Merkmalsextraktion bei Nachrichtenartikeln zur Themenklassifikation am Beispiel von Identitätsdatendiebstahl

Kurzfassung Bachelorarbeit

ausgearbeitet von

Gina Caroline Muuss

3104947

vorgelegt an der

Rheinische Friedrich-Wilhelms-Universität Bonn

Institut für Informatik IV

Arbeitsgruppe für IT-Sicherheit

im Studiengang

Informatik (B.Sc.)

Erstprüfer:	Dr. Felix Boes Universität Bonn
Zweitprüfer:	Prof. Dr. Peter Martini Universität Bonn
Betreuer:	Timo Malderle & Dr. Felix Boes Universität Bonn

Bonn, 30. August 2020

Einleitung

Dies ist eine Kurzfassung der Bachelorarbeit mit dem Titel „Merkmalsextraktion bei Nachrichtenartikeln zur Themenklassifikation am Beispiel von Identitätsdatendiebstahl“ [Muu20] deren Methoden und Ergebnisse in großen Teilen in das in Submission befindliche Paper „Credential Intelligence Agency - A Threat Intelligence Approach to Mitigate Identity Theft“ eingeflossen sind [Mal+on]. In besagter Publikation wurde ein System vorgestellt, dass es ermöglicht Informationen über Identitätsdatendiebstahl aus Nachrichtenartikeln für Analysten einfacher zugänglich zu machen. Das Ziel ist also Nachrichtenartikel zu sammeln, zu filtern und einem Analysten zur Verfügung zu stellen. Der Teil dieses Systems, der die Nachrichtenartikel dahingehend klassifiziert, ob über Identitätsdatendiebstahl berichtet wird, wird in dieser Arbeit vorgestellt.

Der Diebstahl von Identitätsdaten ist kein seltenes Ereignis. Ausgewiesene Quellen erreichen für das Jahr 2019 Werte über 5 Milliarden für die Anzahl an gestohlenen Identitätsdaten[Dav20]. Auch Wikipedia unterhält eine unvollständige Liste von großen Diebstählen sensibler Daten und listet auch die Anzahl der gestohlenen Datensätze. Für das Jahr 2019 sind hier deutlich über 3 Milliarden Datensätze gelistet [Wik20]. Alleine das Unternehmen „Facebook“ hat nach Berichten der Nachrichtenagentur CBS über 540 Millionen Zugangsdaten verloren [Sil].

Um betroffene Nutzer proaktiv zu informieren und vor Identitätsdiebstahl zu schützen müssen die gestohlenen Daten jedoch gesammelt werden. Damit ein Analyst in der Lage ist diese Daten zu finden, muss zunächst festgestellt werden, dass ein Diebstahl der Daten vorgefallen ist. Ein möglicher Ansatz um solche Information zu erhalten ist es Veröffentlichungen von Medien, die über Diebstähle von Identitätsdaten berichten zu beobachten. Beim Sammeln solcher Nachrichtenartikel stellte sich heraus, dass selbst bei nur ausgewählten Quellen ein Analyst ohne Filterung pro Monat etwa 300 Nachrichtenartikel lesen müsste. In unserem Datensatz befassen sich im Schnitt 3,8% tatsächlich mit Identitätsdiebstahl. Daher soll ein System Nachrichtenartikel danach filtern, ob über Identitätsdatendiebstahl berichtet wird. Daraufhin kann ein Analyst nach den gestohlenen Daten suchen.

Die Nachrichtenartikel sind in natürlicher Sprache geschrieben, was eine maschinelle Klassifikation nicht trivial macht. Deswegen sollen Verfahren des *Natural Language Processing (NLP)* verwendet werden, um die Nachrichtenartikel, die über Identitätsdiebstähle berichten, von anderen zu unterscheiden. Das Ziel der Arbeit ist, aus vorhandenen Nachrichtenartikeln *Feature Vektoren* zu extrahieren, so dass danach ein Verfahren des maschinellen Lernens die Artikel klassifizieren kann. Die Forschungsfrage lautet somit:

> Welches Verfahren zur Merkmalsextraktion aus Nachrichtenartikeln eignet sich, um die Artikel mithilfe von maschinellem Lernen möglichst genau dahingehend zu klassifizieren, ob sie über einen Identitätsdiebstahl berichten?

