INSTITUT FÜR INFORMATIK
Datenbanken und Informationssysteme

Universitätsstr. 1 D–40225 Düsseldorf

Heinrich Heine Universität Düsseldorf

Klassifikation von Tweets zur Erkennung von Trollen

Robin Kösters

Bachelorarbeit

Beginn der Arbeit:	18. September 2020
Abgabe der Arbeit:	18. Dezember 2020
Gutachter:	Prof. Dr. Stefan Conrad
	Prof. Dr. Martin Mauve

Erklärung

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

Düsseldorf, den 18. Dezember 2020

Robin Kösters

Zusammenfassung

Hier kommt eine ca. einseitige Zusammenfassung der Arbeit rein.

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

Die US-Präsidentschaftswahl 2016 gilt vielen politischen Beobachtern als eindrucksvollstes Beispiel dafür, wie von staatlicher Seite organisierte Desinformationskampagnen den politischen Diskurs nachhaltig verzerren können. Bereits zwei Jahre nach der Wahl kamen britische Wissenschaftler in Zusammenarbeit mit der Firma Graphika zu der Erkenntnis, dass ein Unternehmen mit dem Namen „Internet Research Agency“ (IRA), welches dem russischen Staat nahesteht, im großen Stil versucht hat, amerikanische Wähler via Fake-Postings in sozialen Netzwerken zu beeinflussen. In ihrem Bericht geben die Forscher an, dass mehr als 30 Millionen Benutzer im Zeitraum von 2015 bis 2017 in Berührung mit von der IRA erstellten Inhalten gekommen sind. Die Urheber jener destruktiver Inhalte werden Trolle genannt. (Howard et al., 2019)
In Twitter, welches das hauptsächlich betrachtete soziale Netzwerk in dieser Arbeit sein soll, machen Trolle besonders von der Hashtag-Setzung Gebrauch. Bei einem *Hashtag* handelt es sich um eine durch den Ersteller eines Beitrags vorgenommene Themenfestlegung bzw. -klassifizierung. Eine nicht durch Leerzeichen unterbrochene Zeichenkette wird durch Voranstellen eines Rautezeichens (engl. *hash*) zu einem Hashtag. In der Folge kann ein *Tweet* (plattformeigene Bezeichnung für einen Beitrag) durch das Benutzen der allgemeinen Suchfunktion gefunden werden. Abbildung 1 zeigt einen beispielhaften Tweet der IRA, welcher dem der Arbeit zugrundeliegenden Datensatz entnommen wurde und die Hashtag-Nutzung durch einen Troll illustriert.

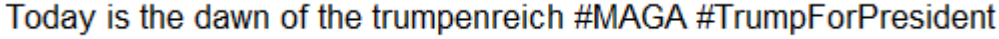

Abbildung 1: Beispiel eines IRA-Tweets

Kommt es innerhalb eines gewissen Zeitraums zur gehäuften Benutzung eines bestimmten Hashtags, so wird dieser in den „Twitter-Trends“, eine Art Ranking der momentan meistgenutzten Hashtags, aufgeführt. Dies macht die Aktualität bzw. Relevanz eines gesellschaftlichen Themas direkt ablesbar. Für Trolle bietet dies aber die Möglichkeit durch geschickte zeitliche Abstimmung bestimmte Hashtags zu Trend-Hashtags zu machen und den Nutzern so aktiv die Relevanz bestimmter Themen vorzutäuschen.
Insgesamt sind mit den genannten Methoden verschiedenste Arten der Manipulation seitens Trollen denkbar. So können Trolle eine politische Partei oder einen Kandidaten unterstützen und versuchen, ihren Mitbewerbern zu schaden. Beispiele dafür gibt es auch in Deutschland. Propagandaforscherin Lisa-Maria Neudert erwähnt in einem Interview mit Deutschlandfunk Kultur, dass das ultrarechte Netzwerk „Reconquista Germanica“ beabsichtigte, „die AfD so gut wie möglich zu verstärken [...] “, und dass seine Trolle die Partei im Endeffekt „größer erscheinen lassen als sie ist“ (Jabs, 2017). Ein weitere tragende Säule von Desinformationskampagnen, welche meist in Verbindung zu Trollen steht, sind die *Fake News*. Hierbei handelt es sich um Falschmeldungen bzw. Nachrichten ohne Wahrheitsgehalt, welche verbreitet werden, um die Öffentlichkeit zu manipulieren. Trolle treten hier in der Regel wechselweise als Urheber und Multiplikatoren solcher Meldungen auf.

