INWT Statistics

Automatische Content-Kuration für Presseportale

Die Modellierung des User Engagement mittels Natural Language Processing

Susanna Rücker & Steffen Wagner

Zum Tagesgeschäft einer Redaktion gehört neben dem Schreiben von Artikeln das Kuratieren derselben. Darunter fallen unter anderem die Auswahl, die Positionierung und das *Pricing* der Artikel. Diese Entscheidungen basieren auf Grundlage der Beurteilung von Relevanz und erwartetem Interesse der Leser*innen:

- Welcher Content soll den Leser*innen wo und wie im Online-Portal angeboten werden?
- Gehört ein Artikel auf die Titelseite oder eignet er sich nur für einen kleinen Teil der Leserschaft und sollte daher auf einer anderen Seite untergebracht werden?
- Soll ein Artikel öffentlich oder nur für Abo-Kund*innen zugänglich sein?
- Bei welchen Artikeln lässt sich möglichst gewinnbringend Werbung schalten?

In dieser Einschätzung und der Optimierung der qualitativen Größen Relevanz und Leser*innen-Interesse liegt die zentrale Herausforderung bei der Aussteuerung der angebotenen Inhalte. Denn die genannten *Key Performance Indicators* (KPIs) haben einen direkten Einfluss auf die *Performance* eines Artikels. Artikel sollen interessieren, also Inhalte fokussieren, die auf die Bedürfnisse der Leser*innen zugeschnitten sind. Die Bemessung der genannten qualitativen Größen erfolgt oft durch subjektive Einschätzung, Erfahrung und Bauchgefühl. Doch moderne Machine-Learning-Methoden eröffnen mittlerweile noch einen zweiten, datengetriebenen Weg: die Modellierung des Zusammenhangs zwischen Artikelinhalten und ihrer *Performance*. Das entsprechende Teilgebiet von Machine Learning, das sich mit der Verarbeitung von Text bzw. Sprache beschäftigt, ist das Natural Language Processing (NLP).

Der Mehrwert des datengetriebenen Ansatzes liegt in der Prozessoptimierung innerhalb von Redaktionen. Automatisierte Machine-Learning-Verfahren sind in der Lage, die Arbeit von professionellen Mitarbeiter*innen enorm zu unterstützen. Beispielsweise können sie eine Vorauswahl von Artikeln treffen, Vorschläge zeitlicher Planung und Platzierung liefern oder sogar im Editierungsprozess einzelner Artikel behilflich sein.

Das vorliegende White Paper beschreibt die Modellierung der Performance von Artikeln mithilfe von Natural Language Processing (NLP). Im Folgenden geben wir zunächst einen Überblick über Methoden aus dem NLP. Anschließend führen wir durch den *Use Case*. Wir stellen die Datengrundlage vor, die als Input für eine Modellierung benötigt wird, in der anhand von Artikeltexten die *Performance* in Form von Verweildauer prognostiziert wird. Wir beschreiben die Ergebnisse und liefern einige

Einblicke zur Interpretierbarkeit auf Textebene im Sinne von *Explainable AI* (XAI).

Der Begriff **Artificial Intelligence** (AI) umfasst das Automatisieren des menschlichen Denkens und Handelns. **Machine Learning** (ML) ist ein Teilgebiet von AI und beschäftigt sich damit, dem Computer selbstständige Entscheidungsfindung oder das Erkennen von Regeln und Zusammenhängen beizubringen. **Deep Learning** (DL) ist ein Machine-Learning-Gebiet, das sich neuronaler Netze zum Anlernen des Computers bedient. **Natural Language Processing** (NLP) ist ebenso ein Teilgebiet von Machine Learning, das sich mit der computergestützten Verarbeitung von Sprache beschäftigt. Verwandte Begriffe zu NLP sind Text Mining, Computerlinguistik oder automatisierte Sprach- oder Textverarbeitung.

Natural-Language-Processing-Methoden

Textdaten (Artikel, Kommentare, etc.) stellen eine wertvolle Informationsquelle dar: Sie enthalten Bewertungen zu Produkten, sie zeigen unterschiedliche Meinungen zu Gesellschaftsthemen, sie vermitteln Wissen. Natural Language Processing bietet eine Möglichkeit diese Information automatisiert nutzbar zu machen. Typische Anwendungsbereiche von NLP sind – um nur ein paar zu nennen – die Klassifikation von Texten (z.B. Spam oder nicht Spam, positive oder negative Rezension, Erkennung des Topics), maschinelles Übersetzen, stetig verbesserte und flexiblere Bearbeitung von Suchanfragen, automatisierte Rechtschreibung- und Grammatikprüfungen, Sprachassistenten wie Alexa und Siri, Chatbots, Textgenerierung, automatisierte Untertitel und Zusammenfassungen, Erstellen von Knowledge Bases, logisches Schlussfolgern, automatisierte Erkennung und Anonymisierung von personenbezogenen Daten oder medizinischen Fachtermini, Erkennung und Kennzeichnung von Hate Speech oder Missinformation. Deutlich wird: Natural Language Processing und die zugehörigen Methoden und Anwendungen sind in der heutigen Zeit nicht mehr wegzudenken, denn sie betreffen einen großen Teil der menschlichen Lebenswirklichkeit: Sprache ist allgegenwärtig.

