Word-Embeddings

Eine Methodik zur Analyse von politischen Geschäften des Schweizer Parlaments

Interdisziplinäre Projektarbeit

Lukas Tobler

Effingerstrasse 41c, 3008 Bern

BM-TAL-18M-S3-Mo-2023

Betreuer: Stefan Brenken

Interdisziplinäre Arbeit in den Fächern Mathematik und Geschichte / Politik

Bern, 20.04.2024

Inhaltsverzeichnis

1 Einleitung

Die Gesellschaft und Politik sind durchgehend in Bewegung, woraus sich gesellschaftliche und politische Gruppierungen bilden, welche sich teils nur leicht, teils jedoch auch stark voneinander unterscheiden. Die Untersuchung der von diesen Gruppen genutzten Sprache bietet Einblicke in die feinen und manchmal deutlichen Differenzen ihrer Ansichten und Überzeugungen. Word-Embeddings sind eine Methodik, um diese sprachlichen Feinheiten zu erfassen und zu deuten. Die Analyse der semantischen Beziehungen zwischen Wörtern in Textkorpora hilft, verborgene Muster zu erkennen und den Wörtern dabei einen einzigartigen Vektor zuzuweisen. Diese Arbeit widmet sich der Nutzung von Word-Embeddings für die Untersuchung von parlamentarischen Texten. Ziel ist, ein grundlegendes Verständnis zu erlangen, wie diese Technologie eingesetzt werden kann, um parlamentarische Dokumente zu analysieren. Die leitende Forschungsfrage lautet:

„Wie können Word Embeddings genutzt werden, um parlamentarische Texte zu analysieren?“

Obwohl es mittlerweile fortgeschrittenere Algorithmen gibt, konzentriert sich diese Arbeit bewusst auf den Word2Vec / Doc2Vec-Algorithmus von Mikolov et al. (2013), da er einen ausgezeichneten Ausgangspunkt für das Verständnis von Word-Embeddings bietet und zudem die Erstellung eines Word2Vec-Modells auch auf weniger leistungsfähigen Computern möglich ist. Diese Arbeit wird zunächst die theoretischen Grundlagen von Word-Embeddings erörtern, gefolgt von einer detaillierten Betrachtung des Word2Vec / Doc2Vec-Algorithmus. Anschließend wird die praktische Anwendung dieser Modelle auf parlamentarische Texte demonstriert. Es wird untersucht, wie Word-Embeddings genutzt werden können, um die Beziehungen zwischen den Parteien und Parlamentariern zu quantifizieren. Des weiteren sollen die Parteien auf populistische Tendenzen untersucht werden. Die Arbeit wird auch die Herausforderungen und Grenzen von Word-Embeddings in der Analyse komplexer Textdaten aufzeigen und mögliche Lösungsansätze vorschlagen. Durch die Kombination von theoretischen Erkenntnissen und praktischen Anwendungen strebt diese Arbeit danach, das Verständnis von Word-Embeddings zur Analyse von parlamentarischen Texten zu verbessern.

2 Hauptteil

2.1 Methodisches Grundkonzept

Alle folgenden Informationen wurden aus unterschiedlichen Quellen zu einen persönlichen Verständnis zusammen gesetzt. Ein grosser Teil basiert jedoch auf den Erläuterungen von Jones (2020).

2.1.1 Begriffsklärung

Damit die folgenden Kapitel gut verständlich sind, werden im folgenden die wichtigsten Begriffe erklärt und eingeordnet.

One-Hot-Vektor

Ein One-Hot-Vektor ist eine Vektorrepräsentation eines Korpus, bei der jeder eindeutige Wert als 0 oder 1 kodiert wird. Werte, die für die Eingabe in ein neuronales Netzwerk ausgewählt werden, erhalten den Wert 1, während alle anderen den Wert 0 haben. Ein Beispiel für einen One-Hot-Vektor ist in Tabelle 2.1 dargestellt.

Eingabeebene

Die Eingabeebene ist die erste Schicht in einem neuronalen Netzwerk, in der die Daten, die verarbeitet werden sollen, eingegeben werden. Die Daten werden typischerweise als One-Hot-Vektoren repräsentiert, die die Dimensionalität des Korpus widerspiegeln.