Alle bisher beschriebenen Strategien sind gemeinsam als integrative Gesamtstrategie dazu geeignet, um die Ausgänge demokratischer Wahlen zu beeinflussen. Die Initiative „ichbinhier e.V." und das Londoner „Institute for Strategic Dialogue" (ISD) kamen in einer Studie beispielsweise zu der Schlüsselerkenntis, dass rechtsextreme Trollnetzwerke im Bundestagswahlkampf 2017 Urheber einer ausgedehnten und erfolgreichen „pro-AfD-Wahlkampagne" waren (Kreißel et al., 2018). Das Ziel der Desinformationskampagnen, gleich ob aus dem Inland oder Ausland gesteuert, ist die Destabilisierung der demokratischen Institutionen bzw. der Demokratie als Ganzes. Aus diesem Grund möchte ich in dieser Arbeit meinen Teil dazu beitragen, dass eine Lösung für dieses Problem gefunden wird.

1.2 Datensätze

Im Rahmen dieser Arbeit soll ein Verfahren entwickelt werden, welches die zuverlässige Erkennung von Troll-Inhalten auf Twitter ermöglicht. Hierbei kommen verschiedenste Techniken der Textklassifikation zum Einsatz. Diese werden auf zwei Datensätze angewandt, auf deren Eigenschaften und Ursprünge an dieser Stelle eingegangen werden soll. Die Grundmenge des ersten Datensatzes ist eine von Linvill und Warren (2018) veröffentlichte Sammlung von rund 3 Millionen Tweets der IRA. Hier wurden die Tweets mit den zehn häufigsten Hashtags (siehe Tabelle 1) entnommen.

Platz	Hashtag	Platz	Hashtag
1	#news	6	#topNews
2	#sports	7	#MAGA
3	#politics	8	#BlackLivesMatter
4	#world	9	#health
5	#local	10	#tcot

Tabelle 1: Hashtags aus Datensatz 1

Beim zweiten Datensatz handelt es sich um im Vorfeld eigens extrahierte Tweets von echten Profilen. Damit thematische Ähnlichkeit besteht wurden nur Tweets, welche die zehn Hashtags aus dem anderen Datensatz enthalten, ausgewählt. Es wurde streng darauf geachtet, dass die Datensätze disjunkt sind. Tabelle 2 zeigt zum Vergleich die wichtigsten Kennzahlen beider Datensätze.

Eigenschaft	Troll	Nichttroll
Anzahl Tweets	303.036	324.873
unterschiedliche Autoren	770	68.706
Zeitraum	01/2015 - 09/2017	01/15 - 05/2018
durchschnittliche Länge	78,64 Zeichen	122,27 Zeichen

Tabelle 2: Vergleich der Datensätze

2 Grundlagen des Text Mining

Im nachfolgenden Kapitel werden jene Konzepte des *Text Minings* erläutert, welche grundlegend für das Verständnis von Methoden der Textklassifikation sind. Abbildung 2 zeigt die Phasen, die beginnend beim Textkorpus (Sammlung der Ausgangstexte) bis zur abschließenden Einstufung durchlaufen werden. Hieran werde ich mich bei den Ausführungen orientieren.

