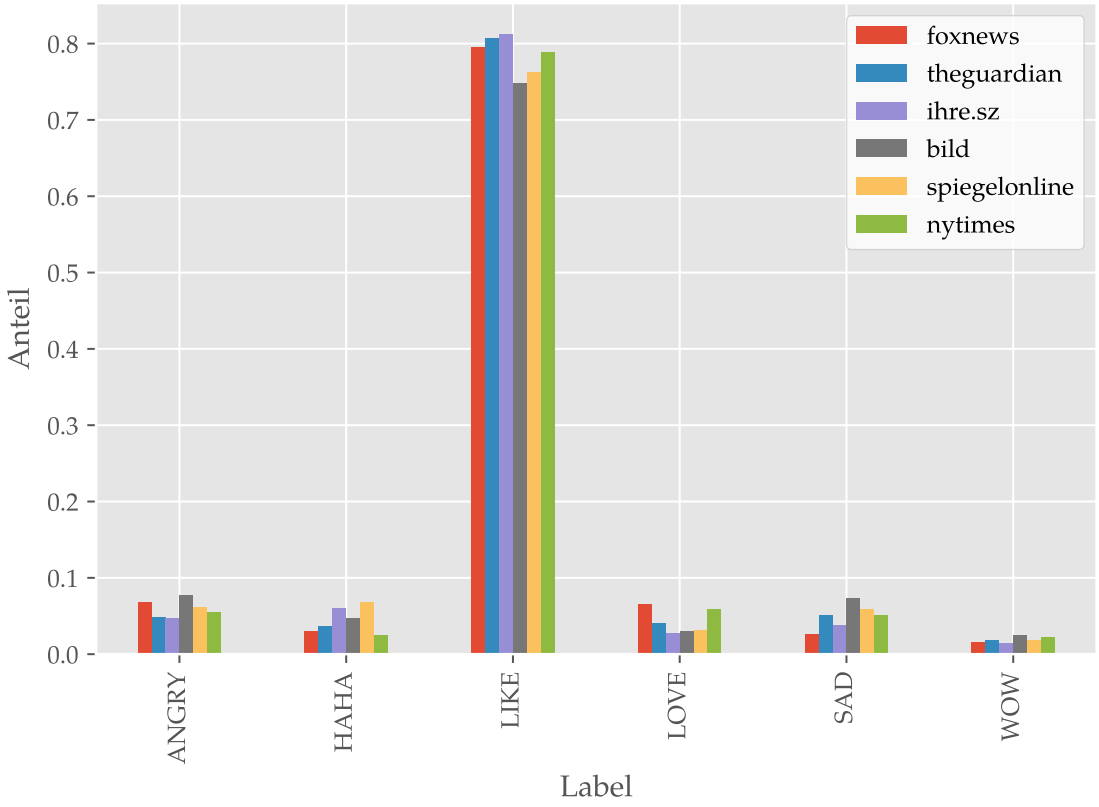

Abbildung 7: Mittlere Verteilung der Labels je Nachrichtenseite

schen den Reaktionen *Like* und *Love* ist ein relativ hoher Zusammenhang von 0.835 zu erkennen, während alle anderen Werte nahe 0 sind und somit die Kategorien paarweise unabhängig agieren. Dadurch ist es notwendig, alle Benutzerreaktionen als unabhängig zu betrachten und Multi-Label/Output-Regressionsalgorithmen zu verwenden.

5.2 Evaluationsmaß

Zur Quantifizierung der Vorhersage-Ergebnisse bietet es sich an, wie bereits in Abschnitt 2.4.2 angedeutet, die absolute Differenz zwischen der Vorhersage $\hat{y}_i$ und dem tatsächlichen Wert y_i zu erheben. Das über die Anzahl der Datenfälle n normierte Differenzmaß wird als *Mean Absolute Error* (**MAE**) bezeichnet.

$$\mathrm{MAE}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Da aufgrund des vorliegenden Datensatzes eine Multi-Output-Vorhersage vorgegeben ist, muss der MAE für jedes Label, also die sechs Typen der Benutzerreaktionen, separat berechnet werden. Anschließend wird der Mittelwert der sechs Werte gebildet um mit nur einem Wert die Vergleiche im folgenden Abschnitt zu vereinfachen.

	ANGRY	HAHA	LIKE	LOVE	SAD	WOW
ANGRY	1.000					
HAHA	0.193	1.000				
LIKE	0.086	0.238	1.000			
LOVE	0.073	0.163	0.835	1.000		
SAD	0.147	0.013	0.173	0.120	1.000	
WOW	0.346	0.213	0.318	0.266	0.147	1.000

Tabelle 3: Korrelationsmatrix der Labels

Einen ähnlichen Ansatz verfolgt das Maß *Root Mean Squared Error* (**RMSE**). Aufgrund der Quadrierung gehen große Fehler stärker in die Gewichtung ein als kleinere Abweichungen.

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Für beide Maße MAE und RMSE gilt, dass die Ergebnisse auf der selben Skala der gemessenen Variablen verortet sind. Dadurch ist die Qualität der Vorhersage einfach zu interpretieren. Dies ist beispielsweise bei dem Maß MSE – also ohne abschließendes Ziehen der Wurzel – nicht der Fall, da die Werte durch die Quadrierung verzerrt sind.

Der Analyse von Hyndman und Koehler zufolge [HK06] wird der RMSE in der Forschung häufig eingesetzt, da er eine große theoretische Relevanz in der statistischen Modellierung inne hat. Aus diesem Grund wird dieses Maß auch für die folgenden Untersuchungen verwendet.

5.3 Aufteilung des Datensatzes

Der übliche Ablauf beim überwachten maschinellen Lernen besteht aus mehreren Phasen. Im ersten Schritt wird ein Teil der Datenfälle als Testmenge separiert. Diese Menge wird letztendlich genutzt, um das Vorhersagemodell zu evaluieren, während der restliche Teil der Daten genutzt werden kann, um das Modell zu konstruieren. Die Motivation dahinter ist, dass die im Test verwendeten Daten vom Modell noch nicht gesehen wurden und somit ein aussagekräftiges Ergebnis erzielt werden kann. Daraus ergibt sich die Fragestellung, in welchem Verhältnis die Zerlegung erfolgen soll. Witten et al. postulieren in [WFH11], dass die Trainingsdaten die größere Menge stellen sollten, um ein besseres Training zu ermöglichen. Im Rahmen dieser Arbeit werden für die Testmenge 20% der Daten zufällig ausgewählt.

In Abbildung 8 ist dieses Vorgehen schematisch dargestellt. Nach der ersten Aufteilung kann die Trainingsmenge zur Modellerstellung genutzt werden. Während dieser Phase werden unterschiedliche Ansätze untersucht um eine gute Vorhersage ermöglichen zu

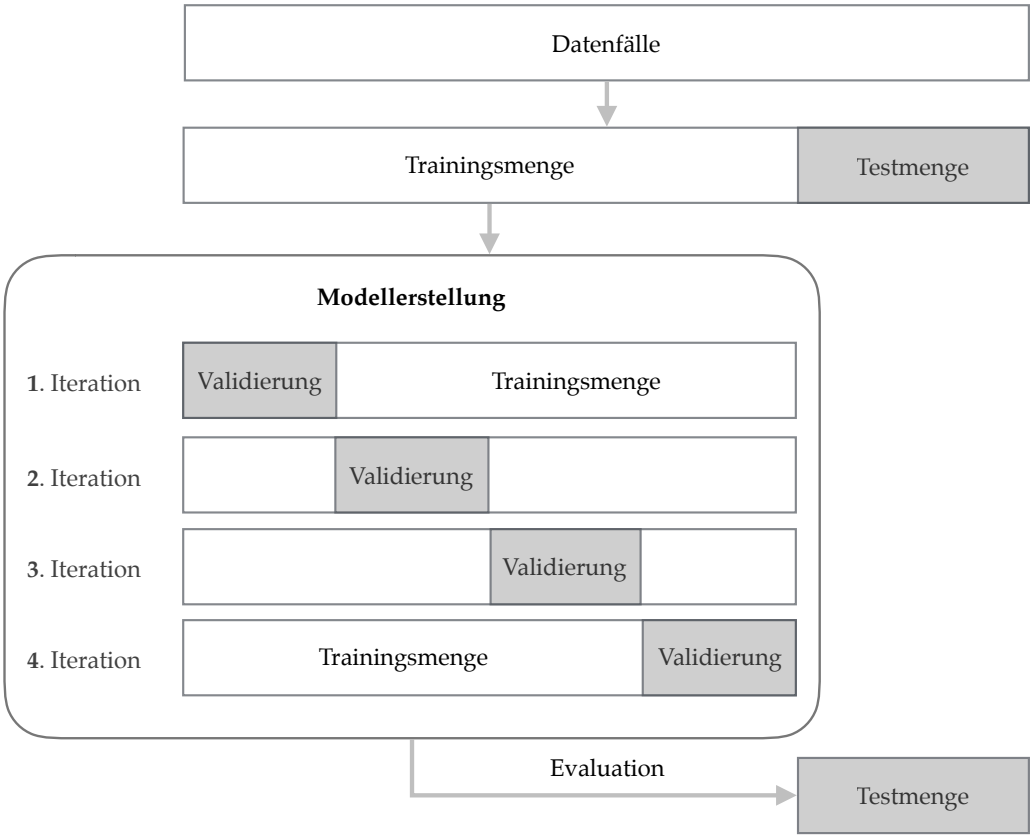

Abbildung 8: Ablauf der Modellerstellung und -validierung

können. Aus diesem Grund ist es auch an dieser Stelle nötig, die Datenmenge zu partitionieren, um das Modell mit Daten validieren zu können, die nicht bereits während des Trainings verwendet wurden.

Die Auswahl der Datenaufteilung hat einen erheblichen Einfluss auf die Qualität des erlernten Modells und dessen Vorhersage. So ist es möglich, zufällig eine Stichprobe zu wählen, die ein überdurchschnittlich gutes oder schlechtes Ergebnis produziert, was zu falschen Schlüssen verleiten kann. Aus diesem Grund wird das Verfahren der **Kreuzvalidierung** eingesetzt. Dabei wird der Datensatz in k gleich große Partitionen zerlegt, wobei eine Partition die Validierungsdaten bildet und der Rest als Trainingsdaten verwendet wird. Dieser Vorgang wird iterativ k-Mal durchgeführt, so dass jede Partition exakt einmal als Testdatenmenge fungiert. In der zuvor gezeigten Abbildung 8 ist dies exemplarisch für $k = 4$ dargestellt.

Üblicherweise wird entweder $k = 3$ gesetzt, was auch als Dreifachfaltung bezeichnet wird, oder die Zehnfachfaltung $k = 10$ eingesetzt. Letztere ist laut Witten et al. die in der Forschung bevorzugte Variante der Kreuzvalidierung, um die beste Fehlerabschätzung zu erhalten. Aus diesem Grund wird auch für die Validierung der Modelle dieser Arbeit die Zehnfachfaltung gewählt.

Schlussendlich stellt sich die Frage, wie die in Abschnitt 5.2 beschriebenen Maße angewendet werden. Bei einer Kreuzvalidierung mit k Faltungen werden auch k Modelle trainiert, die jeweils mit dem gegebenen Maß bewertet werden. Um ein Ergebnis zu erhalten, das unabhängig von der zufälligen Stichprobenwahl ist, kann das Mittel über alle Durchläufe gebildet werden.

5.4 Technische Umsetzung

Um das Training der Vorhersagemodelle realisieren zu können, wird auf die für die Programmiersprache Python entwickelte Bibliothek *scikit-learn* [PVG+11] zurückgegriffen. Diese Bibliothek bietet neben Implementierungen der Regressionsalgorithmen auch einfache Textverarbeitungswerkzeuge. Darüber hinaus wird die NLP-Bilbiothek *spaCy*[10] verwendet, um die Features POS-Verteilung und Word2Vec zu ermitteln.

Die Textstatistiken werden mit der Werkzeugsammlung *textacy*[11] bestimmt. Da diese deutsche Texte nicht unterstützte, wurde eine Funktion zur Berechnung der Wiener Sachtextformel implementiert. Daraufhin wurde die Funktion in die Werkzeugsammlung integriert, damit sie auch durch andere Nutzer genutzt werden kann.

5.5 Bewertung der Vorhersage

Im nächsten Schritt werden mit der zuvor dargelegten Systematik Experimente durchgeführt, um das Modell zu finden, das Vorhersagen mit einem möglichst geringen Fehler ausgeben kann. In den vorherigen Kapiteln wurden verschiedene Elemente vorgestellt, die in das Modell eingehen könnten. Die Herausforderung besteht darin, aus der Kombination folgender Möglichkeiten diejenige mit dem geringsten Vorhersagefehler auszuwählen.

