Word Embeddings für literarische Texte

Masterthesis

von

LEONARD KONLE

Lehrstuhl für Computerphilologie
Institut für deutsche Philologie

JULIUS-MAXIMILIANS-UNIVERSITÄT WÜRZBURG

Studiengang:	Digital Humanities, 7. Fachsemester
Matrikelnr.:	1904133
Erstgutachter:	Prof. Fotis Jannidis
Zweitgutachter:	Dr. Stephan Moser
Ort und Datum:	Würzburg, 26.03.2019

Word Embeddings für literarische Texte

Masterthesis

von

LEONARD KONLE

Lehrstuhl für Computerphilologie
Institut für deutsche Philologie

JULIUS-MAXIMILIANS-UNIVERSITÄT WÜRZBURG

Studiengang:	Digital Humanities, 7. Fachsemester
Matrikelnr.:	1904133
Erstgutachter:	Prof. Fotis Jannidis
Zweitgutachter:	Dr. Stephan Moser
Ort und Datum:	Würzburg, 26.03.2019

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

KAPITEL 1

EINFÜHRUNG

1.1 Einleitung

Die erruptiven Entwicklungen der letzten Jahre im Bereich der maschinellen Sprachverarbeitung wurden durch den Aufstieg des Deep Learning verusacht. Deep Learning bietet eine Vielzahl an Vorteilen gegenüber klassischen Verfahren maschinellen Lernens. Es ist jedoch das Konzept des Word Embeddings, welches das Fundament für jede methodische Innovation bildet (Young et al. 2018). Die distributionelle Repräsentation eines Wortes im Vektorraum ermöglicht es, semantische und syntaktische Eigenschaften maschinenlesbar und kontextsensitiv zu speichern. Diese Eigenschaften betonen den Doppelcharakter von Word Embeddings. Zum einen können sie als Werkzeug Sprache in ein Format bringen, welches weiterverarbeitet zur Lösung von Aufgaben wie Sentiment Analysis[1], Question Answering[2] oder maschineller Textproduktion beiträgt. Zum anderen sind sie aber auch Ressource und selbst Objekt des wissenschaftlichen Interesses. Welche Chancen und Herausforderungen ergeben sich

[1] Untersuchung von Text auf positive oder negative Emotion (Pang, Lee et al. 2008)
[2] Beantwortung von Fragen durch ein System in menschlicher Sprache

hieraus für die Digital Humanities?
Word Embeddings bieten die Möglichkeit eine Vielzahl an Methoden zu ergänzen. Jede quantitative Herangehensweise an Text kann mit ihnen um die semantische Dimension erweitert werden. Denn wo ein ein Wort bisher lediglich als kontextfreie Zahl repräsentiert wurde, ist es nun möglich, diese Zahl durch einen Vektor zu ersetzen, der tatsächlich Information bereithält, schon bevor Berechnungen stattgefunden haben. Zu behaupten Algorithmen könnten Texte durch den Einsatz von Embeddings „verstehen“ ist sicher eine euphemistische Sichtweise, dennoch bedeutet er einen Schritt weg von bloßem „Wörter zählen“ hin zu einem adäquaten Umgang mit Text.
Word Embeddings bieten die Chance literaturwissenschaftliche Fragen zu bearbeiten, für welche bisher aufgrund von mangelndem Zugang zu digitalisierten Texten keine ausreichende Datengrundlage vorhanden ist. Denn die Information über Sprache ist im Gegensatz zu Methoden wie LDA, die Themen aus Texten abstrahieren, um Aussagen über die zugrundeliegenden Strukturen treffen zu können und dabei das Wort durch eine Abstraktion ersetzen, nahezu universell einsetzbar. Denn ihre Bezugsgröße ist nicht das Wort im untersuchten Korpus, sondern die aus einer Vielzahl von externen Kontexten gewonnene Information über ein Wort selbst (Schmidt 2015). Somit können auch kleine Textsammlungen hinreichend untersucht werden.
Gleichzeitig muss betont werden, dass Word Embeddings immer durch die Daten aus denen sie Information gewinnen begrenzt werden. Da zur Erzeugung von Embeddings zum einen leistungsfähige Infrastruktur zur Ausführung komplexer Berechnungen und zum anderen eine immense Menge an Textmaterial benötigt werden, sind die Modelle hauptsächlich für Gegenwartssprache geeignet. Die aktuell verfügbaren Modelle gründen vor allem auf Text aus Webseiten und Zeitungsarchiven. Dies ist für

diejenigen, welche die Modelle erstellen ausreichend, da im Bereich NLP dieses Material zur Evaluation neuer Methoden verwendet wird. Aus Sicht der Digital Humanities ist dieser Umstand ungünstig, da weder ausreichend erforscht ist wie sich Embeddings an historische Sprache anpassen noch ob die Sprache der Literatur hinreichend erfasst werden kann. Daraus leitet sich die Herausforderung für die Digital Humanities ab, eigene Embeddings zu erstellen und deren Verhalten gegenüber ihrem Forschungsgegenstand zu evaluieren. In diesem Zusammenhang ergeben sich durch den Vergleich von Modellen auch Möglichkeiten Domänenspezikika von Sprache in Form von Embeddings gegenüberzustellen. Wie bei jeder Art von Modellierung bietet sich hier die Chance, etwas über den für die Modellierung ausschlaggebenden Datensatz, das Modell selbst und deren gemeinsames Verhältnis zu neuen Daten zu lernen.
Die hier vorliegende Arbeit versucht eine erste Annäherung an das Spannungsfeld Digital Humanities und Embeddings über das hinreichend erforschte Problem der Gattungsklassifikation. Die Methode zur Identifizierung distinktiver Gattungswörter, bekannt als Burrow's Zeta (Burrows 2007, Christof Schöch 2018), wird mit den semantischen Informationen aus Word Embeddings angereichert, um exemplarisch zu zeigen, wie und ob eine Methode durch den Einsatz distributioneller Semantik verbessert werden kann. Ein zweiter Ansatz verzichtet auf die vorhandene Methodik und bietet einen neuen Lösungsweg rein basierend auf Deep Learning. Es wird ein Überblick über gängige Methoden zur Erstellung von Word Embeddings gegeben und die behandelten Typen von Embeddings werden anhand der beiden Methoden evaluiert. Außerdem wird untersucht, in wie weit die Anpassung eines Embeddings an die literarische Domäne Methoden beeinflusst.

1.2 Gattungen

Für eine klare Argumentation ist es notwendig, die Begriffe Genre und Gattung voneinander abzugrenzen. Während diese im anglo-amerikanischen Raum beide unter *genre* subsumiert werden, weisen sie im Deutschen divergierende Bedeutungen auf. Der Gattungsbegriff spielt eine übergeordnete Rolle und vereint zum Beispiel alle Erzähltexte unter der Kategorie Epik. In diesem Zusammenhang bezeichnet Genre die Subkategorien einer Gattung. Diese Einteilung ist jedoch wesentlich unschärfer als die der Gattungen. Zu dieser Problematik Lahn und Jan Christoph 2016:

> „Da ein einzelnes Merkmal für mehrere Genres charakteristisch sein kann, ist eine gewisse Kombination von Eigenschaften für ein Genre spezifisch. Der jeweilige Erzähltext muss aus diesem Merkmalsbündel allerdings nicht jede Eigenschaft realisieren; für die Genreeinordnung ist es ausreichend, wenn eine gewisse Anzahl an Kriterien aufzufinden ist."

Weitergehend wird dieser Zusammenhang als Familienähnlichkeit bezeichnet. Aus dieser Art der Ähnlichkeit leitet sich zusätzlich ab, dass die Bildung von Mischformen möglich ist. Gleichzeitig ist der Begriff des Genres an dieser Stelle zu klein, da bspw. Roman und Parabel Genres der Gattung Epik sind, der Künstler- und Bildungsroman aber ebenfalls als Genre des Romans bezeichnet werden. Da sich diese Arbeit ausschließlich mit Texten des Genres Roman und dessen Subgenres befasst, wird der Begriff Genre hier immer als Subgenre des Romans verwendet.
Der Bereich der Genre-Klassifikation hat in den Digital Humanities in der Vergangenheit bereits eine große Beachtung erfahren (Allison et al. 2011, Jockers 2013, Underwood 2015, Christof Schöch 2017, Hettinger et al. 2016). Die aus einem Fachbereich der Informatik, dem Information Retrieval, hervorgegangenen Methoden zur Klassifikation von Dokumen-

ten (Manning, Raghavan und Schütze 2010, Baeza-Yates, Ribeiro et al. 2011) wurden auf literarische Gattungen, Genres und Subgenres angewandt. Während sich Gattungen sicher unterscheiden lassen, bleibt die Erkennung von Subgenres hinter den Resultaten des Information Retrieval zurück (Hettinger et al. 2016).
Neben der Klassifikation von Genre wird im Feld der Digital Humanities die Zuweisung von Autoren (Burrows 2002b, Christof Schöch 2018, Evert et al. 2015) und in geringerem Maße die automatische Einteilung in Epochen erforscht. Für jede dieser Aufgabenstellungen werden die aus dem Information Retrieval bekannten Methoden verwendet, was auf den ersten Blick überraschen mag, jedoch plausibel wird, wenn man die Fragestellung jeweils als Identifikation von Textähnlichkeiten im Verhältnis zu Gruppen betrachtet. Dieser Umstand wirft allerdings das Problem der Signale auf. Für die hier genannten Anwendungen wären das Genre/Gattungs-, Autorenschafts- und Epochensignal. Es lässt sich an dieser Stelle zurück auf Lahn und Meister (2016) verweisen, da das auszuwertende Signal einem schwer differenzierbarem *Merkmalsbündel* gleichkommt.
Um eine bessere Unterscheidbarkeit aus Perspektive der Zielkategorien zu gewährleisten, ist es demnach essentiell, das gewünschte Signal aus dem Spektrum zu filtern. Für das Signal Autorenschaft sind dabei zwei Ansätze hervorzuheben: Zum einen die Fokussierung auf die häufigsten Wörter innerhalb eines Korpus, für welche nachgewiesen wurde, dass sie bereits einen erheblichen Teil der stilistischen Information eines Textes beinhalten (Evert et al. 2015) und von denen gleichzeitig angenommen werden kann, wenig sensitiv gegenüber Genre zu reagieren. Zum anderen das Gegenüberstellen zweier Textgruppen von Autoren, um anschließend zu ermitteln, welche Wörter als distinktiv für eine der Gruppen bezeichnet werden können (Christof Schöch 2018; siehe Kap. X.X). Beide

Verfahren erreichen gute Ergebnisse im Filtern von Signalen, weisen aber auch Schwächen auf. Die allgemeine Reduktion auf mfw[3] blendet zwar Störsignale aus, allerdings ist der Zusammenhang zwischen Häufigkeit und Informationsgehalt für stilistische Fragestellungen nicht völlig widerspruchsfrei, da es plausibel erscheint, dass auch Wörter unterhalb einer Häufigkeitsgrenze noch eine tragende Rolle spielen können. Bei der Verwendung von Zeta wird dieses Problem zwar gelöst, dafür sind die Ergebnisse extrem abhängig von der Zusammensetzung der Vergleichsgruppe.
Wichtiger für diese Arbeit ist aber, dass beide Methoden semantische Beziehungen zwischen Wörtern ignorieren. Dieser Umstand ist für stilistische Fragen auch von untergeordnetem Interesse, wird jedoch bei der Gattungsklassifikation entscheidend. Zur Verdeutlichung ein Beispiel:

> Gegeben sind 100 Romane aus dem Genre Western. In 90 dieser Romane wird das Wort “Revolver” zur Bezeichnung einer Schusswaffe verwendet. Die übrigen 10 sind von einem Autoren geschrieben, der stattdessen überdurchschnittlich oft das Wort “Colt” verwendet. Dies hat zur Folge, dass die “Colt”-Western aufgrund des Autorensignals in der Gruppe der Western insgesamt unähnlicher zum Durchschnitt gewertet werden.

Dieses Problem lässt sich vermeiden, wenn in die Berechnung von Ähnlichkeiten die semantische Information eingeht, dass “Colt” und “Revolver”, zumindest aus literarischer Sicht, synonym gebraucht werden können. Hieraus ergeben sich zwei Teilfragestellungen:

- 1. Welche Repräsentation von semantischer Ähnlichkeit eignet sich, um Gattungserkennung zu verbessern?

[3] most frequent words: Die häufigsten Wörter eines Textes oder einer Textsammlung

- 2. Wie kann diese in Ergänzung der etablierten Methoden eingesetzt werden?

Die Antwort auf die erste Fragestellung lässt sich bereits dem Titel dieser Arbeit entnehmen und lautet: Word Embeddings. Deren Funktionsweise soll in den nächsten Kapiteln veranschaulicht werden. Die zweite Frage wird in einer Reihe von Experimenten beleuchtet.

1.3 Was sind Word Embeddings?

Word Embeddings sind eine Repräsentation von Wörtern basierend auf deren Semantik. Diese Form der Repräsentation folgt dem Konzept der distributionellen Semantik. Dieses lässt sich auf zwei Grundannahmen zurückführen:

> „Language can be described in terms of a distributional structure, i.e., in terms of the occurrence of parts relative to other parts.“ Harris 1954

> „You shall know a word by the company it keeps.“ Firth 1957

Diesen Ansätzen folgend, lässt sich die Bedeutung eines Wortes also aus dessen Kontext oder Kontexten ermitteln. Nun stellt sich die Frage, wie dieser Kontext skalierbar und vor allem maschinenlesbar repräsentiert werden kann. Für diese Aufgabe wird üblicherweise ein hochdimensionaler Vektorraum genutzt. In diesem „Raum“ wird jedem Wort ein Vektor oder Punkt zugewiesen. Für einen Vektorraum mit drei Dimensionen wäre bspw. „Haus“ als $\langle 4, 1, 5 \rangle$ repräsentiert. Wie entsteht ein solcher Vektor? Dazu folgendes Beispiel mit Dokumenthäufigkeiten:
Gegeben sind zwei Dokumente D und deren Sätze:

D 1: Wir bauen unser Haus. Es wird ein kleines Haus.

D 2: Jetzt haben wir ein kleines Haus.

Aus diesen Sätzen lässt sich eine sog. Document-Term-Matrix erzeugen. Diese enthält die Information darüber, wie oft ein Wort in einem Text enthalten ist.

	wir	ein	bauen	Haus	kleines	unser	jetzt	wird
$D1$	1	1	1	2	1	1	0	1
$D2$	1	1	0	1	1	0	1	0

Tabelle 1.1: Document-Term-Matrix zu Beispiel

Wir sehen, dass „Haus“ in $D1$ zweifach und in $D2$ einmal vorkommt. Daraus lässt sich ein Vektor $\langle 2, 1 \rangle$ ableiten. Da es sich nur um eine 2-dimensionale Abbildung handelt, kann diese auch graphisch betrachtet werden (siehe Abb. 1.1). Aus diesem Beispiel lassen sich natürlich noch

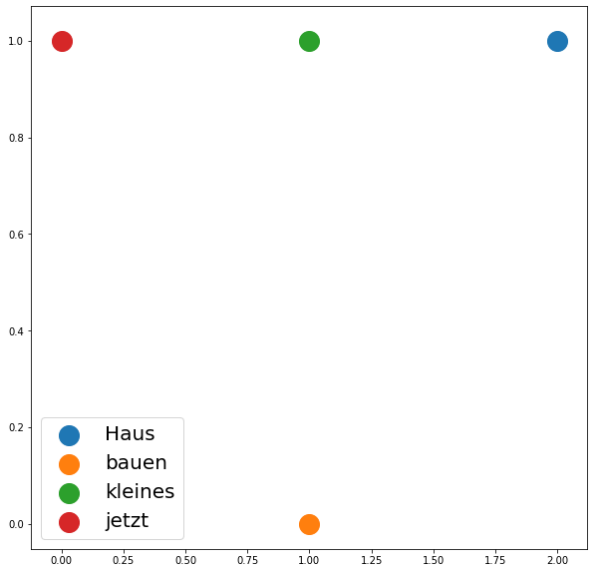

Abbildung 1.1: Plot für Häufigkeits-Vektoren

keine semantischen Informationen lesen, zum einen, da viel zu wenig Text verwendet wurde und zum anderen, weil Dokumenthäufigkeiten keine geeignete Quelle zum Erzeugen von Word Embeddings sind. Stattdessen wird der lokale Kontext eines Wortes innerhalb des Dokuments betrachtet.

Bei einer Kontextgröße von zwei Wörtern ergibt sich mit dieser Methode für das Wort Haus folgende Berechnung:

Kontext1: bauen unser |Haus| Es wird
Kontext2: ein kleines |Haus|
Kontext3: ein kleines |Haus|

Indexiert man anschließend jedes Wort mit einer Zahl und verwendet diese Zahl ergibt sich:

Wort	wir	bauen	unser	haus	Es	wird	ein	jetzt	haben
Index	0	1	2	3	4	5	6	7	8

Tabelle 1.2: Indexierung des Beispielsatzes

In dieser Form lässt sich der Kontext für „Haus“ folgendermaßen schreiben:

Kontext1: 2 3 |Haus| 4 5
Kontext2: 6 7 |Haus|
Kontext3: 6 7 |Haus|

Wort	wir	bauen	unser	haus	Es	wird	ein	jetzt	haben
Index	0	1	2	3	4	5	6	7	8
Häufigkeit im Kontext	0	0	1	1	1	1	2	2	0

Tabelle 1.3: Erzeugen eines Kontextvektors

Nun soll jedes Kontextwort durch eine Dimension des Vektors für „Haus“ repräsentiert werden. Dadurch ergibt sich für Haus der Vektor $\langle 0,0,1,1,1,1,2,2,0\rangle$, woraus sich jetzt bereits ableiten lässt, dass die Wahrscheinlichkeit „kleines“ und „ein“ im Kontext von Haus zu beobachten höher ist als die der übrigen Wörter in den Beispielsätzen.

Die Methoden zur Erzeugung von Word Embeddings sind natürlich wesentlich komplexer. Einige der geläufigsten Modelle werden im folgenden Abschnitt thematisiert.

KAPITEL 2

KONZEPTE UND MODELLE

2.1 Word Embeddings

Das folgende Kapitel gibt einen Überblick über die Entwicklung von Word Embeddings beginnend bei word2vec (Mikolov et al. 2013) über fastText (Bojanowski et al. 2017) bis hin zu den aktuell eingesetzten Embeddings ELMo (Matthew E. Peters et al. 2018) und Bert (Devlin et al. 2018). Es können an dieser Stelle nicht alle Embeddingtypen behandelt werden, der Vollständigkeit halber müssen aber noch gloVe (Pennington, Socher und Manning 2014), flair (Akbik, Blythe und Vollgraf 2018) und ConcpetNet Number Batch (Speer, Chin und Havasi 2017) zumindest erwähnt werden. Dass während der Arbeit an dieser Abhandlung bereits ein neues Embedding Bert als state-of-the-art abgelöst hat, zeigt noch einmal das immense wissenschaftliche Interesse an der Technologie Word Embedding. Da zu diesem Zeitpunkt noch kein Modell verfügbar ist, kann auch gpt-2 (Radford et al. 2019) keine Berücksichtigung finden.

2.1.1 word2vec

Word2vec bezeichnet eine Gruppe von Modellen zur Herstellung von Word Embeddings. Diese Modelle sind flache, zweilagige neuronale Netze, die darauf trainiert sind, sprachliche Zusammenhänge von Wörtern zu rekonstruieren. Word2vec nimmt einen großen Textkorpus als Eingabe und erzeugt einen Vektorraum, typischerweise mit mehreren hundert Dimensionen, wobei jedem einzelnen Wort im Korpus ein entsprechender Vektor im Raum zugeordnet wird. Wortvektoren werden im Vektorraum so positioniert, dass Wörter, die ähnliche Kontexte im Korpus teilen, im Raum in unmittelbarer Nähe zueinander stehen.
Word2vec kann eine von zwei Architekturen verwenden, um Word Embeddings zu erzeugen: Continuous Bag-of-Words (CBOW) oder Continuous Skip-Gramm. Das CBOW-Modell prognostiziert das aktuelle Wort auf Grundlage der umgebenden Kontextwörter. Die Reihenfolge der Kontextwörter hat keinen Einfluss auf die Vorhersage. In der Skip-Gramm-Architektur verwendet das Modell das aktuelle Wort, um das umgebende Fenster von Kontextwörtern vorherzusagen. CBOW oder continuous-bag-of-words Modelle zielen darauf ab, einem Kontext ein Wort zuzuweisen. Skip-gram dagegen ist konzipiert, um für ein Wort einen Kontext zu erzeugen (siehe Abb. 2.1).

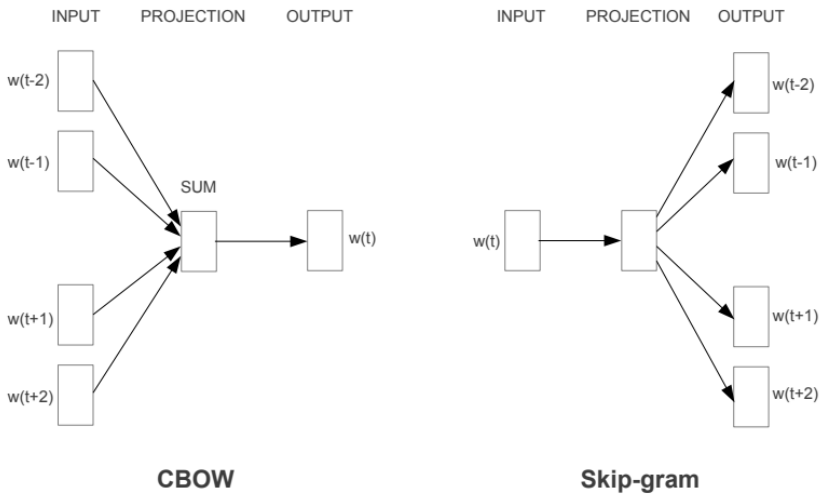

Abbildung 2.1: CBOW und Skip-gram Modelle Mikolov et al. 2013

Um es an einem Beispiel zu verdeutlichen, sei folgender Satz gegeben: „Wir spielen Katz und Maus.“ Für CBOW ergibt sich die Aufgabe aus dem Input „Wir spielen x und Maus“ das Wort „Katz“ für x vorherzusagen. Umgekehrt muss bei Skip-gram für das Wort „Katz“ der Kontext „Wir spielen x und Maus“ errechnet werden. Die Umsetzung der Skip-Gram und CBOW Modelle wird durch ein Feed-Forward[1] Neural Network Bengio et al. 2003 erreicht.