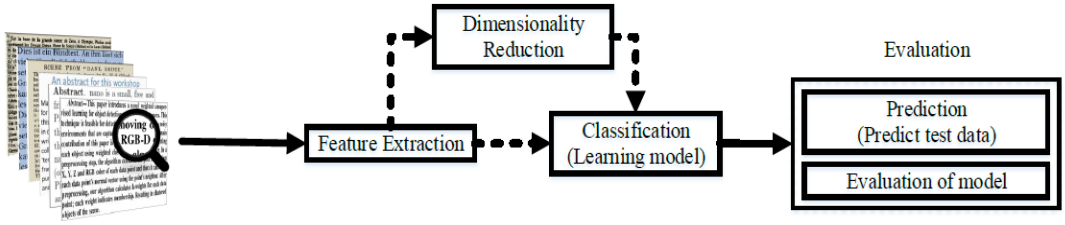

Abbildung 2: Pipeline der Textklassifikation (Kowsari et al., 2019)

2.1 Preprocessing

Der rohe, unbehandelte Text wird beim *Preprocessing* (deutsch: Vorbehandlung) in eine Form überführt, die eine nachfolgende Repräsentation der Daten durch Vektoren und ähnliche Methoden ermöglicht. Je nach gewünschter Anwendung sind hier verschiedene Vorverarbeitungsverfahren in Betracht zu ziehen.
Der Begriff Wortsegmentierung bzw. **Tokenisierung** bezeichnet die Zerteilung des Ausgangstexts in kleinere Einheiten, sogenannte *Token* (Manning und Schütze, 1999). In der Regel handelt es sich hierbei um einzelne Wörter, Zahlen und Satzzeichen. Als grundsätzliches Abgrenzungszeichen gelten in den meisten Sprachen die *Whitespace*-Zeichen (Leerzeichen, Tab, Newline-Zeichen). Dies wird auch auf die zu analysierenden Tweets zutreffen, da diese fast ausschließlich in englischer Sprache verfasst sind.

Er	rief	:	„	Baum	fällt	!	“

Abbildung 3: Tokenisierung eines Beispielsatzes

Eine weitere Methode der Vorverarbeitung ist das Entfernen von **Stoppwörtern**. Dies sind sehr häufig auftretende Wörter wie „der“, „die“, „das“, „und“ oder „von“, welche vornehmlich eine grammatikalische Funktion und keinen Informationsgehalt haben. (vgl. ebd., S. 533)
Bei der **Lemmatisierung** (Airio, 2006) werden mehrere Wörter unterschiedlicher Erscheinungsform, welche aber die gleiche Bedeutung haben, auf eine gemeinsame Grundform

(genannt Lemma) zurückgeführt. So lassen sich beispielsweise die deklinierten Substantive „Wortes“, „Wörter“, „Wörtern“auf „Wort“ zurückführen, während die Grundform der konjugierten Verben „schrieb“, „schreibe“, „schreibst“ und „schreibt“ der Infinitiv „schreiben“ist. Ein verwandtes Konzept wird **Stemming** (Airio, 2006) genannt. Der Unterschied zur Lemmatisierung besteht darin, dass das daraus resultierende Grundwort (hier: Stamm) kein natürlichsprachliches Wort sein muss, sondern in der Regel ein um Präfix und Suffix beschnittes Wort ist. Ein Beispiel hierfür ist die Reduzierung der Wörter „gehen“, „umgehen“und „zugehen“ auf den Stamm „geh-“.
Für die Lemmatisierung eines Textes wird auch das **Part-of-Speech (POS) Tagging** (Kumawat und Jain, 2015) benötigt. Hierbei wird einem Token seine Wortart (z.B. Nomen, Verb, Adjektiv) oder ein anderes Label wie „Interpunktion“ zugewiesen. Werkzeuge, die dieses Verfahren anwenden, werden POS-*Tagger* genannt.

2.2 Feature Extraction

Bevor eine Klassifikation vorgenommen werden kann, müssen die Merkmale aus den vorliegenden Texten gewonnen und mit mathematischen Methoden strukturiert bzw. modelliert werden. Hierbei fällt die Wahl zumeist auf Vektorisierung.
Die einfachste Vektorisierungstechnik ist die **Bag-of-Words** (BoW) (Ramos, 2003). Hier wird ein Text durch einen Vektor mit den Begriffshäufigkeiten (auch: *Term Frequencies* (**TF**)) aller zuvor extrahierten Tokens des Textkorpus repräsentiert. Bei einem Beispielkorpus mit den beiden Texten „my coffee is too hot“ und „my tea is too cold“ und dem zuvor extrahierten Vokabular