Es stellt sich darüber hinaus auch die praktische Frage wie viele Artikel noch gelesen werden müssen, wenn ein solches automatisiertes System eingesetzt wird. Hier stellten wir fest, dass mit dem entwickelten System, nur noch etwa 4 Artikel pro Woche gelesen werden müssen. Geht man

davon aus, dass über jeden Diebstahl mindestens dreimal berichtet wird ist die Wahrscheinlichkeit, dass die Information über einen Diebstahl verloren geht kleiner als 0,28%.

Aber auch für andere Anwendungen im Bereich der IT-Sicherheit kann ein System das Nachrichtenartikel oder andere Texte klassifiziert sinnvoll sein. So lässt sich aus Nachrichtenartikeln oder anderen Texten auch *Threat Intelligence* gewinnen. Es handelt sich um einen innovativen Ansatz um *Indicators of compromise* zu extrahieren, indem nicht nur nach Berichten über Identitätsdatendiebstahl, sondern auch, zum Beispiel, nach dem Namen eines Unternehmens gefiltert wird. Um möglichst viele Anwendungsszenarien zu unterstützen wurde ein System entwickelt, dass es ermöglicht für beliebige Anwendungen das beste Verfahren zur Merkmalsextraktion zu finden.

Experimenteller Aufbau

Das in der Bachelorarbeit entwickelte System führt 288 Experimente durch, wobei in jedem Experiment ein mögliches Verfahren evaluiert wird. Jedes der Verfahren wurde durch 10-fache Kreuzvalidierung getestet. Das bedeutet, dass jedes der Verfahren auf 10 verschiedenen Sätzen von Eingabedaten getestet wurde, also wurden in Summe 2 880 Klassifizierer erstellt. Als Datensatz wurden 15 211 Artikel aus dem Zeitraum von 2007 bis 2019, von denen 1997 über Identitätsdiebstahl berichten, verwendet.

Die Anwendung wurde in C++ unter Zuhilfenahme des *OpenMP*-Standards geschrieben und skaliert daher gut auf Systemen mit mehreren Prozessoren. Für die Administration der Experimente wurden *Python* und *MariaDB* eingesetzt um eine automatisierte Ausführung zu ermöglichen. Die Tests wurden auf einen *Kubernetes*-Cluster in *Docker*-Containern durchgeführt, daher lässt sich auch der vollständige Experiment-Aufbau auf andere Systeme und Anwendungsfälle übertragen.

Diese Arbeit beschränkt sich auf die Evaluation von Verfahren zur Merkmalsextraktion. Ein weiterer Faktor für die Performance des Klassifizierers ist das gewählte Verfahren für maschinelles Lernen. Hier wurde sich für die Evaluation aber auf Support Vektor Machines (SVM) als Verfahren des maschinellen Lernens beschränkt. SVMs wurden ausgewählt, da sie häufig in der betrachteten Referenzliteratur verwendet werden [Reh+15; Bah+18]. Die Anwendung erlaubt jedoch eine einfache Erweiterung und Evaluation weiterer Verfahren des maschinellen Lernens.

Bevor jedoch die Support Vector Machines zum Einsatz kommen, müssen die Texte in sogenannte *Feature Vektoren* konvertiert werden. Hierfür werden zwei Schritte durchgeführt. Zunächst werden beim *Preprocessing*, aus den Zeitungsartikelen einzelne Wörter, die als Token bezeichnet werden extrahiert. Daraufhin wird *Feature Selection* durchgeführt. Dieser Schritt wählt die Tokens aus, die als Features verwendet werden. Das komplette Verfahren für die Evaluation ist in Abbildung 1 zu sehen.