Um den Lernprozess zu verstehen, ist etwas Vorwissen zu neuronalen Netzen nötig. Die kleinste Einheit in neuronalen Netzen ist das Neuron. Ein Neuron hat die Fähigkeit Information aufzunehmen, zu verändern und weiterzugeben. Neuronen sind in Schichten (engl. Layer) organisiert. Ein Netzwerk besteht aus einem Inputlayer (Encoder), welcher Signale von außen aufnimmt, beliebig vielen Hidden-Layers und einem Output-Layer (Decoder), welcher die verarbeiteten Signale wieder ausgibt. Das Verhalten der Neuronen gegenüber Information wird über Gewichte ge-

[1]Feed-Forward bedeutet in diesem Zusammenhang, dass innerhalb des Netzwerkes Informationen lediglich vorwärts, also an die nächste Schicht von Neuronen weitergereicht wird. Der Begriff wird in Abgrenzung zu rekurrenten Netzwerken gebraucht, in deren Architektur Information auch an Neuronen der gleichen oder vorherigen Schicht übergeben werden kann.

steuert. Damit ein Netz eine Aufgabe lösen kann, müssen die Gewichte der Neuronen in den Hidden-Layers Werte annehmen, die zum richtigen Ergebnis an der Übergabe des Output-Layers führen, in Relation zu den Signalen, welche im Input-Layer eingegeben wurden. Die Gewichte der Neuronen werden randomisiert initialisiert und nach jeder Iteration von einem Paket an Information (batch) durch das Netz wird der Abstand der Ausgabe der letzten Schicht mit dem Zielwert verglichen. Dieser Abstand (Loss) wird verwendet, um durch Backpropagation zu ermitteln, welche Gewichte verändert werden müssen, um näher an den Zielwert zu gelangen. Diese Änderung der Gewichte ist der eigentliche Lernvorgang innerhalb eines neuronalen Netzes.
Im CBOW Verfahren erhält der Input-Layer die Eingabe „Wir spielen x und Maus“ und soll im Output-Layer die Ausgabe „Katz“ übergeben. Das Netzwerk, welches für word2vec verwendet wird, hat nur einen Hidden-Layer. Dessen Gewichte werden also solange optimiert, bis tatsächlich das gesuchte Ergebnis für alle Sätze und Zielwörter möglichst richtig berechnet wird. Die Embeddings werden erzeugt, indem für jedes Zielwort der Zustand der Neuronen des Hidden-Layer extrahiert wird, bevor er vom Output-Layer decodiert wird. Die so erzeugten Vektoren können verwendet werden, um arithmetische Rechenoperationen auf semantischen Beziehungen durchzuführen. Das bekannteste Beispiel ist die Rechnung „König“ − „Mann“ + „Frau“, welche zum Ergebnis „Königin“ führt.

2.1.2 Fasttext

Eines der Kernprobleme bei Verwendung von word2vec ist das fest begrenzte Vokabular. Ein Wort, welches im Datensatz mit welchem das Word Embedding trainiert wurde, nicht enthalten ist, kann auch keinen Vektoren zugewiesen bekommen. Analog ist die Repräsentation eines seltenen Wortes unsicherer als die eines Frequenten. Dies ist besonders kritisch für

Sprachen, in denen Wörter stark flektiert werden oder zur Bildung von Komposita neigen, da die Wahrscheinlichkeit für seltene oder überhaupt nicht im Trainingsdatensatz enthaltene Wörter steigt. Selbst wenn der Idealfall, dass im Trainingskorpus jedes denkbare Wort enthalten sein sollte, eintritt, wäre ein Modell, welches auch jedem Wort einen eigenen Vektor zuweist aufgrund seiner Größe kaum prozessierbar.
FastText adressiert diese Probleme, indem es keine Repräsentationen für Wörter, sondern für Ketten von Buchstaben (character n-grams) berechnet. Beispielsweise wird das Wort *Haustür* in FastText als Summe seiner n-gramme[2] <*Ha, Hau, aus, ust, stü, tür, ür*>, <*Haustür*> repräsentiert. Es werden zusätzlich die Zeichen < und > eingeführt, um den Anfang und das Ende eines Wortes zu markieren und so Prä- und Suffixe besser zu erkennen. Außerdem wird immer auch das Wort als Ganzes einbezogen. Für die tatsächliche Berechnung des Embeddings verwendet FastText die mit word2vec eingeführten Skipgram und CBOW Modelle Bojanowski et al. 2017.

2.1.3 ELMo

ELMo (Embeddings from Language Models) (Matthew E. Peters et al. 2018) grenzt sich von word2vec und fastText ab, indem es direkt an das Konzept traditioneller Sprachmodelle anknüpft. Diese Sprachmodelle berechnen gegeben eine feste Anzahl an aufeinander folgenden Wörtern eines Textes, die Wahrscheinlichkeit für das nächste Wort (Seymore, McCallum und Rosenfeld 1999). Für das Training der Embeddings wird allerdings nicht nur der Kontext vor dem Zielwort, sondern auch der folgende Kontext verwendet. Die Aufgabenstellung, also das Vorhersagen eines Wortes aufgrund seines Kontextes, ähnelt zwar dem CBOW

[2]Dieses Beispiel nimmt $n = 3$ an, tatsächlich wird kein fester Wert, sondern ein Bereich angegeben, so dass ein Wort durch seine 3, 4 und 5-gramme gleichermaßen repräsentiert werden kann.

Modell, unterscheidet sich aber darin, dass die Vielzahl an Kontexten eines Wortes nicht genutzt wird, um für jedes Wort einen festen Vektor zu errechnen, sondern den Vektor eines Wortes in Abhängigkeit seines aktuellen Kontextes zu repräsentieren. Man spricht daher von einem kontextsensitiven Embedding.
Gegeben ein Segment von N Token $(t_1, t_2, ..t_N)$ berechnet ein Sprachmodell die Wahrscheinlichkeit für jedes Token k auf Grundlage der vorherigen Token $(t_1, t_2, ..t_{k-1})$. Umgekehrt berechnet ein rückgerichtetes Sprachmodell die Wahrscheinlichkeit auf Basis von $(t_{k+1}, t_{k+2}, ..)$. Um die technische Umsetzung dieses Konzepts verständlich zu machen, ist ein kurzer Exkurs in die Funktionsweise von LSTMs nötig.

LSTMs (Long Short-Term Memory) werden in rekurrenten neuronalen Netzen eingesetzt und ermöglichen der Netzarchitektur Informationen über vergangene Iterationen zu erhalten (Hochreiter und Schmidhuber 1997). Feed-forward Netzwerke passen ihre Gewichte immer anhand der gerade prozessierten Batch an, ohne die Möglichkeit zu erfassen, dass zuvor verarbeitete Signale Einfluss auf die Behandlung der aktuellen Trainigsdaten haben können. Daher sind die für die Vorhersage von abhängigen Sequenzen, wie beispielsweise Entwicklungen über Zeit, nicht geeignet. Ein LSTM Layer ist streng genommen kein Layer, sondern ein eigenes Netzwerk bestehend aus vier neuronalen Schichten. Diese teilen sich in drei Sigmoid (σ)[3] und eine tanh[4] Schicht auf. In Abbildung 2.2 repräsentiert die obere horizontale Linie das Langzeit- (cell state) und die untere Linie das Kurzzeitgedächtnis des Netzwerks. Die erste Sigmoid Schicht ist das *Forget Gate*. Dieses reguliert, wie viel und vor allem welche Information des vorherigen LSTMs an den Cell State weitergegeben

[3] Eine $\sigma - Funktion : \sigma(t) = \frac{e^t}{1+e^t}$ errechnet für jedes Neuron in Abhängigkeit von Gewichten einen Wert zwischen 0 und 1, welcher bestimmt wie viel Information an die nächste Schicht weitergegeben wird.
[4]

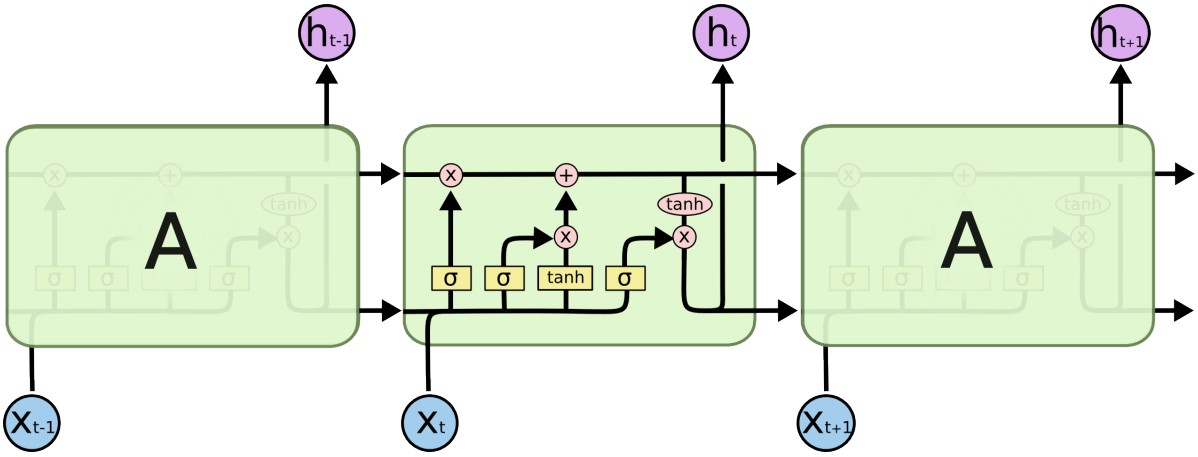

Abbildung 2.2: The repeating module in an LSTM contains four interacting layers. Aus Olah 2015

werden soll. Die nächste Einheit, bestehend aus der zweiten Sigmoid und der tanh-Schicht, bildet das *Input Gate*, welches bestimmt, welche Informationen aus der aktuellen Eingabe hinzugefügt werden. Die letzte Sigmoid-Schicht, das *Output Gate*, errechnet aus der Eingabe und dem Cell State, welche Informationen an die nächste Schicht des Gesamtnetzes, sowie an das nächste LSTM weitergegeben werden.

Die Architektur des ELMo Netzwerks beinhaltet zwei Schichten von LSTMs, welche wiederum in vor- und rückgerichtete Blöcke unterteilt werden. Diese Struktur wird, wie bereits beschrieben, nach dem Konzept der Sprachmodelle trainiert. Um das so generierte sprachliche Wissen produktiv zu nutzen, werden die LSTMs anschließend aus dem Modell herausgelöst, diese bilden das kontextsensitive Embedding. Das Modell kann in anderen neuronale Netzen eingesetzt werden, die Autoren schlagen vor, lediglich noch eine letzte Schicht auf das Embedding aufzusetzen, welche die für die jeweilige Aufgabe relevanten Informationen filtert. Tests lassen vermuten, dass die erste LSTM Schicht mehr Information über grammatikalische und syntaktische Eigenschaften von Sprache beinhaltet, da mit ihren Vektoren bessere Ergebnisse für Aufgaben wie POS-Tagging erzielt

werden können als mit der zweiten Schicht (Matthew E Peters et al. 2018). Diese ist dafür geeigneter für Aufgaben, die semantische Informationen benötigen, wie bspw. Disambiguierung.

2.1.4 Bert

Bert (Bidirectional Encoder Representations from Transformers) Embeddings zählen wie ELMo zu den kontextsensitiven Embeddings. Bert unterscheidet sich von ELMo in drei wesentlichen Punkten: Tokenisierung, Training des Sprachmodells und Netzstruktur (Devlin et al. 2018). Bert verwendet weder eine klassische 1:1 Beziehung zwischen Token und Wort, noch ein generisches *n*-gram Verfahren wie fastText. Stattdessen wird das von Wu et al. 2016 eingeführte Verfahren der *Word-Piece*-Tokenisierung eingesetzt. Hierbei wird Tokenisierung als Optimierungsproblem definiert: Gegeben eine Anzahl zu verwendender character ngrams; Welche müssen ausgewählt werden, um ein Korpus vollständig repräsentieren zu können? Bert verwendet 30.000 pieces. Obwohl das Modell aus linguistischer Sicht fragwürdig erscheint (siehe Beispiel), da es morphologische Strukturen ignoriert, führt seine Verwendung, bspw. in maschinellen Übersetzungen zu besseren Ergebnissen. Aus Wu et al. 2016:

> Word: Jet makers feud over seat width with big orders at stake
> wordpieces: _J et _makers _fe ud _over _seat _width _with _big _orders _at _stake

Das Training der Bert Embeddings erfolgt durch ein maskiertes Sprachmodell. Der Input für das Training besteht aus Segmenten zu je 512 Token[5]. Von diesen Token werden 15% zur Maskierung ausgewählt und zu 80% durch ein spezielles Maskierungswort, zu 10% durch ein zufällig

[5]Hier sind Wörter und Satzzeichen gemeint, die *word-pieces* werden im Netz erstellt

gewähltes Wort und zu wiederum 10% durch sich selbst ersetzt. Diese Aufteilung wirkt zunächst willkürlich, erklärt sich jedoch daraus, dass bei einer Maskierung des Zielwortes zu 100% das Modell keine eigene Repräsentation für nicht maskierte Token erlernt, sondern diese lediglich zur Kontextualisierung der Maskierung nutzt. Werden die übrigen 20% vollständig durch zufällige Token ersetzt, könnte das Modell gar nicht mehr lernen, da sich jede Anpassung aufgrund der maskierten Token als falsch erweisen würde. Das Beibehalten des Zielwortes als Alternative zur Maskierung führt zu einer Voraussage ohne Kontext, lediglich auf dem Tokenembedding (Horev 2018). Das Modell wird im Unklaren darüber gelassen, welches Token ersetzt wurde, sodass für jedes Token eine eigene kontextualisierte Repräsentation vorgehalten werden muss.
Aus dieser Aufgabenstellung ergibt sich, dass eine Architektur mit LSTMs extrem aufwändig wäre, da so für jedes Token im Segment gleichzeitig vorgreifende und zurückliegende Informationen bereitgestellt werden müssten. Daher verwendet Bert keine LSTMs, sondern Transformer.

Exkurs: Transformer Der von Vaswani et al. 2017 eingeführte Transformer-Layer basiert auf dem Konzept der Attention. Attention löst ein Problem, welches in rekurrenten Netzen in Zusammenhang mit weit zurückliegenden Eingaben auftritt. LSTMs erzeugen ihre Ausgabe aus dem letzten *hidden state* und der aktuellen Eingabe. Das Langzeitgedächtnis, also der *hidden state*, muss sämtliche zurückliegende[6] Information bereitstellen, welche für die Verarbeitung der aktuellen Eingabe benötigt wird und das ohne die Eingabe im Vorhinein zu kennen. Dieser Umstand führt dazu, dass LSTMs dazu neigen, weit zurückliegende Informationen zu vergessen, da nicht vorauszusehen ist, ob diese noch benötigt werden.

[6] hier im Sinne der Verarbeitungszeit verwendet, trifft also auch auf Vorgreifendes zu

Attention Mechanismen forcieren dieses Problem, indem sie den Zugriff auf alle zurückliegenden *Hidden States* ermöglichen und gleichzeitig das Filtern der Informationen in Abhängigkeit der Eingabe erlernen. Abbildung 2.3 zeigt die Architektur eines neuronalen Netzes zur maschinellen Übersetzung. Die *Hidden States* der rekurrenten Schichten h_{1-T} werden mittels eines Filters a unter Berücksichtigung der Eingabe t und der zur Verfügung stehenden Information aus h_{1-T} an die nächst höhere Schicht weitergegeben (Bahdanau, Cho und Bengio 2014). Diese Architektur beinhaltet jedoch weiterhin rekurrente Blöcke, welche in Abhängigkeit aller ihrer Vorgänger stehen. Daher eignet sich diese Architektur nicht zur Parallelisierung.

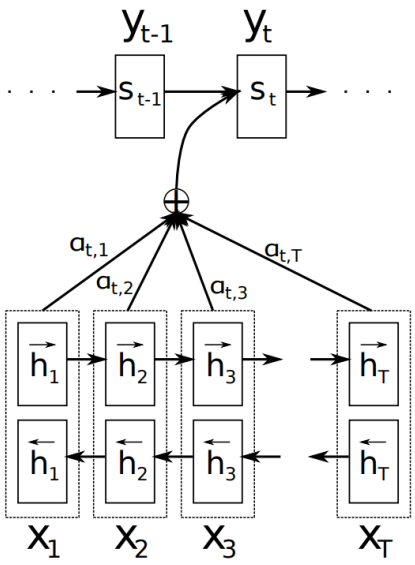

Abbildung 2.3: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence (x_1, x_2, ..., x_T) Aus: Bahdanau, Cho und Bengio 2014

Der Transformer-Layer bietet die Möglichkeit diese rekurrenten Anteile komplett durch Attention zu ersetzen. Er besteht aus einer Encoder und einer Decoder Komponente. Jede dieser Komponenten ist wiederum unterteilt in mehrere Schichten, im Fall von Bert werden 6 Schichten verwendet. Die Encoder Schichten bestehen aus einem Self-Attention Mechanismus und einem Feed-Forward-Network. Die Decoder Schichten haben den gleichen Aufbau, ergänzt durch einen weiteren Attention Mechanismus zwischen Self-Attention und Feed-Forward-Network. Bevor eine Sequenz aus Wörtern den ersten Encoder passiert, wird diese durch ein Embedding in einen Vektor gewandelt. Anschließend folgt die erste Self-Attention Schicht. Self-Attention unterscheidet sich von der im vorherigen Absatz erläuterten Attention dadurch, dass nicht im Fokus steht,

ob ein Wort für das Verständnis eines Satzes oder eine andere Aufgabe relevant ist. Stattdessen wird ermittelt, welche Wörter des Satzes im Bezug auf das aktuell prozessierte Wort von Bedeutung sind.

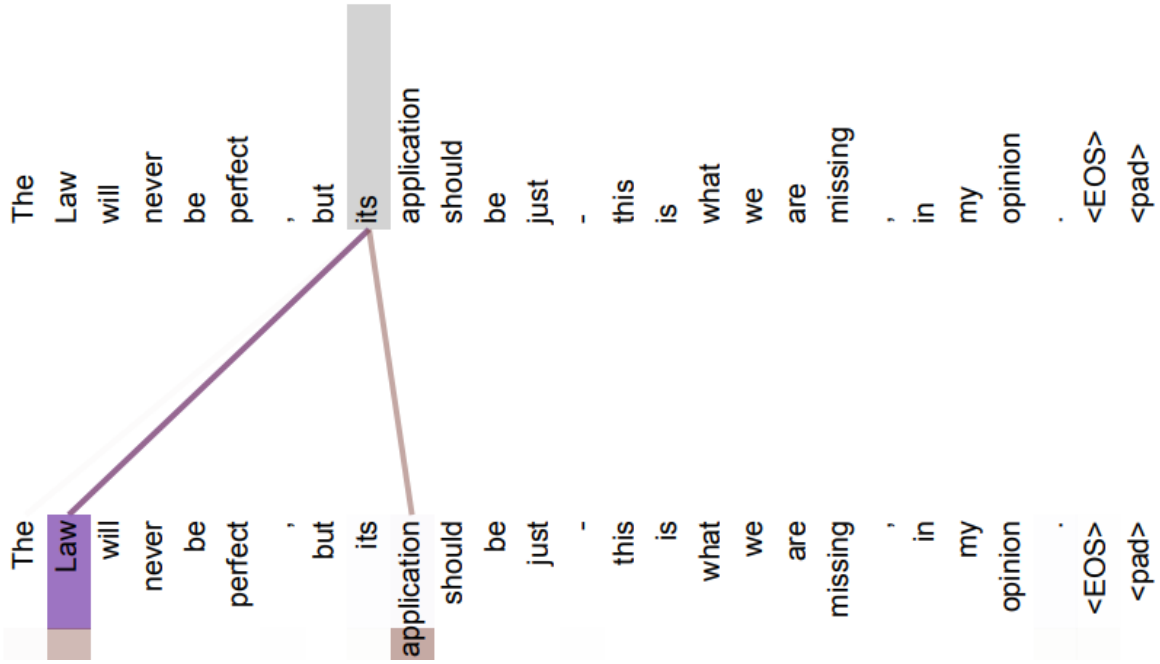

Abbildung 2.4: Isolated attentions from just the word „its“ for attention heads 5 and 6. Aus Vaswani et al. 2017

Abbildung 2.4 zeigt die Self-Attention für das Wort „its“, die Attention wird auf das Wort „Law“ gelegt, auf welches „its“ in diesem Satz referenziert und „application“ was wiederum im Verhältnis zu „Law“ steht. Diese Information wird zusammen mit dem Embedding Vektor an die Feed-Forward Schicht weitergegeben. Diese erzeugt dann eine neue Repräsentation und übergibt sie an den nächsten Encoding Block. In Vaswani et al. 2017 wird neben Self-Attention zusätzlich Multi-Head Attention verwendet. Diese Form der Attention teilt den Vektorraum des Embeddings in Unterräume und ermittelt dann in jedem dieser Unterräume Self-Attention. Auf diese Weise kann ein Transformer Strukturen und Aspekte von Sprache wie Dependenz erkennen und verarbeiten für die ansonsten Parser verwendet werden (Goldberg 2019).