Da Machine-Learning-Modelle auf numerische Eingaben angewiesen sind, sind ihnen Texdaten in ihrer Rohform zunächst nicht zugänglich. Ein wichtiger Teil von NLP-Methoden besteht daher darin, den Text für die Modelle zugänglich zu machen, ihn also in ein numerisches Format zu übersetzen, das den Inhalt des Textes repräsentiert. Diese numerische Repräsentation von Sprache und ihrer Bedeutung macht es dann möglich, Machine-Learning-Fragestellungen zu lösen, beispielsweise eine Prognose darüber zu erstellen, wie gut Artikel bei den Leser*innen ankommen.

Für dieses Überführen in numerische Formate gibt es verschiedene Ansätze, die sich in ihrer Komplexität unterscheiden und damit auch in ihrer Fähigkeit, komplexe Inhaltszusammenhänge abzubilden. Während einfache Ansätze auf Worthäufigkeiten setzen, die aber die tatsächliche Semantik von Sprache nicht transportieren können, zielen andere Verfahren darauf ab, die Bedeutung des Textes bzw. einzelner Textbausteine abzubilden.

Im Folgenden werden die wichtigsten Konzepte im Bereich Natural Language Processing vorgestellt und ihre jeweiligen Stärken und Schwächen diskutiert, eine Übersicht ist in Tabelle 1 zu finden.

NLP-Methoden

NLP-Methoden: Bag of Words

Ein einfacher NLP-Ansatz ist die Verwendung von Worthäufigkeiten: Ein Text wird dabei als Menge der in ihm enthaltenen Wörter betrachtet, was diesem Verfahren die Bezeichnung *Bag of Words* (BOW) verleiht. Die betrachtete Einheit wird als *Token* bezeichnet, dies können neben einzelner Wörter auch Wortgruppen (N-Gramme) oder Satzzeichen sein. Ein Textdokument wird also durch einen Vektor mit einzelnen Tokenhäufigkeiten dargestellt.

Das Bag-of-Words-Verfahren hat einige Vorteile gegenüber komplexeren Verfahren: BOW-Vektoren können inhaltliche oder stilistische Ähnlichkeiten zwischen Texten erstaunlich gut abbilden und sind damit für einige einfache Anwendungen eine gute Wahl. Ihr entscheidender Vorteil liegt aber – neben der schnellen und ressourcensparenden Berechnung – in der hervorragenden Interpretierbarkeit. Gleichzeitig macht man es sich mit BOW-Verfahren oft zu leicht, da einige Textinformationen nicht berücksichtigt werden können: Die syntaktische Abfolge, das Zusammenspiel von Bedeutungen, die sich nur im Kontext ergeben, oder Mehrdeutigkeiten werden nicht erfasst. Bei komplexeren Fragestellungen kann ein BOW-Ansatz also lediglich

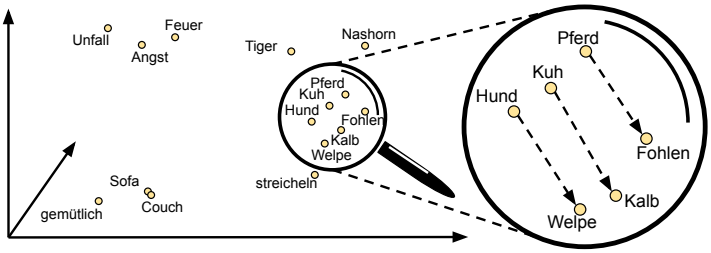

Abbildung 1: Veranschaulichung von Word Embeddings im semantischen Vektorraum.

als Ausgangspunkt dienen, dann sollten Ansätze in Erwägung gezogen werden, die weitere textuelle Ebenen wie die Bedeutung der Sprache erfassen können.

NLP-Methoden: Word Embeddings

Im Gegensatz zum Bag-of-Words-Ansatz erfassen *Word Embeddings* die Bedeutungen der Wörter selbst. Die Bedeutung wird als ein semantischer Vektorraum abgebildet, d.h. Wörter mit ähnlicher Bedeutung liegen in diesem hoch-dimensionalen Raum näher beieinander und Wörter mit unterschiedlicher Bedeutung weiter entfernt voneinander.

Diese Repräsentation von Texten ermöglicht ein gewisses semantisches Verständnis seitens des Modells. In Abbildung 1 ist ein solcher Vektorraum stark vereinfacht dargestellt. Es zeigen sich aber grundlegende Eigenschaften von Word Embeddings:

- Bezeichnungen für eine inhaltlich zusammenhängende Gruppe von Begriffen, hier Tiere (*Hund*, *Pferd*, *Tiger*), liegen in räumlicher Nähe, aber weit entfernt von thematisch sehr unterschiedlichen Begriffen, wie etwa *Sofa*. Auch die drei negativ konnotierten Begriffe *Unfall*, *Angst*, *Feuer* bilden ein gemeinsames Cluster.
- *Sofa* und *Couch* liegen sehr eng beieinander, was die nahezu identische Bedeutung der Begriffe widerspiegelt. Das Wort *gemütlich* findet sich in der Nachbarschaft hierzu.
- Die Relationen, also Abstand und Richtung der Begriffe *Hund–Welpe*, *Kuh–Kalb* und *Pferd–Fohlen* sind fast identisch. Word Embeddings sind also in der Lage, einen Zusammenhang der Form "*B* ist die Bezeichnung für ein junges *A*" zu identifizieren und in der numerischen Darstellung zu berücksichtigen.

Word Embeddings werden von einem ML-Modell anhand großer Sammlungen von unverarbeitetem Text erlernt. Das zentrale Verständnis von Bedeutung besteht in der Annahme, dass Wörter mit ähnlicher Bedeutung tendenziell in ähnlichen Kontexten auftreten. Ihre Verwendung ermöglicht, dass Modelle tatsächlich verblüffend gut Bedeutungszusammenhänge berücksichtigen können.