Das Preprocessing ist der erste Verarbeitungsschritt, den die Texte durchlaufen. Es bekommt die Nachrichtenartikel als Eingabe und produziert als Ausgabe eine Liste von Tokens, die im Text vorkommen. Der Schritt setzt sich aus verschiedenen Verfahren zusammen, wobei 24 verschiedene

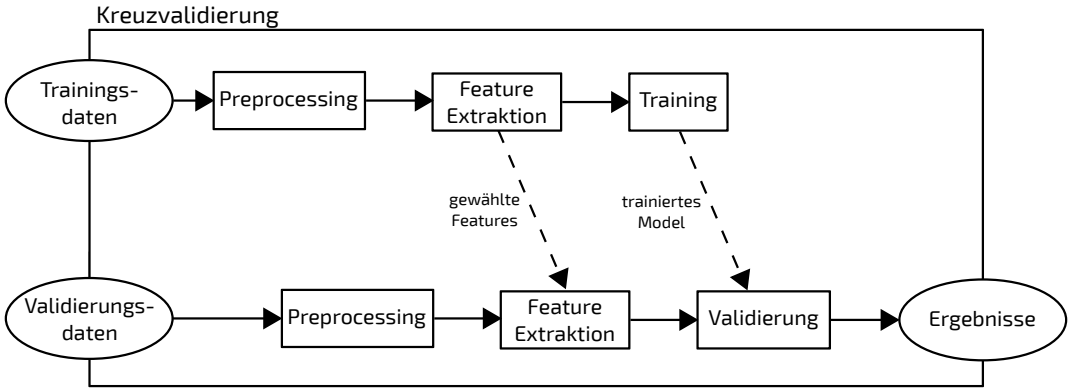


Abbildung 1: Schematische Darstellung des Evaluationsverfahren

Kombinationen (also alle möglichen Pfade durch Abbildung 2) evaluiert werden. Abbildung 2 zeigt die Reihenfolge, in der die Verfahren angewendet wurden. Alle Verfahren, die hier angewendet werden, verfolgen das Ziel überflüssige Tokens, die der Klassifizierung nicht zuträglich sind zu eliminieren. Zudem sollen Tokens, die die gleiche semantische Bedeutung haben, zusammengefasst werden und Tokens auf eine einfache grammatische Form gebracht werden. So wird der Eingaberaum für die folgenden Verfahren verkleinert.

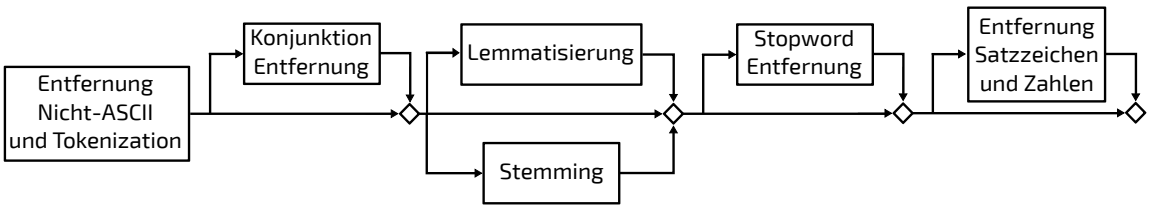


Abbildung 2: Schematische Darstellung der Verfahren zum Preprocessing

Die Feature Selection ist der zweite Verarbeitungsschritt, den die Texte die zuvor in Listen von Tokens verwandelt wurden, durchlaufen. Hierbei sollen die Tokens mit der größten Bedeutung ausgewählt werden und jedem Token ein Gewicht zu sortiert werden. Hierfür wird die Liste von Tokens, die als Eingabe zur Verfügung steht zunächst zusammengefasst, sodass jeder Token nur einmal vorkommt und seine Häufigkeit als Wert zur Verfügung steht.

Daraufhin werden 100 oder 10 000 Tokens ausgewählt, die für alle Dokumente gleich sind, die die Dokumente repräsentieren. Diese Tokens werden durch das *Relative Discriminative Criterion (RDC)* oder das *Multivariate Relative Discriminative Criterion (MRDC)*, was eine Erweiterung von RDC ist, ausgewählt. RDC wählt Features aus, die häufig, aber möglichst nur in einer Klasse von Artikeln vorkommt. MRDC ist eine Erweiterung von RDC und berücksichtigt zusätzlich die Korrelation zwischen den Tokens. Auch ein Verfahren, dass alle Tokens übernimmt wurde, getestet. Aufgrund von Laufzeiterwägungen wurde dieser Ansatz jedoch nicht weiterverfolgt.