Zusätzlich zum Sprachmodell wird eine Voraussage des nächsten Segments trainiert. Hierbei erhält das Netz ein zusätzliches Segment, welches zu 50% ein zufällig aus dem Korpus gewähltes oder das tatsächlich folgende Segment ist. So wird das Erkennen semantischer Ähnlichkeit über einen großen Kontext erlernt.
Um Bert Embeddings als Feature zu verwenden, wird jeder Sequenz von Token in das zuvor trainierte Netz gegeben. Die Token werden anschließend durch die Attentionwerte jedes Transformers und dessen Attentionheads repräsentiert.

2.2 Zeta

Zeta ist eine Methode, welche entwickelt wurde, um die Distinktivität oder engl. *Keyness*, bezeichnet für die Eigenschaft eines Wortes unter einer Fragestellung als Schlüssel zu fungieren, von Wörtern für eine Gruppe von Texten zu ermitteln. Die Verfahren zur Ermittlung von Zeta-Werten stammen aus der Stilometrie, wo die Methode angewandt wird, um distinktive Wörter als Marker für Autorenschaft zu ermitteln. Die Stilometrie beschäftigt sich in der Frage der Autorschaft vor allem mit den häufigsten Wörtern einer Gruppe von Texten. Diese werden zwar mit hoher Wahrscheinlichkeit in jedem Text verwendet, allerdings schwankt das Verhältnis der Wörter untereinander stark genug, um für Autoren typische Muster zu extrahieren und auf deren Basis zu einer Zuweisung von Texten zu gelangen (Burrows 2002a). Man spricht auch vom Schlüsselprofil eines Autoren.
Burrows 2007 argumentiert, dass für Leser die Verteilung von Wörtern des oberen Frequenzspektrums[7] nur schwer zu erfassen ist und es mög-

[7]Die Einteilung von Wörtern in Frequenzbereiche orientiert sich am Zipfschen Gesetz, nachdem die Häufigkeit eines Wortes innerhalb eines Korpus umgekehrt proportional zu seinem

lich sein muss, auch distinktive Wörter aus dem mittleren und unteren Frequenzbereich zu identifizieren. Für den mittleren Frequenzbereich gilt weiter, dass sobald Wörter, welche in allen Texten vorkommen, ausgeschlossen werden, nur solche übrig bleiben, welche von vielen Autoren, dafür aber selten verwendet werden. Verwendet ein Autor einige dieser Wörter häufiger, ist ihr erneutes Auftreten in neuen Texten desselben Autors wahrscheinlich.

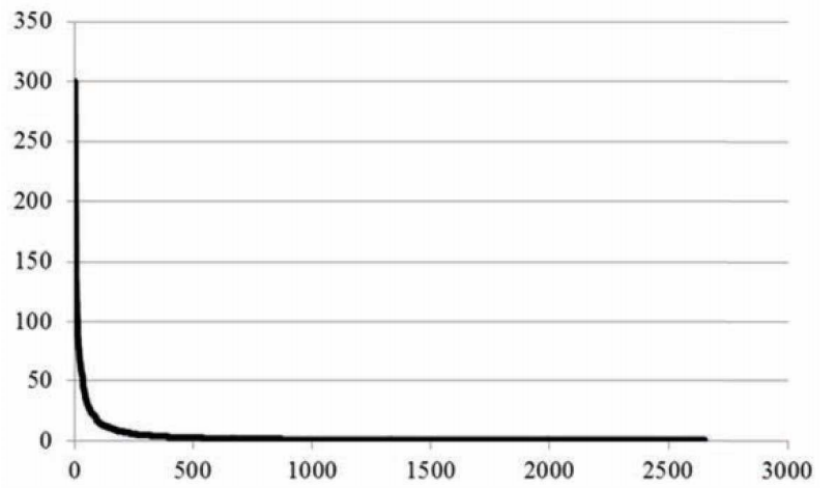

Abbildung 2.5: C Häufigkeitsverteilung der 2.652 Wortformtypen in Kafkas Erzählung „Der Heizer“; y-Achse: Wortformfrequenz, x-Achse: Häufigkeitsrang der Wortformen (Rangdarstellung). Aus: Engelberg 2015

Während das Delta-Verfahren (Burrows 2002b, Burrows 2003) genutzt wird, um aus einer Gruppe von Autoren einen Text seinem Urheber zuzuweisen und seine statistische Aussagekraft vor allem aus den hochfrequenten Wörtern zieht, werden Zeta für das mittlere und Iota für das niedere Frequenzspektrum angewandt, um ausgehend von einem Autoren zu ermitteln, welcher Text am ehesten ihm oder ihr zugeschrieben werden kann.
Die Berechnung von distinktiven Merkmalen aufgrund von Häufigkeits-

Rang in der Frequenztabelle ist. (Zipf 1949) Daraus folgt beispielsweise, dass das häufigste Wort doppelt so oft im Korpus enthalten ist wie das zweithäufigste.(siehe Abbildung2.5)

verteilungen fächert sich nach Christoph Schöch et al. 2018 in vier Ansätze auf:

- Likelihood-Quotienten-Tests
- Transformationen, wie term frequency-inverse document frequency (tf-idf)
- Hypothesentests auf Verteilungseigenschaften (z.b. t-test)
- Dispersionsmaße, welche die Stabilität von Merkmalsverteilungen prüfen

Das von Burrows entwickelte Zeta gehört zur Gruppe der Dispersionsmaße. Um eine Vergleichbarkeit herzustellen werden die untersuchten Texte in gleich lange Segmente eingeteilt. So werden unerwünschte Effekte durch schwankende Textlängen verhindert. Nun wird für jedes Wort die Anzahl der Segmente ermittelt, welche dieses mindestens einmal enthalten. Diese Kennzahl (document proportion, dp) wird mit dem der Vergleichsgruppe subtrahiert, sodass ein Zeta Wert zwischen -1 und 1 ermittelt werden kann. Ein sehr hoher oder niedriger Wert (z) steht für starke Distinktivität eines Wortes (w) für die Unterscheidung zwischen untersuchter (Ug) und Vergleichsgruppe (Vg) (nach Christoph Schöch et al. 2018):

$$z_w = dp(Ug_w) - dp(Vg_w) \tag{2.1}$$

Dieses Vorgehen führt dazu, dass Worte des oberen und unteren Frequenzspektrums durch hohe Werte in beiden oder keiner der Gruppen marginalisiert werden. Gleichzeitig lässt sich aus der Formel bereits ableiten, dass ein Wort nie einen höheren Zeta-Wert als seine Document Proportion erreichen kann, selbst wenn es in der Vergleichsgruppe nicht

vorkommt und somit als distinktiv betrachtet werden kann. (Zur Verdeutlichung siehe: rote Markierung in Abbildung 2.6)

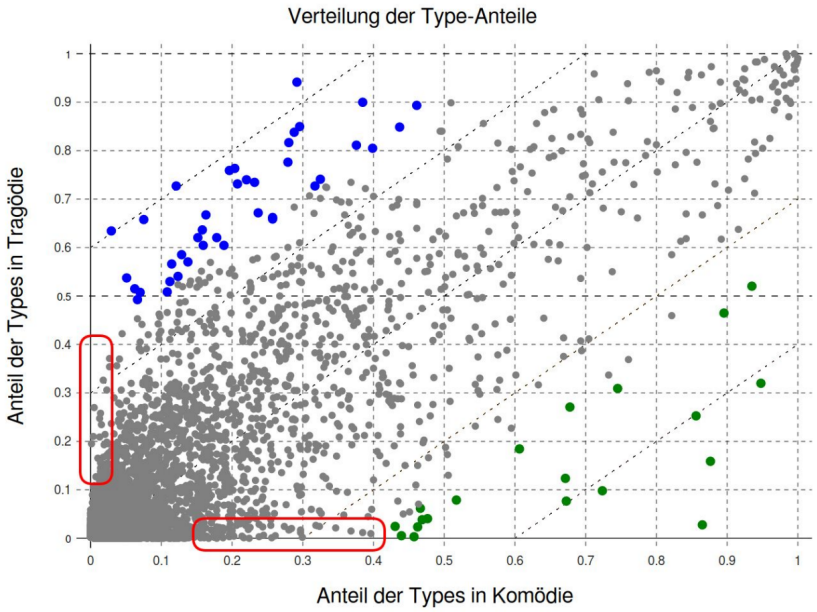

Abbildung 2.6: Scatterplot der Wörter in zwei Textgruppen: „Document Proportions“ der Wörter in zwei Textgruppen (x- und y-Achse) und resultierende Zeta-Werte (Distanz von der Diagonale). Aus Christoph Schöch et al. 2018

.

Christof Schöch et al. 2018 erprobt Variationen von Zeta, um auch diese Wörter zu erfassen. Dafür werden die Document Proportions logarithmisch transformiert und durch relative Häufigkeit ersetzt. Außerdem wird das Verfahren dahingehend verändert, dass die Werte der beiden Gruppen dividiert statt subtrahiert werden. Alle möglichen Kombinationen und ihre Bezeichnungen sind Tabelle 2.1 zu entnehmen. Das klassische Zeta nach Burrows 2007 entspricht hier sd0.

	Document Proprotions		relative Häufigkeiten	
Transformation	keine	log2	keine	log2
Subtraktion	sd0	sd2	sr0	sr2
Division	dd0	dd2	dr0	dr2

Tabelle 2.1: Übersicht über die Varianten von Zeta und ihrer Labels. Aus Christof Schöch et al. 2018

Für eine Unterscheidung zwischen je 12 Romanen aus Spanien und aus Lateinamerika wird gezeigt, dass sd2 mit einer Accuracy von .98 dem klassischen Zeta (.81) überlegen ist. Zusätzlich wird gezeigt, dass die Klassifikation sich mit steigender Segmentgröße verbessert (Christoph Schöch et al. 2018). Die Rechenvorschrift für sd2-zeta enthält eine zusätzliche Variable l, um zu verhindern, dass der Logarithmus von 0 berechnet wird. Sie lautet:

$$z_w = \log_2(dp(Ug_w) + l) - \log_2(dp(Vg_w) + l) \tag{2.2}$$

KAPITEL 3

RESSOURCEN

3.1 Korpus

Die für die folgenden Experimente verwendeten Texte sind, hält man sich an das Schichtmodell bestehend aus Literaten-, Unterhaltungs und Trivialliteratur, der letzten Gruppe zugeordnet. Statt dem Begriff der Trivialliteratur wird hier der weniger vorverurteilende Term der Schemaliteratur zur Beschreibung des Textmaterials verwendet. Statt Literatur nach ihrer Qualität beurteilen, erfolgt eine Einordnung von Werken nach den an sie gestellten Anforderungen. Diese Anforderungen entspringen einer an literarische Texte gerichtete Erwartungshaltung, welche sich wiederum differenzieren lässt. Während von Hoch- oder Literatenliteratur erwartet wird, ein hohes Maß an Variation im Verhältnis zu Vorhergegangenem und damit Innovation zu erzeugen, ist der Anspruch an Schemaliteratur komplimentär. Es ist also konstituierend für diese Literatur, möglichst exakt dem zu entsprechen, was bereits bekannt und damit erwartbar, also strikt einem Schema zuzuordnen ist (Zimmermann 1979, S.36f).
Die Zugehörigkeit der Texte zu dieser Gruppe lässt sich auch aus deren

Publikationsform ableiten. Veröffentlicht wurden die Texte als Heftromane, welche im DIN A5 Format und mit einer festen Länge von 64 Seiten vor allem über Kioske oder über Abonnements und nicht im klassischen Buchhandel vertrieben wurden (Wildberger 1988, S. 48). Während das Phänomen Heftroman in Deutschland bereits Mitte des 19. Jahrhunderts auftaucht[1], konstituiert sich das vorliegende Korpus aus Texten hauptsächlich aus der Zeit zwischen 1970-1990. Es ist zwischen zwei Arten von Heftromanen zu unterscheiden: Serien, welche ein ähnliches Personal um einen Protagonisten in jeder Ausgabe ein abgeschlossenes Abenteuer bestehen lassen, wobei zusätzlich die romanübergreifernde Handlung vorangetrieben wird (z.b. *Perry Rhodan*, *Jason Dark*) und Reihen, deren Gemeinsamkeit nur in der Thematik zu finden ist (z.b. *Alpenglück*, *Die Welt der Hedwig Courths-Mahler*). Oft dienen Reihen als Testblase, um die Installation neuer Serien zu prüfen. So entstammt der Protagonist John Sinclair der Serie Jason Dark ursprünglich aus der Reihe Gespenster-Krimi des Bastei-Verlags, gleiches gilt für Dämonenkiller Dorian Hunter, welcher zuerst in der Reihe Vampir-Horror-Romane (Erich Pabel Verlag) in Erscheinung tritt. In der Regel wird eine Serien-Auskopplung zunächst vom Autor des Pilotromans fortgesetzt und später durch eine wechselnde Gruppe von Autoren unterstützt.
Die historische Entwicklung der Genres innerhalb des Heftromans lässt sich aufgrund der komplexen Publikationsgeschichte, durchzogen von Verlagsübernahmen, Einstellung und Wiederaufnahme von Serien und Reihen, erneutes Publizieren von alten Ausgaben in Sonderheften und Features nur schwer rekonstruieren. Eine Annäherung kann jedoch, zumindest für die Nachkriegszeit, anhand des zeitweise auflagestärksten Verlags Zauberkreis geschaffen werden. Demnach ist das Ursprungsgenre der klassische Liebesroman, welcher ab 1951 in der Reihe *Gold-Roman*

[1] In Form von Kolportage- oder Lieferromanen Huegel 2002

erscheint. Bereits zwei Jahre später wird die Reihe *Silber-Roman* begründet, welche sich an männliche Leser wendet – schon hier ist die bis heute typische Fokussierung auf das Geschlecht der Zielgruppe zu beobachten – und zunächt Kriminal- und kurz darauf auch Western- und Agentenromane beinhaltet. Mitte der 60er Jahre wurde das Portfolio um eine Science-Fiction-Reihe ergänzt. Nachdem in den frühen 1970ern die Veröffentlichung von Gruselromanen innerhalb der Silber-Romane scheiterte, wurde eine eigene Reihe *Silber-Grusel-Krimi* geschaffen (Schnabel 2011).
Dieser Abriss beinhaltet bereits die im Korpus vertretenen Genres, ergänzt werden müssen noch Kriegsromane, hauptsächlich sogenannte *Landser* Hefte (ab 1957, Pabel Moewig Verlag) und Abenteuerromane, welche zum größten Teil von Seefahrern und Piraten handeln. Zu erwähnen ist außerdem der Romantic-Thriller oder Gothic-Romance-Roman. Dieses Mischgenre zwischen Grusel- und Liebesroman, ist zwar kein Einzelfall, aber das am weitesten verbreitetste Mischgenre innerhalb der Heftromane und das einzige, welches männliche Autoren für eine weibliche Zielgruppe zulässt (Gaslicht, Pabel Moewig; Geheimnis-Roman, Bastei) (Käther 2018).
Es stellt sich hier die Frage, warum in dieser ansonsten methodisch orientierten Arbeit ausgerechnet der im literaturwissenschaftlichen Diskurs wenig beachtete Heftroman als Datengrundlage verwendet wird. Zum einen spricht die schlichte Verfügbarkeit der Masse an deutschsprachigem Textmaterial für diese Entscheidung. Der konstante Umfang schließt Effekte, welche beim Vergleich unterschiedlich langer Texte auftreten, aus. Zum anderen wird die ansonsten komplexe Einordnung in Genres hier bereits verlagsseitig übernommen. Zusätzlich bietet der Umstand, dass innerhalb einer Serie mehrere Autoren Romane schreiben, eine ansonsten nur schwer herzustellende Möglichkeit, die Qualität von Autorschaftsat-

tribution zu evaluieren, da inhaltliche Differenzen innerhalb einer Serie als minimal angesehen werden können.

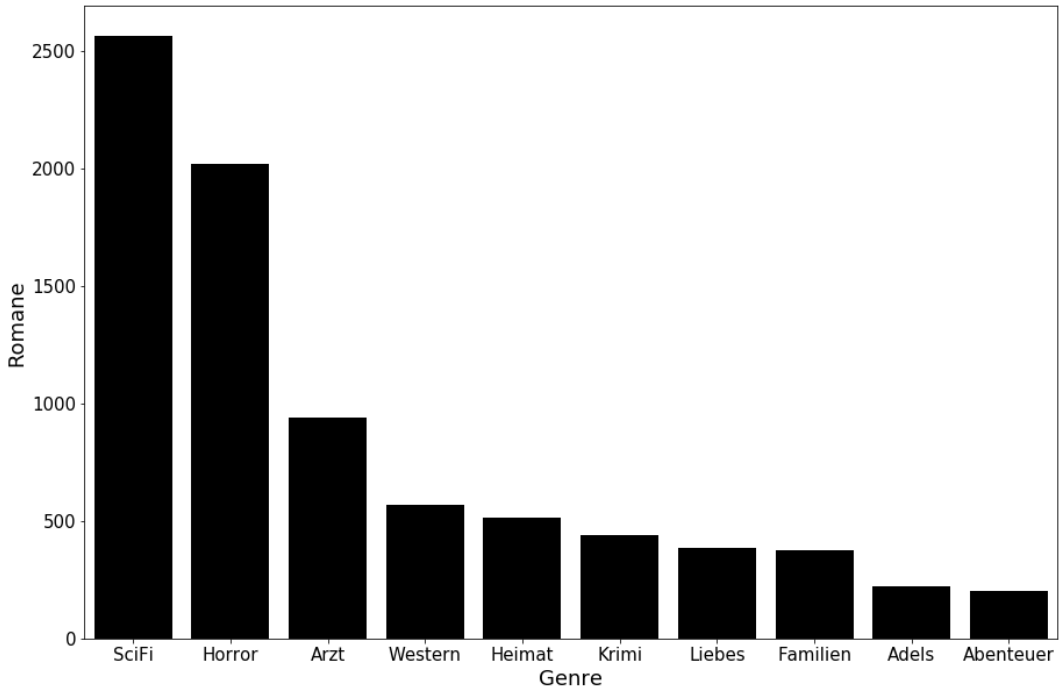

Abbildung 3.1: Anzahl Romane nach Genres

Das Korpus umfasst 8227 Heftromane mit insgesamt ca. 265.000.000 Token. Das Verhältnis der Genres lässt sich Abbildung 3.1 entnehmen. Das Genre des Liebesromans ist hier aufgefächert in den klassischen Liebesroman, sowie Arzt-, Heimat-, Adels- und Familienroman. Streng genommen können diese Texte unter der Bezeichung Frauenroman zusammengefasst werden, was von Verlagen auch so gehandhabt wird, siehe bspw. *Gold-Romane* (Zauberkreis Verlag). Da jede dieser Untergruppen an eine klar definierte Erwartungshaltung gebunden ist, werden sie im folgenden als eigenständige Genres behandelt.