2013 stellte Google den berühmten **Word2Vec-Algorithmus** vor. Seitdem sind Word Embeddings in der Sprachverarbeitung in aller Munde. Für viele Sprachen stehen allgemeinsprachliche vortrainierte *Word Embeddings* öffentlich zur Verfügung (fastText, GloVe, ELMo) und können direkt als Features etwa in statistischen Modellen verwendet werden.

Zwei Aspekte von *Word Embeddings* sind allerdings nicht optimal: Erstens lassen sich die Entscheidungen von Modellen, die *Word Embeddings* als Eingabe enthalten, im Gegensatz zu BOW-Modellen nur schwer interpretieren. Zwar ist die Gesamtbedeutung eines Wortes im Vektor kodiert, die einzelnen Elemente sind aber nicht anschauliche Inhalte, sondern abstrakte Koordinaten in einem Vektorraum. Zweitens operieren *Word Embeddings* analog zum *Bag of Words* maßgeblich auf Wortebene. Das ist besonders dann ein Problem, wenn die Texte Wörter enthalten, die je nach semantischem und syntaktischem Zusammenhang unterschiedliche Bedeutungen aufweisen und daher entsprechend unterschiedliche numerische Repräsentationen erhalten sollten. Diese Lücke lässt sich durch Hinzunahme der nachfolgend vorgestellten Deep-Learning-Ansätze schließen.

NLP-Methoden: Deep Learning

Deep-Learning-Modelle (neuronale Netzwerke) sind aktuell der Standard im Natural Language Processing. Sie schließen an die beschriebene Word-Embedding-Repräsentation von Texten an und nutzen diese als Input. Neuronale Netze lassen sich in unterschiedliche Klassen einordnen, die bestimmte Aspekte der Sprache gezielt berücksichtigen und modellieren. Der Output eines neuronalen Netzes, eine äußerst komplexe numerische Repräsentation von Text, wird dann verwendet um eine konkrete Fragestellung zu beantworten, wie z.B. die Vorhersage eines KPIs, einer Textrubrik oder auch eine Zusammenfassung des ursprünglichen Textes. Die Bausteine der neuronalen Netze (sog. Schichten) basieren auf einer Vielzahl von mathematischen Transformationen und Verknüpfungen, deren zunächst unbekannte Parameter im Modell-Training quantifiziert werden.

Abbildung 2 illustriert drei für den Bereich NLP besonders relevante Netzwerkklassen. Diese unterscheiden sich darin, wie sie die Eingabesequenz einlesen, wie die Informationsverarbeitung im Innern abläuft, und ob sie in der Lage sind, Struktur und Querverbindungen im Text zu erkennen. Der Beispielsatz in Abbildung 2 enthält zur Verdeutlichung zwei Schwierigkeiten, denen bei der Sprachverarbeitung üblicherweise begegnet werden muss: trennbare Verben (*einbrechen* → *brachen … ein*) und Wörter mit verschiedenen Bedeutungen (*Bank*:

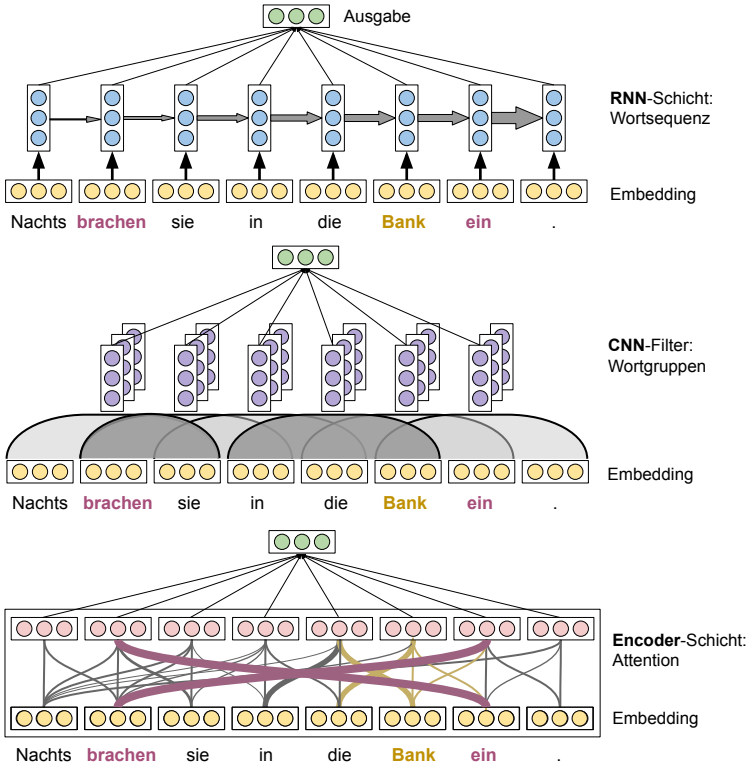

Abbildung 2: Beispielhafte Textverarbeitung mit RNN, CNN und Transformer-Encoder.

Kreditinstitut vs. Möbelstück), die kontextspezifisch aber klar identifizierbar sind.