- Nachrichtenseiten, in unterschiedlicher Sprache
- Teilbereiche der Daten (Titel, Artikeltext, Facebook-Nachricht)
- Regressionsalgorithmen mit jeweils verschiedenen Parametern
- Features, teilweise in unterschiedlichen Ausprägungen

Die Gesamtzahl aller möglichen Kombination ist leider zu hoch, um alle theoretisch konstruierbaren Modelle validieren zu können. Darüber hinaus erschwert die komplexe Problemstellung auch, die Ergebnisse im Rahmen dieser Arbeit übersichtlich visualisieren zu können. Aus diesem Grund werden einzelne Fragestellungen isoliert und nacheinander in den folgenden Abschnitten analysiert.

Das Hauptaugenmerk liegt dabei auf der Ermittlung der erfolgversprechendsten Features und der Frage, mit welchem Regressionsalgorithmus die besten Ergebnisse zu erzielen sind. Die Experimente werden jeweils einmal mit dem kompletten englischsprachigen und deutschsprachigen Datensatz durchgeführt.

[10] http://spacy.io
[11] http://textacy.readthedocs.io

Um bei den folgenden Untersuchungen beurteilen zu können, wie gut ein gemessener Wert tatsächlich ist, wird zum Vergleich die Baseline herangezogen. Dabei handelt es sich um ein statisches Modell, welches unabhängig vom Datenfall immer den Mittelwert aller gesehenen Daten vorhersagt. Ein konstruiertes Vorhersagemodell sollte demnach Ergebnisse mit einem geringeren Fehler als die Baseline liefern können, damit es einen Mehrwert bietet.

Der Datensatz wird zunächst aggregiert betrachtet, um mit der größtmöglichen Menge an Datenfällen die Modelle trainieren zu können. Dafür werden die drei deutschsprachigen und drei englischsprachigen Nachrichtenseiten zusammengefasst, um letztlich jeweils einen Datensatz für beide Sprachen bereitzustellen.

5.5.1 Konfiguration der Bag-of-Words

Um die für den Einsatzzweck beste Konfiguration zu ermitteln, wird im Vorfeld das *Bag-of-Words*-Feature näher untersucht. Im Abschnitt 3.3 wurden verschiedene Optionen der Realisierung vorgestellt. Nun sollen die Fragen beantwortet werden, ob die besten Ergebnisse mit Stammformbildung, Lemmatisierung oder gänzlich ohne Worttransformation erzielt werden, ob Unigramme oder Bigramme zum Einsatz kommen sollen, ob zusätzlich mit TF-IDF gewichtet werden soll und auf wie viele Konzepte (100, 250, 500, 1000) anschließend mittels LSI reduziert werden soll. Um die Anzahl der möglichen Kombinationen an dieser Stelle begrenzen zu können, werden die Experimente nur auf den Artikeltext angewendet, da dieser den größten Textumfang bietet. Als Verfahren kommt dabei zunächst die *Ridge Regression* zum Einsatz, da diese bei den späteren Experimenten, im Vergleich zu anderen Algorithmen, mit relativ geringen Fehler vorhersagte.

In den Abbildungen 9 und 10 sind die Ergebnisse aller Kombinationen dargestellt. Die *Baseline* liefert als statisches Modell einen Fehler von 0.12. Alle getesteten Ansätze liegen dabei im Intervall $[0.015, 0.125]$ (deutsch), bzw. $[0.095, 0.115]$ (englisch). Deutlich sichtbar ist dabei, dass alle Ansätze mit TF-IDF bessere Ergebnisse liefern und Stammformbildung sowie Lemmatisierung kaum zur Verbesserung beitragen. Darüber hinaus schneiden die Optionen mit Bigrammen überwiegend am schlechtesten ab, während die LSI-Konzepte mit Umfang 1000 und 500 aus Unigrammen den geringsten Vorhersagefehler erbringen können.

Im Folgenden wird diese Variante mit allen Regressionsalgorithmen getestet. Zusätzlich wird dabei die Wahl der LSI-Konzeptanzahl offen gelassen, um zu eruieren, ob diese einen Einfluss bei bestimmten Verfahren hat. Die Ergebnisse sind in den Abbildungen 11 und 12 zu sehen. Wie bereits zuvor ähneln sich die Ergebnisse für deutsche und englische Texte sehr. Besonders hervorstechend ist das Verfahren *DecisionTree*, welches unabhängig von der Dimensionalität den größten Fehler aufweist. Ein einfacher Entscheidungsbaum genügt an dieser Stelle nicht, um die Komplexität der Daten abzubilden. Die Ensemble-Methoden *RandomForest* und *ExtraTrees* hingegen sind besser als die Baseline.

Ebenfalls liegen sämtliche *SVR*-Varianten mit nichtlinearen Kernel überwiegend über der Baseline und im Gegensatz zu den meisten anderen Verfahren steigt der Fehler mit der Anzahl der LSI-Konzepte. Dies spricht dafür, dass die Daten überwiegend in einem einfachen linearen Zusammenhang stehen. Komplexere Kernel können die Daten jedoch nicht

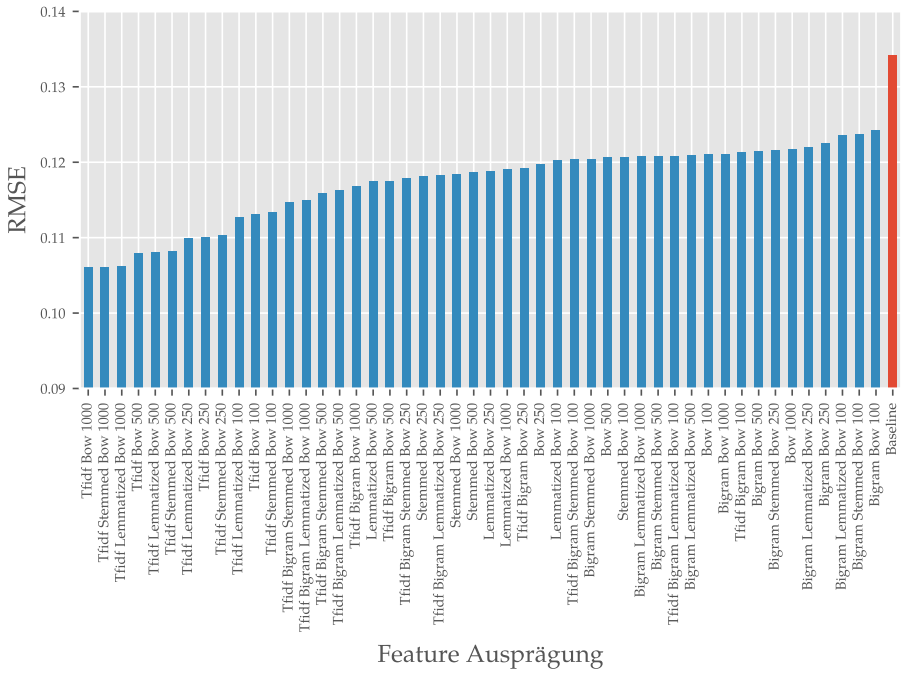

Abbildung 9: Vergleich verschiedener BOW-Ausprägungen (deutschsprachig) verwendet Ridge Regression

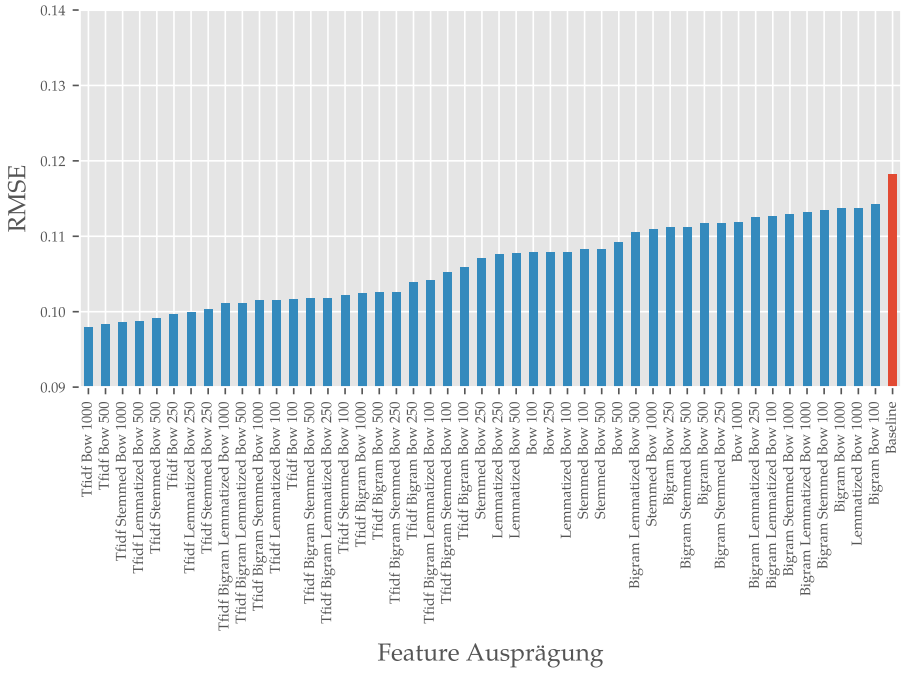

Abbildung 10: Vergleich verschiedener BOW-Ausprägungen (englischsprachig) verwendet Ridge Regression

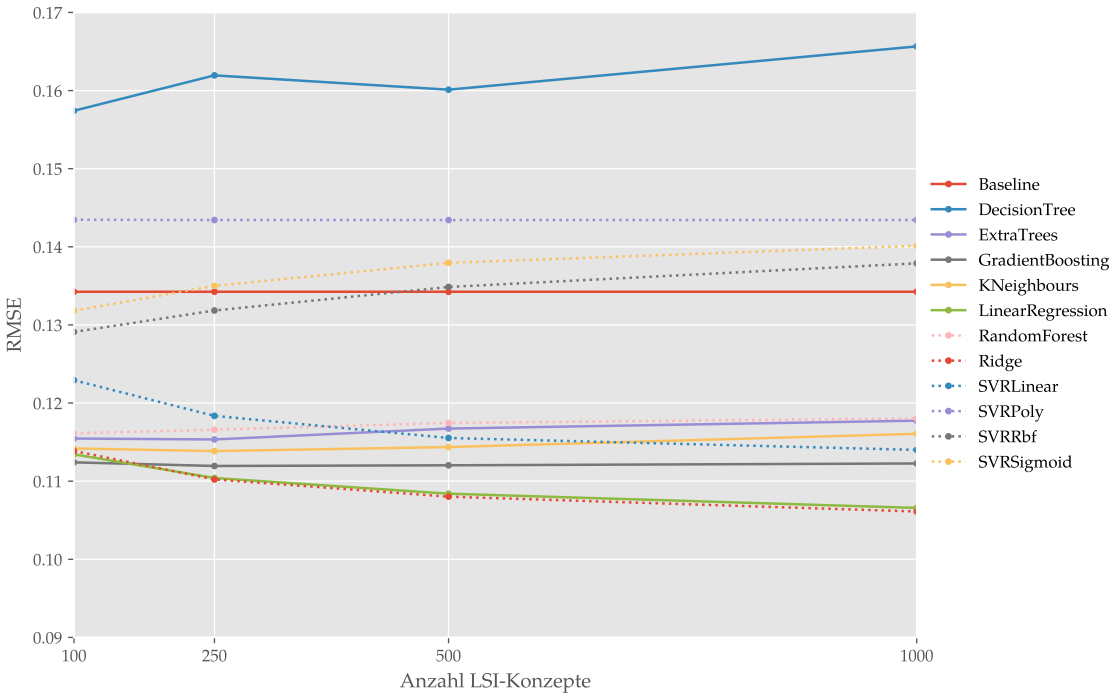

Abbildung 11: Vergleich von Regressoren und LSI-Konzeptgrößen (deutschsprachig)

