Frankfurt University of Applied Sciences

Master-Thesis

Evaluation der Einsatzmöglichkeiten ausgewählter Machine-Learning-Verfahren im Kontext von IT-Governance-Prozessen, insbesondere im Application-Portfolio-Management

zur Erlangung des akademischen Grades Master of Science (M. Sc.)
Studiengang Wirtschaftsinformatik
Fachbereich 2: Informatik und Ingenieurwissenschaften

Von:	Felix Gündling
	Feldbergstraße 7
	60323
E-Mail:	fguendli@stud.fra-uas.de
Matrikelnummer:	1256976

Gutachter:	Univ.-Prof. Dr. Xyz
Betreuer:	Dr. Xyz
Abgabe am:	28.12.2020

Vorwort

Die vorliegende Master-Thesis wurde in Kooperation mit der Dangelmayer & Seemann GmbH (Abk.: DS) erstellt. Die Problemstellung der Arbeit stammte von einem Kunden von DS. Es handelte sich bei dem Kunden um ein multinationales Großunternehmen. DS betreute das Unternehmen in beratender Funktion. Auf die Nennung von Details zum Unternehmen wird verzichtet. Es werden in dieser Arbeit keine vertraulichen Daten veröffentlicht, die einen Rückschluss auf das Unternehmen zulassen würden. Das Vorwort soll neben der Kurzbeschreibung des Umfelds auch für eine Danksagung genutzt werden: Ich bedanke mich herzlich bei allen Mitarbeitern der Dangelmayer & Seemann GmbH, insbesondere bei Olaf Seemann, Dr. Wolf Pfannenstiel, Dr. Nedialka Bubner, sowie bei Andrew Smart für die Unterstützung während der Bearbeitungszeit.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Wenn Menschen von künstlicher Intelligenz (KI) hören, dann denken nicht wenige an Roboter, die sich intelligent verhalten [19, S. 1]. Mit Robotern wird schnell auch die Automatisierung von Abläufen in Verbindung gebracht. Aus Automatisierung folgen Effizienz und Produktivitätssteigerungen. In Folge dessen ist es nicht verwunderlich, dass mehr als die Hälfte der in einer Studie befragten Entscheider künstliche Intelligenz für das eigene Unternehmen als ein Thema mit großer oder sehr großer Relevanz bewerten, wie eine Statista-Studie (vgl. Abbildung 1.1) zeigt.

Abbildung 1.1: Bewertung der Relevanz von künstlicher Intelligenz. Quelle: Statista [59]

Da in der Statista-Studie Experten der Logistikbranche befragt wurden, haben möglicherweise viele Teilnehmer tatsächlich an die Optimierung der Betriebskosten durch den Einsatz von Robotern in den Verteilzentren oder an selbstfahrende Lieferwägen gedacht.

Der Gedanke liegt allerdings nahe, dass auch an Ansätze aus dem Gebiet des maschinellen Lernens (Engl.: Machine-Learning, Abk.: ML) gedacht wurde, um bestimmte organisatorische oder prozessuale Abläufe zu unterstützen. Gemeint sind hier Abläufe in der Verwaltung, im Vertrieb oder im Marketing, oder auch in der IT-Governance.

1.1 Forschungsfrage- und Methode

Der Fokus dieser Arbeit liegt speziell auf dem Application-Portfolio-Management (APM) als Teilbereich der IT-Governance. Es sollen in dieser Arbeit Maßnahmen zur Erreichung bestimmter Ziele aus dem Bereich des APM formuliert werden. Dabei soll ganz konkret evaluiert werden welche Maßnahmen zur Zielerreichung geeignet sind, und welche nicht. Insbesondere die Chancen, die durch den Einsatz bestimmter Machine-Learning-Verfahren entstehen, sind im Kontext des APM zu bewerten. Die zu beantwortende Frage lautet demnach:

Können ausgewählte Machine-Learning-Verfahren sinnvoll im Kontext des APM eingesetzt werden und einen echten Mehrwert erbringen?

Die Forschungsfrage wird durch eigene Untersuchungen und Tests beantwortet. Da in einem Unternehmenskontext gearbeitet wird, sollen verschiedene Verfahren auf reale Problemstellungen angewendet werden. Es wird explorativ vorgegangen. Ansätze werden in Python implementiert und damit ausprobiert, im Anschluss erfolgt eine Bewertung der Ergebnisse. Für die Bewertung verwendet werden objektive Kriterien, die in Abschnitt 3.5 beschrieben werden. Der Forschungsansatz orientiert sich an der Methode Action Research [5].

1.2 Ziele und Aufbau der Arbeit

Das eigentliche Ziel der Arbeit ist eine Untersuchung von ML-Verfahren, die für die Unterstützung des APM in Frage kommen. Dabei wird zunächst beschrieben, wie die jeweiligen Verfahren einen Mehrwert für das APM erbringen können. Die Verfahren sollen dann evaluiert und miteinander verglichen werden, was letztendlich zu einer Bewertung

von verschiedenen Verfahren führt. Es sollen damit Lösungsansätze für den Umgang mit Applikationsportfolios aufgezeigt werden. Wie später gezeigt wird, spielen bestimmte Text-Dokumente eine wesentliche Rolle im APM. Daher kommen insbesondere Verfahren zur Auswertung von Text-Dokumenten in Frage.

Nach der Einführung folgt ein Grundlagenkapitel, das die notwendigen Begriffe definiert und das Thema der Arbeit detailliert beschreibt. Es wird so gut wie möglich auf aktuelle Fortschritte aus der Forschung eingegangen, wobei zu berücksichtigen ist, dass sich das ML-Gebiet in sehr hoher Geschwindigkeit weiterentwickelt [63, S. 146] und aus diversen Richtungen beeinflusst wird. Es soll außerdem beschrieben werden, mit welchen Fragestellungen sich das APM beschäftigt. Eine Abgrenzung zur IT-Governance wird dabei vorgenommen. Im Anschluss an den Theorieteil wird das Umfeld beschrieben, in dem die Evaluation stattfindet. Es wird außerdem beschrieben, auf welche Art und Weise eine Evaluation von ML-Verfahren im gegebenen Umfeld durchgeführt werden kann. Es werden Kriterien für die Evaluation aufgestellt. Die Ergebnisse der Evaluation werden danach beschrieben. Das Fazit beendet die Arbeit.