{ "my", "coffee", "hot", "tea", "cold"}

ergibt sich die folgende Repräsentation:

v_1 = [1, 1, 1, 0, 0]
v_2 = [1, 0, 0, 1, 1]

Eine Erweiterung dieser TF-Methode ist *Term Frequency-Inverse Document Frequency* (**TF-IDF**) (Ramos, 2003). Hier wird die Vorkommenshäufigkeit anders gewichtet, um dem Aspekt gerecht zu werden, dass einige Begriffe überproportional oft vorkommen. Gleichung 1 zeigt die Gewichtung

$$w(d,t) = TF(d,t) \cdot \log\left(\frac{N}{DF(t)}\right) \tag{1}$$

wobei N die Anzahl der Texte im Korpus und $DF(t)$ die Anzahl der Texte, welche den Begriff t enthalten, ist.
Eine Möglichkeit, Wortkombinationen bzw. bestimmte Formulierungen als Merkmal zu berücksichtigen ist das **N-Gramm** (Kowsari et al., 2019). Dies ist die Zusammenfassung von N aufeinanderfolgenden Token in einem Text. Folglich handelt es sich bei den Elementen einer Bag of Words um 1-Gramme. Das nachfolgende Beispiel zeigt die Repräsentation des Textes „Hier sehen Sie ein Beispiel.“ mit 2-Grammen:

{ “Hier sehen “, “sehen Sie“, “Sie ein“, “ein Beispiel“}

Alle vorausgegangenen Methoden der Merkmalsextraktion haben gemeinsam, dass sie keinen Aufschluss über Zusammenhänge der Wortbedeutungen (Semantik) geben. Dieses Problem versuchen die Techniken der **Worteinbettung** zu lösen. Ein prominentes Beispiel ist **Word2Vec** (Mikolov et al., 2013). Die Grundidee ist hier, dass Begriffe, die eine ähnliche Bedeutung haben, in einem ähnlichen Kontext verwendet werden. Auf dieser Basis wird ein Vektor für jedes Wort im Vokabular durch ein neuronales Netz erzeugt. Die Entfernungen im daraus entstehenden Vektorraum geben dabei semantische Ähnlichkeiten wieder.

2.3 Dimensionalitätsreduktion

Bei sehr umfangreichen Datensätzen wie den mehr als 600.000 Tweets in dieser Arbeit werden in der Anwendung der zuvor beschriebenen Verfahren meist hochdimensionale Vektoren erzeugt. In der Folge werden viele Operationen bei späteren Algorithmen der Textklassifikation eine hohe Zeit- und Speicherkomplexität besitzen. Diesem Effekt versucht man im Voraus durch Dimensionalitätsreduktion entgegenzuwirken.
Ein erstes verwendetes Verfahren ist die Hauptkomponentenanalyse bzw. **Principal Component Analysis** (PCA) (Joliffe, 2002). Mit diesem ist es möglich, in der Punktwolke der vorhandenen Vektoren all jene Vektor-Komponenten herauszufinden, die für die größte Varianz verantwortlich sind, also den größten Informationsgehalt haben. Der mathematische Mechanismus dahinter ist die Hauptachsentransformation: Es wird eine Ladungsmatrix aus den Eigenvektoren der Kovarianzmatrix gebildet, aus welcher der Anteil der Varianz jeder Komponente an der Gesamtvarianz ersichtlich ist. In der Folge können Komponenten, welche wenig Varianz beitragen, ohne nennenswerten Informationsverlust verworfen werden.
Eine andere Möglichkeit ist die **Nichtnegative Matrixfaktorisierung** (NMF) (Lee und Seung, 1999). Hier wird aus den n Texten mit insgesamt m Wörtern eine $m \times n$ Matrix gebildet, welche approximativ so faktorisiert wird, dass

$$V \approx WH$$

gilt, wobei W eine $m \times r$ Matrix und H eine $r \times n$ Matrix ist. Die r Spalten von W enthalten semantisch verwandte Wörter, welche zusammen einen Kontext bzw. ein Thema bilden. Voraussetzung für dieses Verfahren ist die Nichtnegativität von V.