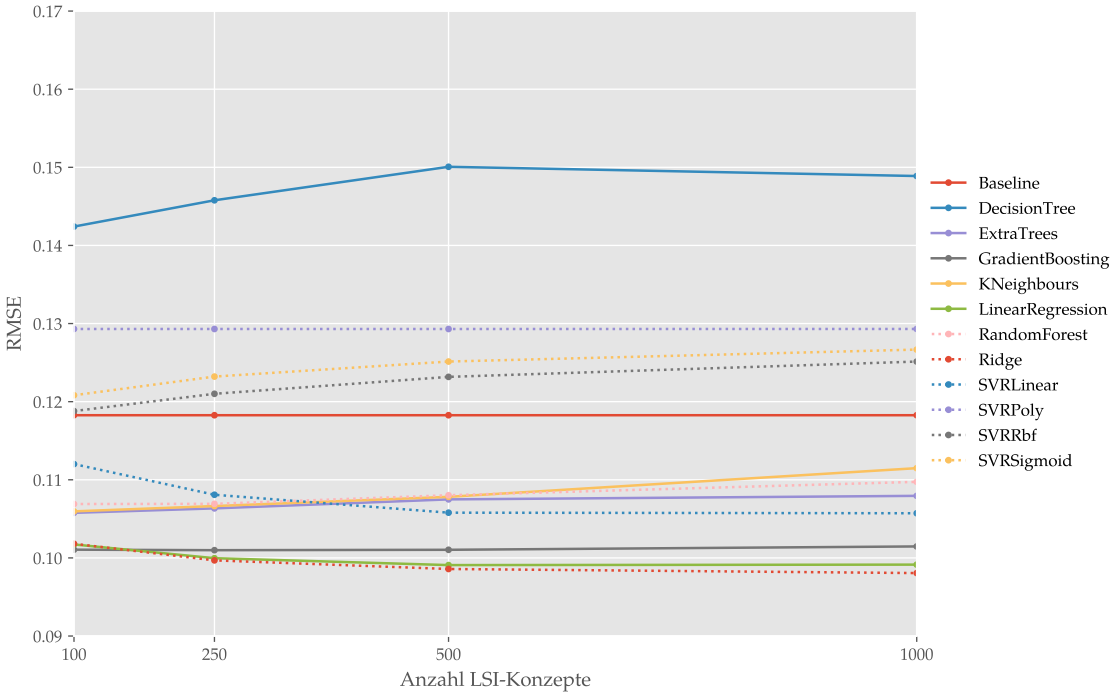

Abbildung 12: Vergleich von Regressoren und LSI-Konzeptgrößen (englischsprachig)

akkurat abbilden. Den geringsten Fehler liefert die *Ridge Regression*, wobei die *Linear Regression* nur unwesentlich schlechter arbeitet.

Aus diesen Untersuchungen ergibt sich, dass die BOW auf dem Attribut „Artikeltext" am besten mit einer Gewichtung durch TF-IDF und 1000 LSI-Konzepten arbeitet. Deshalb wird diese Konfiguration für den anschließenden Vergleich mit anderen Features genutzt. Eine zusätzliche Erhöhung der Konzepte ist angesichts des Trends nicht erstrebenswert, da die Verdoppelung der Konzepte von 500 auf 1000 bereits nur eine minimale Verbesserung erzielt hat.

5.5.2 Vergleich der Features

An dieser Stelle werden die fünf Features, die in Kapitel 3 vorgestellt wurden, gegenüber gestellt. Als Emotionslexika kommen dabei die beiden Varianten EmoLex und DepecheMood zum Einsatz, wobei letzteres nicht für deutsche Texte geeignet ist. Aus Gründen der Vollständigkeit wird dieses Lexikon dennoch für die deutschsprachigen Experimente genutzt, aber dementsprechend sind keine guten Ergebnisse zu erwarten. Das Feature „Readability" enthält einerseits den Lesbarkeitsindex der jeweiligen Sprache, sowie zusätzlich die Textstatistiken.

Für das Feature Bag-of-Words wird die zuvor evaluierte Konfiguration mit Gewichtung durch TF-IDF und 1000 LSI-Konzepten genutzt, da diese den geringsten Fehler lieferte. Alle Features werden jeweils separat mit den drei Attributen „Message", „Titel" und „Artikeltext" getestet. In den Abbildungen 13 und 14 sind die mit dem Verfahren *Ridge Regression* erzielten Ergebnisse dargestellt. Dieses Verfahren hat zuvor bei der Untersuchung des BOW-Features den geringsten Fehler erwirkt.

Die Ergebnisse zeigen unabhängig von der Sprache, dass das BOW-Verfahren auf dem Attribut „Artikeltext" (Text) mit Abstand den geringsten Fehler mit ca. 0.1 produziert. Darauf folgt der Ansatz *Word2Vec* (GloVe) auf dem Textattribut, sowie wiederum das BOW-Verfahren auf den anderen Attributen. Alle weiteren Features, insbesondere die Emotionslexika und die Textstatistiken, liegen dagegen sehr nahe an der Baseline und bieten dadurch kaum einen Mehrwert in der Vorhersage.

Die Emotionslexika liefern insbesondere in Kombination mit den Attributen „Title" und „Message" ein schlechtes Ergebnis. Die dort hinterlegten Texte sind häufig sehr kurz und beinhalten aus diesem Grund nur vereinzelt Wörter, die auch im Lexikon vermerkt sind. Deshalb ist es mit diesen Features nicht möglich, eine sinnvolle Vorhersage zu treffen. Als einziger Ausreißer konnte das Feature DepecheMood auf englischsprachigen Artikeltexten das viertbeste Ergebnis erzielen. Da dieses Emotionslexikon deutlich umfangreicher als EmoLex ist, konnte es häufiger einen passenden Eintrag finden.

Bei Betrachtung des Einflusses der Attribute auf die Vorhersagequalität allgemein kann eine analoge Aussage getroffen werden. Alle Features in Kombination mit dem Attribut „Artikeltext" haben im Vergleich zu den anderen Attributen einen kleineren Fehler produziert. Die einzige Ausnahme bildet der Lesbarkeitsindex (Readability) auf dem englischen Datensatz, der sogar schlechter als die Baseline ist. Die deutsche Variante ist hingegen etwas besser. Möglicherweise lassen sich deutsche Texte einfacher mit Textstatistiken differenzieren.

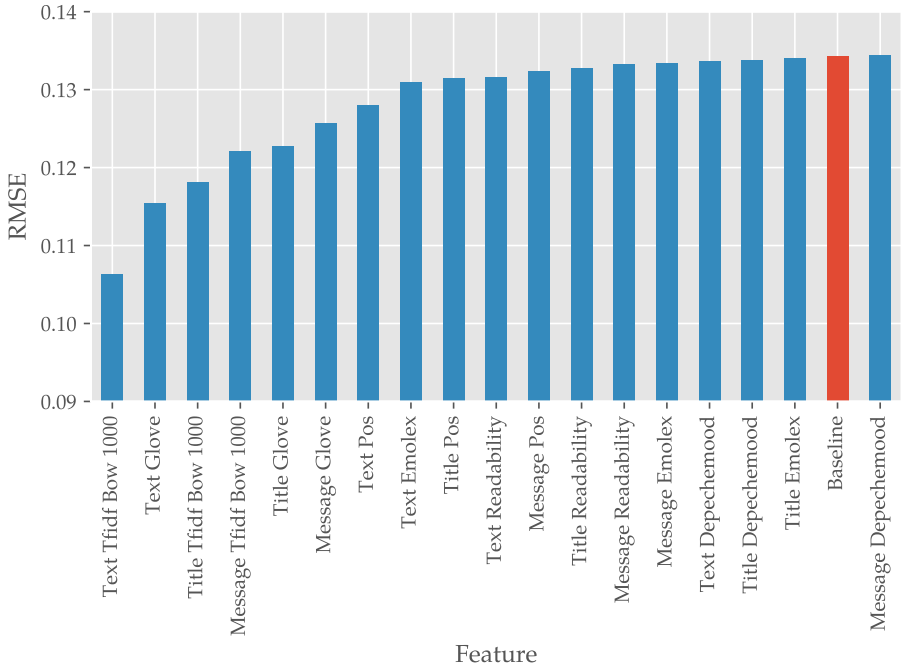

Abbildung 13: Vergleich von Features und Attributen (deutschsprachig) verwendet Ridge Regression

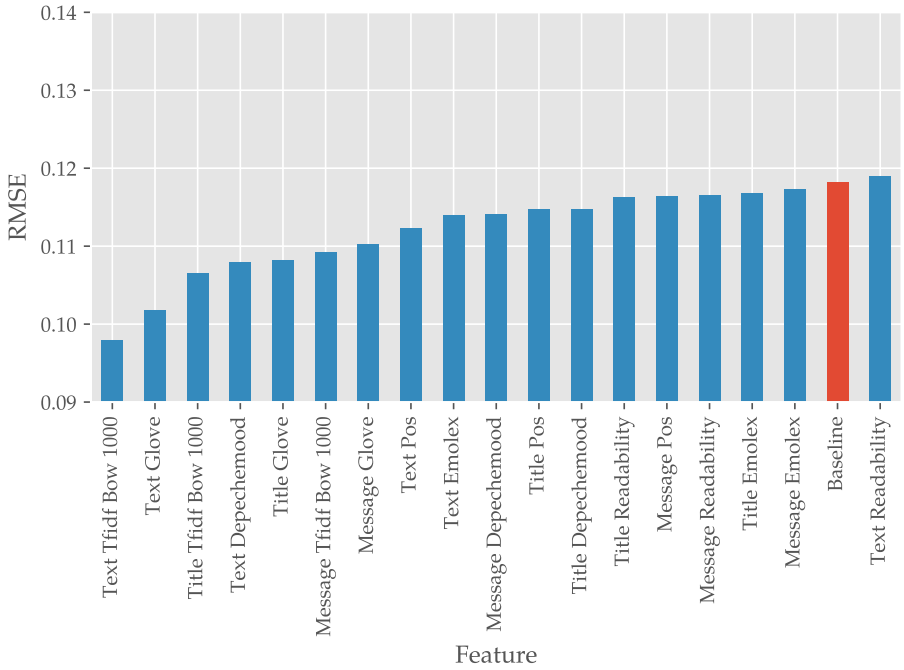

Abbildung 14: Vergleich von Features und Attributen (englischsprachig) verwendet Ridge Regression

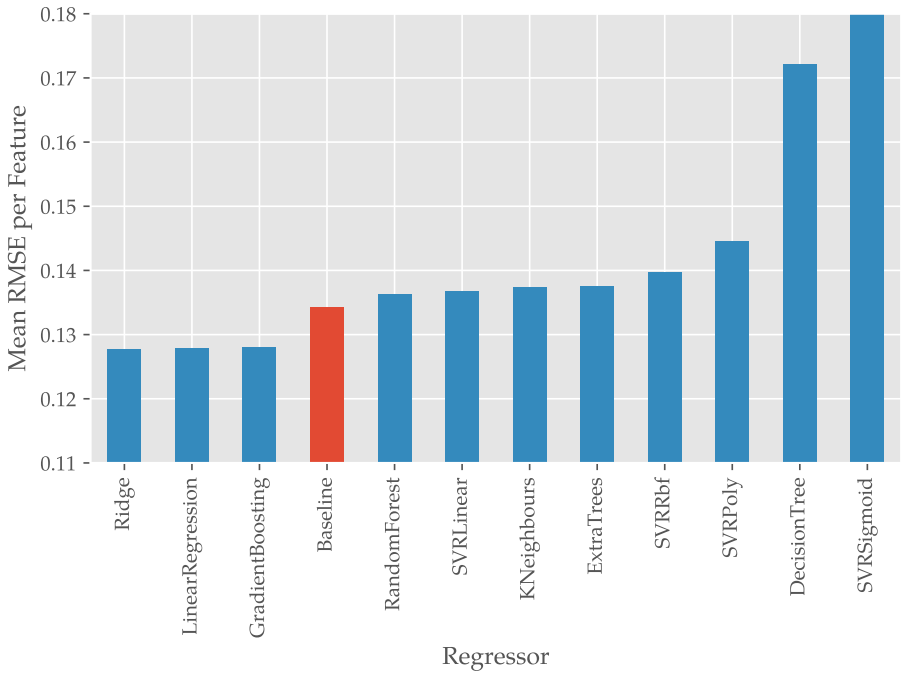

Abbildung 15: Mittlerer Fehler auf allen Features je Regressor (deutschsprachig)