Diesen Herausforderungen begegnen die drei Modellarten in unterschiedlicher Weise. *Recurrent Neural Networks* (RNNs) lesen die Eingabe schrittweise ein und erneuern dabei stets die interne Repräsentation auf Basis der neuen Eingabe[1]. Wörter stehen also nicht losgelöst voneinander, es können semantische und syntaktische Verbindungen zwischen ihnen und Kontextin-

[1] RNN-Modellierungen eignen sich vor allem für *Sequence-to-Sequence*-Anwendungen (z.B. maschinelles Übersetzen, Zusammenfassen, Tagging, Bestimmung von Wortarten oder *Named Entities*). Ein Problem von RNNs ist, dass sie dazu neigen, Information über *weite* Abstände zu vergessen. Es gibt RNN-Erweiterungen, die dieses Problem angehen: *Long Short-Term Memory* (LSTMs) und *Gated Recurrent Units* (GRUs) ermöglichen Direktverbindungen für besonders relevante Information, was dem zusammengesetzten Verb *brachen … ein* zugute kommt.

formationen einfließen: Die Doppelbedeutung von *Bank* kann etwa durch die Nachbarschaft mit *einbrechen* aufgelöst werden, vorausgesetzt der Abstand ist nicht zu groß. Eine weitere klassische Modellart sind *Convolutional Neural Networks* (CNNs)[2]. Ihre Spezialität ist die Verwendung von sog. Filtern, die die Eingabe scannen und auf relevante Muster untersuchen. Bei der Textverarbeitung bedeutet dies die gemeinsame Betrachtung von Wortgruppen: So kann etwa *in die Bank* gemeinsam verarbeitet werden. Das zusammengesetzte Verb hat aber nach wie vor keine Chance, da der Abstand zu groß ist. Über die Wortgruppen hinaus kann die sequenzielle Abfolge nicht berücksichtigt werden.

Viele dieser Probleme werden durch die dritte, aktuell erfolgreichste, Architektur im Natural Language Processing behandelt: der **Transformer**. Ursprünglich für maschinelle Übersetzung entwickelt, dominiert der Encoder-Teil dieser Architektur zur Zeit jede NLP-Anwendung und stellt andere Ansätze in den Schatten. Statt sequenzieller Betrachtung (wie bei RNNs) oder Erkennen von lokalen Mustern (CNN) ist das zentrale Prinzip zur Erkennung von Beziehungen innerhalb der Eingabe die sogenannte *Self-Attention*: Jedes Wort wird dabei mit jedem anderen Wort der Eingabe in Beziehung gesetzt, sodass eine Art Verbindungs- oder Relevanzstärke ermittelt werden kann. Im Beispiel erkennt das Modell etwa den starken Zusammenhang von *brachen* und *ein*. So können Zusammenhänge von Wörtern auch in großem Abstand erkannt und für die Weiterverarbeitung genutzt werden: Die Repräsentation eines Wortes enthält sinnvoll gebündelte Information aus allen Wörtern der Eingabe.

Deep Learning: Transfer Learning mit BERT

Ein zentrales Konzept im Deep Learning ist *Transfer Learning*. Gemeint ist die Nutzung von auf sehr großen Datensätzen vortrainierten Modellen, die dadurch bereits über ein allgemeines Sprachwissen verfügen. Wie in Abbildung 3 konzeptionell dargestellt, wird dieses bestehende, in aller Allgemeinheit vortrainierte Modell auf einen spezifischen Use Case angepasst und auf den eigenen Daten weitertrainiert. Die eigene Modellierung profitiert so von dem allgemeinen Vorwissen des Modells und könnte als spezifische Weiterbildungsmaßnahme betrachtet werden.
Das aktuell prominenteste unter den verfügbaren vortrainierten NLP-Modellen und Vertreter der Transformer-Architektur ist **BERT**: Das *Pretraining* bestand in der Aufgabe, benachbarte Wörter und Sätze möglichst gut vorhersagen bzw. erkennen zu können. Ein deutschsprachiges BERT-Modell wurde bspw. unter

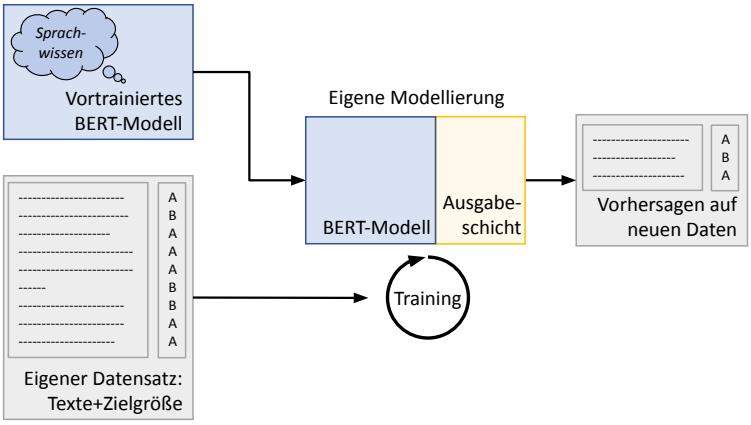

Abbildung 3: Der Ansatz im Transfer Learning: Anwendung eines vortrainierten Sprachmodells.

anderem auf der gesamten deutschen Wikipedia, sowie einem sehr großen News-Datensatz (12GB rohe Textdaten, entspricht ca. 1.7 Milliarden Tokens) neun Tage lang trainiert, das Modell umfasst ca. 110 Mio. Parameter und ist öffentlich verfügbar[3].

