Untersuchung verschiedener Modelle des maschinellen Lernens hinsichtlich ihrer Eignung zur Unterscheidung von Anforderungstexten zu anderen Texten

Timo Schwind **Dirk Hesse**

Fakultät Informatik und Informationstechnik, Hochschule Esslingen

Arbeit durchgeführt bei der Firma Vector Informatik GmbH, Stuttgart

Motivation

Da die meisten Daten unstrukturiert sind, bietet die Fähigkeit, sie zu analysieren und darauf zu reagieren, eine große Chance. [4] Unstrukturiert sind Daten dann, wenn sie in einer nicht formalisierten Struktur vorliegen und somit nicht direkt in SQL-basierten Datenbanken abgelegt werden können. Natural Language Processing (NLP) Methoden helfen dabei, die Bedeutung dieser Daten zu extrahieren. Dies ist gerade in der heutigen Zeit sehr wichtig, Schätzungen zufolge macht dieser Datentyp 80–90 % des gesamten digitalen Datenuniversums aus. Zu finden sind sie in vielerlei Formen, etwa in Dokumenten, E-Mails oder auch Journalen.

Zielsetzung

Das Ziel dieser Arbeit ist es, verschiedene Texte so zu verarbeiten, dass Anforderungen in ihm erkannt und herausgefiltert werden können. Dabei sollen unterschiedliche Modelle des maschinellen Lernens verwendet und miteinander verglichen werden. Als Grundlage dafür wird eine große Textsammlung, im Linguistikumfeld als Textkorpora bezeichnet, erstellt. Diese enthält firmeninterne Anforderungen, welche schlussendlich als solche identifiziert werden sollen.

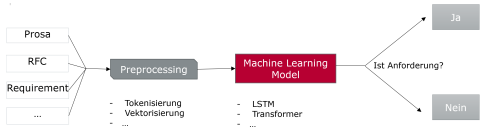

Abb. 1: Bildliche Darstellung der Aufgabe [1]

Um eine saubere Separierbarkeit und auch eine hohe Allgemeingültigkeit zu erreichen, werden den Textkorpora unterschiedliche Textformen hinzugefügt, welche nicht der Klasse der Anforderungen angehören. Beispielsweise wurden mithilfe von Web-Scrapping-Methodiken Requests for Comments (RFCs) in den Textkorpus übernommen. Diese technischen Dokumente sind von den Anforderungstexten aufgrund ihrer semantischen Ähnlichkeit für einen Algorithmus schwer zu unterscheiden.

Die Modelle sollen hinsichtlich ihrer Funktion genau untersucht werden, sodass das Modell nicht als Blackbox betrachtet wird. Mit Methoden der erklärbaren AI (Explainable AI) im NLP-Bereich wird nachvollzogen, welche Faktoren schlussendlich zu der Klassifikation geführt haben. Ein zusätzliches Ziel ist die Weiterverarbeitung der identifizierten Anforderungen. Dabei sollen diese in unterschiedliche Klassen, basierend auf deren Inhalt, unterteilt und zugeordnet werden. Die Klassenanzahl beträgt 15 und schlüsselt die Anforderungen weiter in Unterklassen auf. Beispiele für diese sind etwa 'Test cases' oder 'Terms and definitions'.

Natural Language Processing Überblick

Das Verwenden von maschinellen Lerntechniken ist heutzutage das übliche Vorgehen bei Anwendungen im NLP-Bereich. Dabei erlenen die Systeme auf Grundlage der Trainingsdaten in Form von Textkorpora Muster. Diese Muster entsprechen im Idealfall Sprachkonzepten, die bei der Bearbeitung bestimmter Problemstellungen helfen. Ein großes Problem im NLP-Bereich ist, dass Modelle nicht direkt mit dem Text arbeiten können. Die Satzbausteine müssen zunächst einer passenden und sinnhaften Vektorrepräsentation zugeordnet werden. Nach dieser Einbettung in den Vektorraum erfolgt eine Verarbeitung mit selbstlernenden Algorithmen. Besonders gut hierbei performen sequenzbasierte neuronale Netze, die die Reihenfolge der Wörter beachten. Dafür gibt es heutzutage verschiedene Herangehensweisen. Rekurrente neuronale Netze boten dafür die Grundlagen, welche über die Jahre stetig weiterentwickelt wurden. Aktuelle State-of-the-Art Modelle wie Googles BERT (Bidirectional Encoder Representations from

Transformers) basieren auf Transformerarchitekturen, welche den Aufmerksamkeitsmechanismus nutzen. Diese sind in der Lage, Sequenzen zu verarbeiten und komplexe Abhängigkeiten zu erkennen, die dem Modell dabei helfen, anspruchsvolle Aufgaben wie Textgenerierung oder Sprachübersetzung zu meistern.