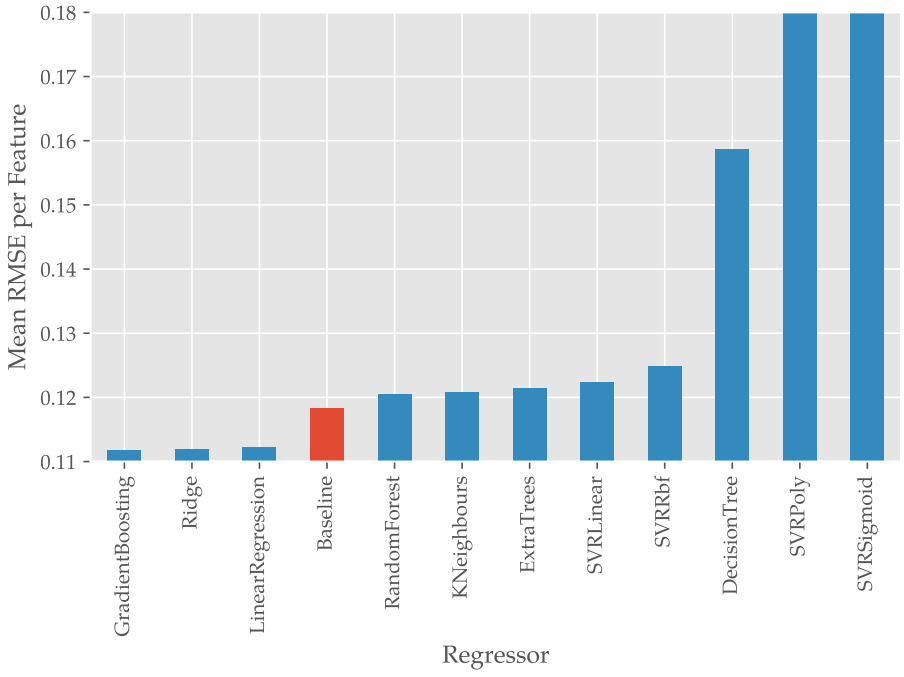

Abbildung 16: Mittlerer Fehler auf allen Features je Regressor (englischsprachig)

Um zu untersuchen, ob einige Features mit bestimmten Regressoren bessere Ergebnisse erzielen, wird obiges Experiment mit allen Regressionsverfahren durchgeführt. Aufgrund der zahlreichen Kombinationen entstehen dabei mehrere Hundert Resultate, weshalb aus Gründen der Übersichtlichkeit in Abbildungen 15 und 16 der mittlere Fehler pro Regressor dargestellt wird. Die vollständigen Ergebnisse sind im Anhang A für den deutschen und den englischen Datensatz separat hinterlegt. Bereits durch die gruppierten Ergebnisse wird sichtbar, dass die Verfahren *Ridge*, *LinearRegression* und *GradientBoosting* fast identisch in ihrer Vorhersagegenauigkeit sind. Alle anderen Verfahren liegen dagegen hinter der Baseline, teilweise überragen sie sogar deutlich das dargestellte Intervall. *SVRPoly* liegt beim englischen Datensatz bei einem Fehler von 0.5, *SVRSigmoid* bei 141.2 (deutsch) und 134.2 (englisch).

5.5.3 Kombination von Features

Im nächsten Schritt werden die erfolgversprechendsten Features BOW, Glove und DepecheMood kombiniert, um gemeinsam ein neues Feature zu bilden. Dahinter steckt die Idee, dass zusammengelegte Features eine noch bessere Vorhersage liefern könnten als ihre isolierten Pendants. Alle Features werden in der Kombination dabei gleich gewichtet.

In den Abbildungen 17 und 18 sind die mit der *Ridge Regression* erzielten Ergebnisse visualisiert. Zum Vergleich sind die einfachen Features ebenfalls in der Grafik hinterlegt. Dadurch wird sichtbar, dass verschiedene Permutationen der Features und den jeweiligen Attributen, die in Abschnitt 5.5.2 auf den ersten vier Plätzen lagen, eine minimal bessere Vorhersage ermöglichen als das Feature „Text Tfidf Bow 1000", welches zuvor das beste Ergebnis lieferte.

Die Features „All Bow" und „All Glove" bezeichnen die Vereinigung der BOW, bzw. Glove auf den drei Attributen „Artikeltext", "Titel" und „Message", die allerdings bereits einen größeren Fehler produzieren als das Feature BOW auf „Artikeltext" alleine. Dies legt die Schlussfolgerung nahe, dass nicht alle Kombinationen von Features das Ergebnis zwangsläufig verbessern. Besonders unerwartet ist die Kombination der BOW und DepecheMood auf dem „Artikeltext" im deutschsprachigen Datensatz. Da DepecheMood keine deutsche Übersetzung hat ist es deshalb für den deutschen Datensatz ungeeignet. Das jeweilige Feature ist in seiner eigenständigen Form auch entsprechend auf dem Niveau der Baseline. Dennoch ist das kombinierte Resultat marginal besser als das BOW-Feature alleine.

5.5.4 Vorhersagequalität differenziert nach Label

An dieser Stelle werden die vorhergesagten Werte im Detail betrachtet, um mögliche Unterschiede der Qualität differenziert je nach Label erkennen zu können. In den Abbildungen 19 und 20 sind jeweils Boxplots für Vorhersagen mit einem BOW-Ansatz inklusive TF-IDF und einem LSI mit 1000 Konzepten auf dem Attribut „Artikeltext" für den deutsch- und englischsprachigen Datensatz zu sehen. Auf der linken Seite sind die Ergebnisse der Baseline dargestellt, während auf der rechten Seite die Ergebnisse der *Ridge*

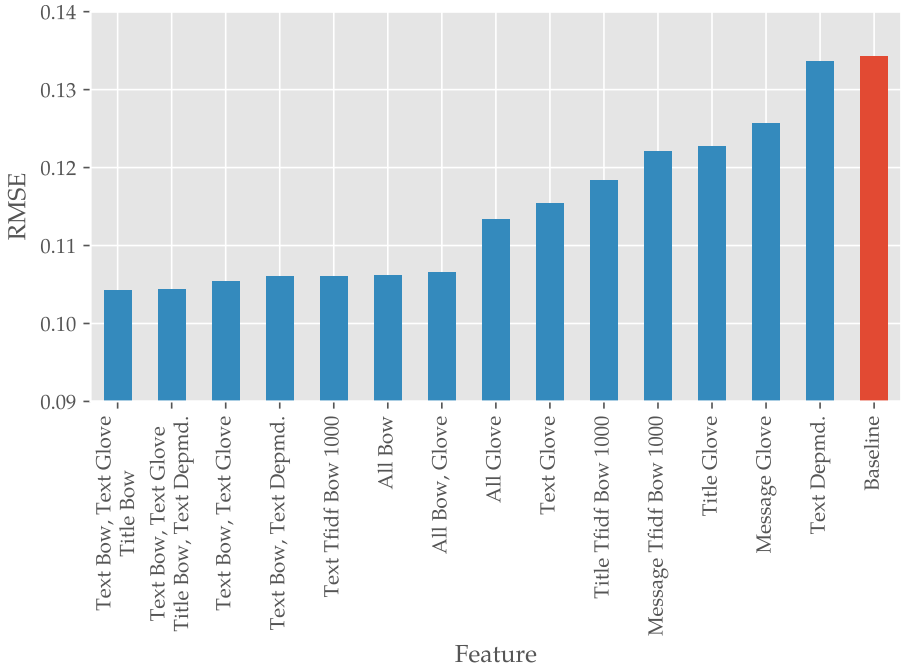

Abbildung 17: Vergleich von kombinierten Features (deutschsprachig) verwendet Ridge Regression

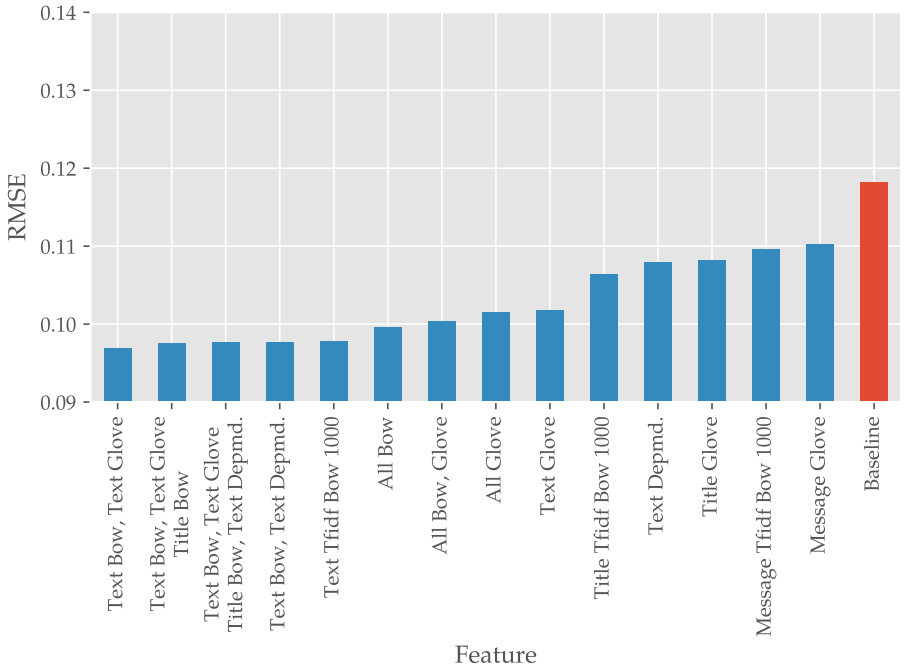

Abbildung 18: Vergleich von kombinierten Features (englischsprachig) verwendet Ridge Regression

Regression abgebildet sind. Der blaue Trennstrich spiegelt den Median der Vorhersagefehler wieder, während das arithmetische Mittel durch ein Dreieck markiert wird.

Inhaltlich unterscheiden sich die Ergebnisse beider Datensätze kaum, weshalb eine pauschale Beurteilung möglich ist. Positiv zu bemerken ist, dass der Fehler pro Datenfall allgemein für über 50% der Daten bei unter 0.2 liegt, bei den von *Like* verschiedenen Labels sogar unter 0.1. In der Kategorie *Wow* ist die Abweichung besonders gering, was allerdings auch zu erwarten ist, da in Abschnitt 5.1.2 gezeigt wurde, dass der mittlere Anteil dieser Reaktion im Datensatz sehr klein ist. Mit der gleichen Begründung ist der hohe Fehler beim Label *Like* zu erklären, da dessen Anteil in den Ausgangsdaten bei bis zu 0.8 liegt. Entsprechend überraschend ist die sehr gute Vorhersage der *Love*-Reaktion, dessen Ausreißer sogar fast alle unter 0.2 liegen. Denn in Abbildung 7 ist zu sehen, dass der Anteil dieser Reaktion ähnlich zu denen aller anderen außer dem *Like* ist.

Der Vergleich mit dem Baseline-Modell zeigt, dass das entwickelte Vorhersagemodell in vielen Fällen eine Abweichung nahe 0 vorweisen kann. Bei der Baseline reichen die unteren Whisker hingegen bei den Labels *Angry*, *Haha* und *Sad* nicht bis zu diesem Wert. Es kann erkannt werden, dass beispielsweise bei der Reaktion *Angry* 50% aller Baseline-Werte einen Fehler im Intervall $[0.08, 0.09]$ haben. Geringere Abweichungen sind an dieser Stelle lediglich als einzelne Ausreißer markiert.

Ein detaillierter Blick darauf, welche Datenfälle in der Regel besonders gut und welche besonders schlecht vorhergesagt werden, zeigt ein Muster. Dabei gilt für alle Labels, dass die Vorhersage für Datenfälle am besten ist, wenn der Anteil des jeweiligen Labels nahe 0 ist. Analog ist der Vorhersagefehler am höchsten, wenn der Anteil eines Labels überproportional hoch ist und die Mehrheit einnimmt. Letztere Konstellation tritt verhältnismäßig selten auf und ähnelt in der Verteilung auch nicht den mittleren Werten aus Abbildung 7. Deshalb liefert das hier verwendete lineare Regressionsmodell für diese Ausreißer ein schlechtes Ergebnis. Dennoch wäre es an dieser Stelle besonders interessant, eine präzise Vorhersage treffen zu können.

5.5.5 Vorhersagequalität differenziert nach Nachrichtenseite

Bisher wurden die Versuche ausschließlich mit den gesamten englischen und deutschen Datensätzen durchgeführt. Allerdings ist auch die Frage interessant, inwieweit die Wahl der Nachrichtenseite Einfluss auf die Vorhersagequalität nimmt. Aus diesem Grund wird das Modell jeweils auf den Datensätzen der einzelnen Nachrichtenseiten trainiert. Wie zuvor wird dabei die *Ridge Regression* und das Feature „Text Tfidf Bow 1000" eingesetzt.

In Abbildung 21 sind die Vorhersageergebnisse unter der Bezeichnung „error" separat nach Nachrichtenseite dargestellt. Zusätzlich wurde auch jeweils die zugehörige Baseline ermittelt. Die Ergebnisse der deutschsprachigen Nachrichtenseiten liegen im Intervall $[0.102, 0113]$. Der Mittelwert der drei Vorhersageergebnisse beträgt 0.109, wohingegen ein Modell, das auf dem kombinierten deutschen Datensatz trainiert wurde, einen Fehler von 0.106 hat. Daraus lässt sich folgern, dass die Verwendung eines größeren und heterogeneren Datensatzes einen positiven Einfluss auf die Ergebnisse hat.