Fazit: die Ansätze im Vergleich

Die Stärken und Schwächen der vorgestellten Ansätze im Natural Language Processing, *Bag of Words*, *Word Embeddings* und *Deep Learning*, sind zusammenfassend in Tabelle 1 aufgeführt. Neuronale Netze sind sehr mächtige Modelle und überzeugen vor allem durch ihre gute Prognosegüte. Ein zentraler Nachteil gegenüber dem einfachen BOW-Ansatz ist die unter dem Stichwort *Black Box* bekannte Problematik: Die Wirkungsweise des Algorithmus und der Einfluss einzelner Textelemente sind nur schwer nachvollziehbar. Während bei klassischeren Ansätzen (z.B. lineare Regression oder Entscheidungsbäume basierend auf BOW-Features) die Wirkungsweise direkt an einzelnen Textelementen festgemacht werden kann, ist dies bei Deep-Learning-Modellen nicht möglich. Außerdem führt die große Zahl an Parametern von Deep-Learning-Modellen die meisten Computer schnell an ihre Grenzen, sie benötigen sehr viel Rechen- und Zeitressourcen und sind für eine gute Anpassung auf recht große Datensätze angewiesen. Der Vorteil von *Transfer Learning* ist dabei aber: Es werden sehr viel weniger Trainingsdaten benötigt. Mithilfe von modernen NLP-Bibliotheken kann in wenigen Schritten ein funktionstüchtiges Modell erzeugt werden!

Es muss individuell nach Use Case und Datengrundlage abgewogen werden, welcher Grad an Komplexität angemessen ist und ob ein einfacher Ansatz ausreicht.

[2] CNNs sind vor allem aus der Bildverarbeitung bekannt, spiel(t)en aber auch im NLP eine Rolle.

[3] `https://huggingface.co/bert-base-german-cased`

NLP-Ansatz	Berücksichtigte Textinformation	direkte Interpretierbarkeit	Ressourcenverbrauch	Vorhersagegüte
BOW	Worthäufigkeiten, kein Kontext, keine Struktur	sehr gut	gering	mittel
Word Embeddings (vortrainiert)	Wortsemantik enthalten, keine Beachtung von Kontextabhängigkeiten	mittel	mittel	mittel
DL-Modelle				
a) RNN/CNN	Wortsemantik, Wortgruppen, Kontext u. Satzstruktur fließt ein	nein	hoch	gut
b) Transformer, vortrainiert (BERT)	allgemeines "Sprachverständnis", Beachtung von komplexen Zusammenhänge zwischen Wörtern möglich	nein	sehr hoch	sehr gut

Tabelle 1: NLP-Ansätze im Überblick: Stärken und Schwächen.

Use Case: Modellierung der Performance von Zeitungsartikeln

Anhand eines Use Cases zeigen wir, wie die *Performance* von Artikeln des Online-Zeitungsportals der **Neuen Osnabrücker Zeitung** (NOZ) mithilfe der dargestellten Methoden modelliert und verstanden werden kann.

Datenbasis

Die Analyse basiert auf ca. 36.000 Artikeln der NOZ. Zusätzlich zu den Texten selbst wird die Information benötigt, welche Performance die einzelnen Artikel erzielt haben und welche *Key Performance Indicators* (KPIs) zur Messung der Performance herangezogen werden können. Mithilfe von On-Site-Tracking können anonymisiert aggregierte Informationen über das Verhalten der Nutzer*innen auf Webseiten erhoben und für die Analyse verwendet werden. Als Kennzahlen eignen sich KPIs wie z.B. Seitenaufrufe, ggf. differenziert nach Ein- und Ausstiegen, *Entrances* und *Exits*. Eine weitere häufig verwendete Kennzahl ist die *Click-Through-Rate* (CTR), also das Verhältnis aus Einblendungen und tatsächlichen Aufrufen eines Artikels. Da Klicks allerdings eher ein Maß für die erste und eventuell auch nur kurzfristige Aufmerksamkeit der Leser*innen sind, enthalten sie nur begrenzte Information über die echte Zufriedenheit über den Inhalt eines Artikels. Unter dem Stichwort *Clickbait* ist dieses Phänomen allgemein bekannt. Andere, explizite Größen wie zum Beispiel die Anzahl von Kommentaren oder Likes pro Artikel gäben zwar mehr Aufschluss, liegen aber nur selten oder nur für wenige Artikel vor.

Für die Messung von Performance im Sinne von Interesse am Artikelinhalt hat sich die Verweildauer (*Dwell Time*) als sehr guter Indikator etabliert. D.h. je länger eine Person auf der Seite mit dem Artikelinhalt verweilt, desto höher wird ihr Interesse am Inhalt interpretiert. Im vorliegenden Use Case liegen uns keine individuellen Daten von spezifischen Nutzer*innen vor, sondern aggregierte Werte über alle Besuche eines Artikels. Wir modellieren also anhand der durchschnittlichen Verweildauer die generelle Performance eines Artikels, nicht das individuelle Interessenprofil einzelner Leser*innen. Sollten individuelle Verweildauern zur Verfügen stehen, kann die im folgenden vorgestellte Analyse problemlos auf die granularere Datenbasis ausgeweitet werden.

Neben dem zu modellierenden KPI, der sog. Zielvariablen, ist zu überlegen, ob über die Artikeltexte hinaus weitere Informationen zur Analyse herangezogen werden können. Im Use Case sind dies die Länge eines Textes und die Rubrik des Artikels. Die Berücksichtigung dieser weiteren Features erlaubt es, unterschiedliche Effekte zu separieren und z.B. den Einfluss der Textlänge von dem des Textinhalts auf die Verweildauer zu unterscheiden. Die Aufbereitung der Textdaten für die Modellierung ist unkompliziert, da die Verwendung des Tranformer-Modells BERT so gut wie keine weitere Vorverarbeitung der Textdaten erforderlich macht.