1.3 Themeneingrenzung

Es wird in der vorliegenden Arbeit nicht vollumfänglich die historische Entwicklung des Themenbereichs KI diskutiert. Ebenfalls nicht behandelt werden philosophische, moralische oder ethische Fragestellungen, sofern diese nicht unmittelbare Auswirkungen haben für das Thema IT-Governance oder das Application-Portfolio-Management. Es soll aber ausdrücklich betont werden, dass die Nutzung von KI-Systemen das Potenzial hat, die Arbeits- und Lebensrealität von Menschen unmittelbar zu beeinträchtigen. Für eine ausführliche Diskussion dazu wird allerdings auf die einschlägige Literatur verwiesen, z. B. auf [14, S. 11-16]. Die Master-Thesis soll sich hingegen inhaltlich auf aktuelle technische und für den Anwendungsfall relevante betriebswirtschaftliche Aspekte konzentrieren. Aufgrund der begrenzten Bearbeitungszeit kann nur eine gewisse Auswahl an ML-Verfahren berücksichtigt werden.

2 Grundlagen

Ein Verständnis der Möglichkeiten ist der Schlüssel [. . .] zur Gestaltung und Durchführung von Text-Analysen [3, S. vii].

2.1 Künstliche Intelligenz, Machine Learning & Deep Learning

Abbildung 2.1 gibt einen groben Überblick darüber, wie die in dieser Überschrift genannten Begriffe zu verstehen sind und in welcher Beziehung sie zueinander stehen.

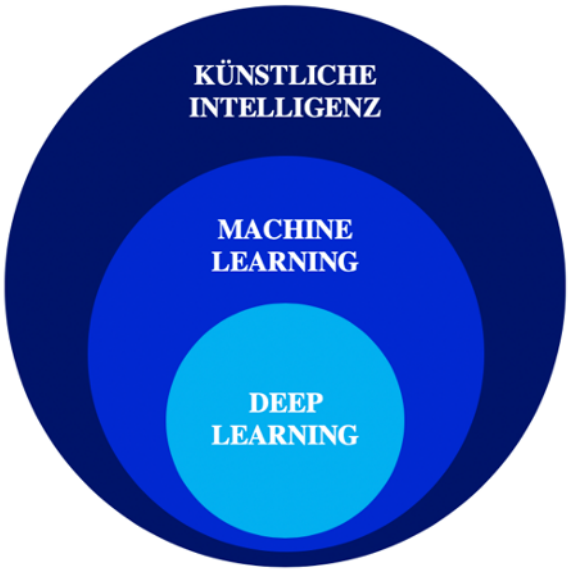

Abbildung 2.1: Zusammenhang von KI, ML und Deep Learning. Eigene Erstellung in Anlehnung an [33]

Das Ziel der künstlichen Intelligenz ist es Systeme aufzubauen, die Aufgaben lösen können, die Intelligenz auf Niveau des Menschen voraussetzen. Machine Learning wird dabei als Teilmenge bzw. Teildisziplin der künstlichen Intelligenz eingeordnet [63, S. 14].

Machine Learning ermöglicht es Computern aus Erfahrungen beziehungsweise aus Aufzeichnungen Regeln zu lernen, ohne explizit programmiert zu werden [52]. Gängig ist im Bereich Machine Learning eine Unterscheidung zwischen drei Arten des Lernens: dem überwachten Lernen, dem unüberwachten Lernen, und dem sogenannten Reinforcement Learning. Beim überwachten Lernen existiert in Trainingsdaten für jedes Beispiel eine Beschriftung. Mit Hilfe der Beschriftungen (oder auch Klassen) kann nun beispielsweise ein Klassifikationsmodell gelernt werden kann. Das so erzeugte Modell wird anschließend genutzt, um eine Vorhersage der Klassen von neuen, unbekannten Datensätzen zu realisieren. Auch die Erstellung von Regressionsatnalysen zählt zur Kategorie des überwachten Lernens. Das unüberwachte Lernen verzichtet auf beschriftete Beispiele. Entsprechende Verfahren können etwa Cluster bilden, die ähnliche Datensätze zusammenfassen (Clustering). Auch die Ausreißererkennung, sowie das Gebiet des Association Rule Learning fallen in die Kategorie des unüberwachten Lernens. Beim Reinforcement Learning werden sogenannte Belohnungen (Rewards) und Strafen (Penalties) verwendet, um beispielsweise Bots das Spielen von Computerspielen beizubringen [19, S. 7-15].

Deep Learning (DL) beschreibt eine Gruppe von Verfahren innerhalb der Domäne des maschinellen Lernens, wobei neuronale Netzwerke eingesetzt werden. Bei künstlichen neuronalen Netzen handelt es sich um eine Gruppe von Machine-Learning-Algorithmen, die inspiriert sind von natürlichen neuronalen Netzen aus der Natur [66, S. 2]. Das Wort deep in Deep Learning bezieht sich auf die Anzahl der Schichten des Netzwerks, durch das Eingaben verarbeitetet werden. Abbildung 2.2 zeigt exemplarisch ein einfaches Artficial Neural Network (ANN). Deep Neural Networks (DNNs) sind stets mit mehr als einem sogenannten Hidden Layer ausgestattet [43].

Es gibt neben ANNs und DNNs noch weitere Architekturansätze zur Realisierung von neuronalen Netzen. Einige davon, wie Recurrent Neural Networks (RNNs), sind für bestimmte Aufgaben besser geeignet als andere. RNNs sind insbesondere für die Verarbeitung von natürlicher Sprache von Bedeutung. Es können verschiedene Formen von RNNs unterschieden werden, wobei für den Anwendungsfall der Sprachverarbeitung insbesondere Encoder-Decoder-RNNs hervorzuheben sind, die auch als seq2seq-Modelle bezeichnet werden und besonders gut mit Sequenzen (z. B. Abfolgen von Wörtern) umgehen können. Gemeint ist hier etwa eine Übersetzung, die eine Sequenz als Eingabe benötigt und eine Übersetzung als Sequenz wieder ausgibt. [66, S. 23–28] Convolutional Neural Networks (CNNs) sind hingegen unter anderem für den Bereich Computer Vision (Bildverarbeitung/