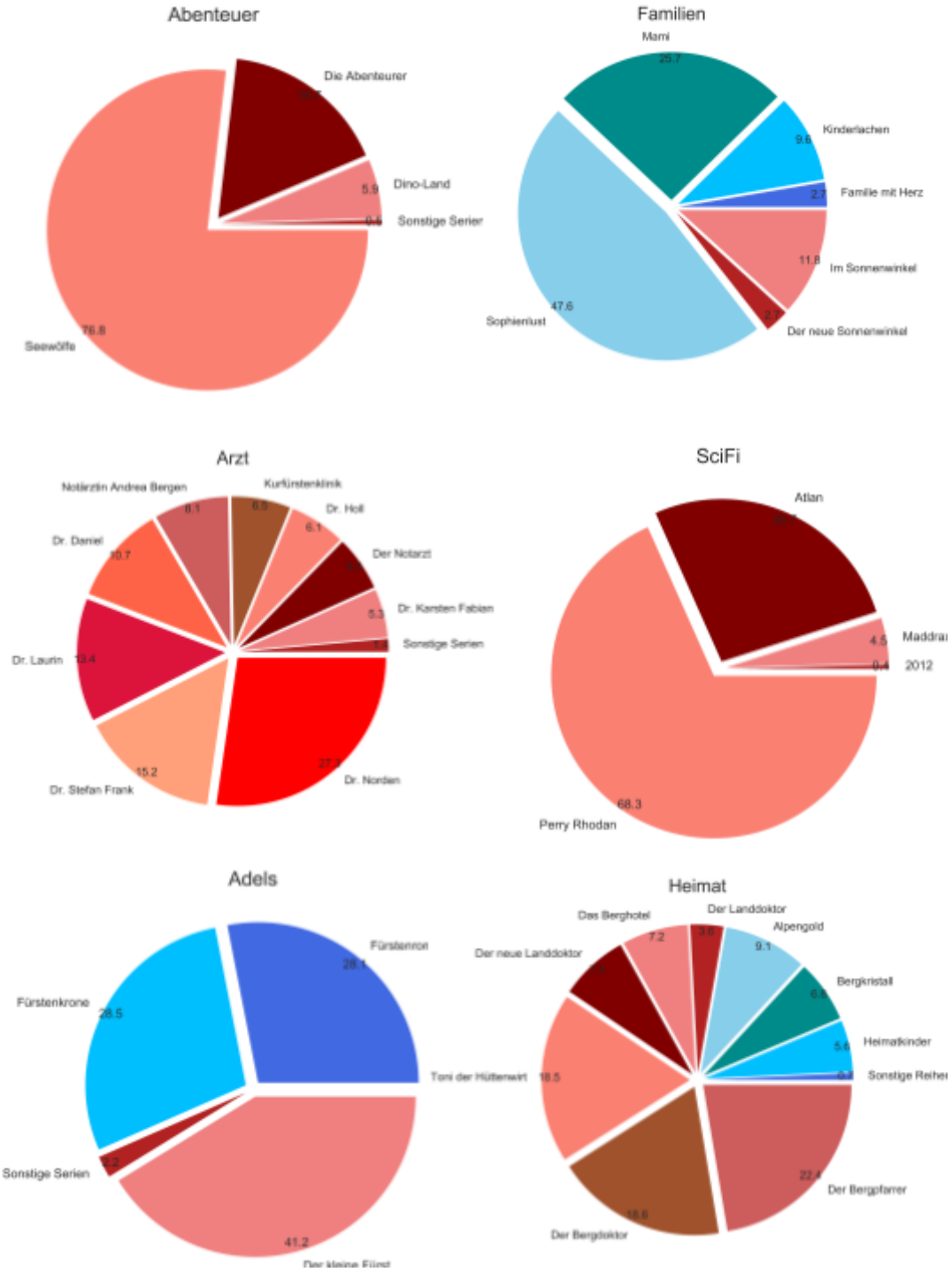

Abbildung 3.2: Serien und Reihen über Genres

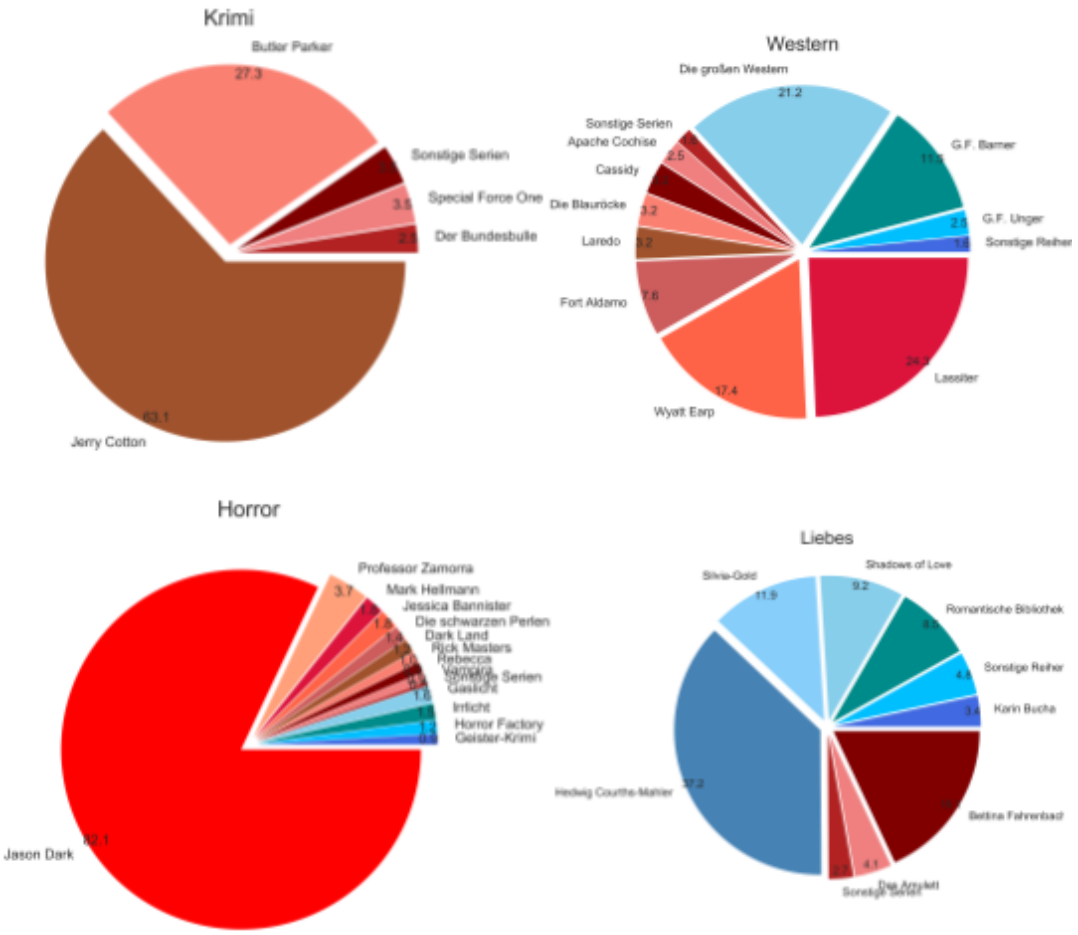

Abbildung 3.3: Serien und Reihen über Genres

Abbildung 3.3 zeigt die Anteile von Reihen und Serien für alle im Korpus enthaltenen Genres. Das Verhältnis ist starken Schwankungen unterlegen: So sind für Liebes- und Familienromane Reihen dominierend, während in SciFi-, Kriminal, Arzt- und Abenteuerromanen ausschließlich serielle Romane zu finden sind. Diese Verteilung birgt zwei Implikationen: Zum einen ist davon auszugehen, dass Serien in sich ähnlicher sind als Reihen und zum anderen, dass ein Genre, welches sich aus wenigen Reihen oder Serien zusammensetzt, homogener als breiter aufgestellte Genres ist. Zusammengefasst gilt aus dieser Perspektive ein Genre aus vielen Reihen und Serien, wobei das Verhältnis stark zugunsten der Reihen ausfällt, als schwer zu klassifizieren. Das Korpus beinhaltet neben

den klar zuzuordnenden Reihen und Serien auch Mischformen: *Cassidy* und *Laredo* laufen unter dem Schlagwort Erotik-Western, *Shadows of Love* als erotische Liebesgeschichten *Der Landdoktor*, *Der neue Landdoktor* und *Der Bergdoktor* in unterschiedlichen Auflagen mal als Heimat- mal als Arztromane; *Irrlicht*, *Gaslicht*, *Jessica Banister*, *Rebecca und Die schwarzen Perlen* werden als Romantic-Thriller beworben. Das Amulett kann als Mischung aus Serie und Reihe bezeichnet werden, da die Protagonistin in jedem Roman wechselt, die Geschichte allerdings durch das namensgebende Amulett zusammengehalten wird.

3.2 Word Embeddings

Um die Qualität der verschiedenen Word Embeddings zu prüfen und um zu ermitteln, wie nutzbringend eine Anpassung an die Domäne der Zieltexte ist, werden beide Experimente mit folgenden Embeddings durchgeführt: Im ersten Experiment werden deutsche Embeddings der Modelle word2vec[2], fastText[3] verwendet. Zusätzlich wird das fastText Modell durch Lernen auf dem gesamten Korpus angepasst. Dieser Vorgang steht nicht in Konflikt mit der späteren Textklassifikation, da die Texte satzweise und ohne Label übergeben werden. Um beide Modelle zu unterscheiden, wird das angepasste Modell als fastText1 und das unveränderte als fastText2 referenziert.
Neben den bereits genannten Modellen werden für das zweite Experiment ein deutsches ELMo[4] und ein mehrsprachiges Bert[5] Embedding eingesetzt. Das ELMo Embedding wird noch weiter differenziert:

[2]Müller 2015 URL: http://cloud.devmount.de/d2bc5672c523b086

[3]Bojanowski et al. 2017 URL: https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.de.300.bin.gz

[4]Fares et al. 2017 URL: http://vectors.nlpl.eu/repository/11/142.zip

[5]Devlin et al. 2018 URL: https://storage.googleapis.com/bert_models/2018_11_23/multi_cased_L-12_H-768_A-12.zip

- EMLo1: Gemittelte Vektoren über alle Schichten
- ELMo2: Die Vektoren aus der ersten LSTM-Schicht
- ELMo3: Die Vektoren aus der zweiten LSTM-Schicht
- ELMo4: Alle Schichten

3.3 Programmbibliotheken

Folgende Bibliotheken werden für die Durchführung der Experimente verwendet:

- scikit-learn (Pedregosa et al. 2011)
- umap (McInnes et al. 2018)
- pytorch (Paszke et al. 2017)
- tensorflow (Martın Abadi et al. 2015)
- keras (Chollet et al. 2015)
- ELMoForManyLangs (Che et al. 2018)
- flair (Akbik, Blythe und Vollgraf 2018)
- gensim (Řehůřek und Sojka 2010)
- fastText (Bojanowski et al. 2017)

Es wird ausschließlich die Promgramiersprache Python verwendet. Details sind dem Repository[6] zu entnehmen.

[6] https://github.com/LeKonArD/master_EmbForLiTtext

KAPITEL 4

EXPERIMENTE

4.1 Methodik

Das folgende Kapitel führt zwei neue Methoden ein. Im Experiment 1 wird eine Verschränkung zwischen Zeta und Word Embeddings operationalisiert und im Experiment 2 die Architektur eines neuronalen Netzes zur Klassifikation von Genre besprochen. Der entstandene Code ist in einem Repository[1] veröffentlicht.

4.1.1 Experiment 1: Zeta-Scores in Word Embeddings

Das erste Experiment folgt dem Gedanken, dass die Qualität der Genre-Klassifikation verbessert werden kann, wenn, statt Wortlisten mit diskreten Zeta-Werten, Wortfelder als Features verwendet werden, um ungesehene Texte zu klassifizieren. Ausgangspunkt für die Berechnung der Wortfelder ist die Wortliste, welche durch die Berechnung von Zeta für zwei Gruppen von Texten entsteht.
Für jedes Wort dieser Liste wird der entsprechende Vektor aus einem der

[1] https://github.com/LeKonArD/master_EmbForLiTtext

Word Embeddings ermittelt. Dieser Vektor wird zusammen mit seinem Zeta-Wert verwendet, um ein Cluster-Verfahren anzuwenden. Es ist nicht *apriori* abzuschätzen welche Anzahl an Clustern für welchen Use-Case sinnvoll ist. Daher entfallen solche Verfahren, welche eine feste Anzahl an Clustern als Parameter benötigen, darunter fällt bspw. K-Means Clustering. Folgende Algorithmen werden für das Experiment in Erwägung gezogen:

Affinity Propagation gehört zur Gruppe der message-passing Algorithmen, diese zeichnen sich durch eine Berechnung aus, bei der jeder Datenpunkt, analog zu einem fully-connected network, Informationen an alle anderen Datenpunkte sendet. Die erste Nachricht einer Iteration beinhaltet die Einschätzung (Responsibility) des sendenden Punktes s, ob dieser als Clusterzentrum für den Empfängerpunkt r im Vergleich mit allen anderen Punkten in Frage kommt. Diese Information wird durch Ähnlichkeiten zwischen den Punkten berechnet, als Maß für Ähnlichkeit dient hier die negative euklidische Distanz. Im zweiten Schritt erhält der Punkt s die Information, wie wahrscheinlich es ist, dass dieser durch Punkt r im Verhältnis zu allen anderen Punkten als Clusterzentrum repräsentiert werden kann (Availability). Für alle folgenden Iterationen wird zusätzlich zur euklidischen Distanz auch die Availability der letzten Iteration zur Berechnung der Responsibility verwendet. Der Algorithmus stoppt, sobald nach mehreren Iterationen keine Änderung der Availability mehr geschieht.
Statt Availability für die erste Iteration zu ignorieren und mit 0 zu initialisieren, kann entweder ein globaler Wert für alle Datenpunkte festgesetzt oder individuelle Werte für jeden Punkt übergeben werden, um bestimmten Punkten eine höhere Wahrscheinlichkeit als Clusterzentrum zuzuweisen (Frey und Dueck 2007).

Mean-Shift wählt eine zufällige Anzahl an Punkten aus einem ebenfalls zufällig positioniertem Fenster mit zuvor definierter Größe und berechnet deren Mittelpunkt. Dieser Vorgang wird so lange wiederholt, bis sich solche Regionen abzeichnen für die besonders oft ein Mittelpunkt berechnet wurde. Diese werden als Clusterzentren ausgegeben. (Fukunaga und Hostetler 1975)

Birch (Balanced Iterative Reducing and Clustering using Hierarchies) ist ein zweistufiges hierarchisches Clusterverfahren. Die erste Stufe ist die Erstellung eines CF-Trees[2]. Diese Modell besteht aus Nodes, welche Cluster repräsentieren und Leafs für Datenpunkte, welche einem Node zugeordnet werden. Der CF-Tree wird aufgebaut, indem iterativ jeder Datenpunkt einem Node zugeordnet wird. Entscheidend für diesem Prozess ist der Schwellenwert T. Ist die Distanz eines Punktes zum Clusterzentrum geringer als T kann dieser zum Cluster hinzugefügt werden. Ist sie zu groß wird ein neuer Node erstellt und die Baumstruktur verzweigt sich weiter.
Im zweiten Schritt wird ein klassisches Clustering z.b. K-Means verwendet, wobei die Nodes hier als Datenpunkte verwendet werden. Innerhalb dieses Experimtes wird jedoch auf den zweiten Schritt verzichtet und mit den Nodes, also Subclustern gearbeitet. (Zhang, Ramakrishnan und Livny 1996)

Die vorgestellten Clusterverfahren werden verwendet, um Clusterzentren für Zetawörter zu berechnen. Diese Clusterzentren ermöglichen nun eine multipolare Ähnlichkeitsanlyse, welche zumindest unter dem Gesichtspunkt von Wortverteilungen das in Kap. 1.2 erläuterte Konzept der Familienähnlichkeit formalisieren. Die Zugehörigkeit eines Textes zu

[2] Cluster-Featur Tree

einer Textgruppe wird folgendermaßen berechnet:
Jedem Wort wird ein Vektor v_k aus dem Embedding zugewiesen. Anschließend wird die Kosinusdistanz cos des Vektors v_k zu allen Clusterzentren c berechnet. Es werden die beiden geringsten Distanzen der beiden Gruppen c^{fokus} und c^{gegen} ermittelt und subtrahiert. Der so berechnete Ähnlichkeitswert wird mit der Worthäufigkeit h_k multipliziert. Der Vorgang wird für jedes Wort eines Textes durchgeführt, summiert und durch die Anzahl der Wörter des Textes n geteilt.

$$\frac{\sum_{k=0}^{n} h_k * (min(\sum_{z=0}^{n} cos(v_k, c_z^{fokus})) - min(\sum_{z=0}^{n} cos(v_k, c_z^{gegen})))}{n} \tag{4.1}$$

Durch die Eigenschaft der Kosinusdistanz können die Ähnlichkeitswerte nur zwischen -1 und 1 liegen. Diese ist genau, wie die Clustereringmethode, nicht zwingend das beste Maß für den Abstand der Wörter zu den Clusterzentren. Kosinus wird hier gewählt, da er sich in stilometrischen Verfahren (Burrows 2002a) durchgesetzt hat. Es müssen jedoch weitere Distanzmaße empirisch erprobt werden.

Dimensionsreduktion Es ist nicht davon auszugehen, dass die gesamte Vielzahl der Dimensionen eines Word Embedding für jede Klassifikationsaufgabe benötigt wird. Beispielsweise enthalten einige Dimensionen hauptsächlich grammatikalische Informationen, wie Wortart, Genus oder Numerus. Es erscheint unter dieser Sichtweise nicht sinnvoll diese gleichrangig mit Dimensionen, welche semantische Information tragen, in die Berechnung einfließen zu lassen. Um eine implizite Gewichtung einzuführen wird eine überwachte[3] Dimensionsreduktion durchgeführt. Diese

[3]Eine Überwachte Dimensionsreduktion kann auch als Metric Learning bezeichnet werden (L. Yang 2007)

erhält die Clusterzentren der Zeta-Embeddings, sowie deren Gruppenzugehörigkeit und reduziert die Vektoren in einen geringer dimensionierten Raum. Da es sich um ein überwachtes Verfahren handelt, werden zusätzlich zwei Zielvorgaben mitberücksichtigt: Clusterzentren einer Gruppe sollen möglichst dicht zusammen liegen und der Abstand der Gruppen zueinander soll möglichst groß sein. Die Wort-Vektoren der Testgruppe werden durch diese optimierte Transformation in den 2-dimensionalen Raum abgebildet und anschließend für die Klassifikation genutzt.

Unabhängig davon, ob eine Dimensionsreduktion durchgeführt wurde, wird mithilfe einer Support-Vector-Machine (SVM) ausgewertet, wobei die Aufgabe hier trivial ist, da lediglich ein Feature (der Ähnlichkeitwert) übergeben wird. Trainiert wird auf den Ähnlichkeitswerten der Texte, welche bereits für die Berechnung der Zeta-Werte verwendet wurden. Getestet wird auf zuvor ungesehenen Daten. (siehe Abb. 3.2)

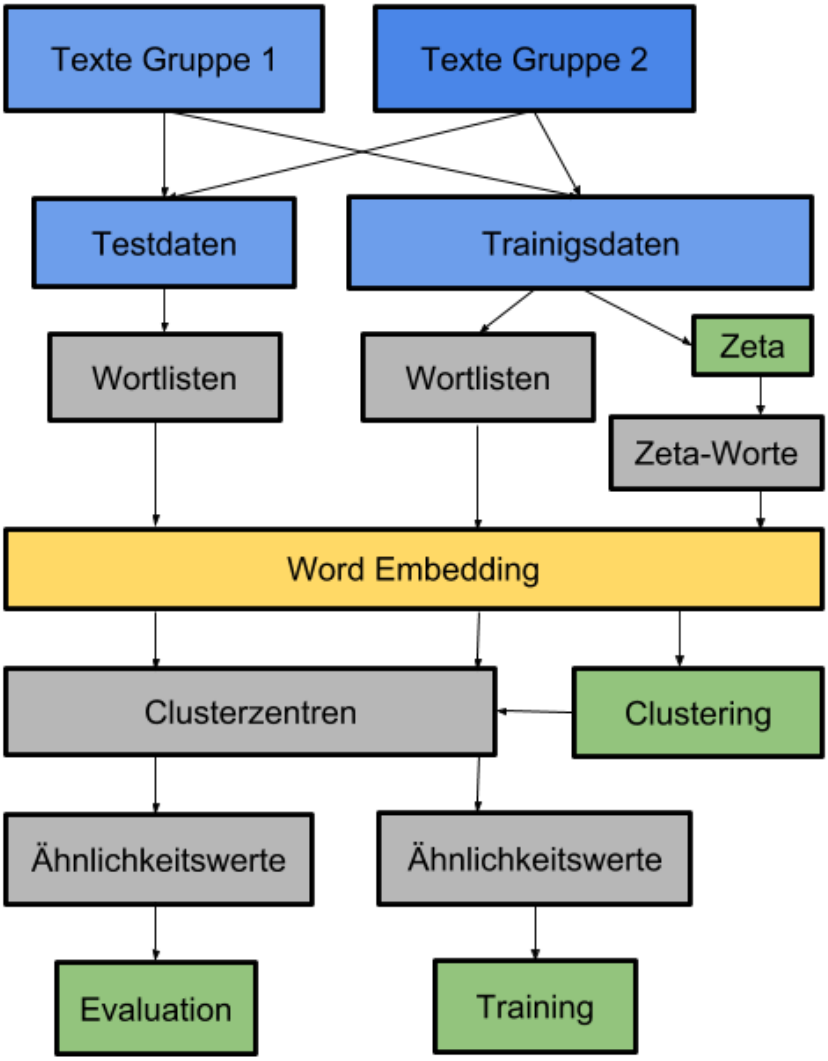

Abbildung 4.1: Schematischer Aufbau des ersten Experiments

4.1.1.1 Vorstudie

Die in Kapitel 1.2 vorgestellte Methode der Kombination von Word Embeddings und Zeta zur Klassifikation von Texten ist zwar konzeptionell definiert, allerdings sind einige Parameter unbestimmt und müssen empirisch untersucht werden. Um nicht jede Kombination von Parametern für jeden Einzelstudie berechnen und evaluieren zu müssen, wird eine Vorstudie zur Bestimmung der besten Parameter angesetzt. Hierfür wird die basale Aufgabe der Unterscheidung zwischen zwei Gruppen von

Genre-Texten als Task definiert. Es werden aber nicht alle 56 möglichen Kombinationen getestet, stattdessen beschränkt sich die Auswahl auf vermeintlich leichte Szenarien Kriminal- vs. Heimatroman, Scifi vs. Liebesroman und Horror vs. Familienroman sowie vermeindlich schwerer zu unterscheidende Paarungen Arzt vs. Adelsroman, Kriminal vs. Westernroman und Heimat vs. Familienroman.
Gesucht wird die beste Kombination aus folgenden Parametern:

- Clusterverfahren: MeanShift, Affinity Propagation oder Birch
- Metric Learning: Ja/Nein
- Distanzmaß: Kosinus-, Manhatten- oder Euklidische Distanz
- Berechnung von Ähnlichkeit durch mittlere Entfernung zu allen Clusterzentren oder minimale Distanz zu Clusterzentren beider Gruppen

Statistischen Schwankungen wird durch den Einsatz einer 20-fold cross-validation[4] vorgebeugt. Wobei hier jeweils nur auf einem fold trainiert und die übrigen 19 zur Evaluation genutzt werden. Eigennamen werden aus jedem Datensatz entfernt. Die Segmentgröße für Zeta wäre ein weiterer Parameter, wird aber hier nicht untersucht, stattdessen wird den Ergebnissen aus Christof Schöch et al. 2018 folgend eine feste Größe von 10.000 Token verwendet.