Modellierung

Bei der Repräsentation des Textes wird auf *Transfer Learning* gesetzt. Verwendet wird daher ein auf deutschen Texten vortrainertes BERT-Modell. Dieses und viele weitere vortrainierten Transformer-Modelle werden von der *transformer*-Bibliothek von *Huggingface* bereitgestellt. An die BERT-Repräsentation schließen wir ein eigenes Ausgabemodell (konkret ein basales *Feed Forward Network*) an, welches die Verweildauer direkt als zu modellierende Zielgröße beinhaltet. Die Separation der Einflüsse von Textlänge und Rubrik sind dem BERT-Modell vorgelagert. Anhand der Trainingsdaten lernt das BERT-Modell also, systematische Zusammenhänge zwischen Inhalt und Verweildauer zu quantifizieren, die nicht al-

R^2	Güte
< 0.2	gering
$0.2 \leq R^2 \leq 0.35$	gut
> 0.35	sehr gut

Tabelle 2: Die Güte des Bestimmtheitsmaßes ist abhängig von der Datenbasis. Bei mikroskopischen, heterogenen Daten wie im vorliegenden Use Case ist ein R^2 von bis zu 0.5 realistisch.

lein durch Länge und Rubrikzugehörigkeit der Texte erklärbar sind.

Prognostizierbarkeit der Verweildauer

Unter Verwendung eines klassischen *Train-Test-Splits* der Daten wird auf zunächst unberücksichtigten Testdaten (ca. 10%) quantifiziert, wie gut das trainierte Modell in der Lage ist, die Verweildauer zukünftiger Texte zu prognostizieren. Die Prognosegüte wird anhand des Bestimmtheitsmaßes R^2 beurteilt, welches angibt, welcher Anteil der beobachteten unterschiedlichen Verweildauern durch Inhalt, Länge und Rubrik der Texte erklärbar ist.

Ergebnisse

Der Use Case zeigt, dass die Verweildauern der Artikel gut prognostizierbar sind. So lässt sich insgesamt 36% der Schwankung der Verweildauern durch die Modellierung erklären. Dies ist ein guter Wert, berücksichtigt man das große Spektrum an möglichen Artikelinhalten und die Modellierung auf mikroskopischer Ebene der einzelnen Texte. Über alle Artikel und Rubriken hinweg setzt sich die Prognosegüte hierbei zu 58% aus Informationen über die Textlänge und Artikelrubrik und zu 42% aus dem eigentlichen Artikeltext zusammen.

Allerdings unterscheiden sich die Prognosegüte und der Einfluss der Textinhalte sehr stark zwischen den einzelnen Textrubriken. Die besten Vorhersagen werden dabei in den Rubriken *Nordrhein-Westfalen* ($R^2 = 0.53$), *Niedersachsen* ($R^2 = 0.46$), *Medien* ($R^2 = 0.40$) und *Politik* ($R^2 = 0.39$) erzielt. In anderen Rubriken, wie z.B. *Kultur* ($R^2 = 0.05$) ist die Prognosegüte sehr gering. Einen Überblick über die pro Rubrik erzielten Ergebnisse gibt Abbildung 4. Es wird deutlich, dass nicht nur der gesamte Erklärungsgehalt zwischen den Rubriken stark variiert, sondern dass auch der Einfluss des eigentlichen Textinhalts je nach Rubrik unterschiedlich dominant ist. So ist der Anteil der Inhalte am Erklärungsgehalt z.B. in den Rubriken *Nordrhein-Westfalen* (63%) und *Politik* (73%) sehr hoch, hingegen in der Rubrik *Sport* (22%) deutlich geringer.

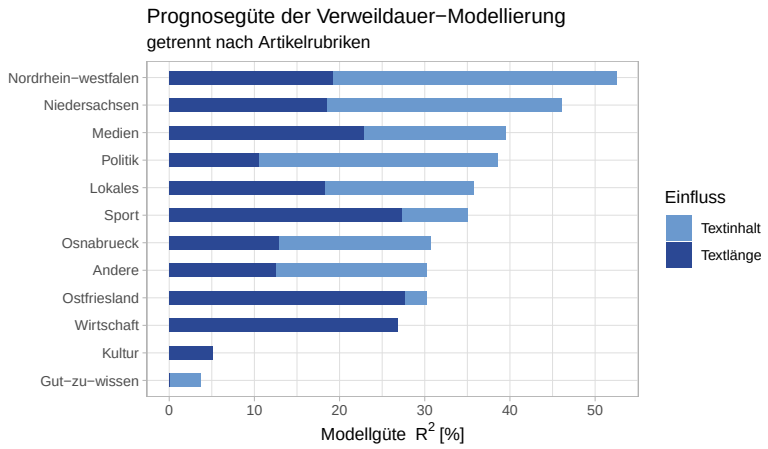

Abbildung 4: Überblick über die erzielten Prognosegüten R^2 getrennt nach Rubrik. Aus der Abbildung wird deutlich, dass der Beitrag der Textinhalte zum Erklärungsgehalt zwischen den Rubriken stark unterschiedlich ist.

Ein weiteres Ergebnis der Modellierung ist, dass bereits allein mit den Inhalten am Artikelanfang die Verweildauern überzeugend vorhergesagt werden können. Dies ist ein Hinweis darauf, dass der Artikelanfang entscheidend für die Performance ist und Inhalte am Artikelende für die Verweildauer eher von nachgelagerter Bedeutung sind.