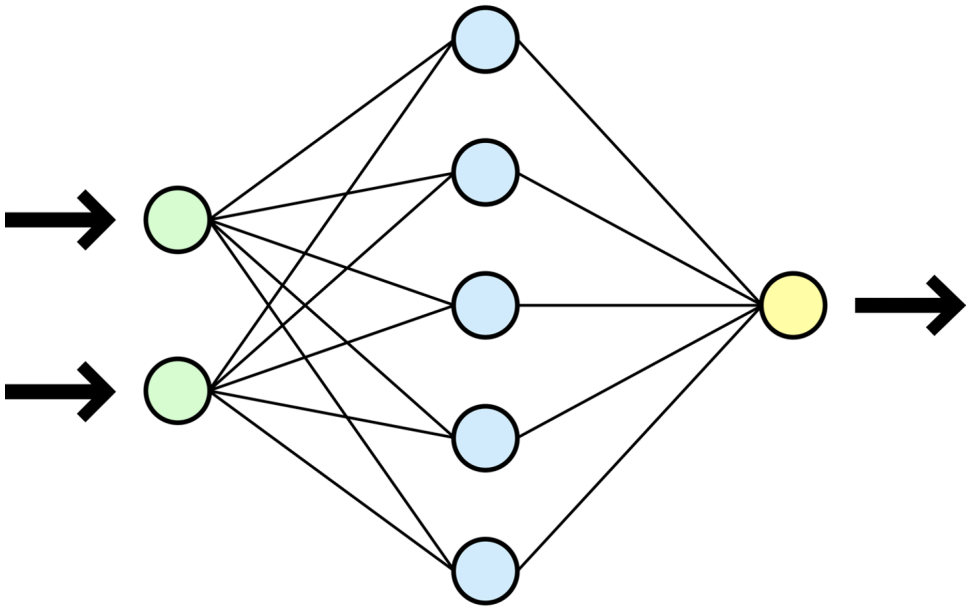

Abbildung 2.2: Ein neuronales Netz mit je einem Input-, Hidden- und Output-Layer, aus [43]

Bilderkennung) von großer Bedeutung. Sie wurden explizit für diese Aufgabe entwickelt [35].

Die verschiedenen neuronalen Netze haben die Gemeinsamkeit, dass sie Neuronen (Knoten) und gewichtete Verbindungen (Kanten) zwischen den Neuronen beinhalten. Neuronen werden aktiviert durch Eingaben und verarbeiten diese mit bestimmten Funktionen weiter, um wiederum eine Ausgabe zu generieren. Das Ermitteln der Gewichtungen dieser Verbindungen wird als Training bezeichnet. Das Training des Netzes besteht aus dem Lösen eines hoch-dimensionalen, nichtlinearen, nicht-konvexen und globalen Optimierungsproblems. Es kann erfolgen unter Zuhilfenahme der Algorithmen Gradient Descent und Backpropagation [66, S. 1–13].

Während bei herkömmlichen Machine-Learning-Ansätzen Features, die während der Verarbeitung benötigt werden, manuell von Menschen gebildet werden müssen, werden diese Features bei Deep-Learning-Ansätzen durch das neuronale Netz erzeugt [2, S. 3-4]. Durch diesen Vorteil sind bessere Ergebnisse möglich [2, S. 11].

2.2 Natural Language Processing

Es geht nachfolgend um Techniken zur Verarbeitung und Auswertung von Texten und natürlicher Sprache (Engl.: Natural Language Processing, Abk.: NLP). NLP kann dabei als Teil des umfangreichen Gebiets Text Mining verstanden werden [46, S. 17]. Das Verständnis von natürlicher Sprache (Engl.: Natural Language Understanding, Abk.: NLU) gilt als Voraussetzung für die Schaffung eines vollständigen KI-Systems, wodurch der Bezug von Sprachverständnis bzw. der Sprachverarbeitung zur KI gegeben ist [67, S. 8].

2.2.1 Vorverarbeitung von Texten

Die Vorverarbeitung (Engl.: Preprocessing) von Texten gilt als Voraussetzung für den Erfolg der späteren Auswertungen [3, S. 46]. Daher werden hier zunächst einige der gängigen Schritte zur Vorverarbeitung von Texten skizziert und grundlegende Methoden erläutert.

Bei der Token-Erstellung (Engl.: Tokenization) wird Text in sogenannte Tokens unterteilt. Im einfachsten Fall bestehen Tokens aus einem Wort – dann handelt es sich um Unigramme (Engl.: unigram). N-Gramme (Engl.: n-grams) stellen eine Alternative zu Unigrammen dar. N-Gramme sind Wort-Sequenzen mit der Länge n [3, S. 48]. Stoppworte dienen zwar einem grammatikalischen Zweck, in Bezug auf den Inhalt von Texten erbringen sie aber nur einen geringen Mehrwert. Aus diesem Grund werden Stoppworte im Rahmen der Vorverarbeitung in der Regel aus Texten entfernt. Dazu kann z. B. ein Wörterbuch verwendet werden, das bekannte Stoppworte wie z. B. [und, er, ein, hat, . . .] enthält [3, S. 50-53]. Die Reduktion von Begriffen auf die Stammform (Engl.: Stemming) verfolgt das Ziel, verschiedene grammatische Varianten des gleichen Wortes zu vereinheitlichen, um die Gesamtanzahl der unterschiedlichen Wörter innerhalb eines Textkörpers zu reduzieren. Die Erzeugung der Stammformen erfolgt z. B. durch den Porter-Stemmer, der Stammformen auf Basis von Regeln bilden kann. Die Lemmatisierung (Engl.: Lemmatization) führt Begriffe auf eine Art „Wörterbuchform“ zurück und kann dadurch besser interpretierbare Ergebnisse liefern als das Stemming [39, S. 30-32] Worttypen wie Adjektive, Verben, Substantive, etc. werden erkannt und markiert durch Part-of-Speech-Tags (Abk.: POS-Tags). Zur Realisierung dieser Markierung können Text-Korpora verwendet werden, bei denen manuell Wörter mit den entsprechenden Wortarten markiert wurden [3, S. 58].

2.2.2 Repräsentation von Texten

Nach den beschriebenen Schritten zur Vorverarbeitung von Textdokumenten kann die Darstellung von Begriffen im Verhältnis zu Dokumenten über eine Term-Document-Matrix (TDM) erfolgen (vgl. Tabelle 2.1). Die transponierte Variante dieser Matrix heißt Document-Term-Matrix (DTM) [3, S. 61ff]. Eine TDM könnte so etwa nach der initialen Vorverarbeitung aussehen.

	Dokument_1	**Dokument_2**	**Dokument_3**	**...**
dog	0	0	2	...
cat	1	1	0	...
bird	3	0	1	...
...	...	...	...	...