4.1.1.2 Test Cases: Genres

Im Folgenden werden weitere Test Cases zur Untersuchung des Verhaltens der Methode aus Kap 1.2 vorgestellt. Wie auch in der Vorstudie wird

[4] Bei einer cross-validation wird der gesamte Datensatz in n gleichgroße Segmente (folds) eingeteilt, um zu verhindern, dass der Algorithmus zu stark auf eine Traings-/Testgruppe angepasst wird und ein allgemeineres Modell erzeugt wird.

mittels 20-fold cross validation, einer Segmentgröße von 10.000 Token und unter Ausschluß von Eigennamen getestet.

Case I: Genres, heterogene Gegengruppe

Ziel des Experimentes ist es, eine Basis für die Klassifikation jedes Genres gegenüber dem gesamten Korpus zu finden. Während eine Gruppe lediglich Texte eines Genres enthält, setzt sich die Gegengruppe aus Texten aller übrigen Genres zusammen. Die innere Verteilung der Gegengruppe wird so gewählt, dass jedes Genre zu möglichst gleichen Anteilen vertreten ist. Theoretisch ist diese Aufgabe schwerer als das Setup der Vorstudie, denn während bei einer Unterscheidung zwischen zwei Genres sowohl die distinktiven Wörter der Fokus als auch die der Gegengruppe genutzt werden können, bestehen die distinktiven Wörter einer heterogenen Gegengruppe lediglich aus negativ distinktiven Wörtern der Fokusgruppe. Daher werden hier schwächere Ergebnisse erwartet. Getestet werden alle Genre des Korpus.

Case II: Genres über Serien und Reihen, heterogene Gegengruppe Dieser Task testet die Fähigkeit zur Generalisierung der Methoden. Der Modus ist äquivalent zu Test Case II, mit dem Unterschied, dass in den Trainingsdaten der Zielgruppe nur eine der im Genre enthaltenen Serien oder Reihen enthalten ist. Die Testgruppe besteht damit aus allen übrigen Reihen und Serien des Genres. Aufgrund der Verteilung von Reihen und Serien in den Genres (siehe: 3.3) ist dies nicht für jedes Genre sinnvoll durchzuführen. Verwendet werden Western mit Die großen Western, Heimat mit Toni der Hüttenwirt und Familienromane mit Sophienlust.

Case III: Serien

Hier wird die Fähigkeit Serien eines Genres zu unterscheiden getes-

tet. Verwendet werden Wyatt Earp vs. Lassiter, Atlan vs. Maddrax und Seewölfe vs. Die Abenteurer.

Case IV: Reihen

Analog zu Case III wird die Möglichkeit Reihen zu unterscheiden überprüft. Getestet wird auf Mami vs. Kinderlachen, Alpengold vs. Bergkristall, Dr. Norden vs. Dr. Fabian und Die großen Western vs. G.F. Barner. Es ist zu erwarten, dass Serien leichter zu unterscheiden sind als Reihen, da diese mehr wiederkehrende Elemente enthalten und von einer kleineren Gruppe von Autoren geschrieben werden.

Case V: Zielgruppe

Verlagsseitig wird für jedes Genre klar festgelegt, welche Zielgruppe angesprochen werden soll. Wichtigstes Unterscheidungsmerkmal ist hier das Geschlecht der Leser. Diese Unterscheidung soll mithilfe von Zeta und Varianten nachvollzogen werden. Die Gruppen teilen sich in Frauenromane: Liebes-, Arzt-, Heimat-, Adels-, und Familienromane und Männerromane: Western-, Kriminal-, SciFi- und Abenteuerromane. Das Genre Horror wird nicht behandelt, da sich dort Romane für beide Zielgruppen finden. Aus jedem Genre werden je 100 Romane gezogen.

4.1.1.3 Vorstudie II

Um zu prüfen, ob die Kombination von Word Embeddings und Zeta tatsächlich stärker generalisieren und damit das Autorensignal unterdrücken wird die Klassifikation von Autorschaft getestet. Die Autorenschaftsattribution innerhalb Serien stellt eine große Herausforderung dar, da sowohl das Genrevokabular als auch das der Serie annähernd in allen Texten extrem ähnlich sein könnte. Da sich hier die Task stark verändert, wird erneut eine Vorstudie zur Parametersuche durchgeführt.

In diesem Experiment werden ausschließlich Autoren betrachtet, welche in der selben Serie veröffentlichen. Da eine Mindestanzahl an Texten für beide Autoren bereitgestellt sein muss, beschränkt sich die Auswahl hier auf:

- Fort Aldamo: Murphy, Bill vs. Callahan, Frank (je 23 Romane)
- Seewölfe: Palmer, Roy vs. McMason, Fred (je 35 Romane)
- Sophienlust: Clausen, Bettina vs. Korten, Aliza (je 29 Romane)

Die Testbedingungen werden aus der ersten Vorstudie übernommen, jedoch wird eine 10-fold cross validation verwendet, da sich die Textmenge drastisch verkleinert.

4.1.1.4 Test cases: Autorschaft

Die folgenden Test Cases überprüfen die Leistungsfähigkeit der Methoden in unterschiedlichen Settings. Hier wird 5-fold cross validation angewandt.

Case VI: Autorschaft in Reihen, heterogene Gegengruppe

Für dieses Testszenario wird die Schwierigkeit noch einmal angehoben, indem die Gegengruppe aus Texten verschiedener Autoren zusammengesetzt wird und somit lediglich die Fokusgruppe relevante Informationen trägt (siehe Test Case I). Verwendet werden:

- Fürstenkrone: Marion Alexi (je 12 Romane)
- Die großen Western: Joe Juhnke (je 14 Romane)
- Mami: Gisela Reutling (je 10 Romane)

Case VII: Autorschaft in Serien, heterogene Gegengruppe

Der letzte Test Case stellt die herausforderndste Aufgabe dar. Die Gegengruppe besteht aus mehreren Autoren, die wiederum Texte für die selbe Serie schreiben aus der die des Autors der Fokusgruppe stammen. Getestet werden folgende Autoren:

- Sophienlust: Patricia Vandenberg (je 12 Romane)
- Seewölfe: Roy Palmer (je 10 Romane)
- Professor Zamorra: Adrian Doyle (je 10 Romane)

4.1.2 Experiment 2: Attention für distinktive Wörter

Das zweite Experiment löst sich vollständig von der Methode Zeta und der Gegenüberstellung von Häufigkeitsverteilungen in Textgruppen. Stattdessen wird das Ziel des Auffindens distinktiver Wörter durch ein neuronales Netz operationalisiert. In das Netzwerk werden Segmente aus Romanen zu je 200 Wörtern gegeben. Diese werden mithilfe vor-trainierter Word Embeddings vektorisiert.

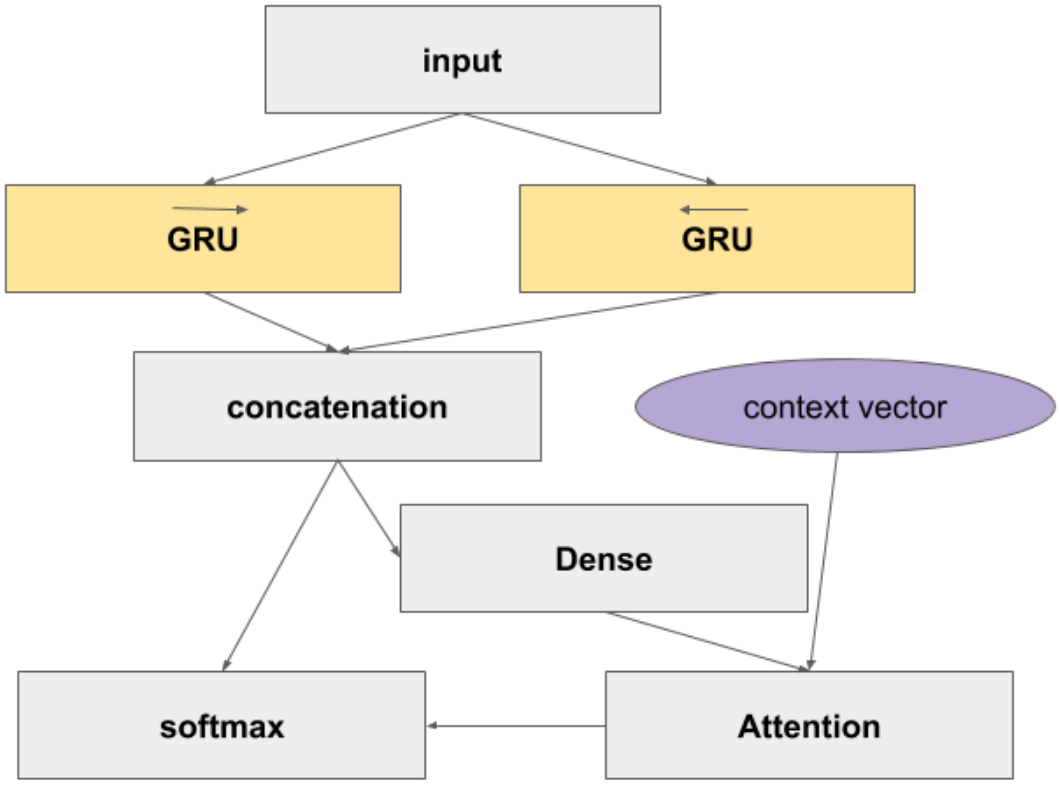

Abbildung 4.2: Architektur des neuronalen Netzes

Die Netzwerkarchitektur (Abb. 4.2) ist angelehnt an das von Z. Yang et al. 2016 eingeführte Hierachical Attention Network. Der Input besteht aus einer Sequenz s_i von Wörtern $w_i t$, welche durch einen Embedding Emb vektorisiert werden. Diese Sequenzen werden in ein bidirektionales

Gated Recurrent Unit[5] GRU mit einer hidden size von 256 units gegeben, welches als Segement Encoder fungiert. Anschließend werden die hidden states h_i beider Leserichtungen zusammengehängt, um die kodierte Sequenz h_{seg} zu erhalten.

$$s_i = Emb(w_{it}) \tag{4.2}$$

$$\overrightarrow{h_i} = \overrightarrow{GRU}(s_i), \tag{4.3}$$

$$\overleftarrow{h_i} = \overleftarrow{GRU}(s_i) \tag{4.4}$$

$$h_{seg} = [\overrightarrow{h_i}; \overleftarrow{h_i}], i = 0, 1, ..n \tag{4.5}$$

An dieser Stelle wäre bereits eine Klassifikation von Textgruppen rein basierend auf h_{seg} möglich. Statt es dabei zu belassen, wird ein Attention-Mechanismus (Chung et al. 2014) in das Netzwerk eingesetzt. Dieser verstärkt zum einen die Fähigkeit des Netzes innerhalb einer Sequenz weit auseinander liegende Kontextbeziehungen zu erkennen und zum anderen eine Rückverfolgung der für die Klassifikation entscheidenden Wörter. Attention wird realisiert, indem h_{seg} durch ein einschichtiges feedforward Netzwerk gegeben wird. Dieses erzeugt eine wiederum abstraktere Repräsentation u_{seg} der hidden states. Der Attentionvektor a_{seg} wird berechnet aus der Ähnlichkeit zwischen u_seg und dem Kontextvektor q_{seg}. Dieser Kontextvektor wird zunächst randomisert initalisiert und mit Gewichten ausgestattet. Diese Gewichte erlauben es, den Vektor mit jeder Iteration anzupassen. Der Attentionvektor a_{seg} wird verwendet, um die hidden states der GRUs zu gewichten. Das Resultat s_{final} wird mittels einer Softmax-Funktion in eine Wahrscheinlichkeitsverteilung über die

[5] GRUs sind rekurrente Netzwerke, ähnlich den LSTMs (siehe Kap. 2.1.3), mit dem Unterschied, dass sie nicht über drei, sondern nur 2 Gates, ein reset und ein update Gate verfügen (Cho et al. 2014)

Anzahl der Klassen umgerechnet.

$$u_{seg} = relu(W_a h_{seg} + b_a) \tag{4.6}$$

$$a_{seg} = \frac{exp(u_{seg}, q_{seg})}{\sum(exp(u_{seg}, q_{seg}))} \tag{4.7}$$

$$s_{final} = \sum a_{seg} h_{seg} \tag{4.8}$$

$$p = softmax(W_b s_{final} + b_b) \tag{4.9}$$

Um annäherungsweise nachvollziehen zu können auf welcher Grundlage das Netzwerk ein Segment einer Gattung zugeordnet, wird es zunächst möglichst austrainiert. Anschließend werden die Vektoren eines Segments mit dem Attentionvektor a_{seg} des Netzes verrechnet, um zu ermitteln welche Wörter für eine Klassifikation entscheidend sind. Da Wörter sich innerhalb eines Segments wiederholen können, wird zunächst die durschnittliche Attention für jedes Wort berechnet. Danach kann die relative Attention ermittelt werden. Dieser Vorgang wird für alle Segemente des Testdatensatzes wiederholt, so dass für jede Klasse eine globale relative Attention auf Wortebene entsteht.
Das Netzwerk wird auf die Unterscheidung der vier Klassen Liebes-, Kriminal-, Science Fiction- und Horrorromanen trainiert. Der Trainingsdatensatz besteht dabei aus 24.000 Segmenten a 200 Token, wobei auf jede Klasse 6000 Segmente entfallen. Die Zeichensetzung wird normalisiert, sodass keine Klassifikation aufgrund der Verwendung besonderer Anführungszeichen oder dem Einsatz von Auslassungspunkten[6] entstehen kann. Namen werden durch das Maskierungstoken <named_entity> ersetzt. Das Testset besteht aus 8000 Segmenten, wovon auf jede Klasse 2000 Beispiele entfallen. Auf das Anlegen einer dritten Gruppe von Trainingsdaten, um eine Anpassung von Hyperparametern auf die Testdaten zu vermeiden, wird verzichtet, da dieses Experiment einen explorativen

[6] Einsatz von ... als Stilmittel

Hintergrund hat und keine Suche nach Hyperparamtern durchgeführt wird. Um zu Prüfen, welchen Einfluss verschiedene Typen von Embeddings auf die Klassifikation haben, wird das Netz mit word2vec, fastText, auf dem gesamtem Datensatz nachtrainiertem fastText, ELMo und Bert Embeddings für jeweils 5 Epochen trainiert. Da die Aufgabe grundsätzlich nicht schwer zu lösen ist und bei Anpassung von Hyperparametern auch für unterkomplexe Embeddings gute Ergebnisse zu erzielen sind, ist für eine Evaluation der Embeddings nicht nur das Ergebnis der Klassifikation auf den Testdaten, sondern auch die Geschwindigkeit mit der das Netz konvergiert[7] entscheidend.

4.2 Ergebnisse

4.2.1 Experiment 1

4.2.1.1 Vorstudie I

Tabelle 4.1 zeigt die Ergebnisse der Parameterstudie für die Textgruppen Arzt und Adelsromane. Erwartungsgemäß fällt die Baseline mit einer Accuracy von .73 eher schwach aus, da die Genres beide Subkategorien des Liebesromans sind. Setzt man die Baseline in Verhältnis zu den Ergebnissen der Methode aus Zeta und Word Embeddings zeigt sich, dass diese zuverlässig erreicht und sogar übertroffen wird. Eine Ausnahme bildet die Kombination aus Metric Learning und MeanShift, welche sehr viel schlechtere Resultate aufweist.

[7] Konvergenz bedeutet in diesem Zusammenhang die iterative Annäherung des Netzes an ein optimales Klassifikationsergebnis auf den Trainingsdaten

Tabelle 4.1: Ergebnisse der Klassifikation von Arzt- vs. Adelsroman gemessen in Accuracy für das Embedding fastText_1

Baseline		Burrow's Zeta: .71 ‖sd2$Zeta$: .73					
Distanz		euc	euc	cos	cos	man	man
min/mean		min	mean	min	mean	min	mean
CA	MeL						
MS	yes	.26	.27	.29	.29	.21	.21
MS	no	.74	.72	.74	.72	.71	.7
AP	yes	.78	.77	.76	.76	.73	.77
AP	no	.75	.73	.74	.73	.74	.73
Birch	yes	.84	.75	.76	.75	.8	.77
Birch	no	.76	.73	.76	.73	.73	.73

Die Ergebnisse für die Genres Familien und Heimatroman (Tabelle 4.2) bestätigen mit einer schwachen Baseline von .83 abermals die Nähe der Subkategorien des Liebesroman untereinander. Auch hier ist die Kombination aus MeanShift und Metric Learning sehr viel schlechter als der Durchschnitt. Die Ergebnisse der neuen Methode können die Baseline in keinem der getesteten Workflows schlagen.

Tabelle 4.2: Ergebnisse der Klassifikation von Familien vs. Heimatroman gemessen in Accuracy für das Embedding fastText_1

Baseline		Burrow's Zeta: .82 ‖sd2$-Zeta$: .83					
Distanz		euc	euc	cos	cos	man	man
min/mean		min	mean	min	mean	min	mean
CA	MeL						
MS	yes	.09	.09	.09	.09	.09	.09
MS	no	.8	.69	.8	.68	.7	.7
AP	yes	.73	.56	.72	.55	.43	.64
AP	no	.72	.72	.75	.74	.59	.64
Birch	yes	.81	.76	.79	.76	.54	.65
Birch	no	.57	.57	.56	.56	.63	.62

Die Unterscheidung zwischen Kriminal- und Heimatromanen (Tabelle

4.3) erscheint gemessen an der Baseline mit 92% Accuracy wesentlich leichter, als die der beiden vorherigen Paarungen. Zusätzlich wird diese von der neuen Methode zuverlässig geschlagen. Vier der Parameterkombinationen erzielen sogar ein perfektes Ergebnis.

Tabelle 4.3: Ergebnisse der Klassifikation von Kriminal- vs. Heimatroman gemessen in Accuracy für das Embedding fastText_1

Baseline		Burrow's Zeta: .92 ‖sd2−$Zeta$: .92					
Distanz		euc	euc	cos	cos	man	man
min/mean		min	mean	min	mean	min	mean
CA	MeL						
MS	yes	.25	.26	.24	.24	.2	.2
MS	no	.96	.92	.94	.89	.98	.99
AP	yes	.99	.99	.99	.99	1.0	1.0
AP	no	.99	.95	.99	.95	.99	.99
Birch	yes	.99	1.0	.99	1.0	.88	.92
Birch	no	.99	.98	.99	.98	.99	1.0

Die letzte Klassifikation der Vorstudie, die Paarung aus Science Fiction- und Liebesromanen, kann als zumindest aus Perspektive eines menschlichen Lesers triviale Aufgabe betrachtet werden. Gegen diese Intuition liegt die Accuracy der Baseline mit 90% Accuracy leicht unter der des zuvor gezeigten Experimentes. Auch hier erzielt die neue Methode bis auf einige Ausnahmen bessere Resultate. Überraschend ist, dass die ansonsten durchweg sehr schwache Kombination aus MeanShift und Metric Learning hier ebenfalls teils gute Ergebnisse erreicht.

Tabelle 4.4: Ergebnisse der Klassifikation von SciFi vs. Liebesroman gemessen in Accuracy für das Embedding fastText_1

Baseline		Burrow's Zeta: .90 ‖sd2−$Zeta$: .89					
Distanz		euc	euc	cos	cos	man	man
min/mean		min	mean	min	mean	min	mean
CA	MeL						
MS	yes	.85	.92	.48	.47	.5	.6
MS	no	.98	.43	.98	.42	.6	.64
AP	yes	.97	.97	.97	.98	.98	.98
AP	no	.98	.79	.98	.79	.89	.98
Birch	yes	.97	.97	.97	.97	.98	.98
Birch	no	.96	.98	.96	.98	.95	.99

Für die Auswertung der Ergebnisse stellt sich die Frage, wie aus dieser Vielzahl an möglichen Kombinationen von Parametern die beste Einstellung gefunden werden kann. Ein einfaches Vergleichen von durchschnittlichen Werten für beispielsweise die verwendeten Distanzmaße ist nicht sinnvoll, da sich alle Parameter auch gegenseitig beeinflussen können. So würde, um den Punkt zu verdeutlichen, Metric Learning sofort ausgeschlossen werden, da die Kombintation mit MeanShift den Durchschnittswert nach unten ziehen würde. Stattdessen wird ein Regression Tree Lewis 2000 verwendet, um die besten Parameter zu finden. Dieser bietet den Vorteil, dass mit jeder Regresssion von der Wurzel angefangen, der Parameter mit der höchsten Einflussgröße gesucht und eliminiert wird. Elimination bedeutet in diesem Fall die Entscheidung für die Abzweigung mit den höheren Klassifikationsergebnissen und den Auschluss aller Ergebnisse der anderen Abzweigung für die nächste Regression. Um die statistische Auswertung robuster zu gestalten werden die Regressionen nicht auf den in den Tabellen gelisteten Durchschnittswerten, sondern basierend auf allen Einzelmessungen der 20-fold cross validation durchgeführt. Die so ermittelten Parameter ergeben die Verwendung von

Birch als Clusterverfahren anschließende Verwendung von Metric Learning und das Berechnen der Ähnlichkeit aufgrund der jeweils geringsten euklidischen Distanz zu einem Cluster der beiden Gruppen. Um zu bestimmten, ob die Anwendung der neuen Methode statistisch signifikant bessere Ergebnisse erzielt, wird ein t-Test für abhängige[8] Stichproben berechnet. Die hier vorgeschlagene Methode unter der Verwendung der nachträglich trainierten fastText-Embeddings, ist mit $p > .05$ signifikant besser als die Verwendung von Zeta ohne Word Embedings.

4.2.1.2 Test Cases

Im Folgenden werden die Ergebnisse für die in Kapitel X definierten Testcases aufgeführt. Als Baseline wird jeweils die bessere Klassifikation aus Burrows Zeta oder sd2-Zeta herangezogen. Die Word Embeddings werden verkürzt referenziert als fasttext1 für das auf dem gesamten Korpus nachtrainierte Embedding, fasttext2 für die von Facebook veröffentlichten deutschen Embeddings und w2v für word2vec Embeddings trainiert auf der deutschen Wikipedia.

