

# Automatische Klassifikation von Anforderungen zur Unterstützung von Qualitätssicherungsprozessen

Jonas Winkler<sup>1</sup>

**Abstract:** Während des Requirements-Engineering werden Anforderungen an zu entwickelnde Systeme und Komponenten in Form von Lastenheften dokumentiert. Diese Lastenhefte enthalten neben rechtlich relevanten Anforderungen weitere Inhalte wie Erklärungen, Zusammenfassungen und Abbildungen. Um zwischen Anforderungen und solchen Zusatzinformationen unterscheiden zu können, werden alle Lastenheftinhalte manuell als Anforderung oder Information eingestuft. Analysen haben ergeben, dass diese Klassifikation nicht konsequent durchgeführt wird. Daher liegt es nahe, diese Klassifikation zu automatisieren. In diesem Beitrag wird ein Ansatz vorgestellt, der diese Klassifikation mithilfe von Techniken aus dem Bereich Data Mining und Machine Learning automatisch durchführen kann.

**Keywords:** Anforderungsmanagement, Qualitätssicherung, Data Mining, Machine Learning, Textklassifikation, Text-Clustering, Neurale Netzwerke

## 1 Motivation

Im Rahmen des Entwicklungsprozesses von Fahrzeugen werden Anforderungen an neu entwickelte Fahrzeugkomponenten in Form von Lastenheften dokumentiert. Diese Dokumente dienen dem Zweck, die Verbindlichkeiten zwischen Auftraggeber (Automobilhersteller) und Auftragnehmer (Lieferant) festzulegen. Vor der Übergabe eines Lastenhefts an einen Lieferanten durchläuft dieses einen intensiven Qualitätssicherungsprozess. Dabei wird unter anderem darauf geachtet, dass die Anforderungen vollständig und eindeutig formuliert sind.

Neben Anforderungen enthalten Lastenhefte weitere Informationen wie beispielsweise Erklärungen, Zusammenfassungen, Beispiele, Abbildungen und Verweise auf andere Dokumente. Diese Elemente sind keine Anforderungen, die der Lieferant in seinem Produkt umsetzen muss. Um zwischen solchen Zusatzinformationen und rechtlich verbindlichen Anforderungen zu unterscheiden, werden alle Elemente eines Dokuments mit speziellen Markierungen versehen, die ein Objekt entweder als Anforderung oder als Information kennzeichnen.

Durch die Analyse mehrerer realer Lastenhefte wurde festgestellt, dass diese Markierung nur selten konsequent durchgeführt wird. Bei der Lastenhefterstellung wird das Attribut oftmals nicht beachtet und die nachträgliche Bearbeitung stellt einen erheblichen Aufwand dar. Dieser Umstand erschwert die Weiterverwendung eines Dokuments in späteren Entwicklungsschritten, beispielsweise bei der Erstellung der Testspezifikation zum Lastenheft.

---

<sup>1</sup> Technische Universität Berlin, DCAITI, Ernst-Reuter-Platz 7, 10587 Berlin, jonas.winkler@dcaiti.com

In diesem Beitrag wird untersucht, inwieweit es möglich ist, die Klassifizierung von Anforderungen und Informationen mithilfe computergestützter Werkzeuge automatisch durchzuführen. Mithilfe eines solchen Werkzeugs wäre es möglich, sowohl nicht klassifizierte Dokumente zu klassifizieren, als auch in bereits klassifizierten Dokumenten den Autor auf möglicherweise falsch klassifizierte Dokumententeile hinzuweisen.

Dazu wurde zunächst eine Feldstudie an mehreren Anforderungsdokumenten durchgeführt, um die Machbarkeit und mögliche Grenzen besser einschätzen zu können und um die Entwicklung eines Ansatzes zur automatischen Klassifizierung planen zu können. Diese Studie, ein daraufhin entwickelter Ansatz und erste Evaluationsergebnisse werden in diesem Beitrag vorgestellt.

## 2 Grundlagen

In diesem Beitrag wird mit Dokumenten gearbeitet, welche in der Dokumentendatenbank IBM Rational DOORS<sup>2</sup> gepflegt werden. In dieser Datenbank besteht jedes *Dokument* aus mehreren *Objekten*. Jedes Objekt eines Dokuments enthält jeweils einen atomaren Dokumentenbaustein wie beispielsweise einen Satz oder eine Abbildung.

Jedes Objekt eines Dokuments besitzt *Attribute*. Diese können von Autoren frei definiert werden. Häufig verwendete und für diesen Beitrag relevante Attribute sind die folgenden:

- **Text:** In diesem Attribut wird der Textkörper des Objekts abgelegt.
- **Schlüssel:** Der Schlüssel identifiziert ein Objekt innerhalb eines Dokuments eindeutig. Wird ein Dokument kopiert, bleiben die Schlüssel erhalten. So kann ermittelt werden, welche Objekte eines Dokuments neu erstellt wurden und welche aus einem anderen Dokument stammen.
- **Typ:** In diesem Attribut wird der Typ des Objekts gespeichert. Für diesen Beitrag sind folgende Typen relevant:
  - **Anforderung:** Der Text des Objekts ist eine verbindliche Anforderung und vom Lieferanten umzusetzen.
  - **Information:** Das Objekt enthält lediglich ergänzende, nicht verbindliche Zusatzinformationen.