3 Klassifikationsverfahren

Nachdem die Tweets vorbearbeitet, die Merkmale gewonnen und die Dimensionen zwecks verbesserter Performance reduziert wurden, können sie nun als Trainingsdatensatz für ein Klassifikationsverfahren verwendet werden. Im nachfolgenden Kapitel werden verschiedene solcher Verfahren vorgestellt und im Hinblick auf Voraussetzungen, Stärken und Schwächen analysiert. Ein besonderes Augenmerk soll in der Analyse auf die Kombination zwischen Algorithmus und den bereits erwähnten Methoden der Merkmalsextraktion gelegt werden, da erwartet wird, dass unterschiedliche Kombinationen sich qualitativ unterscheiden werden.

3.1 k-Nearest-Neighbor-Algorithmus

Der k-Nearest-Neighbor-Algorithmus (KNN) ist eine parameterfreie Methode der Klassifikation (Quelle). Der zu klassifizierende Text wird zunächst analog zu den Trainingstexten vektorisiert. Über ein geeignetes Abstandsmaß werden nun die k räumlich nächsten Nachbarn bestimmt. Der Text wird nun der Klasse zugeordnet, der die Mehrheit der Nachbarn angehören. Abbildung 4 zeigt die k-Nearest-Neighbor-Klassifikation eines Punktes x für $k = 3$. Da die meisten Nachbarn hier dem Troll-Datensatz angehören, würde der zu x gehörige Tweet somit als Troll-Tweet klassifiziert werden.
Die Wahl von k ist entscheidend für die Qualität des Ergebnisses. Entscheidet man sich beispielsweise für ein zu kleines k, so ist möglich, dass vereinzelte Ausreißer die Genauigkeit trüben. Ist es auf der anderen Seite zu groß, so werden wahrscheinlich zu weit entfernte Punkte bei der Klassifikation miteinbezogen, was das Ergebnis wiederum verfälschen kann. Ferner ist bei Vorhandensein von 2 Klassen ein ungerades k zu wählen, da andernfalls ein Unentschieden möglich ist.

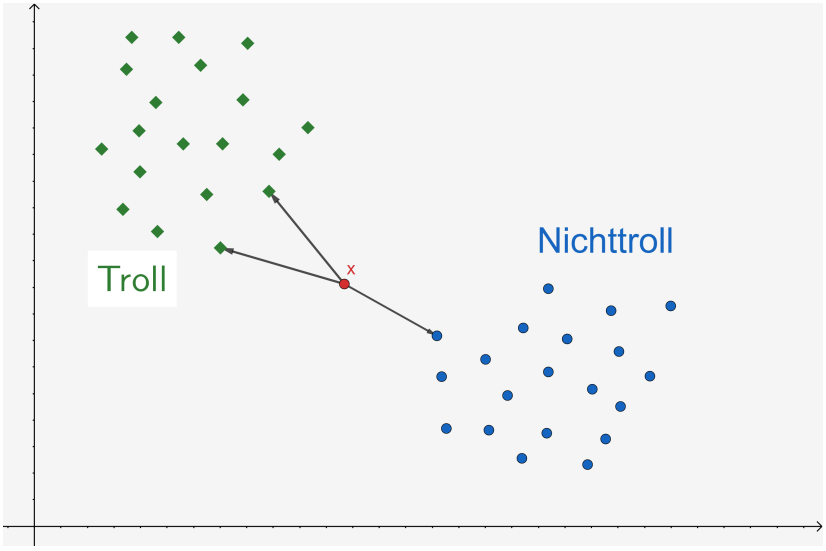

Abbildung 4: KNN-Klassifizierung für $k = 3$

Eine zwingende Voraussetzung, um zuverlässig mit dem KNN-Algorithmus in diesem Projekt arbeiten zu können, ist die Dimensionalitätsreduktion. Die Merkmalsextraktion bringt bei den mehr als 600.000 Tweets hochdimensionale Vektoren hervor. Unbehandelt wären aufgrund der komponentenweisen Abstandsmessung Laufzeiten von einigen Minuten zu erwarten.
Die Vorteile dieses Verfahrens sind die einfache Implementierung und seine Eignung für alle möglichen Ausprägungen von Merkmalsräumen. Als Schwächen werden die bereits angesprochenen Probleme mit der Performance angesehen.