Tabelle 2.1: Beispiel für eine TDM. Fiktives Beispiel in Anlehnung an [3, S. 62]

In dieser Art der Darstellung wird die Menge der Wörter eines Dokuments als Bag-of-Words bezeichnet [3, S. 46]. Die Ausprägungen in den Zellen der Tabelle 2.1 beschreiben in diesem Beispiel die Häufigkeit des Vorkommens der Begriffe in den jeweiligen Dokumenten. Um zur Term Frequency (Abk.: tf) zu gelangen, ist die Anzahl des Begriffs durch die Gesamtanzahl der Begriffe im jeweiligen Dokument zu dividieren. Für den Begriff cat ergibt sich in der ersten Spalte eine tf von $\frac{1}{4}$, unter Vernachlässigung der Spalten und Zeilen ohne kardinale Ausprägungen.

Eine weitere, für die spätere Auswertung relevante Kennzahl heißt Document Frequency (Abk. df). Dabei wird zunächst die Häufigkeit der Begriffe reduziert auf eine binäre Angabe 1 oder 0, die Häufigkeit wird also vernachlässigt und es verbleibt lediglich eine Information darüber, ob ein Wort in den Dokumenten vorkommt oder nicht. Die nun erzeugten Werte werden zeilenweise addiert, um die Kennzahl df zu erhalten. Das Beispiel „cat“ aus Tabelle 2.1 würde zu einer df von 1+1+0=2 führen. Df gibt damit an, in wie vielen Dokumenten ein Begriff vorkommt. Um seltenen, und damit aussagekräftigen Begriffen ein höheres Gewicht zu geben als Begriffen die zwar häufig vorkommen, dafür aber auch weniger Relevanz und Wirkung haben, existiert mit der inversen Dokumenten-Häufigkeit (Engl.: inverse document frequency, Abk.: idf) eine weitere Metrik. Sie wird berechnet als:

$$idf_i = log_2(\frac{n}{df_i}) + 1$$

wobei n die Anzahl der Dokumente ist, und df wie oben beschrieben berechnet wird. Die hier beschriebenen Metriken können kombiniert werden zur tf-idf-Metrik. Dabei wird tf mit idf lediglich multipliziert. Für den Beispielbegriff cat ergibt sich unter Vernachlässigung der Spalten und Zeilen ohne kardinale Ausprägungen ein Wert von $\frac{1}{4} * (log_2(\frac{3}{2}) + 1) = 0,39$ für das erste Dokument. Der TF-IDF-Wert von Begriffen kann nun als Ausprägung in den Zellen einer TDM an Stelle der schlichten Anzahl der Begriffe verwendet werden [6, pp. 61-73].

Die TF-IDF-Metrik ist zusammengefasst eine Grundlage für die Erkennung von wichtigen Begriffen, unter Betrachtung von mehreren Dokumenten. Dabei sollten die Dokumente möglichst einen Bezug zueinander haben. Ein Beispiel wäre eine Menge von Dokumenten über Software-Entwicklungsprojekte. In jenen Dokumenten würde der Begriff „Software“ häufig vorkommen. Die Bedeutung des Begriffs ist allerdings eher gering, da er wahrscheinlich in allen Dokumenten vorhanden sein wird. Beim TF-IDF-Maß erhalten seltene Begriffe einen hohen Wert, wodurch solche Begriffe als interessant identifiziert werden können. [39, S. 107-110]. Auf Basis der einfachen TDM oder auch auf Basis der erweiterten Variante unter Verwendung von TF-IDF-Werten können im Anschluss Auswertungen durchgeführt werden.

2.2.3 Regelbasierte Ansätze

Um ein ML-Modell trainieren zu können, sind Trainingsdaten notwendig (vgl. Kapitel 2.1). Wenn diese nicht in ausreichender Menge vorhanden sind, dann kann ein regelbasierter Ansatz eine Alternative sein. Regelbasierte Systeme zählen zu den ältesten Ansätzen im Bereich NLP und haben sich in der Praxis bewährt. Aus technischer Sicht können für die Realisierung eines regelbasierten Ansatzes etwa String-Vergleiche (z. B. mit regulären Ausdrücken) verwendet werden. Auch mit kontextfreien Grammatiken können Regeln ausgedrückt werden [40]. Ein simpler endlicher Automat kann verwendet werden, um einfache Regeln zu beschreiben [39, S. 219]. Zu den regelbasierten Ansätzen werden hier explizit auch Wenn-Dann-Regeln gezählt [14, S. 23ff.]. Beispiele für Entitäten, die gut über einen regelbasierten Ansatz erkannt werden können, sind etwa IP-Adressen, URLs, oder Telefonnummern [57].

Über Ontologien oder Taxonomien können ebenfalls Entitäten erkannt werden. Solche Wissensbasen galten früher als grundlegender Baustein von KI-Systemen. Ein solcher Bestand

kann zu Beginn eines Projekts erstellt werden und dann im Laufe der Zeit weiterentwickelt werden. Ontologien führen in diesem Kontext zu wiederverwendbaren, erweiterbaren Wissensbeständen. Sie können sowohl von Menschen als auch von Computern verstanden werden und bilden Mengen von explizit definierten Terminologien ab. Die grundlegenden Begriffe, also das Vokabular einer bestimmten Domäne und die Beziehungen der Begriffe untereinander werden in einer Ontologie modelliert [42, S. 37-55]. Das Vokabular könnte in den erwähnten Wenn-Dann-Regeln verwendet werden. Die Begriffe Ontologie und Topic werden in der Literatur und daher auch hier als Synonyme verwendet.

Taxonomien können ebenfalls für solche Klassifizierungszwecke verwendet werden. Eine Taxonomie hat einen Namen und besteht aus Beispielen. Es kann flache bzw. einfache Taxonomien geben, aber auch hierarchische, oder gar vernetzte Taxonomien [27]

Da im Titel dieser Arbeit die Evaluation von ausgewählten ML-Verfahren festgelegt ist, werden regelbasierte Ansätze im praktischen Teil der Arbeit weniger ausführlich betrachtet als “echte“ ML-Ansätze. Sie gelten aber in realen Szenarien als reale Option für bestimmte Anwendungsfälle und wurden daher hier beschrieben.