## 3 Studie

Untersucht wurden mehrere deutsche Lastenhefte eines OEMs von Komponenten aus dem Fahrzeuginnenraum. Die Art des Inhalts ist bei allen Dokumenten identisch, geschrieben wurden die Lastenhefte allerdings von unabhängigen Personen aus verschiedenen Abteilungen. Anhand dieser Lastenhefte wurden folgende Beobachtungen gemacht:

---

<sup>2</sup> <http://www.ibm.com/software/products/de/ratidoor>

1. Viele Lastenheftbestandteile wie beispielsweise die Kapitelstruktur, Erklärungen zu den Lastenheftinhalten und komponentenunabhängige Anforderungen sind in allen Dokumenten nahezu identisch. Dies liegt daran, dass Dokumente stets auf Basis einer gemeinsamen Vorlage erstellt werden. Für alle aus einer Vorlage stammenden Objekte steht die Klassifizierung fest und darf nicht geändert werden.
2. Die meisten komponentenspezifischen Lastenheftinhalte werden in natürlicher Sprache verfasst. Es wurde beobachtet, dass für Anforderungen generell eher präzisere Formulierungen verwendet werden, als für Zusatzinformationen. Weiterhin werden bestimmte Inhalte wie Verweise auf andere Dokumente und Erläuterungen zu einem Kapitel grundsätzlich immer als Information klassifiziert.
3. Einige Lastenheftinhalte werden nicht ausformuliert, sondern in Form von Schlüssel-Wert-Paaren, Aufzählungen, Tabellen und Abbildungen dokumentiert. Einige Inhalte wie Signalspezifikationen und Kontaktflächenspezifikationen werden stets nach dem gleichen Muster erstellt und immer gleich klassifiziert.
4. In vielen Dokumenten ließen sich wiederholt Formulierungen und Formatierungen finden, die für das jeweilige Dokument einzigartig sind. Dies lässt sich auf die unterschiedlichen Autoren der Dokumente zurückführen.
5. Die Klassifizierung wurde in einigen Lastenheften nicht konsequent durchgeführt. Teilweise wurden Hinweise, die offensichtlich keine Anforderungen sind, nicht als Information markiert, teilweise wurde die Klassifizierung gar nicht vorgenommen.

Zur Veranschaulichung werden im Folgenden einige markante Beispiele gegeben. Beispiele für die in (2) genannten natürlichsprachlichen Objekte sind die folgenden Sätze:

- (Anforderung) Es muss ein Resetzähler implementiert werden.
- (Anforderung) Das Relais darf bei einem Spannungseinbruch nicht abfallen.
- (Information) Weitere Informationen sind dem Kapitel „A“ zu entnehmen.
- (Information) Das Schaltbild zeigt schematisch die Bauteile dieser Komponente.

Anforderungen können meist an präzisen Formulierungen („muss“, „darf nicht“) erkannt werden. Informationen werden meist mit anderen Formulierungen geschrieben („Weitere Informationen“, „zeigt schematisch“).

Beispiele für nicht natürlichsprachliche Objekte in (3) sind Pinnings (Spezifikation der Schnittstelle einer elektronischen Komponente) oder auch Ansprechpartner:

Pin Nr: 3	Ansprechpartner
Pin Bezeichnung: Kontrollsignal	Name, Vorname: Mustermann, Max
Nenn-Spannung: 12 V	Abteilung: ABC12
Nenn-Strom: 0,1 A	Telefon: +49 123 456789
Max-Strom: 0,2 A	E-Mail: max@mustermann.de

Alle Objekte, in denen ein Pin spezifiziert wird, werden grundsätzlich als Anforderung klassifiziert. Analog dazu werden Ansprechpartner grundsätzlich als Information klassifiziert. Für jedes dieser Muster lässt sich meist eindeutig eine Klassifizierung bestimmen.

Da die Lastenheftinhalte in sehr unterschiedlichen Formaten vorliegen und die notwendigen Merkmale zur Diskriminierung zwischen Anforderung und Information sehr unterschiedlich sind, kann ein einziger Klassifizierungsalgorithmus keine guten Ergebnisse liefern. Vielmehr muss zwischen den unterschiedlichen Formaten unterschieden werden und je nach Format ein spezialisiertes Klassifizierungsverfahren eingesetzt werden. Innerhalb der betrachteten Dokumente wurden vorrangig folgende Formate identifiziert:

- **Überschrift.** Das Objekt ist eine Überschrift.
- **Satz.** Das Objekt enthält einen oder mehrere wohlgeformte, natürlichsprachliche Sätze.
- **Aufzählung.** Das Objekt enthält eine Aufzählung oder Nummerierung verschiedener Begriffe.
- **Definition.** Eine Definition umfasst eine oder mehrere Zeilen, die jeweils dem Muster „<Schlüssel>: <Wert>“ folgen.
- **Abbildung.** Das Objekt enthält ein Foto, eine schematische Darstellung oder sonstige Abbildung.
- **Tabelle.** Das Objekt enthält eine Tabelle.