Die englischsprachigen Nachrichtenseiten zeigen deutlichere Unterschiede mit Vorhersagefehlern im Intervall $[0.80, 0.122]$. Der Fehler des Modells „theguardian" verortet sich

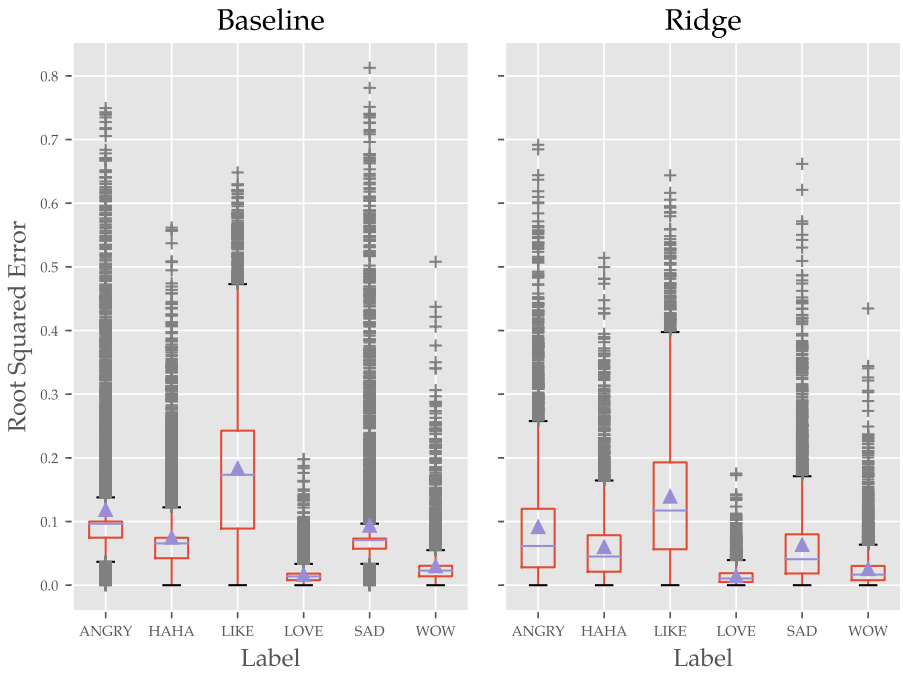

Abbildung 19: Vorhersagefehler je Label (deutschsprachig) verwendet Ridge Regression und Feature „Text Tfidf Bow 1000"

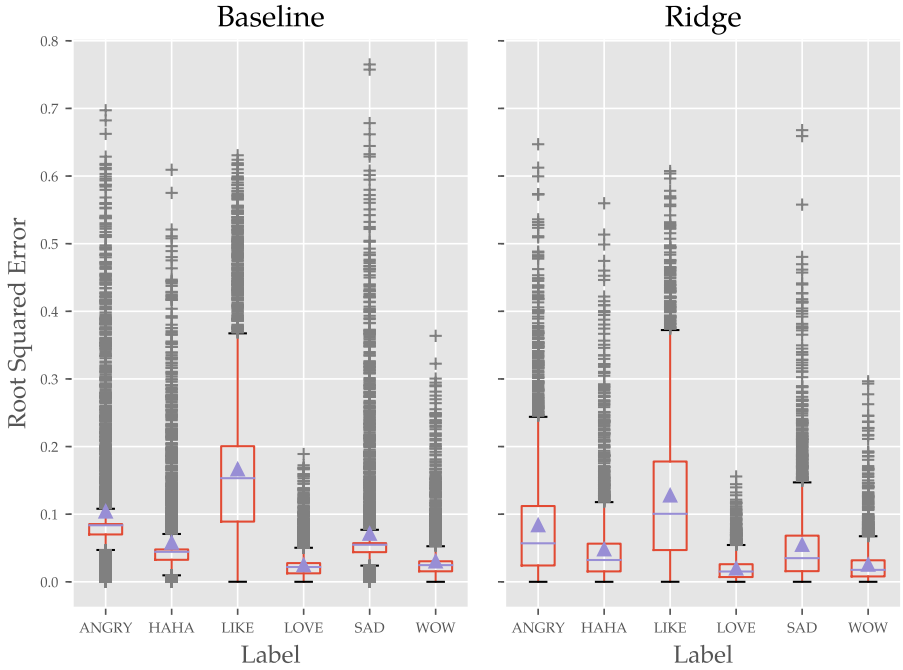

Abbildung 20: Vorhersagefehler je Label (englischsprachig) verwendet Ridge Regression und Feature „Text Tfidf Bow 1000"

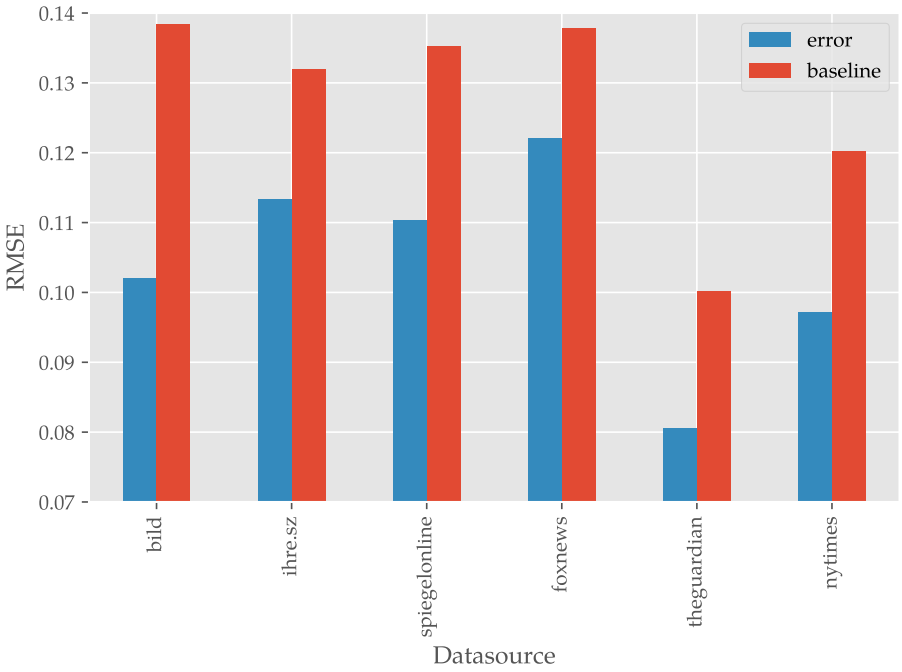

Abbildung 21: Vorhersagefehler je Nachrichtenseite
verwendet Ridge Regression und Feature „Text Tfidf Bow 1000"

am unteren Ende des Intervalls. Allerdings zeigt die Baseline-Vorhersage für diese Nachrichtenseite ebenfalls den geringsten Fehler. Am oberen Ende des Intervalls ist „foxnews" zu sehen. Wie zuvor in Abschnitt 5.1 beschrieben, betreibt diese Nachrichtenseite die populärste Facebook-Präsenz. Aufgrund der größeren Anzahl Benutzerreaktionen ist die Verteilung dieser möglicherweise unvorhersehbarer. Der größere Fehler der Baseline deutet darauf hin, dass die Werte hier stärker gestreut sind. Auch bei englischen Texten zeigt sich, dass der mittlere Fehler aller drei getrennt gelernten Mittel mit 0.100 etwas schlechter ist, als die des kombinierten Modells mit 0.098.

6 Verwandte Arbeiten

In diesem Kapitel werden Forschungsarbeiten vorgestellt, die einen vergleichbaren Kontext haben. Dabei werden einerseits Arbeiten betrachtet, die sich ebenfalls in der Domäne der sozialen Medien bewegen. Andererseits ist die Frage interessant, welche Beiträge andere Forscher zur Emotionsanalyse beigetragen haben.

6.1 Facebook Sentiment: Reactions and Emojis

Tian et al. [TGD+17] haben deskriptiv die Benutzerreaktionen und Emojis in Beiträgen und Kommentaren auf Facebook untersucht. Dabei verfolgten sie einen ähnlichen Ansatz wie in dieser Arbeit und haben Beiträge von Nachrichtenseiten heruntergeladen. Zur Auswahl kamen jeweils ca. vier Nachrichtenseiten aus den USA, Großbritannien, Frankreich und Deutschland. So wurden insgesamt 21 000 Beiträge aus dem Zeitraum August 2016 gesammelt. Zusätzlich wurden die Benutzerreaktionen erhoben und über acht Millionen Kommentare zu den zugehörigen Beiträgen heruntergeladen.

Daraufhin untersuchten die Autoren die Verteilung der Reaktionen. Dabei kamen sie wie in dieser Arbeit zu der Erkenntnis, dass die Reaktion *Like* am häufigsten verwendet wird. Am seltensten wird dagegen die Reaktion *Wow* zum Ausdruck gebracht. Zusätzlich wurde erläutert, dass kleine – aber statistisch signifikante – Unterschiede in der Verteilung in Abhängigkeit des Landes existieren. In dieser Arbeit wurde allerdings in Abschnitt 5.1 gezeigt, dass die Verteilung bereits je nach betrachteter Nachrichtenseite Schwankungen unterliegt. Somit wird die Verteilung der Benutzerreaktionen eines Landes von der Wahl der Nachrichtenseite beeinflusst und nicht, wie im Artikel angedeutet, durch die landestypische Mentalität der Benutzer. Zusätzlich wurden die Beiträge anhand des Verhältnisses der Benutzerreaktionen mittels *K-Means* in vier Cluster eingeteilt. Dadurch kamen unterschiedliche Profile zum Vorschein. Die größte Gruppe der Beiträge beinhaltete überwiegend *Like*-Reaktionen. Die zweitgrößte Gruppe war ähnlich strukturiert, hatte aber einen etwas größeren Anteil an den Reaktionen *Haha* und *Angry*. Die dritte Gruppe bestand, neben dem großen Anteil der *Like*-Reaktionen, fast zur Hälte aus *Angry*-Reaktionen, die vierte Gruppe fast zur Hälfte aus *Sad*-Reaktionen.

Der zweite Teil des Artikels untersuchte die Verteilung von Emojis in Kommentaren. Dazu wurden ohne Nennung einer Begründung lediglich 100 000 zufällige Kommentare ausgewählt. Die Häufigkeiten der Emojis unterlagen einer Zipfschen Verteilung. Am häufigsten war dabei der „Daumen nach oben", gefolgt vom „Smiley mit Herzaugen". Unter Zuhilfenahme eines Sentiment-Lexikons, dass jedem Emoji einen Sentiment-Wert zuordnet, wurde das mittlere Sentiment der zuvor ermittelten vier Cluster berechnet. Das Resultat bestand darin, dass das erste Cluster, welches überwiegend aus *Like*-Reaktionen bestand, das positivste Sentiment ausdrückte.

Der Artikel unterscheidet sich von dieser Arbeit insofern, dass im Artikel neben Benutzerreaktionen auch Emojis in Kommentaren untersucht wurden. Allerdings sind alle Untersuchungen lediglich beschreibend. In dieser Arbeit werden hingegen die Benutzerreaktionen und der verknüpfte Artikeltext tiefergehend betrachtet. Daraus wird dann ein Vorhersagemodell mittels maschinellen Lernens entwickelt.

6.2 EMOTEX: Detecting Emotions in Twitter Messages

Hasan et al. [HRA] verfolgten mit **EMOTEX** eine Aufgabenstellung, der dieser Arbeit sehr ähnelt. Mittels maschinellen Lernens wurde ebenfalls ein Modell entwickelt, dass Emotionen von Beiträgen eines sozialen Netzwerks vorhersagen kann. Allerdings konzentrierten sich die Autoren auf die Plattform Twitter. Dort existieren keine Benutzerreaktionen, allerdings werden auf Twitter im Gegensatz zu Facebook sogenannte Hashtags sehr häufig verwendet. Mit Hashtags können Benutzer ihre Beiträge selbständig einem vorher nicht definierten Thema oder einer Stimmung zuordnen. Diese Hashtags machten sich die Autoren zu Nutze und definierten vier Emotionskategorieren: „Happy-Active", „Happy-Inactive", „Unhappy-Active" und „Unhappy-Inactive". Für jede Kategorie wurden 20 Hashtags ausgesucht, die die Emotionen der Kategorie am besten beschreiben. Daraufhin wurde Twitter nach den ausgewählten Hashtags durchsucht und die zurückgelieferten Beiträge heruntergeladen. Dadurch entstand ein Datensatz bestehend aus über 160 000 Beiträgen. Davon wurden 19% der Beiträge in einer nicht näher beschriebenen Vorfilterung entfernt.