2.2.4 Vektorbasierte Ansätze

Die Begriffe Wort-Vektor und Wort-Einbettung (Engl.: Word Embedding) werden hier als Synonyme verwendet, wie in [66, S. 38]. Als moderner Algorithmus zur Erzeugung von Wort- Einbettungen gilt Word2Vec. Entwickelt wurde das Verfahren von Google-Forschern [41], allerdings sind auch andere Unternehmen an der Weiterentwicklung dieser Ansätze beteiligt. So gibt es etwa mit fastText von Facebook oder mit GloVe der Stanford University ebenfalls Möglichkeiten zur Nutzung von vorbereiteten Word-Vektoren sowie für das Training eigener Wort-Vektoren [15] [45]. Word2Vec versucht, die Probleme des Bag-of-Word-Ansatzes beziehungsweise der One-Hot-Encodings zu vermeiden. Durch diese Repräsentation – bei der das Vorhandensein eines Wortes in einem Textdokument etwa mit einer 1 angezeigt wird und das Fehlen mit einer 0 – entstehen hoch-dimensionale Tabellen mit vielen Zeilen. Dabei kommt häufig der Wert 0 als Ausprägung vor. Damit einher geht ein gewisser Informationsverlust, da Kontextinformationen (z. B. die Position der Wörter innerhalb der Dokumente) verloren gehen und weil viele ML-Verfahren mit Daten, die in dieser Art und Weise repräsentiert sind, schlecht umgehen können. Zu viele Spalten können zu Overfitting und damit zu einem schlechten Modell führen [54, S. 145]. Als

Architektur für das Erzeugen von Wort-Vektoren kommt bei Word2Vec ein voll-verknüpftes neuronales Feed-Forward-Netz zum Einsatz. Eine Eigenschaft des Verfahrens ist, dass bei jedem Wort der Kontext des Worts in gewissem Maße erhalten bleibt. Die Anwendung von Word2Vec führt zu den sogenannten Einbettungen, was letztendlich einer deutlich kompakteren Darstellung der Informationen entspricht, als das etwa bei der Darstellung von Text in Form einer Term-Document-Matrix der Fall ist. Eine Einbettung entspricht einer Darstellung von Begriffen in Vektorform. Die mit Word2Vec erzeugten Wort-Vektoren können als Eingabe für weitere Machine-Learning-Verfahren verwendet werden, wobei diese Repräsentation deutlich bessere Resultate verspricht, als in der unverarbeiteten Form. Um die Vektoren zu erzeugen wird ein neuronales Netz trainiert, dass das Ziel hat, die Wahrscheinlichkeit von bestimmten Kontextwörtern einzelner Wörter vorherzusagen (im sogenannten Skip-Gram-Ansatz). Die internen Gewichtungen des neuronalen Netzes werden letztendlich als Wort-Vektoren verwendet. [54, S. 148-160].

Neben Wort-Einbettungen kommen an dieser Stelle noch Dokumenten-Einbettungen in Betracht. Der Algorithmus dazu heißt Paragraph Vector. Im Vergleich zu Word2Vec werden für das Erstellen der Einbettungen zusätzlich zu den Wörtern auch jeweils Paragraphen in das Training der Vektoren mit einbezogen. Die Paragraphen liefern also zusätzlichen Kontext. Vergleichbar ist der Paragraph mit einer Angabe darüber, wo im Dokument bzw. auch in welchem Dokument sich ein Begriff in den Trainingsdaten befunden hat [34]. Eine Implementierung von Dokument-Einbetttungen findet sich unter dem Namen doc2vec [49].

In Abbildung 2.3 dargestellt sind einzelne Wörter, die als Vektoren abgebildet werden. Die Abkürzung W steht aus mathematischer Sicht für eine Matrix, die einzelnen Wort-Vektoren sind als Spalten in dieser Matrix enthalten. Nach Bildung der initialen Vektoren werden diese konkateniert („aufsummiert“), um den sogenannten Kontext abzubilden. In diesem Beispiel wird ein neuronales Netz verwendet, dass die Wahrscheinlichkeit des Wortes „on“ im Kontext der Begriffe „the cat sat“ ermittelt (Hier dargestellt ist das Continuous Bag of Word-Modell (CBOW), das genaue Gegenteil von Skip-Gram, das oben bereits erwähnt wurde). Bei der Verwendung des Paragraph Vector Ansatzes wird jeder Paragraph als Spalte in einer Matrix D abgebildet. Diese zusätzliche Information ergänzt den beschriebenen Kontext. Was genau als Paragraph definiert wird, kann frei gewählt werden. Es kann sich um längere Abschnitte, ganze Dokumente, oder auch eine Auswahl von Sätzen handeln [34].

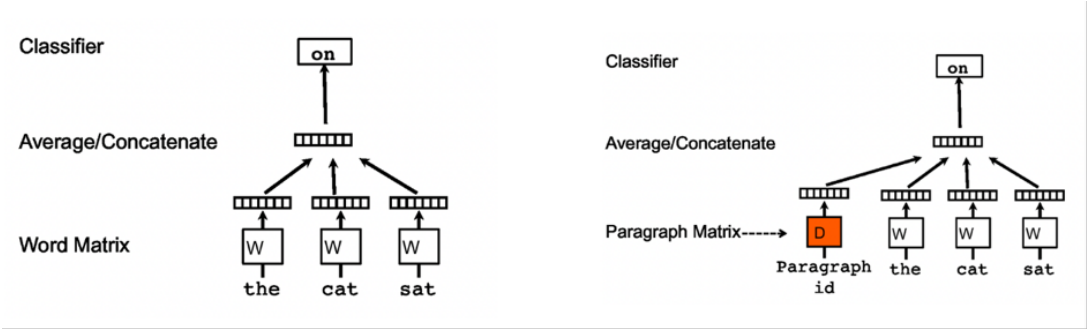

Abbildung 2.3: Funktionsweise von Word2Vec (links) und von Paragraph Vector (rechts), aus [34]

Einbettungen müssen nicht für jede Anwendung neu trainiert werden. Stattdessen können auch vor-trainierte (pre-trained) Einbettungen wiederverwendet werden, wie am Anfang dieses Abschnitts bereits angedeutet wurde. Unten dargestellt (Abbildung 2.4) ist ein Ausschnitt aus dem bereits trainierten GloVe-Modell, der die Vektor-Repräsentation verdeutlichen soll. Es wurde trainiert auf Basis des Korpus Wikipedia 2014 [45].

Abbildung 2.4: Vektoren der Begriffe „the“ & „and“. Eigene Bildschirmaufnahme.