Diese Typisierung wird in den weiteren Abschnitten *Strukturtyp* genannt. Auf Basis der getroffenen Beobachtungen wird im Folgenden ein Ansatz zur automatischen Klassifizierung vorgestellt.

## 4 Ansatz zur automatischen Klassifikation

Alle Objekte eines Dokuments werden einzeln klassifiziert. Jedes Objekt durchläuft während der Klassifizierung mehrere Klassifizierungsschritte. Der Ablauf ist in Abbildung 1 schematisch dargestellt. Jeder Schritt (in der Abbildung grau hinterlegt) untersucht ein Objekt und versucht, mithilfe geeigneter Techniken eine Klassifizierung durchzuführen. Ein Schritt kann ein Objekt entweder in eine Klasse einordnen oder keine Aussage über die Klassifizierung treffen. Wird ein Objekt durch einen Schritt erfolgreich klassifiziert, endet die Klassifizierung mit diesem Schritt. Ist durch einen Schritt keine Aussage über die Klassifikation möglich, wird der nächste Schritt ausgeführt.

Schritte, die sich nur auf wenige Objekte anwenden lassen, dafür allerdings eine sehr hohe Genauigkeit haben, werden zuerst ausgeführt. Schritte, die sich auf sehr viele Objekte anwenden lassen und eine geringere Genauigkeit haben, werden später ausgeführt. Dies hat zur Folge, dass einfach zu klassifizierende Objekte durch einfache Klassifizierungstechniken richtig klassifiziert werden und nicht durch komplizierte Techniken möglicherweise falsch klassifiziert werden.

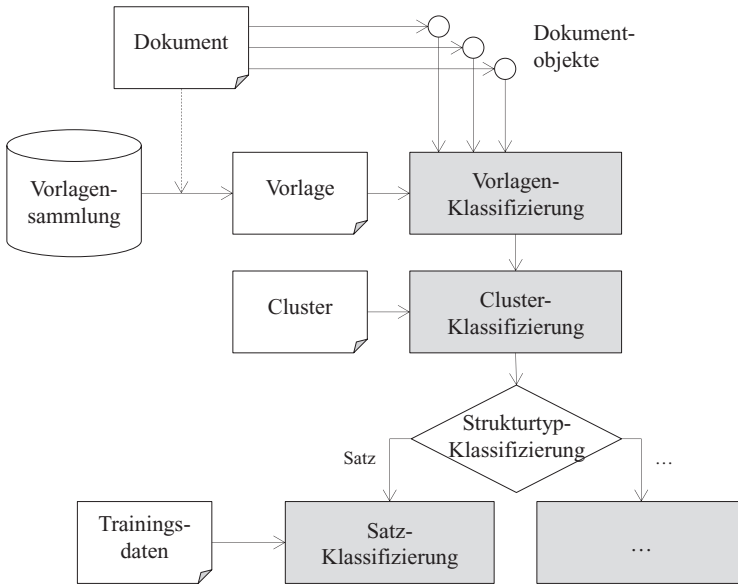


Abb. 1: Übersicht über Klassifizierungsansatz

Die Vorlagen-Klassifizierung überprüft Objekte, die aus einer Vorlage kopiert wurden und klassifiziert entsprechend der Vorlage. Die Cluster-Klassifizierung sucht nach textuell ähnlichen und identisch klassifizierten Objekten und benutzt die daraus entstehenden Cluster zur Klassifizierung häufig wiederkehrender Muster.

Danach wird abhängig vom Strukturtyp eines Objekts ein speziell auf den Strukturtyp ausgerichtetes Klassifizierungsverfahren angewendet. Objekte, deren Strukturtyp nicht bestimmt werden kann, werden nicht weiter klassifiziert. Dadurch werden schwierig zu klassifizierende Objekte und Objekte mit nicht eindeutiger Klassifikation übersprungen. Derzeit existiert lediglich für den Strukturtyp Satz ein Klassifizierungsverfahren.

#### 4.1 Klassifizierung nach Vorlage

In diesem Schritt werden Objekte, welche aus einer Vorlage stammen, identifiziert und klassifiziert. Dazu wird zuerst die dem Dokument zugrundeliegende Vorlage identifiziert und aus einer Vorlagensammlung geholt. Diese Vorlage enthält von allen Objekten der Vorlage deren Text, Typ und Schlüssel.

Anhand des Schlüssels und des Texts eines zu klassifizierenden Objekts wird ermittelt, ob es aus der Vorlage kopiert wurde. Dies ist möglich, da der Schlüssel bei der Kopie eines Objekts nicht verändert wird. Ist dies der Fall, wird das Objekt gemäß der Vorlage klassifiziert. Alle Objekte, welche nicht aus einer Vorlage kopiert wurden, werden durch diesen Schritt nicht klassifiziert.

## 4.2 Klassifikation durch Clustering

Während der Studie in Abschnitt 3 wurde festgestellt, dass viele Arten von Inhalten (bspw. Pinning, Ansprechpartner) meist nach einem einheitlichen Muster dokumentiert werden. Weiterhin werden alle Objekte einer Art immer gleich klassifiziert. Es liegt also nahe, wiederkehrende Muster zu erkennen und entsprechend der für das jeweilige Muster üblichen Klassifizierung zu klassifizieren.