Test Case I

Tabelle 4.5 zeigt die Ergebnisse der Klassifikation eines Genres gegenüber einer Gruppe bestehend aus Texten der übrigen Genres. Unter Verwendung von fastText1 kann die neue Methode die Baseline in 7 von 9 Fällen übertreffen. Die Differenz ist aber bis auf das Western-Genre mit 2-3% sehr gering. Für die Genres Abenteuer und SciFi liegt die Baseline mit einem Prozentpunkt über dem Ergebnis von fastText1. Weniger knapp fällt die Differenz zwischen fastText1 und fastText2 aus. Die Ergebnisse bescheinigen, dass ein Nachtrainieren der Embeddings einen großen Einfluss auf deren Qualität hat. Dieser ist sogar höher als der Unterschied der

[8] Abhängigkeit besteht hier, da jede Messung vom zugrundeliegenden Datensatz beeinflusst wird

durch verschiedene Methoden zur Erzeugung von Embeddings entsteht, was daran zu erkennen ist, dass w2v und fasttext2 sehr dicht beieinander liegen.

Tabelle 4.5: Ergebnisse Test Case I in Accuracy

	Baseline	fastText1	fastText2	w2v
Abenteuer	.88	.87	.77	.73
Adels	.81	.83	.75	.80
Arzt	.89	.92	.86	.91
Heimat	.85	.87	.83	.83
Horror	.88	.90	.83	.88
Krimi	.86	.89	.84	.86
Liebes	.78	.82	.78	.78
SciFi	.95	.94	.93	.90
Western	.90	.96	.95	.92

Test Case II

Tabelle 4.3 zeigt die Ergebnisse der Klassifikation von Genres gegen eine heterogenen Gegengruppe mit dem Zusatz, dass nur auf einer Reihe oder Serie trainiert wird. Für Familien und Westernromane liegt fastText1 deutlich über der Baseline, was tatsächlich als Inidkator für eine ausgeprägtere Abstraktionsfähigkeit gesehen werden kann. Für das Genre des Heimatromans liegen beide Methoden gleichauf, wobei eine Baseline von 99% wenig Spielraum für Verbesserungen lässt. Die Resultate für fastText2 und w2v weisen erneut keine großen Differenzen auf.

Tabelle 4.6: Ergebnisse Test Case II in Accuracy

	Baseline	fastText1	fastText2	w2v
Familien via Sophienlust	.80	.94	.91	.93
Heimat via Toni d. Hüttenwirt	.99	.99	.97	.98
Western via Die großen Western	.88	.98	.88	.90

Test Case III

Die Ergebnisse für Test Case III zur Unterscheidung von Serien lassen wenig Aussagen über die Methoden zu, da die Klassifikationen nahezu identisch sind. Die Baseline wird von fastText1 zweimal eingestellt und einmal knapp unterboten.

Tabelle 4.7: Ergebnisse Test Case III in Accuracy

	Baseline	fastText1	fastText2	w2v
Maddrax/Atlan	.99	.98	.77	.93
Seewölfe/Abenteurer	1.0	1.0	1.0	1.0
Wyatt Earp/Lassiter	.99	.99	.98	.98

Test Case IV

Tabelle 4.8 zeigt die Ergebnisse der Klassifikation von Reihen innerhalb einer Gattung mit homogener Gegengruppe. Aus den Zahlen lassen sich nur schwer Qualitätsunterschiede zwischen Baseline in fastText1 ableiten, da die Klassifikationsqualität nahe beieinander liegt. Auffällig ist jedoch die allgemein gute Unterscheidbarkeit der Arztromane, sowie die Überlegenheit der Word Embedding Methode auf bei Heimatromanen.

Tabelle 4.8: Ergebnisse Test Case IV in Accuracy

	Baseline	fastText1	fastText2	w2v
Alpengold/Bergkristall	.71	.98	.81	.58
Mami/Kinderlachen	.71	.71	.63	.65
Die großen Western/ G.F. Barner	.74	.74	.71	.70
Dr. Norden/ Dr. Fabian	.90	.85	.72	.69

Test Case V

In Test Case V werden die Romane nach Geschlecht der Zielgruppe sortiert. fastText1 übertrift die Baseline hier mit 2 Prozentpunkten.

Tabelle 4.9: Ergebnisse Test Case V in Accuracy

Baseline	fastText1	fastText2	w2v
.95	.97	.94	.94

4.2.1.3 Vorstudie II

In der zweiten Vorstudie sollen die besten Parameter für die Unterscheidung von Autoren gefunden werden, da nicht zwingend davon auszugehen ist, dass die Ergebnisse der ersten Vorstudie zur Gattungsklassifikation übertragbar sind. Tabelle 4.10 zeigt die Ergebnisse der Klassifikation der Autorinnen Bettina Clausen und Aliza Korten aus der Serie Sophienlust der Gattung Familienroman. Zunächst ist hervorzuheben, dass diese Aufgabe trotz ihrer Schwierigkeit, denn es ist davon auszugehen, dass die Texte einer Serie sich in vielen Punkten stark ähneln, von beiden Baselines sehr gut gelöst wird. Die Verwendung von Word Embeddings und Clustering zur Generierung abstrakterer Klassen ist für diese Aufgabenstellung gemessen an den Resultaten eher hinderlich.

Tabelle 4.10: Autorschaftsklassifikation Bettina Clausen vs. Aliza Korten in Accuracy

Baseline		Burrow's Zeta: .97 ‖sd2−$Zeta$: .97					
Distanz		euc	euc	cos	cos	man	man
min/mean		min	mean	min	mean	min	mean
CA	MeL						
MS	yes	.58	.81	.71	.69	.56	.48
MS	no	.4	.4	.38	.39	.82	.82
AP	yes	.71	.53	.72	.56	.84	.84
AP	no	.84	.51	.85	.47	.91	.9
Birch	yes	.92	.59	.9	.59	.54	.84
Birch	no	.64	.64	.71	.71	.8	.75

Tabelle 4.11: Auorschaftsklassifikation Palmer vs. McMason in Accuracy

Baseline		Burrow's Zeta: .96 $\|$sd2$-Zeta$: .96					
Distanz		euc	euc	cos	cos	man	man
min/mean		min	mean	min	mean	min	mean
CA	MeL						
MS	yes	.38	.34	.39	.28	.43	.17
MS	no	.9	.84	.9	.84	.83	.86
AP	yes	.83	.83	.84	.84	.89	.9
AP	no	.95	.71	.91	.72	.84	.87
Birch	yes	.9	.89	.85	.89	.78	.91
Birch	no	.93	.91	.94	.92	.55	.89

Die Ergebnisse für die Serien Seewölfe (Tab. 4.11) und Fort Aldamo (Tab. 4.12) aus den Gattungen Abenteuer und Westernroman replizieren die der ersten Autoren noch einmal und belegen, dass die neue Methode nicht zur Autorenschaftsklassifikation geeignet ist. Um dies statistisch zu untermauern wird ein Regression Tree (siehe Vorstudie 1) verwendet, um die besten Parameter zu finden. Diese sind: Birch Clustering, anschließendes Metric Learning und die kleinste euklidische Distanz eines Wortes zu Clusternzentren als Ähnlichkeitsmaß. Ein t-test ($p > .05$) zeigt, dass auch diese Kombination signifikant schlechter als die Baseline ist.

Tabelle 4.12: Autorschaftsklassifikation Bill Murphy vs. Frank Callahan in Accuracy

Baseline		Burrow's Zeta: 1.0 ‖sd2−$Zeta$: 1.0					
Distanz		euc	euc	cos	cos	man	man
min/mean		min	mean	min	mean	min	mean
CA	MeL						
MS	yes	.54	.54	.56	.57	.21	.21
MS	no	.94	.57	.94	.58	.46	.58
AP	yes	.89	.89	.89	.88	.93	.94
AP	no	.99	.93	.98	.91	.6	.92
Birch	yes	.97	.98	.97	.97	.97	.97
Birch	no	1.0	.92	1.0	.93	.62	.96

Test Case VI

Tabelle 4.13 listet die Ergebnisse der Autorschaftsklassifikation mit heterogener Gegengruppe in den Reihen Fürstenkrone, Die großen Western und Mami unter Verwendung der in Vorstudie II ermittelten Parameter und der drei zu testenden Word Embeddings. Für Marion Alexi und Joe Juhnke setzt sich der Trend der Vorstudie durch; Die Baseline ist hier stärker als die neue Methode. Lediglich die Romane von Gisela Reutling werden mit fastText1 besser klassifiziert. Dies kann als Hinweis gesehen werden, dass die Texte von Gisela Reutling sich innerhalb der Mami-Serie noch über mehr abgrenzen lassen als bloße Autorschaft.

Tabelle 4.13: Ergebnisse Test Case VI in Accuracy

	Baseline	fastText1	fastText2	w2v
Marion Alexi	.98	.89	.82	.81
Joe Juhnke	.68	.47	.46	.39
Gisela Reutling	.7	.77	.61	.63

Test Case VII

Die Ergebnisse des Test Case VII (Tab. 4.14) zeichnen ein leicht anderes

Bild als die der vorherigen Test Case und der Vorstudie. Hier lohnt sich die Einbindnung von Word Embeddings in 2 von 3 Fällen. Ebenfalls überraschend ist die Einsicht, dass die Klassifikation von Autoren in Serien nicht schwerer ist als die in Reihen (Test Case VI).

Tabelle 4.14: Ergebnisse Test Case VII in Accuracy

	Baseline	fastText1	fastText2	w2v
Patricia Vandenberg	.71	.88	.71	.61
Roy Palmer	.90	.79	.63	.70
Adrian Doyle	.96	.98	.97	.92

4.2.2 Experiment 2

In Experiment 2 wird ein neuronales Netz verwendet, um Segmente von 200 Wörtern ihrem Genre zuzuordnen. Die Wörter werden dabei mit Word Embeddings initialisiert. Tabelle 4.15 zeigt die Qualität der Klassifikation in Abhängigkeit zum verwendeten Embedding.

Tabelle 4.15: Ergebnisse des zweiten Experiments (f1 makro auf Testdatensatz)

w2v	fastText1	fastText2	ELMo1	ELMo2	ELMo3	ELMo4	Bert
.81	.96	.91	.93	.91	.92	.90	.93

Die beste Klassifikation mit einem f1 score von .96 wird hier durch das Emebdding fasttext1 erzielt. Danach folgen ELMo und Bert mit jeweils .93. Dieses Ergebnis zeigt zum einen, dass der Qualitätsunterschied zwischen ELMo und Bert sehr gering, in diesem Fall sogar nicht zu messen ist. Zum anderen erweist sich das Nachjustieren eines Embeddings durch Training das auf Daten der untersuchten Domäne als so gewinnbringend, dass auch ein älterer Embeddingtyp wie fastText gegen den state-of-the-art bestehen kann.

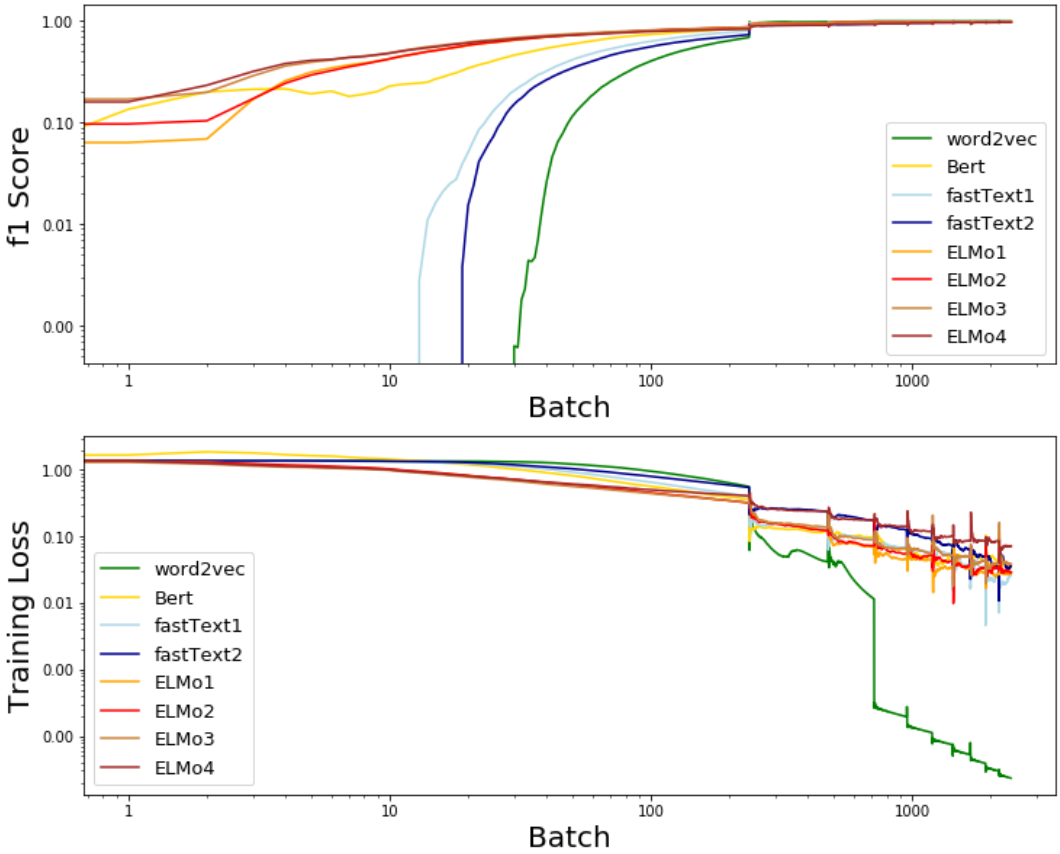

Abbildung 4.3: f1 Score und Training Loss, Verlauf über Trainingsprozess

Abbildung 4.3 zeigt den Verlauf von f1 Score und Loss über die Trainingsdauer für alle verwendeten Embeddings. Aus der Betrachtung des f1 Score leitet sich ein grundsätzlicher Unterschied zwischen festen und kontextsensitiven Embeddings ab. Während word2vec und fasttext genuin noch über keine Information zur Unterscheidung der Klassen verfügen, scheint diese in ELMo und Bert leicht für das Netz interpretierbar vorhanden zu sein, da bereits ab der ersten Batch ein f1 Score von ca. 0.1 erreicht wird. Mit fastText wird dieses Niveau erst bei Batch 30 erreicht, nach der Verarbeitung von 3000 Beispielsegmenten. Der f1 Score der ersten Batches für die ELMo Variationen zeigt, dass semantische Information zu großen Teilen im zweiten LSTM Layer repräsentiert wird, da hier ELMo3 und ELMo4 besser abschneiden. Es scheint aber für die hier gestellte Aufgabe ausreichend zu sein, einen gemittelten Vektor über alle Schich-

ten zu verwenden (ELMo1), was auch daran liegen kann, dass die Tiefe des Netzes nicht ausreicht, um mit dem gesamten Embedding (ELMo4) sinnvoll umzugehen. Der Trainings Loss zeichnet ein homogenes Bild für alle Embeddings, bis auf word2vec. Dessen Loss fällt rapide und deutet auf ein starkes Overfitting hin.
Alle nachfolgenden Ergebnisse beziehen sich auf die Klassifikation mit fastText2.

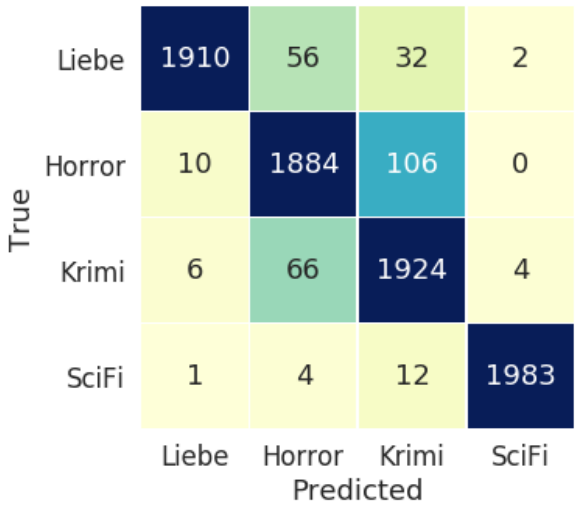

Abbildung 4.4: Confusionmatrix der Klassifikation mit fastText1

Abbildung 4.4 ermöglicht einen Einblick in die Fehlklassifikationen des Modells. Auf der x-Achse sind die tatsächlichen Label der Segmente aufgetragen und auf der y-Achse die vom Modell ermittelten. Science Fiction ist hier als das in sich geschlossenste also am leichtesten zuzuordnende Gerne zu lesen, denn von 2000 Segmenten entfallen lediglich 17 auf anderen Klassen. Am häufigsten werden Kriminal- und Horrorromane verwechselt und stehen sich zumindest nach diesem Befund am nächsten. Neben der Klassifikation der Genres kann das Netzwerk Aufschluss darüber geben, welche Wörter Entscheidungen wie stark beeinflusst haben. Um zu demonstrieren, dass der Attention Mechanismus sinnvolle Ergebnisse hervorbringt folgt ein Ausschnitt aus einem Science Fiction Roman[9].

[9]Kurt Mahr(1962): Perry Rhodan Band 47: Gom antwortet nicht.

Die Wörter[10] sind rot hervorgehoben, wobei eine hohe Sättigung für einen hohen Attention Wert steht:

> [...] Vor ihm lag schließlich nicht nur das gesamte Triebwerk mitsamt dem Empfängermechanismus , der die Fernsteuerungssignale aufnahm und verarbeitete , sondern auch der Generator zur Erzeugung des künstlichen Schwerefeldes in der Kabine , die Fernbildkamera , die ihre Impulse auf den Bildschirm des Fernsteuernden abstrahlte , und schließlich das Feldaggregat zweier schwerer Desintegratoren , die starr in den Außenmantel des Raumschiffes eingebaut waren. NE sah, dass er einen Fund gemacht hatte. Er sah allerdings auch, dass es nun in erster Linie darauf ankam, ob er ihn behalten konnte. Die Entfernung von NE nach NE war für ein Fahrzeug dieser Art in weniger als einer Stunde zu überwinden. Wenn er also nicht in gefährlicher Nähe des feindlichen Stützpunktes geraten wollte, dann musste er schnell handeln. Mit ein paar raschen Griffen löste er die Zuleitung zum Fernsteuer-Empfänger und unterbrach die Kontakte , so dass das Triebwerk von keinem von außen kommenden Signal mehr zu beeinflussen war. Dann untersuchte er das Triebwerk selbst und stellte fest, dass es im gleichen Augenblick aufgehört hatte, sich zu bewegen. [...]

Es ist augenscheinlich, dass ein Großteil der markierten Worte aus dem Vokabular für die technische Beschreibung von Raumschiffen besteht. Dies ist natürlich nachvollziehbar, da diese Wörte in keinem der

[10] statt des Maskierungstoken < *named_entity* > wird hier NE verwendet, um die Lesbarkeit zu erhöhen

anderen Genres zu finden sein werden. Die schwache Attention, welche auf allen maskierten Token für Eigennamen liegt, ist weniger intuitiv zu interpretieren. Eine mögliche Erklärung könnte sein, dass die direkte Umgebung von Eigennamen, also Interaktion von Personen und Personenbeschreibungen in besondere Weise nützlich für eine Klassifikation ist. Leider bietet die Datengrundlage für die Prüfung dieser Fragestellung keine Möglichkeit.

Tabelle 4.16: Distinktive Wörter für Genres nach Attention

Liebe	Horror	Krimi	SciFi
Tränen	Mumie	Gangster	NE
NE	Teufel	Detective	Mutanten
Baby	Dämonen	Cops	Mausbiber
küsste	Friedhof	Agent	Aras
Kuss	Magie	Dollar	Robotgehirn
sinnlichen	Hexe	Mord	Teleporter
T-Shirt	Detektivin	Mörder	Raumanzug
zog	Gral	Whisky	Energieschirm
Haushälterin	Hexen	Boss	Imperium
Sex	Werwolf	Kerl	Thermostrahler
Wut	Bösen	Revolver	Echsen
sexy	NE	Maschinenpistole	Helmlampe
Party	Killer	Rauschgift	Roboter
Familie	Teufels	Komplizen	positronischen
Slip	magischen	Spielclub	Energiekuppel

Die Einsicht, dass ein neuronales Netz eine bessere Klassifikation erzeugt als die Verwendung von Zeta-Werten, ist nicht sehr überraschend[11]. Schon allein der Tatsache wegen, dass Zeta hauptsächlich zum Auffinden distinktiver Wörter im mittleren Frequenzspektrum konzipiert ist. Um also auch diese Funktion mittels Deep Learning umzusetzen werden

[11] Verwendet man ein Mehrheitsvoting der Segmente eines Textes, um auf dessen Genre zu schließen, wird eine perfekte Klassifikation erreicht.

die Attention Werte aller Segmente für jedes Wort des Testkorpus ermittelt. Ein reines Berechnen der relativen Attention ist zunächst wenig Aufschlussreich, da auf diese Weiße lediglich Stopwords wie Pronomen, Konjunktionen und Hilfsverben in den oberen Rängen platziert werden. Dies ist auf den Umstand zurückzuführen, dass diese Funktionswörter zuverlässig im mittleren Bereich der Verteilung der Attention liegen. Um aussagekräftigere Wörter zu finden, wird der Datensatz vorgefiltert, um nur solche Worte zu berücksichtigen, welche tatsächlich ausschlaggebend sind. Hierfür werden nur die Wörter im oberen Quartil der Attentionverteilung innerhalb ihres Segments für die Berechnung zugelassen. Im Anschluss wird für diese Wörter die relative Attention errechnet. Das Ergebnis für die vier Genres ist in Tabelle[12] 4.16 dargestellt und ist zumindest qualitativ nicht schlechter als die von Zeta erzeugten Wortlisten.