Die größeren NLP-Modelle, etwa von spaCy, beinhalten standardmäßig nutzbare Wortvektoren. Eine eigene Untersuchung ergibt, dass diese Modelle auch gewisse fachspezifische Wörter berücksichtigen, etwa die Begriffe Shipments oder containerization. Allerdings existieren für echte domänenspezifische Begriffe wie Projekt- oder Systemnamen keine Wortvektoren. Solche Begriffe nennt man out-of-vocabulary [58]. Das erschwert die Verwendung von Modellen, die „gebrauchsfertig“ zur Verfügung stehen.

2.2.5 Transformer-Ansätze

Transformer werden erstmals eingeführt Ende 2017. Laut den Autoren der Veröffentlichung ist der Ansatz in der Lage bessere Ergebnisse in bestimmten Bereichen zu erzielen als bisherige Ansätze auf Basis von RNNs oder CNNs; zudem ist das Training parallelisierbar und damit schneller als bei früheren Ansätzen [64]. Transformer-Ansätze können für viele NLP-Teilbereiche verwendet werden, etwa für Named-Entity-Recognition, Übersetzungen, das Generieren von Zusammenfassungen, und weitere Anwendungsfälle. [63, S. 25-26]. Die Nutzung von vortrainierten Modellen wird als Transfer-Learning bezeichnet. Transformer können dies ermöglichen. Gewisse Gewichtungen innerhalb von neuronalen Netzen können dabei übernommen werden und müssen nicht neu trainiert werden (z. B. grundlegende Sprach-Eigenschaften). Vortrainierte Modelle sind eine Option, wenn nur wenige Daten für das Training zur Verfügung stehen. Die vor-trainierten Modell erfahren eine Fein-Abstimmung (Engl.: Fine-Tuning), wodurch die Modelle an den jeweiligen Anwendungsfall angepasst werden [2, S. 45-56]. Ein konkreter Transformer-Ansatz ist BERT (Bidirectional Encoder Representations from Transformers). Es gibt eine Pre-Training-Phase und eine Fine-Tuning-Phase, durch die das finale Modell erzeugt wird. Es wurden Ergebnisse erreicht, die als state-of-the-art bezeichnet werden können [12]. GPT (Generative Pre-trained Transformer) und seine Nachfolger GPT-2 und GPT-3 sind weitere Beispiele für Transformer-Modelle, die insbesondere für die Text-Erzeugung eingesetzt werden können [8]

2.3 Das Application-Portfolio-Management als Teil der IT-Governance

IT-Governance soll sicherstellen, dass Informationstechnologie (IT) die Strategien und Ziele von Unternehmen unterstützt. Die IT soll so angepasst und transformiert werden, dass unterschiedliche Anforderungen erfüllt werden. Dazu zählen insbesondere regulatorische und gesetzliche Anforderungen [51, S. 5]. Gemeint sind damit etwa Gesetze und Rechtsverordnungen, wie die Datenschutzgrundverordnung (DSGVO), der Sarbanes-Oxley-Act (SOA), oder auch die bankaufsichtlichen Anforderungen an die IT (BAIT) im Bankenumfeld [18, S. 274-275].

Außerdem kann davon ausgegangen werden, dass Manager stets auch die eigenen Interessen im Sinn haben, wenn die Notwendigkeit von IT-Governance begründet werden soll. Risiken, die mit dem Betrieb von IT-Systemen einhergehen, sollten dem Management bekannt sein, damit entsprechende Maßnahmen getroffen werden können. Außerdem ist von Interesse, welchen Mehrwert einzelne Systeme tatsächlich bringen und welche Geschäftsbereiche durch die Systeme unterstützt werden. Der große Vorteil von IT-Governance ist erhöhte Transparenz. Die Transparenz macht Kosten sichtbar und ermöglicht sinnvolle Entscheidungen im Zusammenhang mit der Unternehmens-IT [26, S. 5-7]. Mit den angesprochenen Risiken können (u. a.) wirtschaftliche Verluste im Falle von Systemausfällen verbunden sein, aber auch der langfristige Verlust der eigenen Wettbewerbsfähigkeit, wenn zu lange mit der Erneuerung von Systemen gewartet wird [26, S. 20]. IT-Governance ist ein Bestandteil eines übergeordneten Corporate-Governance-Ansatzes, bei dem es um die Steuerung und Überwachung eines gesamten Unternehmens geht [18, S. 2].

2.3.1 COBIT als mögliche Grundlage für das APM?

Das Rahmenwerk (Engl.: Framework) COBIT kann als methodische Grundlage für IT-Governance dienen. Die Abkürzung steht für Control Objectives for Information and Related Technology. Die aktuelle Version (während der Bearbeitungszeit dieser Arbeit) ist COBIT 2019 [18, S. 9-12].

COBIT definiert in seinem Kern-Modell (vgl. Abbildung 2.5) sogenannte Governance- und Management-Ziele [18, S. 67]. Ziel APO05 heißt Managed Portfolio. Anhand dieses Beispiels soll verdeutlicht werden, dass der Umgang mit bestimmten Portfolios ein wesentlicher Bestandteil von Governance-Tätigkeiten sein kann. Die Bezeichnung Application Portfolio wird hier allerdings nicht verwendet, es geht stattdessen um Projekte und Programme. Dabei müssen einzelne Projekte oder Programme priorisiert werden, damit strategische Ziele erreicht werden können [30, S. 87-92].

Statt einem eigenen Management-Ziel zum Umgang mit Applikationen exisitert in COBIT die Empfehlung zu Managed Assets (“Betriebsmitteln“, vgl. Ziel BAI09) [30, S. 209-214], wozu auch der Umgang mit Software-Lizenzen zählt. Unter dem Ziel Managed Configuration (BAI10) wird empfohlen, Daten zu “wichtigen Ressourcen“ zu erheben und zu verwalten [18, S. 73]. Unter Ziel APO03 wird das Managen der Unternehmensarchitektur empfohlen, was die Einrichtung einer Referenzarchitektur beinhaltet, in der u. a. auch

Anwendungen und Technologien berücksichtigt werden können [18, S. 70]. In welchen Variationen diese Empfehlungen in Unternehmen ganz konkret umgesetzt werden, ist diesen stets selbst überlassen [18, S. 65].