Gewichtsmatrix

Die Gewichtsmatrix enthält die Gewichte, welche die Verbindungen zwischen den Neuronen in verschiedenen Schichten des neuronalen Netzwerks darstellen. Die Gewichte werden durch einen Lernalgorithmus angepasst und optimiert, um die Genauigkeit des Netzwerks zu verbessern.

Projektionsebene

Die Projektionsebene ist eine Zwischenschicht in einem neuronalen Netzwerk, die zwischen der Eingabe- und der Ausgabeebene liegt. In komplexeren Modellen kann diese Schicht aus mehreren sogenannten versteckten Schichten bestehen, die jeweils eine bestimmte mathematische Funktion ausführen. Man unterscheidet zwischen linearen und nicht-linearen Schichten, je nachdem, welche Art von Funktion verwendet wird.

Ausgabeebene

Die Ausgabeebene gibt die berechneten Werte für die jeweilige Eingabe aus. Die Grösse dieser Ebene hängt mit der Anzahl an möglichen Klassifikationen zusammen. Die Ausgabeebene verwendet eine Aktivierungsfunktion, um die Werte in eine Wahrscheinlichkeitsverteilung zu konvertieren. Die Aktivierungsfunktion kann je nach Problem variieren. Häufig verwendete Funktionen sind die Softmax-Funktion oder die Sigmoidfunktion.

2.1.2 Bisherige Forschungsarbeiten

Die distributionelle Semantik beschäftigt sich mit der Frage, wie man die Bedeutung von Wörtern anhand ihrer Verteilung in Texten erfassen kann. Ein Ansatz, der in diesem Bereich viel Aufmerksamkeit erregt hat, ist die Verwendung von neuronalen Netzwerken, um Wörtern numerische Vektoren zuzuordnen, die ihre semantischen Eigenschaften widerspiegeln. Ein Pionier dieser Idee waren Bengio et al. (2000), die ein neuronales Netzwerk mit einer linearen Projektionsschicht und einer nicht linearen versteckten Schicht vorschlugen, um Wortvektoren aus einem großen Textkorpus zu lernen. Dieser Ansatz war jedoch sehr rechen- und datenintensiv und erforderte viel Zeit und Ressourcen, um gute Ergebnisse zu erzielen.

Um diese Herausforderung zu überwinden, präsentierten Mikolov et al. (2013) das Word2Vec Modell, das eine wesentliche Verbesserung in der effizienten Erzeugung von Wortvektoren darstellte. Der Hauptunterschied zu dem Modell von Bengio et al. (2000) war, dass Mikolov et al. (2013) auf die nicht lineare versteckte Schicht verzichteten und damit die Anzahl der zu lernenden Parameter drastisch reduzierten. Sie stellten zwei Varianten ihres Modells vor, die beide auf dem Prinzip des "predictive learning"basierten. Die erste Variante war das ’Contious bag of word (CBOW)’ Verfahren, bei dem das Netzwerk versuchte, das Zielwort anhand der umgebenden Wörter vorherzusagen. Die zweite Variante war das ’Skip-Gram’ Verfahren, bei dem das Netzwerk versuchte, die umgebenden Wörter anhand des Zielworts vorherzusagen. In beiden Fällen wurde nur ein begrenzter Kontext (zum Beispiel 2 Wörter links und rechts vom Zielwort) aus dem Textkorpus berücksichtigt.

2.1.3 Berechnung

Im folgenden Abschnitt wird das Skip-Gram-Verfahren detaillierter beschrieben, um die Methodik und Anwendung anhand eines konkreten Beispiels zu verdeutlichen. Der verwendete Korpus setzt sich aus einer Sammlung spezifischer Wörter zusammen, die jeweils durch einen eindeutigen One-Hot-Vektor repräsentiert werden. Diese Vektoren dienen als numerische Repräsentation der Wörter und ermöglichen es dem Skip-Gram-Modell, Beziehungen zwischen den Wörtern im Kontext zu lernen.