Dazu wurde zunächst ein hierarchischer Complete-Link-Clusterer auf eine sehr große Anzahl bereits klassifizierter Objekte aus vielen Dokumenten angewendet. Als Textdistanzmaß wurde die Levenshtein-Distanz verwendet. Dieses Verfahren bildet aus einer Menge von Zeichenketten Cluster, wobei die Distanz jeweils aller Paare zweier Zeichenketten innerhalb eines Clusters nicht größer ist als ein festgelegter Schwellenwert. Die Elemente eines Clusters sehen sich daher sehr ähnlich. Details zu den verwendeten Verfahren werden in [AZ12b] beschrieben.

Ein geeigneter Schwellenwert für das Textdistanzmaß wurde empirisch ermittelt.

Die erzeugten Cluster enthielten sowohl die erwarteten Cluster für Muster wie Pinning und Signalspezifikation, allerdings auch Cluster ähnlicher, aber sehr unterschiedlich klassifizierter Sätze. Weiterhin enthielten einige Cluster Objekte aus lediglich einem oder wenigen Dokumenten. Solche Cluster sind zur Erkennung von Mustern in anderen Dokumenten nicht hilfreich. Daher wurden die erkannten Cluster mithilfe zweier Metriken gefiltert.

Sei  $C$  ein Cluster. Sei  $c_{types}(C)$  eine Liste, wobei diese Liste so viele Elemente hat, wie in  $C$  einzigartige Klassifizierungen vorkommen. Jedes Element stellt die Anzahl der Vorkommen einer Klassifizierung dar. Bei einem Cluster  $C$ , in dem 10 Objekte als Anforderung, 2 Objekte als Information und 1 Objekt nicht klassifiziert sind, ist beispielsweise  $c_{types}(C) = [10, 2, 1]$ . Analog dazu sei  $c_{sources}(C)$  eine Liste, wobei jedes Element die Anzahl der Objekte enthält, welche aus demselben Dokument stammen. Bei einem Cluster  $C$  mit 4 Objekten aus Dokument A und 6 Objekten aus Dokument B wäre somit  $c_{sources}(C) = [4, 6]$ . Mithilfe der Funktion

$$f(L) = \frac{\sum L}{\max L}$$

kann ein Cluster  $C$  nun wie folgt bewertet werden:

- Ist  $f(c_{types}(C)) = 1$ , wird für alle Objekte die gleiche Klassifizierung verwendet. Je größer der Wert von  $f$ , desto mehr unterschiedliche Klassifizierungen werden verwendet.
- Ist  $f(c_{sources}(C)) = 1$ , stammen alle Objekte eines Clusters aus demselben Dokument. Je größer der Wert von  $f$ , aus desto mehr unterschiedlichen Dokumenten stammen die Objekte eines Clusters.

Wünschenswert sind Cluster mit kleinem  $f(c_{types}(C))$  und hohem  $f(c_{sources}(C))$ . Schwellenwerte wurden wiederum empirisch ermittelt. Einige der erkannten Cluster sind in Tabelle 1 dargestellt. Neben den erwarteten Clustern für Signalspezifikationen, Pinning und Ansprechpartner wurden auch viele Cluster gefunden, die ähnlich formulierte Sätze zusammenfassen.

Clustername	Anzahl	$f(c_{types}(C))$	$f(c_{sources}(C))$	Klassifikation
Ansprechpartner	177	1.208	4.750	Information
Signalspezifikation	84	1.201	3.448	Anforderung
Pinning	130	1.127	4.057	Anforderung
Kontaktierung	60	1.089	4.357	Anforderung
Verweis auf anderes Dokument	59	1.594	2.204	Information
Identifikationsnummer	198	1.200	2.250	Information

Tab. 1: Erkannte Cluster

Auf Basis dieser Cluster kann nun ein einfaches und effektives Klassifizierungsverfahren konstruiert werden. Soll ein Objekt klassifiziert werden, wird die Distanz des Texts des Objekts zu jedem Cluster ermittelt. Dabei wird dasselbe Textdistanzmaß eingesetzt, welches auch zur Erstellung der Cluster verwendet wurde. Das Objekt wird demjenigen Cluster zugeordnet, zu dem die Distanz am kleinsten ist. Dabei werden allerdings nur Cluster betrachtet, zu denen die Distanz kleiner ist, als der während der Cluster-Erstellung festgelegte Distanz-Schwellwert. Wird ein Cluster gefunden, wird das Objekt entsprechend der im Cluster am häufigsten vorkommenden Klassifizierung klassifiziert. Wird kein passendes Cluster gefunden, trifft dieser Klassifizierungsschritt keine Aussage über die Klassifizierung.