Nachdem die Tokens ausgewählt wurden werden diese für jedes Dokument gewichtet. Diese Gewichtung kann eine binäre Darstellung (also ob das Token vorhanden war) die Häufigkeit der

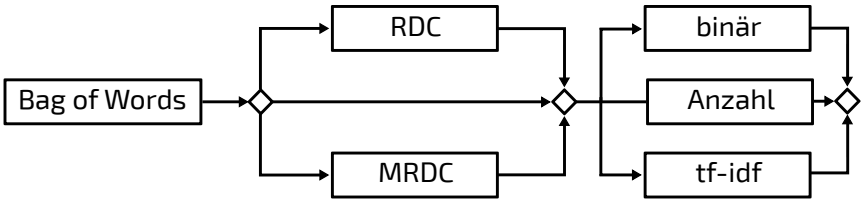


Abbildung 3: Schematische Darstellung der Verfahren zur Feature Selection

Vorkommnisse eines Tokens oder ein Maß namens *tf-idf* sein. *tf-idf* gibt an wie oft ein Token in einem Dokument, im Verhältnis zur Häufigkeit des Tokens in allen Dokumenten, vorkommt. Eine Übersicht über diesen Schritt findet sich in Abbildung 3.

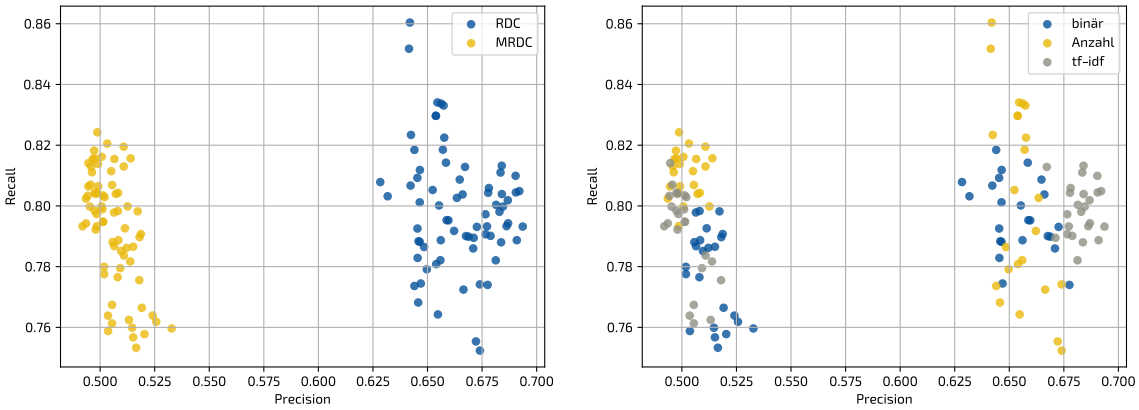
Nachdem die Tokens ausgewählt wurden werden die Artikel von einer Support Vektor Maschine verarbeitet. Wenn 100 Tokens ausgewählt wurden wird eine SVM mit Radial-Basis-Function (RBF)-Kernel verwendet. Wurden jedoch 10 000 Tokens ausgewählt wird eine linear SVM verwendet. Die Parameter für die SVM werden durch Gittersuche bestimmt. Daraufhin wird die SVM trainiert und das trainierte Model auf den Validierungsdaten evaluiert.

Evaluation

Für die Evaluation werden vier Metriken betrachtet, die sich alle aus der Konfusionsmatrix berechnen: Recall, Precision, F_1 und F_{19}. F_{19} ist ein F_β-Score mit einem $\beta = 19$, diese Metrik stellt also das gewichtete Harmonische Mittel von Recall und Precision dar, wobei der Recall mit 95% und die Precision mit 5% gewichtet wird. Diese Metrik wird betrachtet, da sie den Recall stärker gewichtet als die Precision. Der Recall ist für die hier betrachtete Anwendung relevanter, da ein Analyst sehr einfach einen falschen Artikel aussortieren kann. Das aktuelle Konzept sieht jedoch keine Möglichkeit vor, wie ein verworfender Artikel einen Analysten noch erreichen kann.