Auf Grundlage dieser Daten entwickelten die Autoren ein Klassifikationsmodell, dass Beiträge von Twitter einer der vier zuvor definierten Kategorien zuordnen kann. Als Features kam dabei eine Kombination aus Emotionslexikon, Erkennung von Emoticons, Häufigkeit des Auftretens von Satzzeichen und Negationen zum Einsatz. Daraufhin wurden Vorhersagemodelle mit den Algorithmen *Naive Bayes*, *Support Vector Maschine*, *Decision Trees* und *k-Nearest-Neighbour* trainiert. Die Ergebnisse wurden unter anderem anhand des Maßes „F-Measure" ermittelt und für jedes Feature einzeln, als auch kombiniert aufgelistet. Die Ergebnisse zeigten, dass die höchste Genauigkeit von 90.2 mit KNN und der Kombination aus Emotionslexikon und Negationen erzielt werden konnten. Allerdings war die Genauigkeit der SVM mit dem Emotionslexikon und der Decision Trees mit einer Kombination aller Features mit 90.0 fast genauso gut. Mit Naive Bayes konnte mit allen Features lediglich eine Genauigkeit von 86.9 erzielt werden.

Diese Arbeit verfolgt einen zum Artikel vergleichbaren Ansatz um Emotionen in Beiträgen sozialer Netzwerke vorherzusagen. Als Grundlage wurden allerdings Beiträge von Facebook sowie die Benutzerreaktionen betrachtet, anstatt Beiträge von Twitter gezielt nach Hashtags auszusuchen. Weiterhin wurden die Beiträge im Artikel in selbst definierte Emotionsklassen eingeteilt, die sich nicht an den im Abschnitt 2.2 vorgestellten Emotionsmodellen orientieren. Die Benutzerreaktionen auf Facebook entsprechen dahingegen teilweise den Basisemotionen. Im Gegensatz zum Artikel kommt keine Klassifizierung sondern eine Regression zum Einsatz, um das Verhältnis der Benutzerreaktionen vorhersagen zu können.

7 Fazit

Zu Beginn der Arbeit wurde in Kapitel 1 die Motivation dargelegt. Es sollte gezeigt werden, dass die Forschung an der Emotionserkennung, welcher es an frei verfügbaren Datensätzen mangelt, mithilfe von Facebook vorangetrieben werden kann. Tatsächlich konnte in Abschnitt 5.1 ein Datensatz aus sechs exemplarisch gewählten Nachrichtenseiten bezogen werden, wobei es möglich ist, beliebig weitere Daten zu crawlen. Zum einen können mehr Nachrichtenseiten ausgesucht werden, eventuell um sogar zusätzliche Sprachen abzudecken, zum anderen werden auf dem sozialen Medium Facebook seit der Einführung der Benutzerreaktionen Ende Februar 2016 kontinuierlich weitere Daten generiert.

Als Ziel wurde daraufhin in der Motivation die Entwicklung eines automatischen Vorhersagemodells für eben jene Benutzerreaktionen gesetzt. Zu diesem Zweck wurde von den auf Facebook betrachteten Nachrichtenseiten die zugehörigen Originalartikel bezogen, um sie als Eingabe für das Vorhersagemodell zu nutzen. Zur Verdeutlichung, wie dies realisiert werden kann, wurden zunächst die Grundlagen des maschinellen Lernens und des Natural Language Processings erläutert. Darauf aufbauend wurde dann anhand verschiedener Features aufgezeigt, wie die Texte aus dem Artikel und von Facebook in eine Vektordarstellung bestehend aus reellen Zahlen überführt werden. Unter Zuhilfenahme der NLP-Werkzeuge *NLTK* und *spaCy* konnte diese Aufgabe automatisiert werden.

Daraufhin stand die Konstruktion des Vorhersagemodells im Fokus. In Kapitel 4 wurden verschiedene Ansätze für Regressions-Verfahren vorgestellt. Dabei hat sich die Software-Bibliothek *scikit-learn* von großer Nützlichkeit erwiesen. Damit konnten alle Verfahren mit geringen Aufwand implementiert und eingesetzt werden.

Im darauf folgenden Kapitel 5 wurde zur Evaluation übergeleitet. Dabei wurde begründet, warum die Aufteilung des Datensatzes in eine Trainings-, Test- und Validationsmenge nötig ist und wie die Kreuzvalidierung eingesetzt werden kann, um das Vorhersagemodell zu optimieren. Um quantifizieren zu können, welches Regressions-Verfahren das bessere Ergebnis liefert, wurde das Maß des *Root Mean Squared Errors* definiert. Dieses Maß wurde genutzt um die entwickelten Vorhersagemodelle auf ihre Verlässlichkeit zu überprüfen. Hierzu wurde die Abweichung zwischen der Vorhersage und dem realen Wert gemessen.

7.1 Ergebnisse

Bevor in verschiedenen Experimenten die Features miteinander verglichen werden konnten, wurde das Feature Bag-of-Words auf die optimale Konfiguration eingestellt. Dabei stellte sich heraus, dass mit der zusätzlichen Gewichtung durch TF-IDF und der Dimensionsreduktion auf 1000 Konzepte durch das LSI das beste Ergebnis zu erzielen ist. Letztendlich erzielte das BOW-Feature einen Fehler von unter 0.11 für deutsche Texte und sogar unter 0.10 für englische Texte auf dem Attribut „Artikeltext" und dem Regressor *Ridge*. Allerdings ist dabei anzumerken, dass die Baseline, welche konstant den Mittelwert vorhersagt, in beiden Fällen lediglich um 0.02 schlechtere Ergebnisse liefert.

Daraufhin wurden die Vorhersageleistungen der einzelnen Features mit den Attributen „Artikeltext", „Titel" und „Message" gegenübergestellt. Als Regressor kam zunächst wieder die *Ridge-Regression* zur Anwendung. Das zuvor beschriebene Feature BOW mit TF-IDF und 1000 LSI-Konzepten auf dem „Artikeltext" erzielte an dieser Stelle das beste Ergebnis bei beiden Sprachen. Die zweitbeste Vorhersage lieferte Glove ebenfalls auf dem „Artikeltext", gefolgt vom BOW-Feature auf den beiden anderen Attributen. Dahingegen waren die Fehler der Emotionslexika und Lesbarkeitsindizies fast identisch mit der Baseline. Als Ausnahme davon zeigte sich DepecheMood auf dem „Artikeltext" im englischsprachigen Datensatz. Grundsätzlich zeigten die beiden anderen Attribute „Titel" und „Message" größere Vorhersagefehler, da dessen Textumfang deutlich geringer ist.

Dieses Resultat motivierte weitere Versuche zur Minimierung des Vorhersagefehlers. Im Vergleich der Regressoren wurde festgestellt, dass das Verfahren *Ridge* bereits gemeinsam mit dem *GradientBoosting* und der *LinearRegresion* das beste Ergebnis liefert. Das einfache Verfahren *DecisionTree*, sowie die *SVR* mit nicht-linearen Kernel waren dahingegen schlechter als die Baseline. Dies ließ darauf schließen, dass in Ansätzen ein linearer Zusammenhang zwischen den Texten und den Benutzerreaktionen besteht.

Daraufhin wurde untersucht, inwieweit die Kombination verschiedener Features die Vorhersage verbessern könnte. Ein kleiner Erfolg konnte dabei erzielt werden, als beispielsweise die Kombination der BOW und dem Word2Vec-Ansatz *GloVe* auf dem Attribut „Artikeltext" eine minimale Verbesserung gegenüber dem reinen BOW-Ansatz zeigten. Andere Kombinationen der erfolgversprechendsten einzelnen Features zeigten dahingegen kaum einen kleineren Fehler oder sogar einen größeren.

Zur weiteren Detailanalyse wurde untersucht, wie sich der Vorhersagefehler auf die einzelnen Benutzerreaktionen verteilt. Dabei wurde ersichtlich, dass, bis auf die Reaktion *Like*, mehr als die Hälfte der Datenfälle mit einem Fehler von unter 0.1 vorhergesagt werden können. Der größere Fehler bei der Reaktion *Like* ist daraufhin zurückzuführen, dass diese Reaktion im Mittel einen Anteil von 0.8 im gesamten Datensatz einnimmt. Deshalb existieren dort häufiger stärkere Schwankungen des Anteils, die schwieriger vorherzusagen sind.

Bei der anschließenden Betrachtung der Vorhersagequalität bezogen auf die einzelnen Nachrichtenseiten haben sich Unterschiede zwischen den englischsprachigen Portalen gezeigt. Während der Fehler für Artikel von „theguardian" am kleinsten war, konnte auf dem Datensatz, der nur aus von „foxnews" bezogenen Artikeln besteht, die schlechteste Vorhersage erzielt werden.

Abschließend kann zusammengefasst werden, dass die erzielten Ergebnisse des Vorhersagemodells einen grundlegenden Beitrag zur Emotionserkennung liefern. Allerdings wurde in der Motivation postuliert, ein tatsächlich produktiv einsetzbares Werkzeug daraus zu entwickeln. Dafür ist die Vorhersage jedoch noch nicht genau genug.

7.2 Ausblick

Als möglicher Anknüpfungspunkt für zukünftige Forschungen kann beispielsweise das Parameter-Tuning der Regressoren noch intensiver betrieben werden. Dabei kann im Ver-

lauf der Kreuzvalidierung durch das automatische Ausloten verschiedener Konfigurationen der Regressionsverfahren die Beste ausgewählt werden. Aufgrund der bereits sehr hohen Anzahl an Möglichkeiten das Vorhersagemodell zu konstruieren (vgl. 5.5), wurde das Tuning lediglich manuell umgesetzt. Da die Wahl der Parameter letztlich auch nur marginale Veränderungen des Vorhersagefehlers bewirkten, ist an dieser Stelle kein ausschlaggebendes Verbesserungspotential zu vermuten.

Aus diesem Grund könnten weitere Features konstruiert und evaluiert werden. Denkbar wäre eine Vorverarbeitung der aus Facebook stammenden Texte des Attributs „Message". Durch eine gesonderte Betrachtung der Domäne der sozialen Medien wäre es möglicherweise sinnvoll, Emojis miteinzubeziehen. Weiterhin können weitere Daten bezogen werden, indem von Facebook die Kommentare zu den Beiträgen heruntergeladen werden. Dadurch existiert dann eine weiteres Feature, dass mit den Features kombiniert werden kann.

Darüber hinaus wäre die Wahl eines deutlich anderen Ansatzes vielversprechend. In vielen Bereichen mit Anwendungen des maschinellen Lernens ist das Verfahren des *Deep Learning* [DY14], welches auf künstlichen neuronalen Netzen basiert, sehr erfolgreich. Beispielsweise hat Google für ihren Übersetzer ein neues Modell eingeführt, dass auf rekurrenten neuronalen Netzen basiert [WSC+16]. Dadurch wurde die Qualität des Übersetzers im Vergleich zum vorherigen statistischen Modell verbessert. Die Anwendung von Deep Learning zur Emotionserkennung bietet daher eine interessante Aufgabenstellung für zukünftige Forschungen.