### 4.3 Strukturtyp-Klassifizierung

Zur weiteren Klassifikation muss der Strukturtyp eines Objekts ermittelt werden. Dazu wurde ein einfaches Verfahren entwickelt, welches ein Objekt schrittweise auf bestimmte Merkmale untersucht:

1. Besitzt das Objekt eine Kapitelnummer, wird es als Überschrift eingestuft.
2. Enthält das Objekt eine Grafik, so wird es als Abbildung eingestuft.
3. Enthält das Objekt eine Tabelle, so wird es als Tabelle eingestuft.
4. Umfasst das Objekt mehrere Zeilen und beginnt ein Großteil der Zeilen mit demselben Zeichen oder beginnt ein Großteil der Zeilen mit den für Nummerierungen typischen Zeichen (1, 2, 3, ...; a, b, c,...), so ist es eine Aufzählung.
5. Umfasst das Objekt eine oder mehrere Zeilen, die dem Muster „<Schlüssel>: <Wert>“ folgen, so ist das Objekt eine Definition.

Alle weiteren Objekte sind potentiell Sätze oder schwer zu klassifizierende Sonderformate. Zur Erkennung von Sätzen wird der Text der übriggebliebenen Anforderungen mithilfe des

Stanford-Parsers [KM03] analysiert. Dieser ordnet jedem Wort ein sogenanntes Part-of-Speech-Tag zu und erkennt unter anderem Sätze, Nominalphrasen und Adverbialphrasen. Alle Objekte, deren Text durch den Stanford-Parser als vollständiger Satz eingestuft wird (erkennbar an den Markierungen „S“ und „CS“ für *Sentence* und *Coordinated Sentence*), werden im weiteren Verlauf als Satz behandelt.

Alle durch diese Schritte nicht erfassten Objekte werden derzeit nicht betrachtet.

#### 4.4 Satz-Klassifikation

Zur Klassifikation von Sätzen werden Convolutional Neural Networks (CNN) eingesetzt [Ki14]. Diese bieten gegenüber etablierten Textklassifizierungsverfahren wie beispielsweise Naive Bayes [AZ12a] und Support-Vector-Machines [Kw98] die folgenden Vorteile:

- Andere Verfahren nutzen meist die Bag-of-Words-Technik. Bei dieser gehen Informationen über die Reihenfolge der Wörter verloren. Dies ist bei CNNs nicht der Fall.
- Durch Verwendung eines Wortvektorisierungsverfahrens (hier: word2vec [Mi13]) können auch Muster erkannt werden, bei denen die verwendeten Wörter leicht unterschiedlich sind.
- Im Vergleich zu anderen Verfahren hat diese Art von neuronalen Netzwerken bei diversen Textklassifizierungsaufgaben bessere Ergebnisse erzielt [KGB14].

Das Netzwerk wurde gemäß den in [ZW15] gegebenen Hinweisen erstellt. Die verwendete Netzwerkarchitektur ist mit der dort vorgestellten Architektur identisch. Die Vorbereitung der Trainingsdaten, die generelle Funktionsweise und der Trainingsprozess des Netzwerks werden im Folgenden kurz beschrieben. Für detaillierte Erklärungen zur Funktionsweise von CNNs sollte [Ki14] und [ZW15] konsultiert werden.

Ausgangsbasis sind alle Objekte aus sämtlichen zur Verfügung stehenden Anforderungsdokumenten, die durch die Strukturtyp-Klassifizierung als Satz eingestuft wurden und entweder als Anforderung oder Information klassifiziert wurden. Die Trainingsmenge umfasst etwa 10000 Sätze aus 89 Dokumenten. Alle Sätze werden vorverarbeitet, um das Training des Netzwerks zu vereinfachen: Klammerausdrücke und Sonderzeichen werden entfernt, Großbuchstaben werden in Kleinbuchstaben umgewandelt. Die daraus resultierende Datenmenge wird zur Erstellung einer Word-Vektor-Abbildung gemäß [Mi13] genutzt. Nach diesem Schritt besitzt jedes Wort aus der Datenmenge eine Vektordarstellung und damit jeder Satz eine Matrixdarstellung. Diese Matrizen dienen als Eingabe für das neurale Netzwerk.

Kernbestandteil des Netzwerks sind sogenannte Filter: Diese haben eine festgelegte Länge und werden wie ein Fenster über die Matrix eines Satzes geschoben. Diese Filter ermöglichen es dem Netzwerk, Muster in Wortfolgen entsprechend der Länge der Filter zu erlernen. Für diese Arbeit wurden die Filterlängen 2, 3, 4, 5 und jeweils 64 Filter pro Filterlänge verwendet.



Das Netzwerk besitzt für die zwei Klassen Information und Anforderung zwei Ausgangsneuronen. Diese sind jeweils vollständig mit allen Filtern verbunden. Über diese Neuronen zeigt das Netzwerk die Klassifizierung einer Eingabe an.

Zum Training des Netzwerks wurde wiederum die vorverarbeitete Datenmenge verwendet. Diese wurde in 90% Trainingsdaten und 10% Testdaten geteilt. Das Netzwerk wurde mithilfe der Trainingsdaten solange optimiert, bis es die Trainingsdaten zu 100% korrekt abbilden konnte.

Die Leistung des Netzwerks zur Klassifikation von Sätzen wurde mithilfe der Testdaten evaluiert. Dazu ist in Abbildung 2 die Entwicklung der Genauigkeit der Klassifikation der Trainingsdaten (blau) und Testdaten (rot) während des Trainingsprozesses dargestellt. Nach etwa 1000 Trainingsschritten ist das Netzwerk in der Lage, die Trainingsdaten zu 100% korrekt zu klassifizieren. Die dabei aus den Trainingsdaten erlernten Muster ermöglichen es dem Netzwerk, insgesamt ca. 81% der Testdaten richtig zu klassifizieren.