Wort	One-Hot-Vektor
Der	$\vec{v} = (1, 0, 0, 0, 0, 0, 0)$
Film	$\vec{v} = (0, 1, 0, 0, 0, 0, 0)$
Oppenheimer	$\vec{v} = (0, 0, 1, 0, 0, 0, 0)$
erhielt	$\vec{v} = (0, 0, 0, 1, 0, 0, 0)$
sehr	$\vec{v} = (0, 0, 0, 0, 1, 0, 0)$
gute	$\vec{v} = (0, 0, 0, 0, 0, 1, 0)$
Kritik	$\vec{v} = (0, 0, 0, 0, 0, 0, 1)$

Tabelle 2.1. One-Hot-Vektoren des Beispielkorpus

Die Tabelle zeigt die One-Hot-Vektoren für den Beispielkorpus. Jedes Wort des Korpus wird durch einen Vektor dargestellt. So hätte das Wort 'Oppenheimer' beispielsweise dem Vektor $(0, 0, 1, 0, 0, 0, 0)$

Der Kontext K wird auf ± 2 Wörter festgelegt, was bedeutet, dass für jedes Eingabewort die zwei vorhergehenden und die zwei nachfolgenden Wörter als Kontextwörter betrachtet werden. Die Variable i bezeichnet die Position eines Wortes im Korpus, während j für die Position innerhalb des Kontexts steht. Das erste Wort im Korpus entpricht stets dem Index 0.

Eingabeebene

Die Eingabeebene beim Skip-Gram Verfahren in einem Word2Vec Modell besteht aus der One-Hot-Repräsentation des Eingabeworts. Diese Repräsentation wird anschliessend an die Projektionsebene weitergeleitet, wo sie in einen dichteren Vektor umgewandelt wird. Die Dimensionalität des Eingabevektors ist gleich der Anzahl der einzigartigen Wörter im Korpus. Dies bedeutet bedeutet, dass jeder Vektor die gleiche Dimensionalität hat, die auf der Größe des Korpus basiert. Zum Beispiel wird das Wort 'erhielt' in einem Korpus mit sieben einzigartigen Wörtern als Vektor mit sieben Dimensionen repräsentiert, wobei die vierte Dimension den Wert 1 hätte und alle anderen Dimensionen 0 wären.

$$\vec{v_e} = \begin{bmatrix} d_1 = 0 \\ \dots \\ d_4 = 1 \\ \dots \\ d_7 = 0 \end{bmatrix}$$

Projektionsebene

Die Eingabeschicht führt den Eingabevektor an die Projektionsschicht weiter. Innerhalb dieser Schichten wird die Gewichtsmatrix G_e verwendet, deren Spaltenanzahl den Dimensionen D_K des Eingabevektors V_e entspricht. Deren Zeilenanzahl ist variabel, wodurch die Dimensionalität D_W des Wortvektors V_W bestimmt wird. Üblicherweise werden 100 bis 300 Dimensionen für D_W gewählt, aber im Beispiel werden nur 3 Dimensionen verwendet, was zu einer Gewichtsmatrix der Größe (3×7) führt.

$$G_e = \begin{bmatrix} -1 & -0.8 & -0.6 & -0.4 & -0.2 & 0 & 0.2 \\ 0.4 & 0.6 & 0.8 & 1 & 0.2 & 0.4 & 0.6 \\ 0.8 & 1 & -0.2 & -0.4 & -0.6 & -0.8 & -1 \end{bmatrix}$$

Anfänglich werden die Werte der Gewichtsmatrix zufällig zwischen -1 und 1 festgelegt und während des Trainingsprozesses optimiert.

Die Projektionsschicht eines Word2Vec-Modells ist stets linear und hat die Funktion, den Eingabevektor V_e mittels Matrizenmultiplikation auf die Dimensionalität D_W zu projizieren.

Für das Wort 'erhielt' als Eingabevektor wird beispielsweise folgende Berechnung durchgeführt:

$$G_e \times V_e = \vec{V_W} \begin{bmatrix} -0.4 \\ 1.0 \\ -0.4 \end{bmatrix}$$

Ausgabeebene

Die Projektionsebene gibt den Wortvektor V_W über K Gewichtsmatrizen weiter an die Ausgabeebene. Mithilfe von Matrizenmultiplikationen werden K Vektoren z berechnet, welche D_K Dimensionen aufweisen. Jede Position i im Vektor entspricht einer nicht normalisierten Wahrscheinlichkeit, dass das Wort i an der entsprechenden Position j steht. Im Anschluss werden diese durch die SoftMax Funktion in normalisierte Wahrscheinlichkeiten umgewandelt.