A Anhang

Detaillierte Ergebnisse zum Vergleich der Features (Deutsch)

Feature	Regressor	RMSE
Text Tfidf Bow 1000	Ridge	0.106
Text Tfidf Bow 1000	LinearRegression	0.107
Text Tfidf Bow 1000	GradientBoosting	0.112
Text Tfidf Bow 1000	SVRLinear	0.114
Text Glove	Ridge	0.115
Text Glove	LinearRegression	0.115
Text Glove	GradientBoosting	0.116
Text Tfidf Bow 1000	KNeighbours	0.116
Text Tfidf Bow 1000	RandomForest	0.117
Title Tfidf Bow 1000	Ridge	0.118
Text Tfidf Bow 1000	ExtraTrees	0.118
Title Tfidf Bow 1000	LinearRegression	0.119
Text Glove	KNeighbours	0.121
Message Tfidf Bow 1000	Ridge	0.122
Title Tfidf Bow 1000	GradientBoosting	0.122
Title Glove	Ridge	0.123
Title Glove	LinearRegression	0.123
Message Tfidf Bow 1000	LinearRegression	0.123
Text Glove	SVRLinear	0.123
Title Glove	GradientBoosting	0.123
Message Glove	Ridge	0.126
Text Glove	ExtraTrees	0.126
Message Glove	LinearRegression	0.126
Message Glove	GradientBoosting	0.126
Text Glove	RandomForest	0.126
Title Tfidf Bow 1000	SVRLinear	0.126
Message Tfidf Bow 1000	GradientBoosting	0.126
Text Pos	GradientBoosting	0.127
Text Pos	Ridge	0.128
Text Pos	LinearRegression	0.128
Message Tfidf Bow 1000	SVRLinear	0.129
Title Glove	KNeighbours	0.130
Title Tfidf Bow 1000	ExtraTrees	0.130
Title Pos	GradientBoosting	0.130
Title Glove	SVRLinear	0.130
Title Tfidf Bow 1000	RandomForest	0.130
Text Emolex	GradientBoosting	0.130
Title Tfidf Bow 1000	KNeighbours	0.131
Text Emolex	LinearRegression	0.131
Text Emolex	Ridge	0.131
Title Glove	ExtraTrees	0.131
Title Depechemood	GradientBoosting	0.131
Title Pos	Ridge	0.131
Title Pos	LinearRegression	0.131
Text Readability	GradientBoosting	0.131
Text Readability	LinearRegression	0.131
Text Readability	Ridge	0.132
Message Pos	GradientBoosting	0.132
Title Glove	RandomForest	0.132
Text Depechemood	GradientBoosting	0.132
Title Depechemood	RandomForest	0.132
Message Pos	Ridge	0.132
Message Pos	LinearRegression	0.132
Message Depechemood	GradientBoosting	0.133
Title Readability	LinearRegression	0.133
Title Readability	Ridge	0.133
Title Readability	GradientBoosting	0.133
Message Glove	SVRLinear	0.133
Message Tfidf Bow 1000	ExtraTrees	0.133
Message Readability	LinearRegression	0.133
Message Readability	Ridge	0.133
Title Depechemood	ExtraTrees	0.133
Title Glove	SVRRbf	0.133
Message Emolex	GradientBoosting	0.133
Message Emolex	LinearRegression	0.133
Message Emolex	Ridge	0.133
Message Readability	GradientBoosting	0.134
Text Depechemood	Ridge	0.134
Text Depechemood	LinearRegression	0.134
Text Glove	SVRRbf	0.134
Title Depechemood	LinearRegression	0.134
Title Depechemood	Ridge	0.134
Title Emolex	GradientBoosting	0.134
Title Emolex	Ridge	0.134
Title Emolex	LinearRegression	0.134
Title Tfidf Bow 1000	Baseline	0.134
Message Depechemood	Ridge	0.134
Message Depechemood	LinearRegression	0.134
Title Glove	SVRSigmoid	0.134
Message Glove	ExtraTrees	0.134
Text Pos	RandomForest	0.134
Text Pos	ExtraTrees	0.134
Message Tfidf Bow 1000	RandomForest	0.135
Message Depechemood	RandomForest	0.135
Message Depechemood	ExtraTrees	0.135
Message Glove	RandomForest	0.135
Title Emolex	RandomForest	0.135
Message Tfidf Bow 1000	KNeighbours	0.135
Text Glove	SVRSigmoid	0.136
Title Emolex	ExtraTrees	0.136
Text Pos	SVRRbf	0.136
Title Emolex	DecisionTree	0.136
Message Glove	KNeighbours	0.136
Message Glove	SVRRbf	0.136
Message Glove	SVRSigmoid	0.137
Text Tfidf Bow 1000	SVRRbf	0.138
Message Emolex	KNeighbours	0.139
Text Pos	SVRLinear	0.139
Text Pos	KNeighbours	0.139
Title Depechemood	DecisionTree	0.139
Message Emolex	RandomForest	0.139
Title Pos	SVRRbf	0.139
Text Depechemood	RandomForest	0.139

Text Emolex	SVRRbf	0.139
Text Depechemood	ExtraTrees	0.140
Message Depechemood	KNeighbours	0.140
Text Readability	SVRRbf	0.140
Text Tfidf Bow 1000	SVRSigmoid	0.140
Message Emolex	ExtraTrees	0.140
Text Emolex	SVRLinear	0.141
Message Pos	SVRRbf	0.141
Text Depechemood	SVRRbf	0.141
Text Emolex	RandomForest	0.141
Title Depechemood	SVRRbf	0.141
Text Readability	SVRLinear	0.141
Title Pos	SVRLinear	0.141
Title Pos	RandomForest	0.141
Text Readability	RandomForest	0.141
Title Depechemood	KNeighbours	0.142
Text Emolex	KNeighbours	0.142
Title Tfidf Bow 1000	SVRRbf	0.142
Message Pos	RandomForest	0.142
Title Readability	SVRRbf	0.142
Text Readability	ExtraTrees	0.142
Message Tfidf Bow 1000	SVRRbf	0.142
Message Pos	SVRLinear	0.142
Title Depechemood	SVRPoly	0.142
Title Pos	KNeighbours	0.142
Text Depechemood	KNeighbours	0.142
Title Readability	SVRLinear	0.142
Message Depechemood	DecisionTree	0.142
Message Readability	SVRRbf	0.142
Message Depechemood	SVRRbf	0.142
Title Readability	SVRPoly	0.142
Title Tfidf Bow 1000	SVRSigmoid	0.142
Text Emolex	SVRPoly	0.142
Message Tfidf Bow 1000	SVRSigmoid	0.143
Message Readability	SVRLinear	0.143
Text Emolex	ExtraTrees	0.143
Message Depechemood	SVRPoly	0.143
Text Readability	KNeighbours	0.143
Title Emolex	KNeighbours	0.143
Message Emolex	SVRLinear	0.143
Text Depechemood	SVRLinear	0.143
Title Emolex	SVRRbf	0.143
Message Emolex	SVRRbf	0.143
Text Depechemood	SVRPoly	0.143
Title Emolex	SVRLinear	0.143
Title Depechemood	SVRLinear	0.143
Title Glove	SVRPoly	0.143
Message Glove	SVRPoly	0.143
Message Pos	ExtraTrees	0.143
Text Glove	SVRPoly	0.143
Message Depechemood	SVRLinear	0.143
Text Tfidf Bow 1000	SVRPoly	0.143
Title Tfidf Bow 1000	SVRPoly	0.143
Message Tfidf Bow 1000	SVRPoly	0.143
Title Pos	ExtraTrees	0.144
Message Pos	KNeighbours	0.144
Title Readability	KNeighbours	0.145
Message Readability	SVRPoly	0.145
Title Pos	SVRPoly	0.145
Message Emolex	DecisionTree	0.146
Message Pos	SVRPoly	0.146
Text Readability	SVRPoly	0.147
Message Readability	KNeighbours	0.147
Message Readability	RandomForest	0.148
Title Readability	RandomForest	0.150
Message Readability	ExtraTrees	0.155
Text Pos	SVRPoly	0.156
Title Readability	ExtraTrees	0.159
Text Tfidf Bow 1000	DecisionTree	0.161
Title Pos	DecisionTree	0.172
Text Glove	DecisionTree	0.174
Title Tfidf Bow 1000	DecisionTree	0.176
Message Tfidf Bow 1000	DecisionTree	0.180
Title Glove	DecisionTree	0.181
Text Pos	DecisionTree	0.182
Title Readability	DecisionTree	0.184
Message Glove	DecisionTree	0.185
Message Pos	DecisionTree	0.186
Text Emolex	DecisionTree	0.186
Text Depechemood	DecisionTree	0.188
Text Readability	DecisionTree	0.190
Message Readability	DecisionTree	0.191
Title Pos	SVRSigmoid	58.103
Message Pos	SVRSigmoid	59.637
Text Pos	SVRSigmoid	75.796
Title Emolex	SVRSigmoid	81.402
Message Emolex	SVRSigmoid	137.247
Message Depechemood	SVRSigmoid	141.578
Title Depechemood	SVRSigmoid	155.222
Text Depechemood	SVRSigmoid	173.750
Text Emolex	SVRSigmoid	215.951
Title Readability	SVRSigmoid	386.197
Text Readability	SVRSigmoid	510.709
Message Readability	SVRSigmoid	545.844

Detaillierte Ergebnisse zum Vergleich der Features (Englisch)

Feature	Regressor	RMSE
Text Tfidf Bow 1000	Ridge	0.098
Text Tfidf Bow 1000	LinearRegression	0.100
Text Glove	GradientBoosting	0.101
Text Tfidf Bow 1000	GradientBoosting	0.102
Text Glove	Ridge	0.102
Text Glove	LinearRegression	0.102
Text Glove	KNeighbours	0.105
Text Tfidf Bow 1000	SVRLinear	0.106
Text Depechemood	GradientBoosting	0.106
Title Tfidf Bow 1000	Ridge	0.106
Text Glove	ExtraTrees	0.107
Text Tfidf Bow 1000	RandomForest	0.107
Text Depechemood	Ridge	0.108
Text Depechemood	LinearRegression	0.108
Title Tfidf Bow 1000	LinearRegression	0.108
Title Glove	Ridge	0.108
Title Glove	LinearRegression	0.108
Title Glove	GradientBoosting	0.109
Text Glove	RandomForest	0.109
Message Tfidf Bow 1000	Ridge	0.109
Text Tfidf Bow 1000	ExtraTrees	0.110
Title Tfidf Bow 1000	GradientBoosting	0.110
Message Glove	GradientBoosting	0.110
Message Glove	Ridge	0.110
Message Glove	LinearRegression	0.111
Message Tfidf Bow 1000	LinearRegression	0.111
Text Glove	SVRLinear	0.111
Text Tfidf Bow 1000	KNeighbours	0.111
Text Pos	GradientBoosting	0.111
Text Depechemood	RandomForest	0.112
Message Tfidf Bow 1000	GradientBoosting	0.112
Text Emolex	GradientBoosting	0.112
Text Pos	Ridge	0.112
Text Pos	LinearRegression	0.112
Text Depechemood	ExtraTrees	0.112
Text Depechemood	KNeighbours	0.113
Message Depechemood	GradientBoosting	0.113
Title Depechemood	GradientBoosting	0.114
Title Tfidf Bow 1000	SVRLinear	0.114
Text Emolex	Ridge	0.114
Text Emolex	LinearRegression	0.114
Message Depechemood	Ridge	0.114
Message Depechemood	LinearRegression	0.114
Title Pos	GradientBoosting	0.114
Title Pos	LinearRegression	0.115
Title Pos	Ridge	0.115
Title Depechemood	Ridge	0.115
Title Depechemood	LinearRegression	0.115
Text Readability	GradientBoosting	0.115
Message Pos	GradientBoosting	0.116
Title Tfidf Bow 1000	KNeighbours	0.116
Title Glove	KNeighbours	0.116
Title Readability	Ridge	0.116
Title Readability	LinearRegression	0.116
Title Emolex	GradientBoosting	0.116
Title Glove	ExtraTrees	0.116
Message Pos	Ridge	0.116
Message Pos	LinearRegression	0.116
Title Glove	SVRLinear	0.116
Message Readability	Ridge	0.117
Message Readability	LinearRegression	0.117
Message Glove	KNeighbours	0.117
Title Readability	GradientBoosting	0.117
Title Tfidf Bow 1000	ExtraTrees	0.117
Message Emolex	GradientBoosting	0.117
Message Readability	GradientBoosting	0.117
Message Tfidf Bow 1000	SVRLinear	0.117
Title Emolex	Ridge	0.117
Title Emolex	LinearRegression	0.117
Title Tfidf Bow 1000	RandomForest	0.117
Message Emolex	Ridge	0.117
Message Emolex	LinearRegression	0.117
Text Emolex	ExtraTrees	0.118
Text Pos	RandomForest	0.118
Text Pos	ExtraTrees	0.118
Text Depechemood	SVRRbf	0.118
Title Tfidf Bow 1000	Baseline	0.118
Title Glove	RandomForest	0.118
Message Tfidf Bow 1000	ExtraTrees	0.118
Text Readability	LinearRegression	0.119
Message Glove	ExtraTrees	0.119
Text Emolex	RandomForest	0.119
Text Readability	Ridge	0.119
Message Glove	SVRLinear	0.119
Title Depechemood	RandomForest	0.119
Message Depechemood	ExtraTrees	0.119
Message Glove	RandomForest	0.120
Message Tfidf Bow 1000	RandomForest	0.120
Message Depechemood	RandomForest	0.120
Title Depechemood	ExtraTrees	0.120
Text Emolex	KNeighbours	0.120
Text Pos	KNeighbours	0.120
Text Glove	SVRRbf	0.120
Title Glove	SVRRbf	0.120
Text Depechemood	SVRLinear	0.121
Title Glove	SVRSigmoid	0.121
Text Glove	SVRSigmoid	0.122
Message Depechemood	KNeighbours	0.122
Text Pos	SVRRbf	0.122
Title Depechemood	KNeighbours	0.123
Message Glove	SVRRbf	0.123
Text Emolex	SVRRbf	0.123
Text Readability	ExtraTrees	0.124
Text Readability	RandomForest	0.124
Message Glove	SVRSigmoid	0.124
Message Pos	ExtraTrees	0.124
Title Pos	KNeighbours	0.124
Message Pos	RandomForest	0.125
Message Depechemood	SVRRbf	0.125