In der Evaluation stellt sich zunächst heraus, dass die Auswahl von nur 100 Features stets deutlich schlechtere Performance als die Verfahren mit 10 000 Features zeigt. Daher werden für die restlichen Evaluationen nur noch die Verfahren mit 10 000 Features betrachtet. Die Evaluation wird hier beispielhaft für zwei Verfahren durchgeführt. Zunächst werden die verschiedenen Verfahren zur Feature Extraktion betrachtet. Ein Vergleich der beiden getesteten Verfahren zur *Feature Extraction* findet sich in Abbildung 4a. Es ist für jedes Verfahren ein Punkt in das Diagramm gezeichnet, welcher anhand des verwendeten Feature Extraktions Verfahrens eingefärbt wurde. Hier zeigt sich eine klare Trennung der beiden Punktwolken. Der Recall der Methoden befindet sich, bis auf einige Ausreißer, in der gleichen Größenordnung. Es ist eine leichte Tendenz zu einem besseren Recall für das RDC-Verfahren zu erkennen. Zudem ist die Precision für das RDC-Verfahren in allen Fällen deutlich besser. Daher liefert das RDC-Verfahren hier deutlich bessere Ergebnisse.

Anders sieht der Vergleich für das zweite betrachtete Verfahren aus. Es handelt sich hier um die verschiedenen Feature Gewichtungsverfahren. Für die Gewichtung der Features wurden drei Varianten betrachtet. Eine binäre Darstellung, die Anzahl der Vorkommnisse und *tf-idf*. Ein Vergleich



(A) Recall und Precision nach Feature Extractions-Verfahren

(B) Recall und Precision nach Feature Gewichtungsverfahren

Abbildung 4: *Darstellung Auswertung*

der Performance findet sich in Abbildung 4b. Hier zeigt sich, dass die Verfahren nicht trivial zu ordnen sind. Dennoch ist zu sehen, dass in Kombination mit anderen Verfahren die Gewichtung nach Anzahl einen besseren Recall erreicht, als die anderen Techniken. Auch zeigt sich, dass eine Gewichtung mit *tf-idf* eine bessere Precision ermöglicht.

Für alle betrachteten Verfahren wurde die Nützlichkeit im Anwendungsfall evaluiert. Für einige der Verfahren lässt sich für diesen und ähnliche Anwendungsfälle eine Empfehlung aussprechen. So zum Beispiel sollte stets RDC statt MRDC verwendet werden. Für andere Verfahren sind solche Aussagen jedoch nicht möglich. Hier ist für den individuellen Anwendungsfall eine Evaluation notwendig. Ein Beispiel hierfür ist die Auswahl des Feature Gewichtungsverfahrens. Hier ist Expertenwissen über die Anwendung für die Auswahl notwendig. Wenn das System für andere Anwendungsfälle verwendet werden soll, sollte erneut bestimmt werden welche Verfahren dann am besten funktionieren.

Ergebnis

Die Frage, welches Verfahren sich am besten eignet, um Nachrichtenartikel bezüglich ihres Inhalts zu klassifizieren, lässt sich also nicht trivial beantworten. Für diesen konkreten Anwendungsfall liefern viele der betrachteten Verfahren akzeptable Ergebnisse und unterscheiden sich nur in der Ausprägung der Fehlerklassen. Allgemein lässt sich lediglich sagen, dass in jedem Fall die Verwendung von 10 000 Features der von 100 vorgezogen werden sollte, da die Ergebnisse für 100 gewählte Features deutlich schlechter sind. Für die anderen betrachteten Verfahrensschritte lassen sich solche allgemeinen Aussagen nicht unbedingt treffen. Tendenziell lässt sich sagen, dass RDC gegenüber MRDC vorgezogen werden sollte, für alle anderen Verfahrensschritte muss für den jeweiligen Anwendungsfall erneut bestimmt werden, ob sie sinnvoll sind.

Für den konkreten Anwendungsfall der Klassifizierung von Nachrichtenartikeln im Bezug darauf, ob sie von Identitätsdiebstahl handeln, findet das System das folgende Verfahren aus den 288 betrachteten:

1. Entfernen von Nicht-ASCII Zeichen
2. Tokenisierung
3. Stemming mit den Porter-Stemmer
4. Entfernen von Satzzeichen und Ersetzen von Zahlen durch die Zeichenkette „NUMBER“
5. Auswahl von 10 000 Features mit den *Relative Discriminative Criterion*
6. Gewichtung der Features über die Häufigkeit eines Features im Text
7. Normalisierung der Feature Vektoren auf Länge 1
8. Training einer *Support Vector Machine* mit einem linearen Kernel

Diese Verfahren erreicht auf den Testdaten in 10-facher Kreuzvaliderung eine Recall von 0,86, eine Precision von 0,64, F_1 von 0,74 und F_{19} von 0,86. Dies sind sowohl für den Recall als auch für F_{19} die höchsten erreichten Werte.