3.2 Naiver Bayes-Klassifikator

Der Naive Bayes-Klassifikator (NB) ist ein statistisches Verfahren der Klassifikation. Die Grundlage der Berechnung bildet hier der Satz von Bayes, bekannt aus der Wahrscheinlichkeitstheorie. In seiner herkömmlichen Interpretation beschreibt dieser die Berechnung der Wahrscheinlichkeit, dass ein Ereignis dem anderen vorausgegangen ist. Wendet man dies auf einen gegebenen Text $t \in T$ und die Klasse $k_i \in K$ an, so erhält man die Formel für die Wahrscheinlichkeit, dass t der Klasse k_i angehört (siehe Gleichung 2).

$$P(k_i|t) = \frac{P(t|k_i) \cdot P(k_i)}{P(t)} \tag{2}$$

Der Klassifikator bestimmt nun diejenige Klasse k_i, für die der Wert dieser Formel maximal ist. Mathematisch formuliert:

$$k = arg \max_{k_i \in K} P(k_i|t) = arg \max_{k_i \in K} \frac{P(t|k_i) \cdot P(k_i)}{P(t)} \tag{3}$$

3.3 Support Vector Machine

Lorem ipsum dolor sit amet.

3.4 Weiteres Verfahren

Lorem ipsum dolor sit amet.

4 Evaluation

Lorem ipsum dolor sit amet.

4.1 Gütekriterien

Lorem ipsum dolor sit amet.

4.2 Ergebnisse

Lorem ipsum dolor sit amet.

Literatur

Eija Airio (2006). „Word normalization and decompounding in mono-and bilingual IR“. In: *Information Retrieval* 9.3, S. 2–3.

Philip N Howard, Bharath Ganesh, Dimitra Liotsiou, John Kelly und Camille François (2019). *The IRA, social media and political polarization in the United States, 2012-2018*, S. 3.

Thorsten Jabs (2017). *„AfD hat 30 Prozent des Social-Media-Traffics ausgemacht“*. URL: https://www.deutschlandfunkkultur.de/rueckblick-auf-den-wahlkampf-im-netz-afd-hat-30-prozent-des.1008.de.html?dram:article_id=396594.

I.T. Joliffe (2002). *Principal Component Analysis*. 2. Aufl. Springer Series in Statistics. Springer.

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes und Donald Brown (2019). „Text Classification Algorithms: A Survey“. In: *Information* 10.4.

Philip Kreißel, Julia Ebner, Alexander Urban und Jakob Guhl (Juli 2018). *Hass auf Knopfdruck*. Rechtsextreme Trollfabriken und das Ökosystem koordinierter Hasskampagnen im Netz. URL: http://www.isdglobal.org/wp-content/uploads/2018/07/ISD_Ich_Bin_Hier_2.pdf.

Deepika Kumawat und Vinesh Jain (Mai 2015). „POS Tagging Approaches: A Comparison“. In: *International Journal of Computer Applications* 118.6, S. 32.

Daniel D. Lee und H. Sebastian Seung (1999). „Learning the parts of objects by non-negative matrix factorization“. In: *Nature* 401, S. 788–791.

Darren L. Linvill und Patrick L. Warren (2018). *Troll Factories: The Internet Research Agency and State-Sponsored Agenda Building*.

Christopher D. Manning und Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, S. 124–125.

Tomas Mikolov, Kai Chen, Greg Corrado und Jeffrey Dean (2013). „Efficient Estimation of Word Representations in Vector Space“. In: arXiv: 1301.3781.

Juan Ramos (2003). *Using TF-IDF to Determine Word Relevance in Document Queries*.

Abbildungsverzeichnis

Tabellenverzeichnis