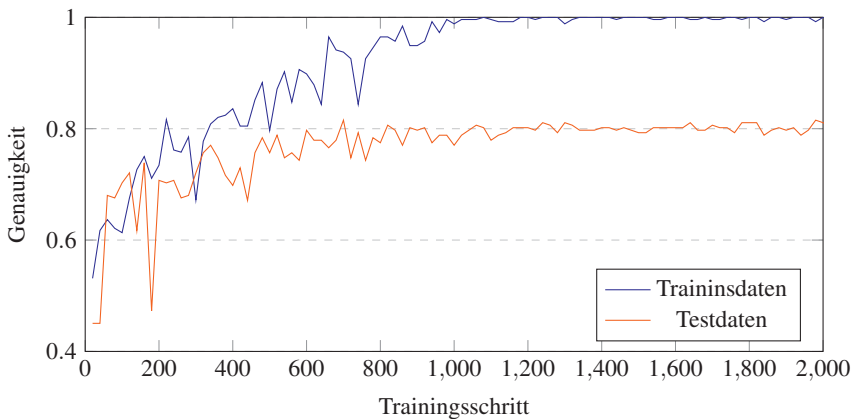


Abb. 2: Evaluation Lernprozess CNN

Zur Klassifizierung eines Objekts mithilfe des trainierten Netzwerks wird dieses genauso wie die Trainingsdaten vorverarbeitet, mithilfe der Wort-Vektor-Abbildung in eine Matrix konvertiert und als Eingabe für das Netzwerk verwendet. Das Netzwerk zeigt an den Ausgangsneuronen die Klassifizierung an.

## 5 Evaluation

Zur Evaluation wurde das Verfahren auf ein ausgewähltes Lastenheft angewendet. Dieses wurde weder zum Training verwendet, noch waren die Inhalte während der Konstruktion des Ansatzes bekannt. Die Ergebnisse jedes einzelnen Klassifizierungsschritts werden im Folgenden vorgestellt.

Das Lastenheft enthält insgesamt 747 Objekte. 20,7% dieser Objekte sind Überschriften. Insgesamt wurden 3,6% durch die Vorlagen-Klassifizierung, 11,6% durch die Cluster-

Klassifizierung und 41,5% durch die Satz-Klassifizierung abgedeckt. Die verbleibenden 22,6% konnten nicht klassifiziert werden.

Zur Evaluation der Strukturtyp-Klassifizierung wurde der Strukturtyp aller Objekte des Lastenhefts manuell gemäß der in Abschnitt 3 definierten Strukturtypen bestimmt und mit der automatischen Klassifizierung verglichen. Die Ergebnisse sind in Tabelle 2 dargestellt. Die Klassifizierung von Überschriften ist erwartungsgemäß fehlerfrei. Die Genauigkeit der Erkennung von Sätzen ist akzeptabel, könnte durch den Einsatz anderer Verfahren ggf. noch verbessert werden. Die beiden Strukturtypen Aufzählung und Definition konnten nur mit mäßiger Genauigkeit erkannt werden, dies liegt vor allem an inkonsistenten und teilweise sehr unterschiedlichen Formatierungen im Lastenheft. Für die restlichen 136 von insgesamt 747 Objekten konnte kein Strukturtyp ermittelt werden.

Strukturtyp	Anzahl	Recall	Precision	f1-Score
Überschrift	155	1.0	1.0	1.0
Satz	383	0.937	0.970	0.954
Aufzählung	13	1.0	0.813	0.897
Definition	60	0.817	0.645	0.721
Unbekannt	136			

Tab. 2: Evaluation Strukturtyp-Erkennung

Zur Evaluation der Genauigkeit der einzelnen Klassifizierungsschritte wurde die von den Schritten ermittelte Klassifizierung mit der tatsächlichen Klassifizierung verglichen. Jeder Schritt wurde dabei anhand der Objekte evaluiert, die durch den jeweiligen Schritt klassifiziert wurden. Die Ergebnisse sind in Tabelle 3 dargestellt.

Klassifizierer	Typ	Recall	Precision	f1-Score
Vorlage	Anforderung	1.0	1.0	1.0
	Information	1.0	1.0	1.0
Cluster	Anforderung	0.982	0.964	0.973
	Information	0.938	0.968	0.952
CNN	Anforderung	0.828	0.951	0.885
	Information	0.870	0.626	0.728
Gesamt	Anforderung	0.862	0.955	0.906
	Information	0.906	0.737	0.813

Tab. 3: Evaluation Klassifizierungsschritte

Die Genauigkeit der Vorlagenklassifizierung ist erwartungsgemäß perfekt. Von den 747 Objekten werden allerdings nur 27 durch diesen Schritt abgedeckt. Die Cluster-Klassifizierung hat im Lastenheft einige Muster und viele Sätze erkannt, die häufig in anderen Dokumenten in ähnlicher Form vorkommen und hat die Klassifizierung übernommen. Dies funktioniert erstaunlich gut. Die Satz-Klassifikation mittels CNNs hat 83% der klassifizierten Objekte richtig klassifiziert. Dies deckt sich mit der in Abschnitt 4.4 an den Testdaten ermittelten Genauigkeit. Ein Problem ist die niedrige Präzision von Informationen: Viele Anforderungen wurden durch das Netzwerk als Information klassifiziert. Die drei Klassifizierungsschritte konnten insgesamt 87.5% der Objekte richtig klassifizieren.