$$G_{a^j} = \begin{bmatrix} 0.54 & -0.72 & 0.98 \\ -0.88 & -0.78 & -0.21 \\ -0.10 & 0.69 & -0.61 \\ -0.01 & -0.75 & -0.41 \\ -0.96 & 0.29 & 0.56 \\ 0.50 & -0.59 & 0.06 \\ 0.11 & 0.52 & -0.97 \end{bmatrix}$$

$$G_{a^j} \times V_W = \vec{V_{a^j}} \begin{bmatrix} -1.328 \\ -0.344 \\ 0.974 \\ -0.582 \\ 0.45 \\ -0.814 \\ 0.864 \end{bmatrix}$$

Die Softmax-Funktion ist eine normalisierte Exponentialunktion mit der Funktionsgleichung

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{n=1}^{D_K} e^{z_n}} \quad f\text{ür } i = 1, 2, \ldots, D_K$$

Daraus resultiert der Ausgabevektor $\vec{V_a}$ an der Position j. Jeder Wert i in diesem Vektor entspricht einem Wert zwischen 0 und 1, der als Wahrscheinlichkeit für das Wort i an der Position i vom Korpus interpretiert werden kann. Daraus schliesst:

$$P(W_k|W_e) = \sigma(W_e)$$

Dabei gilt: W_k = Kontextwörter und W_e = Eingabewort

$$\sigma(V_{a^j}) = \begin{bmatrix} 0.031 \\ 0.082 \\ 0.309 \\ 0.065 \\ 0.183 \\ 0.052 \\ 0.277 \end{bmatrix}$$

Somit wäre die Wahrscheinlichkeit nach dem ersten Durchlauf, dass an der Position j im Bezug zum Eingabewort das Wort 'Oppenheimer' kommt, 30.9 Prozent.

Backpropagation

In Word2Vec-Modellen beginnt der Lernprozess mit zufällig initialisierten Gewichtsmatrizen, was zu ersten ungenauen Ergebnissen führt. Die Methode der Backpropagation, auch bekannt als Fehlerrückführung, dient dazu, die Gewichte systematisch anzupassen, um die Genauigkeit des Netzwerks zu verbessern. Die Verlustfunktion L misst dabei die Abweichung zwischen der tatsächlichen Ausgabe V_a und der gewünschten Ausgabe. Anschließend optimiert die Backpropagation die Gewichte so, dass der Wert der Verlustfunktion minimiert wird. Dieser Prozess wird iterativ durchgeführt, bis das Netzwerk ein festgelegtes Leistungsziel erreicht. Da der genaue Prozess der Backpropagation den Stoff der Berufsmatura übersteigt, wird darauf nicht genauer eingegangen.

2.1.4 Auswertung

Die Verwendung von Wortvektoren ermöglicht eine Vielzahl von Berechnungen, um den zugrunde liegenden Korpus zu analysieren. Diese Vektoren sind nützlich, um die Beziehungen zwischen den Wörtern im gelernten Korpus zu verstehen und zu interpretieren.

Addition / Subtraktion

Die Vektorisierung von Wörtern ermöglicht es, mathematische Operationen wie Addition und Subtraktion auf sie anzuwenden. So kann die semantische Bedeutung eines Wortes durch diejenige eines anderen ergänzt oder reduziert werden, was zur Bildung eines neuen Begriffs führt.

Im gegebenen Beispiel würde daher die Addition von ’Film’ und ’gut’ das Resultat ’Oppenheimer’ liefern.

Ähnlichkeit

Die Kosinus-Ähnlichkeit ist ein Maßstab, der dazu dient, die Ähnlichkeit zwischen zwei Vektoren im Vektorraum zu bestimmen. Im Kontext der Verarbeitung natürlicher Sprache ermöglicht diese Metrik die Berechnung der Nähe zwischen zwei vektorisierten Wörtern. Dieser Ansatz ist besonders nützlich, um die Beziehungen zwischen Wörtern in verschiedenen Textkorpora zu analysieren. Durch den Vergleich der Kosinus-Ähnlichkeitswerte lassen sich die Wörter identifizieren, die in den Korpora am ähnlichsten sind, was wiederum Rückschlüsse auf thematische oder kontextuelle Überschneidungen zulässt.