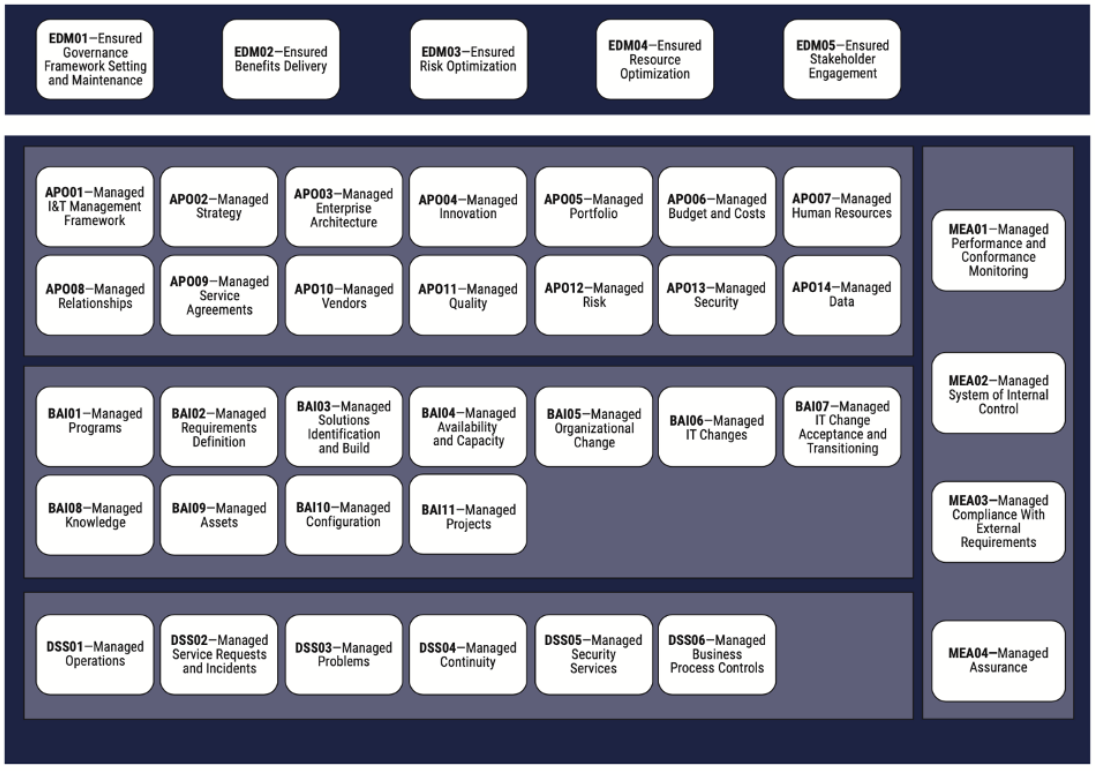

Abbildung 2.5: COBIT-Kernmodell, aus [29, S. 21]

Neben COBIT gibt es weitere Frameworks, bzw. Modelle zum Umgang mit einzelnen Governance-Themen. In [18, S. 127-133] werden beispielsweise das CMMI - Capability Maturity Model (Integration) sowie die IT Infrastructure Library (ITIL) erwähnt.

Außerdem gibt es spezielle (Enterprise-)Architektur-Frameworks. Auf einige davon wird in einem späteren kurz eingegangen.

Literatur

[1] M. Allahyari, S. Pouriyeh, M. Assefi & al. *Text Summarization Techniques: A Brief Survey*. 2017.

[2] M. Alom, T. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. Nasrin, M. Hasan, B. Van Essen, A. Awwal & V. Asari. *A State-of-the-Art Survey on Deep Learning Theory and Architectures*. Electronics, 2019.

[3] M. Anandarajan, C. Hill & T. Nolan. *Practical Text Analytics: Maximizing the Value of Text Data*. Springer, 2019.

[4] *Application Portfolio Management*. The Art of Service - Application Portfolio Management Publishing, 2019.

[5] D. Avison, F. Lau, M. Myers & P. A. Nielsen. *Action Research*. Communications of the ACM, 1999.

[6] I. Beltagy, M. E. Peters & A. Cohan. *Longformer: The Long-Document Transformer*. 2020.

[7] BITKOM. *Enterprise Architecture Management – neue Disziplin für die ganzheitliche Unternehmensentwicklung*. 2011.

[8] T. B. Brown, B. Mann, N. Ryder & al. *Language Models are Few-Shot Learners*. 2020.

[9] A. M. Dai, C. Olah & Q. V. Le. *Document Embedding with Paragraph Vectors*. 2015.

[10] H. T. Dang. *Overview of DUC 2005*. Proceedings of the Document Understanding Conf. Wksp. 2005 (DUC 2005), 2005.

[11] scikit-learn developers. *sklearn.metrics.f1-score*. Abgerufen am 20. November 2020. URL: `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html`.

[12] J. Devlin, M.-W. Chang, K. Lee & K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019.

[13] N. Dilawar, H. Majeed, M. O. Beg & al. *Understanding Citizen Issues through Reviews: A Step towards Data Informed Planning in Smart Cities*. Applied Sciences 8(9):1589, 2018.

[14] W. Ertel. *Introduction to Artificial Intelligence*. Springer, 2017.

[15] *fastText. Library for efficient text classification and representation learning*. Facebook Inc. Abgerufen am 15. August 2020. URL: `https://fasttext.cc/`.

[16] T. Frey. *Governance Arrangements for IT Project Portfolio Management*. Springer Gabler, 2014.

[17] A. Gadatsch & E. Mayer. *Masterkurs IT-Controlling*. Springer Vieweg, 2014.

[18] M. Gaulke. *Praxiswissen COBIT. Grundlagen und praktische Anwendung in der Unternehmens-IT*. dpunkt.verlag, 2020.

[19] A. Géron. *Hands-On Machine Learning with Scikit-Learn, Keras & Tensorflow*. O'Reilly, 2019.

[20] C. Gliedman. *Defining IT Portfolio Management*. Forrester Research, 2004.

[21] O. Grisel, L. Buitinck & C.-K. Yau. *Topic extraction with Non-negative Matrix Factorization and Latent Dirichlet Allocation*. Abgerufen am 20. November 2020. URL: https://scikit-learn.org/0.18/auto_examples/applications/topics_extraction_with_nmf_lda.html.

[22] huggingface. *Longformer*. Abgerufen am 28. Oktober 2020. URL: https://huggingface.co/transformers/model_doc/longformer.html.

[23] huggingface. *Model: bert-base-uncased*. 03. Dezember 2020. URL: https://huggingface.co/bert-base-uncased.

[24] huggingface. *Model: t5-small*. Abgerufen am 30. Oktober 2020. URL: https://huggingface.co/t5-small.

[25] huggingface. *Transformers*. Abgerufen am 13. Oktober 2020. URL: https://huggingface.co/transformers/.