Um die Performance der Anwendung zu evaluieren wurde es auf einem handelsüblichen Computer mit einem Intel Core i7 8700k und 16 GiB RAM getestet. Die Trainingsphase verbraucht 8:25 min Echtwelt-Zeit und 20:30 min CPU-Zeit. Der maximale Hauptspeicherverbrauch war 2,6 GiB, die durchschnittliche CPU-Auslastung 260% und die maximale CPU-Auslastung 710%. Die Architektur der C++-Anwendung ist modular gehalten und aufgeteilt in eine Bibliothek, die die eigentlichen Verarbeitungsschritte bereitstellt und einfach wiederverwendet werden kann. Zudem gibt es eine Command-Line-Anwendung, die die Verarbeitungsschritte nacheinander auf den Daten ausführt. Dieses Design ermöglicht eine einfache Weiterverwendung der Bibliothek für verschiedene Anwendungen, auch mit anderen Anwendungsszenarien.

Die Ergebnisse des Verfahrens bedeuten für einen Analysten nun Folgendes: Von den 300 Artikeln, die zuvor pro Monat gelesen werden mussten, handeln in unserem Datensatz 3,8% von Identitätsdiebstahl. Von den 11 Artikeln, die also über Identitätsdiebstahl berichten gibt das System 9 aus. Zusätzlich werden noch 5,5 Artikel, die nicht über Identitätsdiebstahl berichten ausgegeben. Der Analyst muss also pro Woche nur noch etwa 4 Zeitungsartikel lesen, im Gegensatz zu etwa 75, die zuvor gelesen werden mussten. Wenn man davon ausgeht, dass über einen gegebenen Identitätsdiebstahl zwei Artikel veröffentlicht werden ist die Wahrscheinlichkeit, dass beide diese Artikel verpasst werden 1,9%. Geht man von drei Artikel aus ist die Wahrscheinlichkeit sogar kleiner als 0,28%. Das System stellt also eine deutliche Zeitersparnis für einen Analysten dar, ohne dass eine wesentlichen Menge relevanter Informationen verloren geht.

Literatur

[Bah+18] Said Bahassine, Abdellah Madani, Mohammed Al-Sarem und Mohamed Kissi. „Feature selection using an improved Chi-square for Arabic text classification“. In: *Journal of King Saud University-Computer and Information Sciences* (2018).

[Dav20] David McCandless, Tom Evans, Paul Barton, Dr Stephanie Starling, Duncan Geere. *World's Biggest Data Breaches & Hacks*. https://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/. [Online; accessed 23-August-2020]. 2020.

[Mal+on] Timo Malderle, Felix Boes, Gina Muuss, Matthias Wübbeling und Michael Meier. „Credential Intelligence Agency - A Threat Intelligence Approach to Mitigate Identity Theft“. In: *Springer Book of International Conference on Information Systems Security and Privacy* (in submission).

[Muu20] Gina Muuss. „Merkmalsextraktion bei Nachrichtenartikeln zur Themenklassifikation am Beispiel von Identitätsdatendiebstahl“. Bachelorarbeit. Rheinische Friedrich-Wilhelms-Universität Bonn, 2020.

[Reh+15] Abdur Rehman, Kashif Javed, Haroon A Babri und Mehreen Saeed. „Relative discrimination criterion–A novel feature ranking method for text data“. In: *Expert Systems with Applications* 42.7 (2015), S. 3670–3681.

[Sil] Jason Silverstein. *Facebook data breach: Hundreds of millions of records exposed on Amazon server, according to UpGuard cybersecurity research firm - CBS News*. 2019 CBS Interactive Inc. https://www.cbsnews.com/news/millions-facebook-user-records-exposed-amazon-cloud-server/. Accessed: 2020-03-05.

[Wik20] Wikipedia contributors. *List of data breaches — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=List_of_data_breaches&oldid=973980100. [Online; accessed 23-August-2020]. 2020.