Clustering

Der KMeans-Algorithmus, auch bekannt als Lloyd-Forgy-Algorithmus, ist ein weit verbreiteter Ansatz in der Clusteranalyse, der darauf abzielt, eine vordefinierte Anzahl von Clustern innerhalb eines

Datensatzes zu identifizieren. Der Anwender bestimmt die Anzahl der Cluster, die dann durch zufällig ausgewählte Zentroide repräsentiert werden. Diese Zentroide dienen als Startpunkte für den Algorithmus, der in iterativen Schritten die Summe der euklidischen Distanzen zwischen den Datenpunkten und den Zentroiden minimiert. Dieser Prozess führt zu einer effektiven Gruppierung der Daten, wobei die Zentroide die Clusterzentren darstellen. Der KMeans-Algorithmus ist besonders nützlich für große Datensätze, da er eine schnelle Annäherung ermöglicht und die Datenstruktur auf eine Weise offenlegt, die sowohl intuitiv als auch visuell zugänglich ist. Allerdings stößt der Algorithmus an seine Grenzen, wenn die Daten keine sphärische Clusterstruktur aufweisen. In solchen Fällen können alternative Methoden wie die hierarchische Clusteranalyse, die auch nicht-sphärische Cluster erkennen kann, herangezogen werden. Diese beginnt mit der Annahme, dass jedes Objekt sein eigenes Cluster bildet. Sie fusioniert diese schrittweise basierend auf ihrer Ähnlichkeit, bis alle Objekte in einem einzigen Cluster vereint sind. Während KMeans für seine Effizienz bei großen Datensätzen bekannt ist, bietet die hierarchische Clusteranalyse eine flexiblere Herangehensweise an die Clusterbildung (DataScientest, 2022).

2.2 Vor- und Nachteile

Ein wesentlicher Nachteil des Word2Vec-Algorithmus liegt in seiner Unfähigkeit, unterschiedliche Bedeutungen desselben Wortes zu unterscheiden. Wörter wie ’Bank’, die mehrere Bedeutungen haben können, erhalten im Modell nur einen einzigen Vektor, unabhängig von ihrem Kontext. Dies bedeutet, dass der Algorithmus bei Wörten mit mehreren Bedeutungen die Nuancen und Kontextunterschiede nicht adäquat berücksichtigen kann.

Auf der anderen Seite steht der signifikante Vorteil des Algorithmus in Bezug auf seinen effizienten Umgang mit Ressourcen. Word2Vec benötigt vergleichsweise wenig Daten und Rechenkapazität für das Training eines effektiven Modells. Diese Effizienz macht Word2Vec besonders attraktiv für Einzelpersonen oder Projekte mit begrenzten Ressourcen. Es ermöglicht eine effektive Analyse und Verarbeitung von Sprache, auch ohne den Einsatz umfangreicher Datensätze oder leistungsstarker Rechensysteme.

2.3 Analyse parlamentarischer Daten

In der folgenden Analyse soll gezeigt werden, wie ein Word2Vec angewandt werden kann, um parlamentarische Daten auszuwerten. Es ist nicht das Ziel belastbare, Ergebnisse zu liefern. Es geht vielmehr darum ein Anwendungsbeispiel mit verschiedenen Auswertungsmöglichkeiten aufzuzeigen.

Im Rahmen der Untersuchung wurden zwei zentrale Forschungsfragen entwickelt:

1. Lässt sich die Parteizugehörigkeit von Parlamentariern anhand ihres legislativen Verhaltens bestimmen?
2. Welche Ähnlichkeiten weisen die Parteien zu populistischen (Rooduijn & Pauwels, 2011) Aussagen auf ?

Diese Fragen zielen darauf ab, die politischen Muster und Tendenzen zu ergründen, die das Handeln von Abgeordneten im parlamentarischen Kontext prägen.

Alle verwendeten Python-Scripts sind auf GitHub [1] zugänglich.