Es sei angemerkt, dass die vorgestellten Ergebnisse allein auf der Verwendung von textueller Informationen beruhen, also noch keinerlei weitere Angaben wie etwa das Vorhandensein von Bildern, Werbung oder Kommentarspalten – die starken Einfluss auf die Verweildauer haben können – berücksichtigen. In Anbetracht der komplexen Fragestellung zeigen die Ergebnisse daher bereits sehr deutlich, dass die Texte selbst schon genügend Information beinhalten, um deren Performance nach der Ausspielung vorherzusagen. Damit stellt das trainierte Modell die Basis für eine automatisierte und optimierte Aussteuerung von Artikelinhalten dar.

Modelle aus der klassischen Statistik weisen eine extrem gute **Interpretierbarkeit** auf. Dafür kommen sie bei der **Modellgüte** oft nicht ganz an komplexere Machine-Learning-Ansätze heran. Aufgrund der höheren Komplexität sind ML-Modelle aber eine **Black Box** – d.h. der Zusammenhang zwischen den Einflussgrößen und der aus dem Modell resultierenden Entscheidung ist ohne weiteres nicht direkt interpretierbar. Auf den Use Case bezogen lässt sich also nicht erschließen, warum ein bestimmter Text viel höhere Performance aufweist als ein anderer.

Tieferes Verständnis mit Explainable AI

Der beschriebene NLP-Ansatz kann die Verweildauer eines Artikels gut modellieren. Die Problematik der *Black Box* verwehrt allerdings zunächst genauere Einblicke oder Schlussfolgerungen auf Textebene, also Antworten auf Fragen wie z.B.: **Warum** wird für einen bestimmten Text eine hohe bzw. geringe Verweildauer vorhergesagt? Das Gebiet der *Explainable AI* beschäftigt sich mit der Interpretierbarkeit von Black-Box-Modellen. Mithilfe von *SHAP* – einem sehr mächtigen Verfahren zur Bestimmung der Relevanz bestimmter Features – werfen wir im Folgenden einen Blick hinter die Kulissen der eingesetzten Machine-Learning-Algorithmen und schlüssel auf, welche Texteigenschaften Einfluss auf die Performance der Artikel haben.

Features sind im Falle von Textdaten die einzelnen Wörter (genauer die daraus abgeleitete Tokens), die in die NLP-Modellierung eingehen. *SHAP* fährt einen lokalen Interpretationsansatz, untersucht also für jeden einzelnen Text, welche Relevanz die im Text enthaltenen Features auf die Prognose haben. Die aggregierte Betrachtung dieser *Einzeltext*-Ergebnisse lassen dann Rückschlüsse auf generelle Tendenzen zu, die für alle Texte gemeinsam gelten.

Interessant ist diese Betrachtung sowohl über alle Texte als auch innerhalb einzelnen Textrubriken. Über alle Rubriken hinweg zeigt sich, dass Zitate (identifizierbar durch die verwendeten Anführungszeichen) einen positiven Einfluss auf die Verweildauer aufweisen. Innerhalb einzelner Rubriken lassen sich besonders relevante Tokens identifizieren, die bei Leser*innen verstärkt Interesse wecken und mit längeren Verweildauern einherge-

Rubrik	Bereich	Relevante Tokens
politik	Parteipolitik	*cdu, spd, csu, 2017*[4]*, linke, angela*
	Außenpolitik	*eu, usa, donald*
osnabrueck	Stadtgeschehen	*zoo, kita, hochschule, ziel, uhr, preis*
gut-zu-wissen	Medizin	*arzt, schwitzen, tod*
	Ernährung	*ei, fisch, gemüse*
	Familie/Haushalt	*tier, baby, brett, euro, kosten*

Tabelle 3: Relevante Schlagworte, die zu höheren Verweildauern führen, für ausgewählte Rubriken.

hen. Beispielhaft sind hierfür Ergebnisse in Tabelle 3 aufgeführt.
Der SHAP-Ansatz ermöglicht es auch, gezielt den Einfluss einzelner Begriffe oder Themen zu untersuchen. In der Rubrik *Politik* ergibt die Suche nach in den Texten genannten politischen Parteien und ihrem Einfluss auf deren Verweildauern das in Abbildung 5 dargestellte Ergebnis: Das Auftreten der Kürzel *spd*, *cdu* und *csu* wirkt sich positiv auf die Verweildauer aus, wohingegen das Kürzel *afd* mit einer kürzeren Verweildauer einhergeht. Die Begriffe *linke* und *grüne* liegen im neutralen Bereich.
Auch ein einzelner Artikel kann entsprechend analysiert werden. Abbildung 6 zeigt beispielhaft einen Ausschnitt eines Artikels aus der Ratgeber-Rubrik (*gut-zu-wissen*). Die Farbhinterlegung der einzelnen Tokens gibt Auskunft darüber, ob das Token laut Modell zu einer höheren (blau) bzw. niedrigeren (orange) Verweildauer beigetragen hat oder diesbezüglich neutral (keine

[4] 2017 fand eine Bundestagswahl statt.

Abbildung 5: Negative (links) und positive (rechts) Einflüsse von Parteikürzeln auf die Verweildauern bei Artikeln aus der Rubrik *Politik*.