[12]Das Wort „Aras“ sollte durch das Maskierungstoken ersetzt werden, vermutlich wurde es aufgrund der gleichnamigen Vogelart nicht erkannt

KAPITEL 5

DISKUSSION

5.1 Experiment 1

Die für diese Arbeit motivierende Hypothese, dass die Verbindung von Word Embeddings mit Zeta für die Suche nach distinktiven Wörtern in Genres nutzbringend ist, wird als bestätigt angesehen. Tatsächlich kann die semantische Information in Word Embeddings verwendet werden, um bei der Klassifikation von Texten Autorschaftssignale zu unterdrücken, indem die Ähnlichkeit aufgrund von Worthäufigkeitsverteilungen auf eine abstraktere Repräsentation basierend auf Wortfeldern gehoben wird. Diese Wortfelder, den Topics erzeugt durch LDA (blei2003latent) ähnlich, werden als Clusterzentren von distinktiven Wörtern im Vektorraum des Embeddings formalisiert. Dieses Vorgehen führt zu höheren Ähnlichkeitswerten für Texte mit gleicher Thematik (vgl. Vorstudie I, Test Cases I-V), bei gleichzeitig zunehmender Unschärfe bei der Unterscheidung von Autorschaft (vgl. Vorstudie II, Test Cases VI-VIII).
Um die Methode näher zu beleuchten folgt ein Beispiel anhand der binären Klassifikation von Adels- und Familienromanen unter der Verwen-

dung der in Vorstudie I ermittelten Kombination aus Parametern.

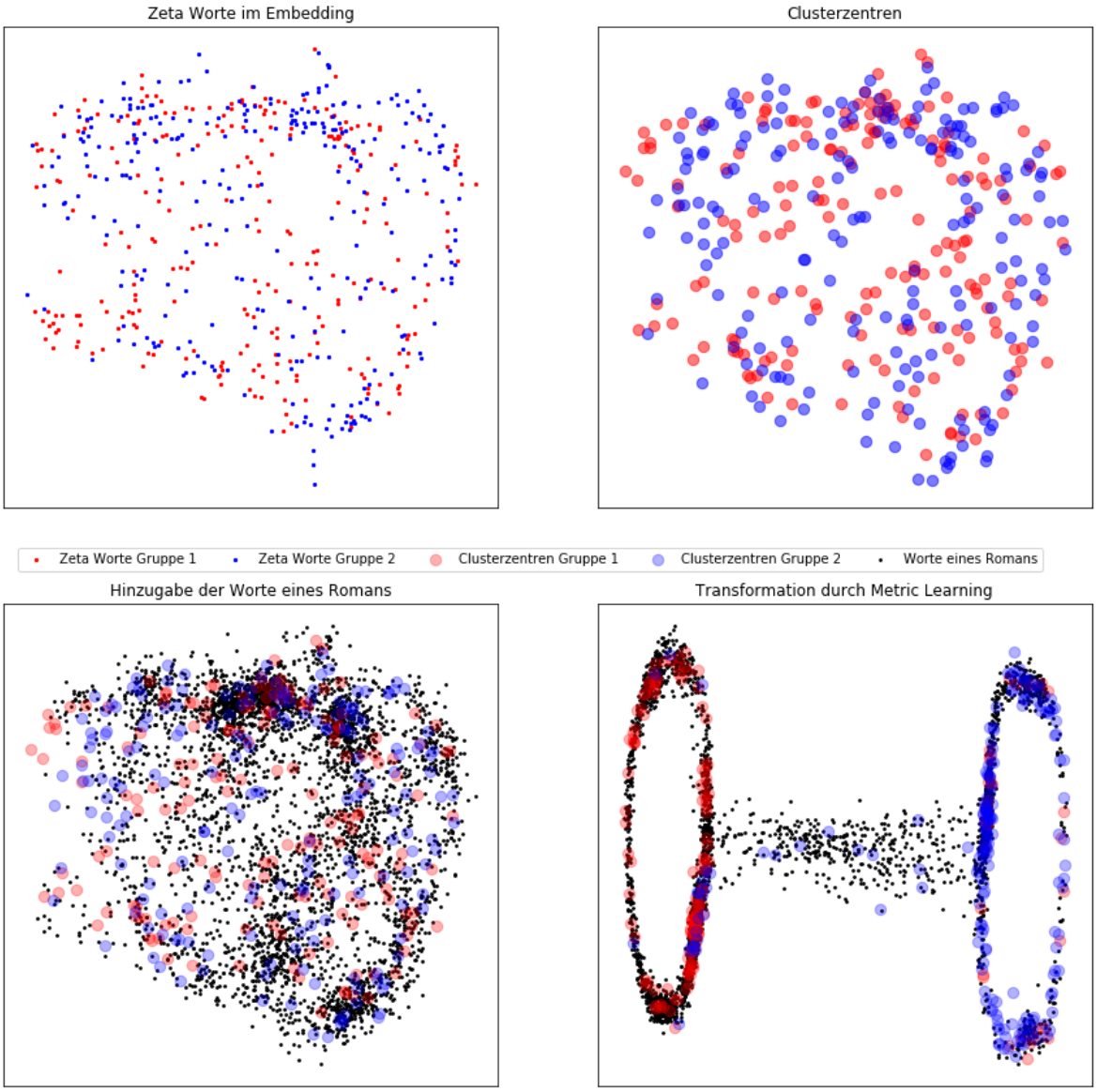

Abbildung 5.1: Anwendung von Clustering und Metric Learning

Abbildung 5.1 veranschaulicht den Workflow der Methode. Der erste Plot zeigt die aus den Trainingsromanen ermittelten distinktiven Wörter für beide Textgruppen in den 2-dimensionalen Raum. Der Vektorraum verfügt tatsächlich über 300 Dimensionen, die Reduktion dient hier nur der Anschaulichkeit. Der zweite Plot zeigt die Clusterzentren, welche mithilfe des Birch-Algorithmus aus den Wortvektoren berechnet wurden.

In der dritten Abbildung werden die Worte eines Romans über die Clusterzentren gelegt. Um von dieser Darstellung zur nächsten zu gelangen, wird eine Dimensionsreduktion der Clusterzentren durchgeführt, welche nicht nur den Erhalt von möglichst viel Information der Ausgangsdimensionen zum Ziel hat, sondern auch dahingehend optimiert wird, dass die Zentren einer Gruppe sich untereinander nahe stehen und gleichzeitig große Abstände zu Zentren der Gegengruppe entstehen. Naturgemäß ist keines dieser drei Ziele in Gänze zu erfüllen. So ist erkennbar, dass Clusterzentren beider Gruppen auch im Großcluster der Gegengruppe zu finden sind.
Tatsächlich profitiert die Methode von diesem Ungenügen sogar. Denn die Gruppenzugehörigkeit eines Wortes ergibt sich aus der Differenz zu den Clusterzentren beider Gruppen mit minimaler Distanz. Wenn also ein Cluster und ein Cluster der Gegengruppe sehr dicht stehen, wird die Differenz des Abstandes eines Wortes zu diesen Gruppen immer relativ gering ausfallen und somit wenig Einfluss auf die Klassifikation des Dokuments ausüben. Was nur sinnvoll ist, da räumliche Nähe hier semantische Nähe repräsentiert und damit beide Clusterzentren nicht diskriminativ für ihr Genre sind.
Kritisch zu sehen ist die Identifikation des Birch Algorithmus als favorisiertes Clusterverfahren, denn es ist nicht klar, ob tatsächlich die Methode sinnvollere Cluster bildet, da die Klassifikationsgüte mit der Anzahl an ermittelten Clustern korreliert. Am Eindrücklich wird dies unter Verwendung von MeanShift. Die Methode erzeugt konsequent weniger Cluster als die anderen Verfahren und die Kombination mit Metric Learning scheint die Information in selbigem weiter zu reduzieren, sodass keine sinnvollen Ergebnisse mehr erreicht werden können (vgl. bspw. Vorstudie I). Dieser Befund zeigt, dass das Potential der Methode noch nicht zur Gänze ausgereizt ist und eine weitere Parametrisierung der Clusterverfahren

basierend auf Eigenschaften der Zeta-Wörter angestrebt werden sollte.

Tabelle 5.1: 5 Cluster zur Unterscheidung von Adels und Familienromanen

Aus dem Genre Adelsroman				
Tee	Tod	Baron	Reiten	Hauptportal
trinken	Trauer	Stallmeister	Pferde	Suite
Kamillentee	Unglück	Herr	Stall	Raum
Kuchen	Verlust	Fürst	Hengst	Ostflügel
Torte	Grab	Kriminalrat	Stute	Eingangshalle
Aus dem Genre Familienroman				
Haus	Wagen	Kinder	zärtlich	Spinat
Häuschen	Auto	Babys	sanft	Appetit
Nachbarn	Bus	Familien	strich	Eier
Villa	Bollerwagen	Mütter	behutsam	Brot
Hinterhof	Fahrrad	erziehen	küßte	Fleisch

Tabelle 5.1 zeigt je fünf Cluster, welche als Nebenprodukt der Unterscheidung zwischen Adels- und Familienroman entstanden sind. Qualitativ sind diese weder besser noch schlechter als die Ergebnislisten des klassischen Zeta Verfahrens. Sie unterscheiden sich jedoch darin, dass ein konkreter Wert analog zum Zeta Wert fehlt. Dieser könnte indirekt durch den Abstand jedes Wortes eines Clusters zu dessen Clusterzentrum berechnet werden. Ob ein solches Ranking sinnvoll ist muss noch erprobt werden. Dieser Nachteil wird durch die Gruppierung der Worte in Clustern ausgeglichen. So ist es auf diese Weise möglich herauszufinden, welche Eigenschaften, hier Wortfelder, für eine Textgruppe konstituierend und welche eher optional sind. Auf dieser Basis können dann wiederum Subgenres identifiziert und Aussagen über Einzelromane getroffen werden. Beispielsweise lässt sich so beantworten wie prototypisch ein Roman für sein Genre ist. Als Prototyp kann hier entweder ein Roman mit großer Abdeckung an Clusterzentren oder einer Verteilung über die Zentren, welche der Gesamtverteilung aller Romane am nächsten kommt, definiert

werden.
Vergleicht man die Methode mit dem Konzept der Familienähnlichkeit (siehe Kap 1.2) kann festgestellt werden, dass die Modellierung ihrem literaturtheoretischen Ideal nahe kommt. Definiert man die Clusterzentren als Merkmale, so ist die Bedingung, dass ein Cluster für mehrere Genres charakteristisch sein kann, erfüllt. Es handelt sich im Modell zwar nicht de facto um die gleichen Cluster, sondern um zwei sich lediglich semantisch nahe stehende, jedoch können diese gleichzeitig charakteristisch und nicht diskriminativ sein. Gleichzeitig muss ein Text nicht sämtliche Cluster einer Gruppe bedienen, um dieser zugeordnet zu werden.

5.2 Experiment 2

Die Verwendung von Deep Learning zur Klassifikation von Heftromangenres erweist sich als grundsätzlich sinnvoll, dennoch ist auch dieses Verfahren nicht perfekt. Die Confusion Matrix[1] der Klassifikation gibt Anlass sich mit den falsch eingeordneten Segmenten zu beschäftigen, da auf diese Weise sowohl Erkenntnisse über die Funktion und Schwächen des Netzes, als auch Aussagen über die Datengrundlage ermöglicht werden. Es folgt ein Segemt aus einem Horrorroman[2], welches als aus dem Liebes-Genre stammend klassifiziert wurde:

> [...] „Langweilst du dich?“, erkundigte sich NE lächelnd und beugte sich über ihren Mann , der auf der Couch im Wohnzimmer lag und zur Decke starrte . Er schlang seine Arme um sie und zog sie an sich . Doch ehe er sie küssen konnte, meldete sich der Nachwuchs . Seufzend richtete sich NE

[1] siehe Kap. 4.2.2 Abb. 4.4

[2] Richard Wunderer (hier als Jason Dark): John Sinclair Folge 46: Die Dämonenschmiede. 1976. Batei

wieder auf. „Es wäre so schön gewesen. Aber im Ernst , wenn du nicht weißt, was du tun sollst, kannst du mir helfen. Der Staubsauger steht in der Abstellkammer .“ Während sie sich um ihren Sohn kümmerte , stemmte sich NE NE von der Couch hoch und machte sich an die Arbeit. Er konnte seiner Frau nicht sagen, was ihn bedrückte . Er langweilte sich tatsächlich ein wenig , obwohl er mehr als genug zu tun hatte . Er sehnte sich nach gefährlichen Abenteuern , aber NE passte auf, dass er sich auf keine risikoreichen Unternehmen einließ . Dazu liebte sie ihn zu sehr . Und er liebte seine Frau so sehr, dass er sich an ihre Bitten hielt .[...]

Tatsächlich ist dieser Textabschnitt auch für einen menschlichen Leser kaum als Teil eines Gruselromans zu identifizieren. An der Markierung der Attention lässt sich nachvollziehen, dass das Modell auf Wörter wie *Wohnzimmer*, *küssen*, *Staubsauger*, *Nachwuchs* und *Sohn*, die Szene als Alltagsbeschreibung eines Liebesromans wertet. Die mit dem Vorwissen, dass es sich um einen Ausschnitt aus einem Horrorroman handelt, distinktiv wirkenden Abschnitte *gefährlichen Abenteuern*, *riskoreichen Unternehmen*, können im Kontext eines Liebesromans auch auf die Anbahnung eines Seitensprungs gedeutet werden und verstärken das Signal für eine falsche Klassifikation noch zusätzlich.
Das nächste Segment stammt aus einem Liebesroman[3], wird aber dem Genre Krimi zugeordnet:

[...] „Das hat dir vorher doch auch nichts ausgemacht“, warf er ein. „Weil das vorher auch alles war, was ich hatte“ , gab sie zurück. NE so konnte ich überhaupt in deiner

[3]Charlotte Phillips (2014): Julia Extra 399: Spiel nicht mit der Liebe!. Cora Verlag.

Nähe sein. „Aber jetzt reicht mir das nicht mehr. Und nach dem letzten Wochenende dachte ich, dir würde es genauso ergehen.“Der Kellner führte zwei Männer in Geschäftsanzügen zu ihrem Tisch . „Bitte geh jetzt nicht“, bat er leise. „Lass uns die Besprechung mit den beiden schnell hinter uns bringen, damit wir dies hinterher besprechen können.“ NE lachte trocken und hob abwehrend die Hände .„Das ist genau der Punkt “, sagte sie mit sarkastischem Unterton. „Du denkst tatsächlich , wir würden uns erst mit deinen Geschäftsfreunden zusammensetzen – und hinterher besprechen, was aus uns beiden werden soll . “ Sie zuckte die Schultern. „ Nicht mit mir . Was immer das auch zwischen uns war – platonische Freundschaft oder Kurzaffäre – es ist aus.“[...]

Leider zeigt dieses Beispiel, dass die Gründe für eine Fehlklassifikation nicht immer nachvollziehbar sind. Passagen welche eindeutig auf ein Beziehungsdrama hindeuten „Aber jetzt reicht mir das nicht mehr. Und nach dem letzten Wochenende dachte ich, dir würde es genauso ergehen“, „was aus uns beiden werden soll“ oder „es ist aus“ erhalten nur geringe Attention Werte. Stattdessen konzentriert sich das Modell auf *Kellner*, *Geschäftsanzügen* und *Geschäftsfreunden* und ordnet die Szene als reines Geschäftsessen ein, ohne die hintergründigen Beziehungsprobleme zu erkennen. Der stärkste Hinweis auf Wortebene „Affäre“ , wird im Kontext des Geschäftsessens als geschäftliche Affäre und nicht wie im Text angelegt als Seitensprung gedeutet.
Die nächste Analyse beschäftigt sich mit einem kompletten Liebesroman, welcher in Segmente unterteilt und klassifiziert wird. Um das Ergebnis richtig deuten zu können, hier eine Zusammenfassung der Handlung: Die

junge Lehrerin Anne wird von ihrer Mutter auf dem Sterbebett gebeten in der Türkei nach ihrer Schwester Eva zu suchen. Die Schwestern haben seit langem keinen Kontakt mehr, da Eva mit Armin, Annes damaligen Partner, durchgebrannt ist. In der Türkei trifft Anne den schwedisch-deutschen Generaldirektoren Erik von Bergen, welcher ihr bei der Suche hilft. Die beiden verlieben sich, müssen aber erfahren, dass Eva sich mit zwielichtigen Gestalten umgeben und schließlich verstorben ist. Allerdings hat sie ein Kind hinterlassen, welches Anne und Erik adoptieren.
Abbildung 5.2 zeigt die Wahrscheinlichkeit der Segmente des Romans zu einem der Genres zu gehören. Die Reihenfolge der Segmente entspricht der natürlichen Reihenfolge des Textes, beginnend von unten. Von den 172 Segmenten werden 100 aus einem Liebes-, 57 aus Horrorgenre und 15 aus einem Kriminalroman stammend identifiziert. Science Fiction hat für keinen der Textabschnitte die höchste Wahrscheinlichkeit. Der Roman würde also bei einem Mehrheitsvoting richtig klassifiziert. Während die Klassifikation von Segmenten als Kriminalroman auch als Zufallsverteilung gedeutet werden kann, gibt die Verteilung des Horrorgenres durch die Häufung dieser Segmente zu ganzen Textabschnitten Spielraum für Interpretation. Exemplarisch sind in Abb 5.2 vier dieser Abschnitte markiert. Tatsächlich scheinen die Abschnitte nicht zufällig zu entstehen, sondern markieren vielmehr Passagen mit besonders negativer emotionaler Färbung. Tod der Mutter und anschließende Trauer, sowie die Nachricht von Evas Tod lassen sich hier sehr plausibel einordnen. Interessanter ist die Ankunft in Istanbul, die Beschreibung einer fremden Umgebung, welche auf die Protagonistin tatsächlich bedrohlich wirkt, scheint nicht typisch für das Genre des Liebesroman zu sein.

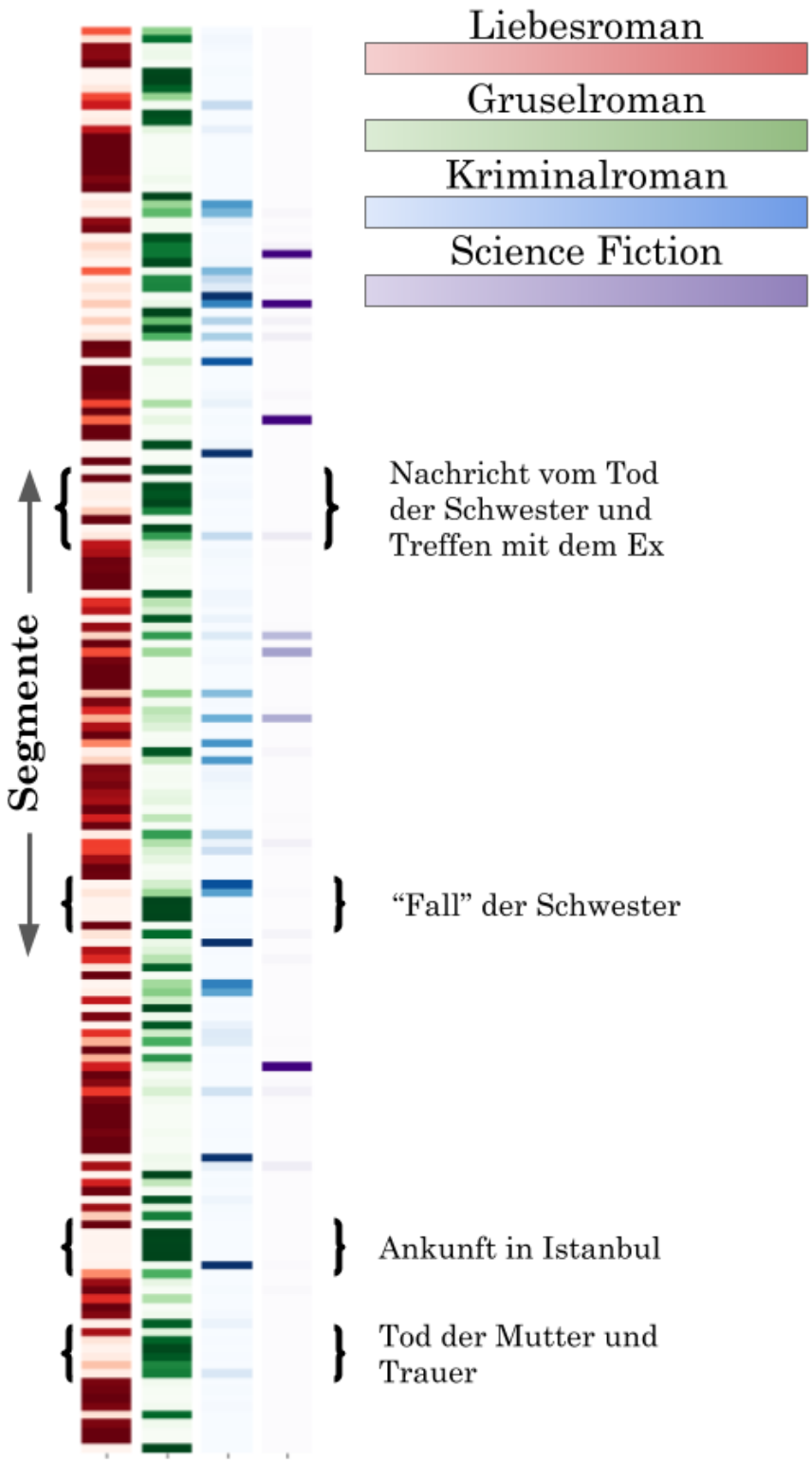

Abbildung 5.2: Zeitstrahl der Segmente aus *Kein leichtes Leben* und deren Wahrscheinlichkeit zu einem Genre zu gehören

Der folgende Abschnitt stammt aus dem ersten Drittel des Romans und hat eine relativ hohe Wahrscheinlichkeit dafür aus dem Science Fiction Genre zu stammen[4]. Er schildert die Fahrt Annes vom Flughafen zu von Bergens Anwesen:

> [...] „So viel Schönheit erhitzt die Gemüter“, fuhr er fort. Es klang ernst. Es klang auch wie eine leise Warnung, in diesem fremden Land äußerste Vorsicht zu wahren. „Die Sonne erhitzt nicht nur die Stadt , auch das Blut . Es wäre besser, Sie würden sich noch ein paar Tage Ruhe gönnen. Der Körper muss sich auf dieses Klima einstellen. Und außerdem haben Sie Urlaub“, NE seufzte leise, dachte an die leere Wohnung, an ein mit Kränzen und Blumen geschmücktes Grab – und wurde traurig. Es schien, als spüre er den jähen Umsturz ihrer Gefühle. Er nahm seine Hand vom Steuer und legte sie auf ihre Hände . Lächelnd schlug er vor : „Wir sollten diesen ersten Tag noch nicht beenden. Ich kenne ein Hotel, in dem es sehr behaglich ist . Darf ich Sie dorthin einladen – zu einem Drink ?“ „Dieser Tag ist zu Ende “, sagte sie kühl. NE von NE antwortete nicht.[...]