2.3.1 Datengewinnung und Reinigung

Die Datengrundlage für diese Analyse bildete der offene Zugang zu den Parlamentsdaten [2], wo alle Geschäftsdaten ab der 43. Legislaturperiode abrufbar sind. Diese wurden im .json Format abgerufen und heruntergeladen. Anschliessend wurden Satzzeichen, Zahlen und Sonderzeichen entfernt, um diese mithilfe des Treetaggers [3] zu taggen und lemmatisieren. Da die Partei der Person, die das Geschäft eingereicht hat, nicht auf aufgeführt war, musste diese zusätzlich mithilfe des Personenregisters und Personennummer hinzugefügt werden. Zusätzlich wurden nur Fragen, Anfragen, Interpellationen, Motionen und Postulate miteinbezogen, die von einer einzelnen Person eingereicht wurden. Ausserdem wurden nur Geschäfte von den sechs grössten Parteien berücksichtigt.

2.3.2 Deskriptive Analyse des Korpus

Um einen ersten Eindruck von der Datengrundlage zu erhalten, wurden erste, einfache Untersuchungen gemacht. So zeigt die Tabelle 2.1, dass die neueren Geschäfte deutlich stärker im Korpus vertreten sind als ältere. Diese Entwicklung lässt sich durch die veröffentlichten Statistiken des Parlamentes bestätigen (Parlamentsdienste, 2023). Die Tabelle 2.2 zeigt die Anzahl der eingereichten Geschäften pro Partei. Darin ist ersichtlich, dass die Polparteien (SP und SVP) am stärksten vertreten sind.

[1] https://github.com/blackrvn/300_IDPA
[2] https://ws-old.parlament.ch
[3] https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