[26] IMPACT. *IT Governance. Developing a successful governance strategy. A Best Practice Guide for decision makers in IT*. The National Computing Centre, 2005.

[27] B. Inmon. *Turning Text into Gold: Taxonomies and Textual Analytics*. Technics Publications, 2016.

[28] *Integration Architecture*. LeanIX GmbH. Abgerufen am 13. November 2020. URL: https://docs.leanix.net/docs/integration-architecture.

[29] ISACA. *COBIT 2019 Framework Introduction and Methodology*. 2018.

[30] ISACA. *COBIT 2019 Framework. Governance and Management Objectives*. 2018.

[31] J. Jung. *Purpose of Enterprise Architecture Management: Investigating Tangible Benefits in the German Logistics Industry*. 2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW), 2019.

[32] J. Jung & R. Schlör. *Elaborating Potential for Improving Application Landscapes in Logistics*. 2018 IEEE 22nd International Enterprise Distributed Object Computing Workshop, 2018.

[33] J. D. Kelleher. *Deep Learning*. The MIT Press, 2019.

[34] Q. Le & T. Mikolov. *Distributed Representations of Sentences and Documents*. 2014.

[35] Y. LeCun, L. Bottou, Y. Bengio & P. Haner. *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, 1998.

[36] C.-Y. Lin. *ROUGE: A Package for Automatic Evaluation of Summaries*. Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), 2004.

[37] H. P. Luhn. *The Automatic Creation of Literature Abstracts*. IBM Journal of Research und Development, 1958.

[38] L. J. P. van der Maaten & G. E. Hinton. *Visualizing High-Dimensional Data Using t-SNE*. Journal of Machine Learning Research 9(Nov):2579-2605, 2008.

[39] C. D. Manning, P. Raghavan & H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[40] M. Mayo. *The Main Approaches to Natural Language Processing Tasks*. Abgerufen am 20. Juli 2020. URL: https://www.kdnuggets.com/2018/10/main-approaches-natural-language-processing-tasks.html..

[41] T. Mikolov, K. Chen, G. Corrado & J. Dean. *Efficient Estimation of Word Representations in Vector Space*. 2013.

[42] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator & W. R. Swartout. *Enabling Technology for Knowledge Sharing*. AI Magazine Volume 12 Number 3, 1991.

[43] M. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

[44] D. O'Connel. *Integrated Portfolio Management: Better Visibility, Easier Decisions, Lower Costs*. Aite Group/ Software AG, 2020.

[45] J. Pennington, R. Socher & C. D. Manning. *GloVe: Global Vectors for Word Representation*. URL: https://nlp.stanford.edu/projects/glove/.

[46] J. M. C. Pérez. *Augmenting Humans*. TU München (Dissertation), 2017.

[47] C. Raffel, N. Shazeer, A. Roberts & al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2020.

[48] P. C. Rao, A. Reedy & B. Bellman. *Certified Enterprise Architect*. McGraw-Hill Education, 2019.

[49] R. Řehůřek. *Doc2vec paragraph embeddings*. Abgerufen am 05. August 2020. URL: https://radimrehurek.com/gensim/models/doc2vec.html.

[50] R. Řehůřek. *Word2Vec Model*. Abgerufen am 08. Oktober 2020. URL: https://radimrehurek.com/gensim/auto_examples/tutorials/run_word2vec.html#sphx-glr-auto-examples-tutorials-run-word2vec-py.

[51] M. Reiss. *Dokumentationsmanagement – Basis für IT-Governance*. Springer Vieweg, 2018.

[52] A. L. Samuel. *Some studies in machine learning using the game of checkers*. IBM Journal of research und development, 3(3), 210-229., 1959.

[53] *ServiceNow Application Portfolio Management*. Datasheet. ServiceNow, Inc., 2020.

[54] R. Silipo & V. Tursi. *From Words to Wisdom*. KNIME Press, 2018.

[55] D. Simon, K. Fischbach & D. Schoder. *Application Portfolio Management - An Integrated Framework and a Software Tool Evaluation Approach*. Communications of the Association for Information Systems: Vol. 26 , Article 3, 2010.

[56] spacy.io. *Doc (Class)*. Abgerufen am 01. September 2020. URL: https://spacy.io/api/doc.

[57] spacy.io. *Rule-based matching*. Abgerufen am 02. August 2020. URL: https://spacy.io/usage/rule-based-matching.

[58] spacy.io. *Word Vectors and Semantic Similarity*. Abgerufen am 02. August 2020. URL: https://spacy.io/usage/vectors-similarity.

[59] Statista. *Statista-Expertenbefragung BVL & Statista Logistikmonitor 2018*. Abgerufen am 20. Juli 2020. URL: https://de.statista.com/prognosen/943357/expertenbefragung-zur-kuenstlichen-intelligenz-in-der-logistikbranche.

[60] M. Tang, P. Gandhi & M. A. Kabir. *Progress Notes Classification and Keyword Extraction using Attention based Deep Learning Models with BERT*. 2019.

[61] *The Definitive Guide to Application Portfolio Management*. LeanIX GmbH.

[62] *The TOGAF Standard 9.2*. The Open Group. URL: https://pubs.opengroup.org/architecture/togaf92-doc/arch/index.html.

[63] S. Vajjala, B. Majumder, A. Gupta & H. Surana. *Practical Natural Language Processing*. O'Reilly, 2019.

[64] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser & I. Polosukhin. *Attention Is All You Need*. 2017.

[65] A. Vogelsang & J. Winkler. *Automatic Classification of Requirements Based on Convolutional Neural Networks*. 2016 IEEE 24th International Requirements Engineering Conference Workshops (REW), 2016.

[66] L. White, R. Togneri, W. Liu & M. Bennamoun. *Neural Representations of Natural Language*. Springer, 2018.

[67] R. Yampolskiy. *Turing Test as a Defining Feature of AI-Completeness*. In: Yang X.-S. (Herausgeber). Artificial Intelligence, Evolutionary Computing und Metaheuristics. Studies in Computational Intelligence, vol 427. Springer, 2013.

Eidesstattliche Versicherung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen sind, sind als solche kenntlich gemacht. Die Arbeit hat in gleicher Form noch keiner anderen Prüfbehörde vorgelegen.

Frankfurt am Main, im Dezember 2020

Anhang

Der Python-Code, der für die in dieser Arbeit beschriebenen Evaluationen verwendet wurde, ist in einem Zip-Archiv zu finden, das mit der Ausarbeitung abgegeben wird.