Die Evaluation hat gezeigt, dass die Klassifikation von Sätzen mittels CNNs nur mäßig funktioniert. Um die Stärken und Schwächen der Klassifizierung von Sätzen durch CNNs einschätzen zu können, werden in Tabelle 4 einige prägnante Beispiele gegeben, bei denen die Klassifikation besonders gut oder schlecht funktioniert.

Satz	Typ	Korrekt?	Ausgabe Inf / Anf
Der Wandler muss entsprechend der Strategie in den Selbstschutz gehen.	Anf	ja	-1.655 / 0.986
Der Strom des Heizelements muss mit einer Auflösung von ... gemessen werden .	Anf	ja	-3.860 / 1.483
Für die LED ist das Signal ... zu verwenden.	Anf	ja	-1.701 / 0.343
Diese Funktion ist in der Systembeschreibung ... im Detail beschrieben.	Inf	ja	3.386 / -4.359
Kurzschlüsse im Geräteinnern durch mutwilligen Eingriff dürfen zur Zerstörung führen.	Inf	nein	-0.919 / -0.551
Die Funktionsanteile für die Komponente sind im Systemlastenheft ... beschrieben.	Inf	nein	-0.898 / 0.302

Tab. 4: Beispiele

Die Tabelle enthält zu jedem Satz die jeweils richtige Klassifizierung und die Information, ob das CNN den Satz richtig klassifizieren konnte. Die letzte Spalte enthält die Ausgabe des Netzwerks für die beiden Klassen Information und Anforderung. Der jeweils größere Wert ist ausschlaggebend für die Ausgabe des Netzwerks. Je größer / kleiner eine Ausgabe, desto sicherer ist sich das Netzwerk, dass der Satz der jeweiligen Klasse angehört / nicht angehört.

Beispiel 1 und 2 konnten durch das CNN richtig klassifiziert werden. Anhaltspunkte für die richtige Klassifizierung dieser Beispiele sind möglicherweise die Wörter und Wortgruppen „muss“, „gemessen werden“ oder auch „entsprechend der Strategie“. Die Klassifizierung durch das Netzwerk ist bei diesen beiden Beispielen glaubwürdig, da die Ausgabewerte hinreichend groß bzw. klein sind.

Beispiel 3 wurde ebenfalls richtig klassifiziert. Die Ausgabe des Netzwerks für die Klasse Anforderung ist bei diesem Beispiel allerdings deutlich kleiner. Ursache dafür ist möglicherweise das Nichtvorhandensein ausschlaggebender Formulierungen im Satz. Die Glaubwürdigkeit dieses Ergebnisses muss daher hinterfragt werden.

Beispiel 4 ist eine Zusatzinformation, die durch das CNN mit großer Sicherheit als solche erkannt wurde. Angezeigt wird dies durch die sehr großen bzw. kleinen Ausgabewerte.

Beispiel 5 ist ebenfalls eine Zusatzinformation, die durch das Netzwerk fälschlicherweise als Anforderung eingestuft wird. Bei Betrachtung der Ausgabewerte fällt allerdings auf, dass keine der beiden Klassen auf diesen Satz zutreffen. Dieses Ergebnis deutet auf ungünstig gewählte Formulierungen des Satzes hin, oder auch auf eine zu kleine Trainingsdatenmenge.

Der in Beispiel 6 gegebene Satz wurde ebenfalls falsch klassifiziert, obwohl er große Ähnlichkeit zu dem sehr gut klassifizierbaren Beispiel 4 besitzt. Die Gründe dafür müssen weiter untersucht werden.

Anhand dieser Beispiele lässt sich erkennen, dass bei der Klassifikation durch CNNs in vielen Fällen erkennbar ist, ob das Ergebnis der Klassifikation glaubwürdig ist. Allerdings ist es unklar, welche Teile der Sätze für die jeweilige Klassifikation ausschlaggebend sind, da das Netzwerk hierüber keine Informationen preisgibt. Ob und wie solche Aussagen anhand des Netzwerks getroffen werden können, muss in weiteren Arbeiten genauer untersucht werden.

Die von keinem Schritt klassifizierten Objekte (22.6%) konnten nicht klassifiziert werden, da entweder der Strukturtyp nicht erkannt wurde, oder für den erkannten Strukturtyp kein Klassifizierungsschritt zur Verfügung steht.

## 6 Verwandte Arbeiten

Die Autoren von [HLR14] stellen ein graphisches Werkzeug zur Analyse von Anforderungen vor und haben für dieses Werkzeug mehrere Komponenten zur Klassifikation von Anforderungen entwickelt. Das Werkzeug kann unter anderem temporale Anforderungen von nicht temporalen Anforderungen unterscheiden. Auf Basis von Fallstudien zeigen die Autoren, dass Techniken aus dem Bereich Machine Learning zur Lösung von Problemen des Requirements-Engineering grundsätzlich anwendbar sind.