Verwendete Klassifikationsmodelle

- **Random Forest Classifier:** Klassifikationsverfahren, welches mehrere unkorrelierte Entscheidungsbäume aus verschiedenen Dateneigenschaften erstellt. Diese Bäume wachsen während des Lernprozesses. Entschieden wird durch die Mittelwertbildung der Ergebnisse der einzelnen Entscheidungsbäume.

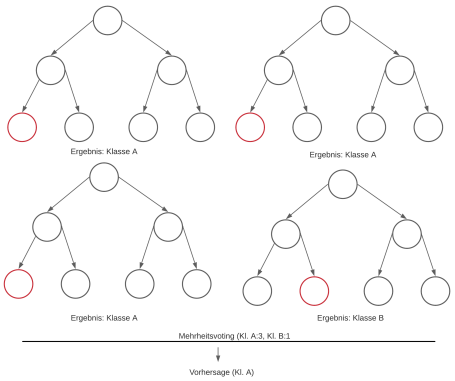

Abb. 2: Random Forest Classifier [1]

- **Naive Bayes Classifier:** Aus einer Reihe von Trainingsdaten werden Statistiken erhoben, sodass für ein neues Objekt festgestellt werden kann, wie es in diese Wahrscheinlichkeitsverteilungen hineinpasst. Der Naive Bayes Algorithmus geht davon aus, dass alle Attribute unabhängig voneinander sind.
- **Support Vector Classifier:** Das vektorbasierte Modell beschreibt (Hyper-) Ebenen, welche die gegebenen Trainingsdaten linear separieren. Ist dies in den Dimensionen der Daten nicht möglich, werden diese in einen höherdimensionalen Raum transformiert. Mithilfe einer Support-Vektor-Maschine wird ein Klassifikator gefunden, der die Datenpunkte möglichst gut voneinander trennt.
- **Feed forward neuronale Netze:** Die Topologie eines Feed Forward neuronalen Netzes lässt sich durch Knoten bzw. Recheneinheiten beschreiben, die ohne Zyklen über Schichten hinweg mit gerichteten und gewichteten Kanten verbunden sind. Jede Recheneinheit erhält und verarbeitet den gewichteten Output der Einheiten der vorherigen Schicht. Neuronale Netze sind dazu in der Lage, jede beliebige Funktion in beliebiger Genauigkeit zu approximieren. Diese Funktion kann dann zur Klassifikation verwendet werden.

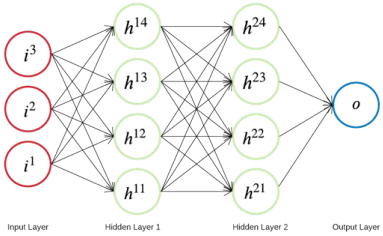

Abb. 3: Feed forward neuronales Netz [1]

- **Rekurrente Neuronale Netze und Long Short Term Memory Netzwerke (LSTMs):** Eine Erweiterung des Feed Forward Neuronalen Netzes, welche Zyklen innerhalb der Netzwerkverbindungen enthält. So können Sequenzen wie etwa Texte oder Sprache besser verarbeitet werden. Ein LSTM ist eine Form eines rekurrenten Netzes, welches Speicherzellen einführt, um Eingangssignale einzuordnen und entsprechend ihrer Bedeutung beurteilt. So können wichtige Informationen auf längere Distanz beibehalten werden.
- **Transformerarchitekturen:** Deep Learning Modelle, welche auf Zyklen verzichten und stattdessen für die Sequenzverarbeitung den sogenannten Aufmerksamkeitsmechanismus verwenden. Dieser prüft in jedem Schritt der Input Sequenz, welche anderen Teile des Satzes für den spezifischen Input von Bedeutung sind. Transformer funktionieren über einen Encoder-Decoder-Stack, wobei der Encoder die Sequenz zu einer internen Repräsentation verarbeitet. Diese wird anschließend vom Decoder benutzt, der letztendlich den Output liefert. Der von Google entwickelte und vortrainierte Encoder Bidirectional Encoder Representations from Transformers (BERT), der auch für deren Suchmaschine verwendet wird [3] , hat in vielen NLP-Benchmarks neue Maßstäbe gesetzt. [2]