Ausschlaggebend für das Resultat sind die Wörter „erhitzt“, „Sonne“, „Klima“ und „Umsturz“. Es scheint so, dass dieses aus dem Kontext genommen technisch/naturwissenschaftliche Vokabular hier die Wahrscheinlichkeit für Science Fiction erhöht. Auch das Topos des Fremden, gefährlichen Landes/Planeten kann an dieser Stelle das Ergebnis beeinflusst haben.
Das nächste Segment ist Teil einer Unterhaltung zwischen von Bergens Sekretärin und weiteren Mitgliedern seines Personals über Annes Verhältnis zu Erik und ihre möglichen Absichten:

[4]Die Wahrscheinlichkeit für Liebesroman liegt dennoch höher.

> [...] „Ein Eisblock ist sie! Oh , wäre sie nie in dieses Haus gekommen!“ „Ich habe von NE bereits gewarnt“, sagte ein älterer Kollege . „Aber er hat mich ausgelacht.“ „Auch das ist verdächtig “, fand NE . „Nie zuvor hat er sich von einem weiblichen Wesen so beeindrucken lassen.“ „Sie ist bezaubernd und intelligent .“ „Alles Vorzüge, die der Spionage dienen.“ „Hier wird sie nichts finden. Und sie weiß auch nicht, wo der Safe ist.“ „Soll sie doch fahren und nicht mehr wiederkommen“ , murmelte NE . Sie gestand sich nicht ein, dass Eifersucht sie so reden ließ. Für NE war es nicht leicht, dem Fahrer des Taxis klarzumachen , wo ihr Reiseziel lag . Er verstand ein paar Brocken Englisch , aber seine Gesten und Blicke deuteten darauf hin, dass er es ähnlich wie jener Pferdekutscher machen würde.[...]

Der Abschnitt wurde zum Krimi Genre zugehörig klassifiziert. Ausschlaggebend sind hier vor allem die Keywords „Spionage“ und „Safe“.
Die Fehlklassifikationen des Modells werden hauptsächlich durch fehlenden Kontext verursacht. Die offensichtliche Antwort auf diesen Befund ist die Vergrößerung der Teilsegmente. Gleichzeitig erlauben gerade die Fehler bei der Klassifikation kleiner Segmente aufschlussreiche Einblicke in die Struktur von Einzeltexten und Beziehungen von Genres untereinander. Die Möglichkeiten zur Analyse von Romanen des Modells sind mit der Klassifikation von Genre noch nicht ausgereizt. Denkbar wären Klassifikationen, welche den Inhalt von Segmenten näher bestimmen sollen. Die Identifikation von emotional negativ geladenen Passagen durch die Verwendung des Genres Horror als Proxy funktioniert zwar überraschend gut, es wäre aber methodisch sinnvoller ein Korpus aufzubauen, welches dezidiert auf Handlungselementen wie Trauerfällen trainiert werden

kann.

KAPITEL 6

ZUSAMMENFASSUNG

Ziel dieser Arbeit war das Hervorheben neuer Chancen und Herausforderungen, welche die Technologie Word Embeddings für die textwissenschaftlichen Digital Humanities bedeutet. Unter einer Vielzahl von möglichen Anwendungen wurde die Aufgabe der Klassifikation von Genre in Heftromanen gewählt, um an ihr die Erweiterung vorhandener und die Möglichkeit zur Entwicklung neuer Methoden exemplarisch durchzuführen.

Theoretisch aufbereitet und getestet wurden die Embeddings word2vec, fastText, ELMo und Bert. Diese konnten in Kombination mit Burrow's Zeta zeigen, dass der Einsatz von Embeddings in Textklassifikationen das Autorensignal unterdrückt und im Gegenzug eine Unterscheidung auf thematischer bzw. semantischer Ebene verbessert. Die Methode Zeta, entwickelt für die Identifikation von distinktiven Wörtern des mittleren Frequenzspektrums, wurde so modifiziert, dass statt einer Ausgabe von Wörtern und zugehörigen Distinktivitätswerten Cluster von semantisch ähnlichen distinktiven Wortfeldern ermittelt werden können.

Um zu zeigen, welche neuen methodischen Wege durch Embeddings er-

schlossen werden können, wurde die Aufgabe der Klassifikation und der Identifikation von Distinktivität neu operationalisiert. Statt auf Zeta und damit letztlich auf der Verteilung von Worthäufigkeit innerhalb von Textgruppen zu gründen, wurde ein Ansatz basierend auf Deep Learning erprobt. Aus dem Bereich der maschinellen Sprachverarbeitung ist bekannt, dass Deep Learning für Textklassifikation grundsätzlich anderen Methoden überlegen ist. Daher wäre es unzureichend, und wenig innovativ, ein neuronales Netz nur als „Black Box[1]“ zur Lösung einer Klassifikationsaufgabe zu nutzen. Um die zweite Anforderung, also das Finden distinktiver Wörter, zu ermöglichen, wurde ein Attention-Mechanismus im Netz implementiert, so dass im Nachgang zur Klassifikation ermittelt werden kann, auf Grundlage welcher Wörter eine Entscheidung getroffen wurde. Die Attentionwerte zusammen mit deren Häufigkeit können in ein Ranking überführt werden, welches dem Zetas ähnlich ist. Während Zeta eine stabile Klassifikation erst ab einer Segmentgröße von ca. 10.000 Worten erreicht, leistet das neuronale Netz eine solide Klassifikation bereits für 200 Worte. Diese feingranulare Auflösung kann verwendet werden, um Muster in Einzeltexten zu erfassen. Dies wurde anhand des Liebesromans *Kein leichtes Leben* durchexerziert und ein zumindest qualitativer Zusammenhang zwischen Klassifikation von Textabschnitten und Handlung gefunden.

Die Evaluation der Embeddingtypen hat gezeigt, dass die Fortschritte, welche jede Methode zur Erstellung von Embeddings eingeführt hat, auch bei der Untersuchung von literarischen Texten zu Verbesserungen führt. So sind ELMo und Bert als kontextsensitive Embeddings fastText und word2vec überlegen. Gleichzeitig, und hierbei handelt es sich um die vielleicht wichtigste Erkenntnis dieser Arbeit, ist der positive Effekt der Anpassung eines Embeddings an die Textdomäne stärker als der Einfluss

[1] Ausdruck der auf die Unzulänglichkeit hinweißt, dass Deep Learning zwar Aufgaben lösen kann, die internen Ablöufe jedoch undurchsichtig bleiben.

der verwendeten Methode zur Erstellung des Embeddings. Hieraus ergibt sich ein deutlicher Handlungsbedarf zur Erstellung von an literarische und auch historische Sprache angepassten Word Embeddings für die Digital Humanities.

KAPITEL 7

AUSBLICK

Diese Arbeit konnte im Rahmen ihrer Möglichkeiten die Chancen und Herausforderungen der Technologie Word Embeddings für die Digital Humanities anhand von Textklassifikation aufzeigen. Damit ist aber nur ein kleiner Teil an Methodik abgedeckt. Word Embeddings bieten natürlich auf vielen anderen Feldern ebenfalls Innovationpotential. Darunter fallen für die Literaturwissenschaft Sentiment Analysis, Emotionsdetektion, Analyse von Plot und Figurenkonstellation, um nur einige zu nennen.
Auch die eingeführten Methoden, sollten sie auf Anklang stoßen, müssen weiter evaluiert werden, so ist betreffend der Kombination von Zeta und Word Embeddings nicht zwingend jeder Parameter optimal gewählt. So konnten aus einer Vielzahl an Clusterverfahren lediglich drei Methoden erprobt werden. Gleiches gilt für Metric Learning und die Frage ob dieses vor oder nach dem Clustering zu verwenden ist. Es handelt sich hier jedoch um methodische Detailfragen und die Anhebung der Accuracy um wenige Prozentpunkte sollte nicht im Vordergrund stehen. Wichtiger ist die Frage wie die ermittelten distinktiven Wortcluster verwendet werden können, um Rückschlüsse auf die Textgrundlage zuzulassen und wie diese

in ein Zeta ähnliches Ranking überführt werden können. Eine Studie zu Subgenres anhand von distinktiven Clustern und ihrem Auftreten in Romanen wäre wünschenswert und realistisch umzusetzen.
Selbiges gilt für die Deep Learning basierte Methode. Auch hier ist eine Anpassung der Parameter eher zweitranging. Im Vordergrund sollte der Umgang mit den Ergebnissen stehen. Die Berechnung des Rankings nach Distinktivität ist hier nur eine basale Herangehensweise und sollte statistisch genauer untersucht werden. Wesentlich spannender sind die Befunde zur Untersuchung von Einzeltexten. Diese sind in der Arbeit nur exemplarisch durchgeführt worden und müssen in einen größeren Kontext eingeordnet werden. Vielversprechend ist die Untersuchung von mehreren Einzeltexten, um nach Mustern zu suchen. Denkbar wären hier Fragen wie: Benötigt ein Liebesroman mit negativen Emotionen behaftete Passagen, um emotionale Kontraste zu erzeugen oder umgekehrt, wie viel Text verwendet ein Autor von Science Fiction Romanen, um eine Liebesgeschichte zu erzählen.
Die Evaluation der Embeddings hat vor allem gezeigt, dass Domänenadation großen Einfluss auf die Qualität eines Modells hat. Mit diesem Hintergrundwissen als Motivation sollte nun untersucht werden, wie viel Text benötigt wird, um Word Embeddings anzupassen und wie stark der Anpassungsgrad verschiedener Emebeddingtypen skaliert. Das hier vorgestellt Korpus aus Heftromanen kann nur die literarische Dimension des Problems abbilden, für eine Studie hinlänglich historischer Sprache muss ein weiterer Datenbestand gefunden oder aufgebaut werden.

LITERATUR

Akbik, Alan, Duncan Blythe und Roland Vollgraf (2018). "Contextual String Embeddings for Sequence Labeling". In: *COLING 2018, 27th International Conference on Computational Linguistics*, S. 1638–1649.

Allison, Sarah et al. (2011). "Quantitative Formalism: An Experiment". In: *Stanford Literary Lab Pamphlet* 1.

Baeza-Yates, Ricardo, Berthier de Araújo Neto Ribeiro et al. (2011). *Modern information retrieval*. New York: ACM Press; Harlow, England: Addison-Wesley.

Bahdanau, Dzmitry, Kyunghyun Cho und Yoshua Bengio (2014). *Neural machine translation by jointly learning to align and translate*. eprint: arXiv:1409.0473.

Bengio, Yoshua et al. (2003). "A neural probabilistic language model". In: *Journal of machine learning research* 3.Feb, S. 1137–1155.

Bojanowski, Piotr et al. (2017). "Enriching word vectors with subword information". In: *Transactions of the Association for Computational Linguistics* 5, S. 135–146.

Burrows, John (2002a). "Delta: a Measure of Stylistic Difference and a Guide to Likely Authorship". In: *Literary and Linguistic Computing* 17.3, S. 267–287. DOI: 10.1093/llc/17.3.267.

— (2002b). "Delta: a measure of stylistic difference and a guide to likely authorship". In: *Literary and linguistic computing* 17.3, S. 267–287.

— (2003). "Questions of Authorship: Attribution and Beyond: A Lecture Delivered on the Occasion of the Roberto Busa Award ACH-ALLC

2001, New York". In: *Computers and the Humanities* 37.1, S. 5–32. ISSN: 00104817. URL: http://www.jstor.org/stable/30204877.

Burrows, John (2007). "All the Way Through: Testing for Authorship in Different Frequency Strata". In: *Literary and Linguistic Computing* 22.1, S. 27–47. DOI: 10.1093/llc/fqi067.

Che, Wanxiang et al. (2018). "Towards Better UD Parsing: Deep Contextualized Word Embeddings, Ensemble, and Treebank Concatenation". In: *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Brussels, Belgium: Association for Computational Linguistics, S. 55–64. URL: http://www.aclweb.org/anthology/K18-2005.

Cho, Kyunghyun et al. (2014). *On the properties of neural machine translation: Encoder-decoder approaches*. eprint: arXiv:1409.1259.

Chollet, François et al. (2015). *Keras*. https://keras.io.

Chung, Junyoung et al. (2014). *Empirical evaluation of gated recurrent neural networks on sequence modeling*. eprint: arXiv:1412.3555.

Devlin, Jacob et al. (2018). *Bert: Pre-training of deep bidirectional transformers for language understanding*. eprint: arXiv:1810.04805.

Engelberg, Stefan (2015). "Quantitative Verteilungen im Wortschatz. Zu lexikologischen und lexikografischen Aspekten eines dynamischen Lexikons". In: *Sprachwissenschaft im Fokus. Positionsbestimmungen und Perspektiven*. Hrsg. von Ludwig M. Eichinger. De Gruyter, S. 205–230. URL: http://nbn-resolving.de/urn:nbn:de:bsz:mh39-34810.

Evert, Stefan et al. (2015). "Towards a better understanding of Burrows's Delta in literary authorship attribution". In: *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, S. 79–88.

Fares, Murhaf et al. (2017). "Word vectors, reuse, and replicability: Towards a community repository of large-text resources". In: *Proceedings of the 21st Nordic Conference on Computational Linguistics*. Gothen-

burg, Sweden: Association for Computational Linguistics, S. 271–276. URL: http://www.aclweb.org/anthology/W17-0237.

Firth, J. R. (1957). "A synopsis of linguistic theory 1930-55." In: *Studies in Linguistic Analysis (special volume of the Philological Society)*. Bd. 1952-59. Oxford: The Philological Society, S. 1–32.

Frey, Brendan J und Delbert Dueck (2007). "Clustering by passing messages between data points". In: *science* 315.5814, S. 972–976.

Fukunaga, Keinosuke und Larry Hostetler (1975). "The estimation of the gradient of a density function, with applications in pattern recognition". In: *IEEE Transactions on information theory* 21.1, S. 32–40.

Goldberg, Yoav (2019). "Assessing BERT's Syntactic Abilities". In: *CoRR*. arXiv: 1901.05287.

Harris, Zellig (1954). "Distributional structure". In: *Word* 10.23, S. 146–162.

Hettinger, Lena et al. (2016). "Classification of Literary Subgenres". In: *DHd 2016*.

Hochreiter, Sepp und Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, S. 1735–1780.

Horev, Rami (2018). *BERT Explained: State of the art language model for NLP*. URL: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270.

Huegel, Hans Otto (2002). "Kommunikative und ästhetische Funktion des Romanhefts". In: *Handbuch zur Medienwissenschaft 3. Teilband*. Hrsg. von Joachim-Felix Leonhardt und Hans-Werner Ludwig, S. 1621–1631.

Jockers, Matthew L (2013). *Macroanalysis: Digital methods and literary history*. University of Illinois Press.

Käther, Matthias (2018). *Gothic Romance: Das unterschätztes Genre*. URL: https://www.zauberspiegel-online.de/index.php/phantastisches/

gedrucktes-mainmenu-147/33953-gothic-romance-das-unterschaetzte-genre-teil-1-ein-faszinierenden-genre-1 (besucht am 09. 01. 2018).

Lahn, Silke und Meister Jan Christoph (2016). *Einführung in die Erzähltextanalyse*. J.B. Metzler.

Lewis, Roger J (2000). "An introduction to classification and regression tree (CART) analysis". In: *Annual meeting of the society for academic emergency medicine in San Francisco, California*. Bd. 14.

Manning, Christopher, Prabhakar Raghavan und Hinrich Schütze (2010). "Introduction to information retrieval". In: *Natural Language Engineering* 16.1, S. 100–103.

Martın Abadi et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. URL: https://www.tensorflow.org/.

McInnes, Leland et al. (2018). "UMAP: Uniform Manifold Approximation and Projection". In: *The Journal of Open Source Software* 3.29, S. 861.

Mikolov, Tomas et al. (2013). *Efficient estimation of word representations in vector space*. eprint: arXiv:1301.3781.

Müller, Andreas (2015). *GermanWordEmbeddings*. URL: https://github.com/devmount/GermanWordEmbeddings.

Olah, Christopher (2015). *Understanding LSTM Networks*. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Pang, Bo, Lillian Lee et al. (2008). "Opinion mining and sentiment analysis". In: *Foundations and Trends® in Information Retrieval* 2.1–2, S. 1–135.

Paszke, Adam et al. (2017). *Automatic differentiation in PyTorch*.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, S. 2825–2830.

Pennington, Jeffrey, Richard Socher und Christopher Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of the*

2014 conference on empirical methods in natural language processing (EMNLP), S. 1532–1543.

Peters, Matthew E. et al. (2018). “Deep contextualized word representations”. In: *Proc. of NAACL*.

Peters, Matthew E et al. (2018). *Dissecting contextual word embeddings: Architecture and representation*. eprint: arXiv:1808.08949.

Radford, Alec et al. (2019). *Language Models are Unsupervised Multitask Learners*.

Řehůřek, Radim und Petr Sojka (2010). “Software Framework for Topic Modelling with Large Corpora”. English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. http://is.muni.cz/publication/884893/en. Valletta, Malta: ELRA, S. 45–50.

Schmidt, Ben (2015). *Vector Space Models for the Digital Humanities*. URL: http://bookworm.benschmidt.org/posts/2015-10-25-Word-Embeddings.html.

Schnabel, Uwe (2011). *Der Zauberkreisverlag*. URL: https://www.zauberspiegel-online.de/index.php/durchblick-hintergrnde-mainmenu-15/druck-und-buch-mainmenu-295/6475-der-zauberkreis-verlag (besucht am 09. 01. 2018).

Schöch, Christof (2017). “Topic Modeling Genre: An Exploration of French Classical and Enlightenment Drama.” In: *DHQ: Digital Humanities Quarterly* 11.2.

— (2018). “Zeta für die kontrastive Analyse literarischer Texte”. In:

Schöch, Christof et al. (2018). “Burrows Zeta: Varianten und Evaluation”. In: *DHd 2018*.

Schöch, Christoph et al. (2018). “Burrows’ Zeta: Exploring and Evaluating Variants and Parameters”. In: *DH 2018 Book of Abstracts*. ADHO.

Seymore, Kristie, Andrew McCallum und Roni Rosenfeld (1999). “Learning hidden Markov model structure for information extraction”. In:

AAAI-99 workshop on machine learning for information extraction, S. 37–42.

Speer, Robert, Joshua Chin und Catherine Havasi (2017). "Conceptnet 5.5: An open multilingual graph of general knowledge". In: *Thirty-First AAAI Conference on Artificial Intelligence*.

Underwood, William (2015). *Understanding Genre in a Collection of a Million Volumes*. DOI: 10.6084.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in Neural Information Processing Systems*, S. 5998–6008.

Wildberger, KasparNikolaus (1988). *Beates blondes Haar oder Linguistische Aspekte von Schemaliteratur*. Peter Lang Verlag.

Wu, Yonghui et al. (2016). *Google's neural machine translation system: Bridging the gap between human and machine translation*. eprint: arXiv:1609.08144.

Yang, Liu (2007). "An overview of distance metric learning". In: *Proceedings of the computer vision and pattern recognition conference*.

Yang, Zichao et al. (2016). "Hierarchical attention networks for document classification". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, S. 1480–1489.

Young, Tom et al. (2018). "Recent trends in deep learning based natural language processing". In: *IEEE Computational intelligenCe magazine* 13.3, S. 55–75.

Zhang, Tian, Raghu Ramakrishnan und Miron Livny (1996). "BIRCH: an efficient data clustering method for very large databases". In: *ACM Sigmod Record*. Bd. 25. 2. ACM, S. 103–114.

Zimmermann, Hans Dieter (1979). *Schemaliteratur: Ästethische Norm und literarisches System*. W. Kohlhammer Verlag.

Zipf, George K. (1949). *Human Behavior and the Principle of Least Effort. An Introduction to Human Ecology*. Addison-Wesley.

EIGENSTÄNDIGKEITSERKLÄRUNG

Hiermit versichere ich, dass ich die vorliegende Arbeit in allen Teilen selbstständig angefertigt und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt habe. Sämtliche wörtlichen oder sinngemäßen Übernahmen und Zitate sind kenntlich gemacht und nachgewiesen.

_____

Datum, Unterschrift