In [Ot13] wird ebenfalls ein Ansatz zur Klassifizierung von Anforderungen vorgestellt. In diesem Beitrag werden Naive-Bayes und Support Vector Machines zur Bestimmung des Themas von Anforderungen eingesetzt, um Reviewprozesse zu unterstützen.

## 7 Zusammenfassung und Ausblick

In diesem Beitrag wurde ein Ansatz zur Klassifikation der Objekte von Lastenheften in Anforderung und Information vorgestellt. Der Ansatz kombiniert unterschiedliche Klassifizierungstechniken, um eine möglichst genaue Klassifizierung zu gewährleisten.

Die vorgestellte Cluster-Klassifizierung sucht nach textuell ähnlichen, identisch klassifizierten Objekten und benutzt die daraus entstehenden Cluster zur Klassifizierung. Die erzielten Ergebnisse sind bereits sehr gut, ggf. können allerdings mit anderen Cluster-Techniken oder Textdistanzmaßen noch bessere Ergebnisse erzielt werden.

Ein weiterer Bestandteil des Ansatzes ist die Unterscheidung von Strukturtypen: Abhängig vom Format der in einem Objekt vorkommenden Inhalte wird ein geeigneter Klassifizierungsschritt ausgewählt. Dies wurde für den Strukturtyp Satz gezeigt. Es muss weiter untersucht werden, wie andere Strukturtypen klassifiziert werden können und wie die Erkennung unterschiedlicher Strukturtypen ggf. durch den Einsatz neuronaler Netzwerke verbessert werden kann, um noch mehr Objekte klassifizieren zu können.

Die Klassifizierung von Sätzen mittels Convolutional Neural Networks liefert akzeptable Ergebnisse. Größtes Verbesserungspotential liegt hier in der Trainingsdatenmenge: Durch weitere Vorverarbeitung und Filterung der Datenmenge kann das Netzwerk möglicherweise besser lernen.

Neben der Verbesserung der verwendeten Techniken ist der nächste Schritt die Integration in ein GUI-Werkzeug zur Benutzerunterstützung. Dieses soll den Benutzer auf falsch klassifizierte Objekte hinweisen. Für den Nutzer ist dabei neben einer hohen Genauigkeit wichtig, dass gegebene Hinweise zu falsch klassifizierten Objekten begründet werden. Insbesondere bei neuronalen Netzwerken ist eine derartige Begründung schwierig, da der Entscheidungsprozess innerhalb des Netzwerks sehr komplex ist. Im Rahmen einer Werkzeugintegration muss dieses Problem weiter untersucht werden.

## Literaturverzeichnis

- [AZ12a] Aggarwal, Charu C.; Zhai, ChengXiang: A Survey of Text Classification Algorithms. In (Aggarwal, Charu C.; Zhai, ChengXiang, Hrsg.): Mining Text Data, S. 163–222. Springer US, 2012.
- [AZ12b] Aggarwal, Charu C.; Zhai, ChengXiang: A Survey of Text Clustering Algorithms. In (Aggarwal, Charu C.; Zhai, ChengXiang, Hrsg.): Mining Text Data, S. 77–128. Springer US, 2012.
- [HLR14] Hayes, Jane H.; Li, Wenbin; Rahimi, Mona: Weka meets TraceLab: Toward Convenient Classification: Machine Learning for Requirements Engineering Problems: A Position Paper. In: 1st IEEE International Workshop on Artificial Intelligence for Requirements Engineering. AIRE, S. 9–12, 2014.
- [KGB14] Kalchbrenner, Nal; Grefenstette, Edward; Blunsom, Phil: A Convolutional Neural Network for Modelling Sentences. arXiv preprint, abs/1404.2188, 2014.
- [Ki14] Kim, Yoon: Convolutional Neural Networks for Sentence Classification. arXiv preprint, abs/1408.5882, 2014.
- [KM03] Klein, Dan; Manning, Christopher D.: Accurate Unlexicalized Parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1. ACL '03, Association for Computational Linguistics, Stroudsburg, PA, USA, S. 423–430, 2003.
- [Kw98] Kwok, James Tin-yau: Automated Text Categorization Using Support Vector Machine. In: In Proceedings of the International Conference on Neural Information Processing (ICONIP. S. 347–351, 1998.
- [Mi13] Mikolov, Tomas; Chen, Kai; Corrado, Greg; Dean, Jeffrey: Efficient Estimation of Word Representations in Vector Space. arXiv preprint, abs/1301.3781, 2013.
- [Ot13] Ott, Daniel: Automatic Requirement Categorization of Large Natural Language Specifications at Mercedes-benz for Review Improvements. In: Proceedings of the 19th International Conference on Requirements Engineering: Foundation for Software Quality. REFSQ, Springer-Verlag, Berlin, Heidelberg, S. 50–64, 2013.
- [ZW15] Zhang, Ye; Wallace, Byron C.: A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. arXiv preprint, abs/1510.03820, 2015.