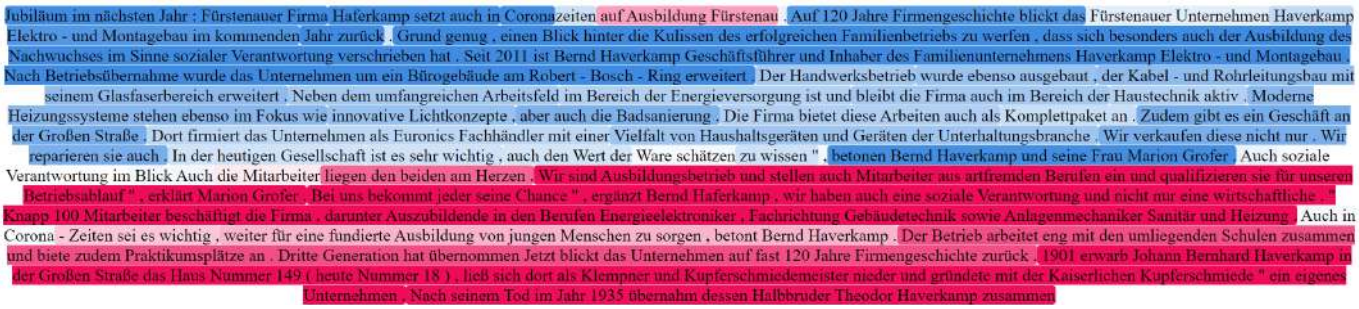

Abbildung 6: Beiträge einzelner Wörter zur Vorhersage: Blau entspricht negativer, rot positver Auswirkung auf die Verweildauer, die Farbintensität spiegelt die Stärke der Auswirkung wider.

Färbung) war. Im Beispiel zeigt sich, dass Begriffe aus den Themenfeldern Familie, Gesundheit und Haushalt positiven Einfluss auf die Verweildauer haben.

Fazit und Ausblick

Mit Methoden aus dem Bereich Machine Learning, spezifischer der automatisierten Sprachverarbeitung bzw. Natural Language Processing (NLP), können wertvolle Informationen, die in Textdaten (z. B. Artikeln) vorliegen, nutzbar gemacht werden. Die modellierte und zu prognostizierende Zielgröße kann spezifisch für den jeweiligen Use Case gewählt werden, solange eine Verknüpfung zwischen KPI und Text möglich ist. Der vorgestellte Use Case macht dies am Beispiel der Performance von Zeitungsartikeln deutlich, der Ansatz bietet sich aber für beliebige Fragestellungen an.

Als erfolgversprechend erweist sich die Kombination aus unstrukturierten Textdaten (Content) und strukturierten Metadaten, wie hier im Use Case die Textrubrik und -länge. Abbildung 4 zeigt deutlich: Die Information des Contents spielt eine entscheidende Rolle. Bei Bedarf und Vorhandensein weiterer Metadaten kann die Modellierung problemlos entsprechend erweitert werden. Im Bereich des Zeitungswesens naheliegend wäre bspw. die zusätzliche Berücksichtigung von Informationen über das Vorhandensein von Bild- oder Videoelementen, Kommentaren, die Positionierung des Artikels, Autor*in oder Erscheinungszeitpunkt naheliegend. Auch eine Analyse getrennt nach Leser*innen-Segmenten oder sogar auf Individualebene ist möglich, wenn entsprechende Daten vorhanden sind.

Die Wahl der zu modellierenden Performance-Metrik richtet sich direkt nach der erwünschten Fragestellung. Denkbar wären z.B. auch die Anzahl der Aufrufe, Kommentare, abgeschlossene Abos oder Daten über Scrolling- und Klickverhalten. Neben der Vorhersage der Performance können aber auch andere für Redaktionen relevante Fragestellungen mittels Natural Language Processing bearbeitet werden. So kann etwa eine automatisierte Unterstützung bei der Identifikation von *Hate Speech* in Kommentaren umgesetzt werden. Kommentarspalten können eine automatisierte Vorsortierung durchlaufen, oder das erwartbare Volumen an problematischen Kommentaren pro Artikel vorhergesagt werden. Weitere Szenarien sind beispielsweise das automatisierte Zusammenfassen von Texten, die Identifikation und Anonymisierung von personenbezogenen Daten, oder schlicht automatisiertes Tagging von Artikeln mit thematischen Überbegriffen oder Schlagwörtern.

Somit können die vorgestellten datengetriebenen Methoden der Unterstützung und Entlastung von Redaktionen und Journalist*innen dienen und interessante Informationen für sie herausarbeiten. Damit wird aus einem generellen *content matters* ein datengetriebenes und datenunterstütztes *specific content matters*, was für Redaktionen bei Planung und Aussteuerung von Inhalten eine wertvolle Zusatzinformation darstellt.

Ihre Ansprechpartner*innen

Susanna Rücker (M.A.)

Im Rahmen ihrer computerlinguistischen Masterarbeit an der Universität Jena hat Susanna mit INWT zusammengearbeitet und sich dort im Bereich Natural Language Processing (NLP) eingebracht. Auf den Ergebnissen der Masterarbeit aufbauend ist in anschließender Zusammenarbeit das vorliegende White Paper entstanden. Ab März 2022 ist Susanna Doktorandin im Bereich NLP an der Humboldt-Universität zu Berlin.

Prof. Dr. Steffen Wagner

Steffen beschäftigt sich schwerpunktmäßig mit den Themen Predictive Analytics, Online Marketing und Customer Relationship Management. Neben seiner langjährigen Erfahrung in der Data-Science-Beratung hat er seit September 2021 eine Professur für *Angewandte Statistik* an der BHT Berlin inne. Steffen ist Mitgründer von INWT.

- **Tel.:** +49 30 1208231-58
- **E-Mail:** steffen.wagner@inwt-statistics.de

INWT Statistics GmbH
Hauptstraße 8
Meisenbach Höfe, Aufgang 3a
10827 Berlin

+49 30 1208231-0

info@inwt-statistics.de

www.inwt-statistics.de