Text Readability	KNeighbours	0.125
Title Pos	SVRRbf	0.125
Message Tfidf Bow 1000	KNeighbours	0.125
Text Pos	SVRLinear	0.125
Text Tfidf Bow 1000	SVRRbf	0.125
Text Readability	SVRRbf	0.125
Title Pos	RandomForest	0.126
Title Emolex	RandomForest	0.126
Title Emolex	KNeighbours	0.126
Text Emolex	SVRLinear	0.126
Message Depechemood	SVRLinear	0.126
Title Depechemood	SVRRbf	0.126
Title Depechemood	SVRLinear	0.127
Text Tfidf Bow 1000	SVRSigmoid	0.127
Title Pos	SVRLinear	0.127
Message Pos	KNeighbours	0.127
Message Pos	SVRRbf	0.127
Title Readability	SVRRbf	0.127
Title Readability	KNeighbours	0.127
Message Readability	SVRRbf	0.127
Message Emolex	SVRRbf	0.128
Title Tfidf Bow 1000	SVRRbf	0.128
Title Readability	SVRLinear	0.128
Title Readability	SVRPoly	0.128
Text Readability	SVRLinear	0.128
Title Emolex	SVRRbf	0.128
Message Pos	SVRLinear	0.128
Message Tfidf Bow 1000	SVRRbf	0.128
Message Readability	SVRLinear	0.128
Title Emolex	SVRLinear	0.128
Message Emolex	RandomForest	0.128
Message Readability	SVRPoly	0.128
Message Readability	KNeighbours	0.128
Title Tfidf Bow 1000	SVRSigmoid	0.128
Message Readability	RandomForest	0.128
Message Emolex	SVRLinear	0.128
Message Tfidf Bow 1000	SVRSigmoid	0.129
Title Glove	SVRPoly	0.129
Message Glove	SVRPoly	0.129
Text Glove	SVRPoly	0.129
Title Tfidf Bow 1000	SVRPoly	0.129
Text Tfidf Bow 1000	SVRPoly	0.129
Message Tfidf Bow 1000	SVRPoly	0.129
Message Emolex	KNeighbours	0.130
Title Emolex	ExtraTrees	0.130
Message Emolex	ExtraTrees	0.130
Title Pos	ExtraTrees	0.131
Title Emolex	SVRPoly	0.131
Title Depechemood	SVRPoly	0.131
Message Readability	ExtraTrees	0.133
Title Readability	RandomForest	0.134
Text Emolex	SVRPoly	0.135
Title Pos	SVRPoly	0.135
Title Readability	ExtraTrees	0.142
Title Emolex	DecisionTree	0.144
Text Pos	SVRPoly	0.147
Text Tfidf Bow 1000	DecisionTree	0.148
Message Pos	SVRPoly	0.150
Text Depechemood	DecisionTree	0.150
Text Glove	DecisionTree	0.154
Message Depechemood	SVRPoly	0.156
Title Depechemood	DecisionTree	0.157
Message Emolex	DecisionTree	0.157
Title Tfidf Bow 1000	DecisionTree	0.158
Title Glove	DecisionTree	0.158
Title Pos	DecisionTree	0.160
Text Pos	DecisionTree	0.160
Text Emolex	DecisionTree	0.161
Message Tfidf Bow 1000	DecisionTree	0.161
Title Readability	DecisionTree	0.161
Message Glove	DecisionTree	0.163
Message Depechemood	DecisionTree	0.163
Message Emolex	SVRPoly	0.165
Message Pos	DecisionTree	0.165
Text Readability	DecisionTree	0.166
Message Readability	DecisionTree	0.170
Text Depechemood	SVRPoly	0.171
Text Readability	SVRPoly	6.729
Title Pos	SVRSigmoid	25.778
Message Pos	SVRSigmoid	38.633
Text Pos	SVRSigmoid	71.948
Title Depechemood	SVRSigmoid	99.468
Title Emolex	SVRSigmoid	162.338
Message Emolex	SVRSigmoid	166.185
Message Depechemood	SVRSigmoid	167.891
Text Emolex	SVRSigmoid	201.161
Title Readability	SVRSigmoid	290.321
Text Depechemood	SVRSigmoid	323.111
Text Readability	SVRSigmoid	431.943
Message Readability	SVRSigmoid	437.328

B Abkürzungsverzeichnis

BOW Bag-of-Words

GloVe Global Vectors for Word Representation

LSI Latent Semantic Indexing

ML Machine Learning, Maschinelles Lernen

NLP Natural Language Processing

POS Part-of-Speech

SVR Support Vector Regression

TF-IDF Termfrequenz - inverse Dokumentfrequnz

Literatur

[Alt92] ALTMAN, N. S.: An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. In: *The American Statistician* 46 (1992), Nr. 3, S. 175–185

[BEPW08] BACKHAUS, Klaus ; ERICHSON, Bernd ; PLINKE, Wulff ; WEIBER, Rolf: *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung*. 12., vollständig überarbeitete Auflage. Springer, Berlin, 2008

[BFOS84] BREIMAN, Leo ; FRIEDMAN, J. H. ; OLSHEN, R. A. ; STONE, C. J.: *Classification and Regression Trees*. Wadsworth Publishing Company, 1984 (Statistics/Probability Series)

[BH16] BUECHEL, Sven ; HAHN, Udo: Emotion Analysis as a Regression Problem - Dimensional Models and Their Implications on Emotion Representation and Metrical Evaluation. In: *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, 2016, S. 1114–1122

[BH17] BUECHEL, Sven ; HAHN, Udo: EMOBANK: Studying the Impact of Annotation Perspective and Representation Format on Dimensional Emotion Analysis. In: *EACL 2017* (2017), S. 578

[Bre01] BREIMAN, Leo: Random Forests. In: *Mach. Learn.* 45 (2001), Oktober, Nr. 1, S. 5–32

[BV84] BAMBERGER, R. ; VANECEK, E.: *Lesen-Verstehen-Lernen-Schreiben: die Schwierigkeitsstufen von Texten in deutscher Sprache*. Jugend und Volk, 1984

[CEE+09] CARSTENSEN, Kai-Uwe ; EBERT, Christian ; EBERT, Cornelia ; JEKAT, Susanne ; KLABUNDE, Ralf ; LANGER, Hagen: *Computerlinguistik und Sprachtechnologie: Eine Einführung*. 3. Spektrum Akademischer Verlag, 2009

[DY14] DENG, Li ; YU, Dong: Deep Learning: Methods and Applications. (2014), May

[Ekm92] EKMAN, Paul: An Argument for Basic Emotions. In: *Cognition and Emotion* 6 (1992), Nr. 3-4, S. 169–200

[ELLS11] EVERITT, Brian ; LANDAU, Sabine ; LEESE, Morven ; STAHL, Daniel: *Cluster analysis*. 5th. Wiley, 2011

[Fri00] FRIEDMAN, Jerome H.: Greedy Function Approximation: A Gradient Boosting Machine. In: *Annals of Statistics* 29 (2000), S. 1189–1232

[GEW06] GEURTS, Pierre ; ERNST, Damien ; WEHENKEL, Louis: Extremely randomized trees. In: *Machine Learning* 63 (2006), Nr. 1, S. 3–42

[Gun52] GUNNING, R.: *The Technique of Clear Writing*. McGraw-Hill, 1952

[HJ15] HONNIBAL, Matthew ; JOHNSON, Mark: An Improved Non-monotonic Transition System for Dependency Parsing. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, September 2015, S. 1373–1378

[HK06] HYNDMAN, Rob J. ; KOEHLER, Anne B.: Another look at measures of forecast accuracy. In: *International Journal of Forecasting* (2006), S. 679–688

[HL04] HU, Minqing ; LIU, Bing: Mining and Summarizing Customer Reviews. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2004 (KDD '04), S. 168–177

[HRA] HASAN, Maryam ; RUNDENSTEINER, Elke ; AGU, Emmanuel: EMOTEX: Detecting Emotions in Twitter Messages. In: *2014 ASE BigData/SocialCom/CyberSecurity*

[JM00] JURAFSKY, Daniel ; MARTIN, James H.: *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 1st. Prentice Hall PTR, 2000

[KI06] KENNEDY, Alistair ; INKPEN, Diana: Sentiment classification of movie reviews using contextual valence shifters. In: *Computational intelligence* 22 (2006), Nr. 2, S. 110–125

[KLC17] KAHLERT, Roland ; LIEBECK, Matthias ; CORNELIUS, Joseph: Understanding Trending Topics in Twitter. In: *Datenbanksysteme für Business, Technologie und Web (BTW 2017), Workshopband*, 2017, S. 375–384

[KP98] KOHAVI, Ron ; PROVOST, Foster: Glossary of terms. In: *Machine Learning* 30 (1998), S. 271–274

[LC15] LIEBECK, Matthias ; CONRAD, Stefan: IWNLP: Inverse Wiktionary for Natural Language Processing. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Association for Computational Linguistics, 414–418

[Lov68] LOVINS, Julie B.: Development of a stemming algorithm. In: *Mechanical Translation and Computational Linguistics* 11 (1968), S. 22–31

[Mil95] MILLER, George A.: WordNet: A Lexical Database for English. In: *Commun. ACM* 38 (1995), November, Nr. 11, S. 39–41. – ISSN 0001–0782

[MRS08] MANNING, Christopher D. ; RAGHAVAN, Prabhakar ; SCHÜTZE, Hinrich: *Introduction to Information Retrieval*. Cambridge University Press, 2008

[MSC+13] MIKOLOV, Tomas ; SUTSKEVER, Ilya ; CHEN, Kai ; CORRADO, Greg S. ; DEAN, Jeff: Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*, 2013, S. 3111–3119

[MT13] MOHAMMAD, Saif M. ; TURNEY, Peter D.: Crowdsourcing a Word-Emotion Association Lexicon. 29 (2013), Nr. 3, S. 436–465

[PDM12] PETROV, Slav ; DAS, Dipanjan ; MCDONALD, Ryan: A Universal Part-of-Speech Tagset. In: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, European Language Resources Association (ELRA), may 2012

[Plu80] PLUTCHIK, Robert: A general psychoevolutionary theory of emotion. In: *Emotion: Theory, Research, and Experience* 1 (1980), S. 3–31

[Por01] PORTER, M. F.: Snowball: A language for stemming algorithms. (2001)

[PSM14] PENNINGTON, Jeffrey ; SOCHER, Richard ; MANNING, Christopher D.: GloVe: Global Vectors for Word Representation. In: *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543

[PVG+11] PEDREGOSA, F. ; VAROQUAUX, G. ; GRAMFORT, A. ; MICHEL, V. ; THIRION, B. ; GRISEL, O. ; BLONDEL, M. ; PRETTENHOFER, P. ; WEISS, R. ; DUBOURG, V. ; VANDERPLAS, J. ; PASSOS, A. ; COURNAPEAU, D. ; BRUCHER, M. ; PERROT, M. ; DUCHESNAY, E.: Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research* 12 (2011), S. 2825–2830

[Qui86] QUINLAN, J. R.: Induction of Decision Trees. In: *MACH. LEARN* 1 (1986), S. 81–106

[RM77] RUSSELL, James A. ; MEHRABIAN, Albert: Evidence for a three-factor theory of emotions. In: *Journal of Research in Personality* 11 (1977), September, Nr. 3, S. 273–294. – ISSN 00926566

[San90] SANTORINI, Beatrice: Part-of-speech tagging guidelines for the Penn Treebank Project. (1990), Nr. MS-CIS-90-47

[SG14] STAIANO, Jacopo ; GUERINI, Marco: DepecheMood: a Lexicon for Emotion Analysis from Crowd-Annotated News. In: *CoRR* abs/1405.1605 (2014)

[TGD+17] TIAN, Ye ; GALERY, Thiago ; DULCINATI, Giulio ; MOLIMPAKIS, Emilia ; SUN, Chao: Facebook Sentiment: Reactions and Emojis, 2017

[Tik43] TIKHONOV, Andrey N.: On the stability of inverse problems. In: *Doklady Akademii Nauk SSSR* 39 (1943), Nr. 5, S. 195–198

[Vap95] VAPNIK, Vladimir N.: *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., 1995

[WFH11] WITTEN, Ian H. ; FRANK, Eibe ; HALL, Mark A.: *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd. Morgan Kaufmann Publishers Inc., 2011

[WSC+16] 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, Xiaobing ; KAISER, Lukasz ; 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: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. In: *CoRR* abs/1609.08144 (2016)

Abbildungsverzeichnis

Tabellenverzeichnis