Zwischenstand

Ersten Ergebnissen zufolge ist die Separierbarkeit der Anforderungen mit den beschriebenen Modellen sehr gut möglich. Ein Grund hierfür ist die spezifische Sprachdomäne der firmeninternen Anforderungen, welche, gekennzeichnet durch viele charakteristische N-Grams, die Zuordnung erleichtern. N-Grams sind aufeinanderfolgende Wörter in einem Dokument, beispielsweise "Machine Learning". Dies wäre ein Bigram, da die Anzahl der aufeinanderfolgenden Wörter $N = 2$ ist. Das in der Abbildung dargestellte Streudiagramm zeigt jeweils auf den Achsen die klassenspezifisch (Anforderung/keine Anforderung) häufig auftretenden N-Grams (Unigram, Bigram, Trigram), welche durch Punkte dargestellt werden. Häufig in der jeweiligen Klasse auftretende Worte sammeln sich im "Frequent"-Bereich der dazugehörigen Achse. Besonders charakteristisch sind sie, wenn ihre Auftrittsfrequenz nur bei einer Klasse hoch ist. Dies ist der Fall bei den Punkten, welche sich nah an den Achsen sammeln.

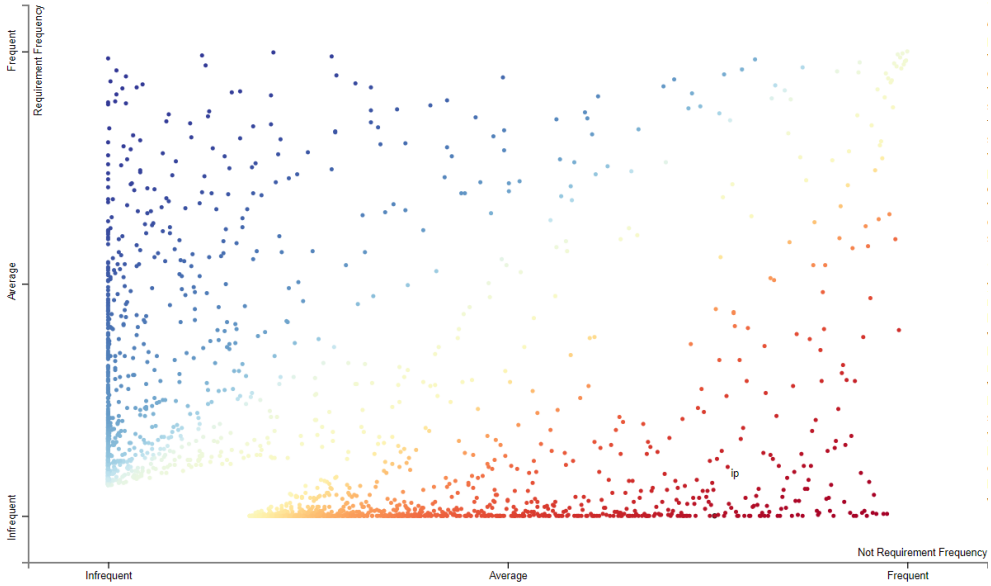

Abb. 4: Streudiagramm der Worthäufigkeiten pro Klasse [1]

Das Modell kann unter anderem durch das Erlernen eben solcher charakteristischen Stichwörter oder der Analyse semantischer Eigenschaften eine Klasseneinteilung vornehmen. Dies führt zu einer durchschnittlichen Testgenauigkeit von 99,2% auf dem Validierungsdatensatz, erreicht mit BERT als Encoder und einem kleinen neuronalen Netz, welches mit dem Encoderoutput arbeitet und auf die spezifische Klassifikationsaufgabe trainiert wurde.

Ausblick

Die Arbeit liefert die Grundlage für eine automatisierte Verarbeitung von Anforderungen. Fernziel ist es, durch künstliche Intelligenz Anforderungsdokumente zu verarbeiten. Dadurch soll eine Prüfung der Qualität realisiert werden, mit der Fehler erkannt und infolgedessen auch Verbesserungsvorschläge erstellt werden können. Durch die vorliegende Arbeit wird die grundlegende Struktur der Anforderungen erkannt und untersucht. Weitergehende Arbeiten können diese nutzen, um daraus konkrete Maßnahmen zur Verbesserung abzuleiten.

Literatur und Abbildungen

[1] Eigene Darstellung.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186. Association of Computational Linguistics, 2019.

[3] Pandu Nayak. Understanding searches better than ever before. https://blog.google/products/search/search-language-understanding-bert/, 10 2019.

[4] Mikey Shulman. Tapping the power of unstructured data. https://mitsloan.mit.edu/ideas-made-to-matter/tapping-power-unstructured-data, 02 2021.