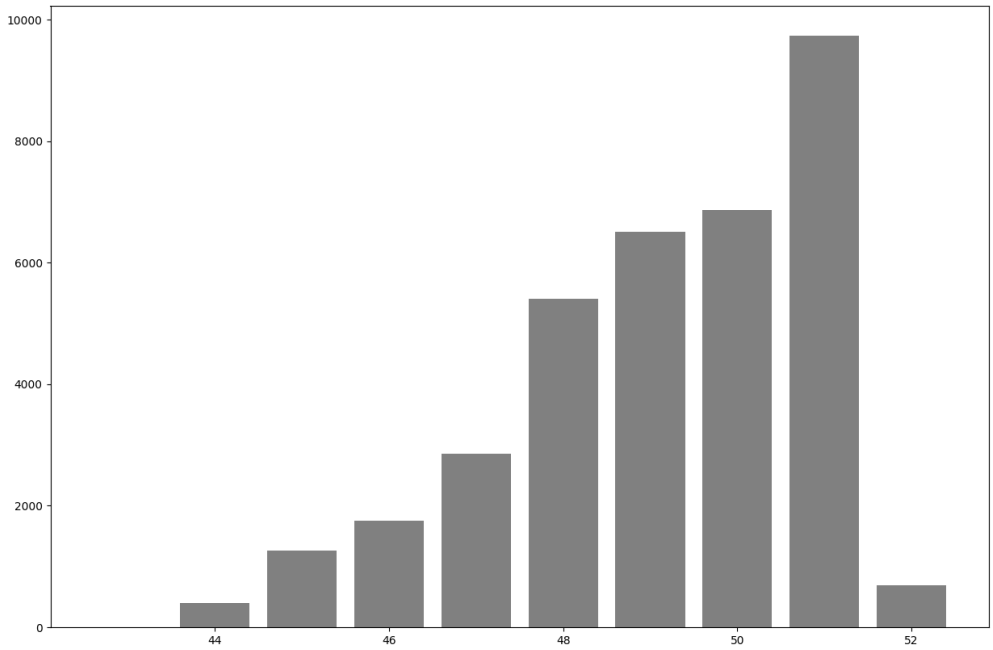

Abbildung 2.1. Anzahl der Geschäfte nach Legislaturperiode

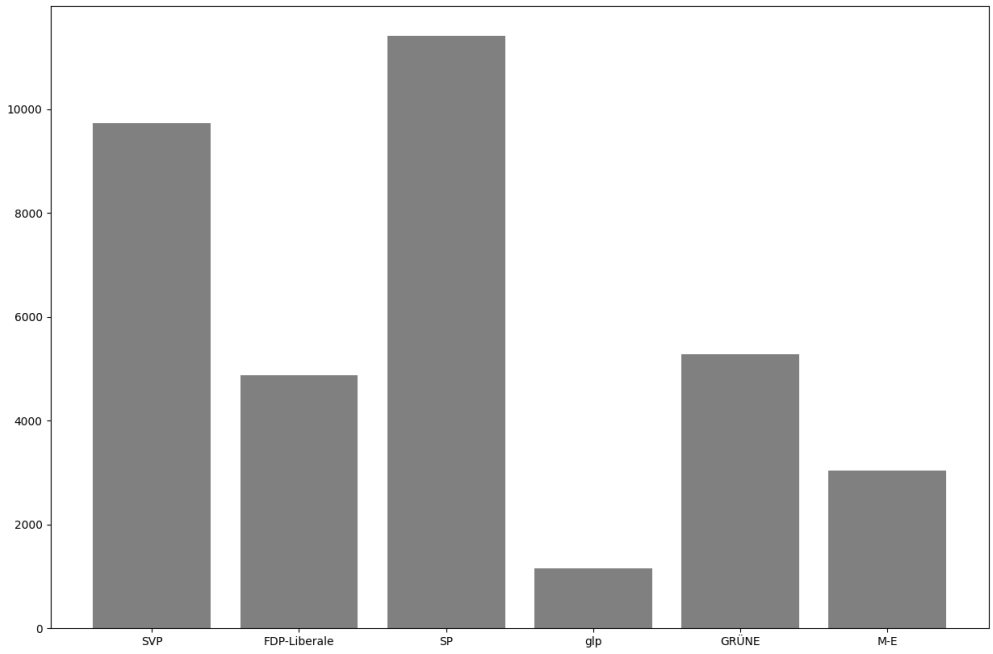

Abbildung 2.2. Anzahl der Geschäfte nach Parteizugehörigkeit

2.3.3 Analyse mit Doc2Vec

In den folgenden Analysen kam das Doc2Vec-Modell (Le & Mikolov, 2014) zum Einsatz, welches eine Erweiterung des im vorherigen Kapitel beschriebenen Word2Vec-Modells ist. Im Gegensatz zum Word2Vec-Modell, das sich auf einzelne Wörter konzentriert, teilt Doc2Vec den Textkorpus in 'Dokumente' auf (zum Beispiel eine eingereichte Motion), wobei jedes Dokument eine eigene Nummer erhält, die hier als 'Label' bezeichnet wird. Dieser Ansatz ermöglicht es, Vergleiche zwischen den Dokumenten effizienter durchzuführen. Für die Umsetzung wurde die Gensim-Bibliothek [4] genutzt. Das Modell wurde mit dem beschriebenen Korpus trainiert. Dabei wurden unterschiedliche Gewichtung der Parteien im Korpus vernachlässigt und nur die sechs grössten Parteien berücksichtigt.[5].

Um die beiden Fragestellungen möglichst effizient beantworten zu können, wurden für jede Parlamentarierin und jeden Parlamentarier und für jede Partei ein Vektor im selben Vektorraum trainiert. Das Modell umfasst somit insgesamt 531 Vektoren für die Parlamentarier und Parlamentarierinnen und 6 Vektoren für die Parteien. Die Hyperparameter[6] , die für das Modell verwendet wurden, sind in Tabelle 2.2 aufgeführt. Sie wurden optimiert, um die Beziehungen zwischen den Parteien und Parlamentarierinnen und Parlamentarier möglichst realistisch abzubilden. Der Hyperparameter 'Trim-Rule' kam zur Festlegung einer Obergrenze für die Wortfrequenz zum Einsatz, wodurch nur solche Wörter im Training berücksichtigt wurden, die weniger als 100 Mal im gesamten Textkorpus auftauchten.

Parametername	Wert
vector_size	70
min_count	5
epochs	30
window	5
hs	1
negative	0
workers	4
dm	0
dbow_words	1
trim_rule	function

Tabelle 2.2. Verwendete Hyperparameter

Die Bedeutung der einzelnen Hyperparameter können in der Dokumentation von Gensim[7] nachgeschlagen werden.

[4] https://radimrehurek.com/gensim
[5] https://www.ch.ch/de/wahlen2023/resultate-der-wahlen/parteistarke/#weitere-informationen
[6] https://datascientest.com/de/hyperparameter-was-ist-das-wozu-dienen-sie
[7] https://radimrehurek.com/gensim/models/doc2vec.html

Erste Fragestellung

Die erste Fragestellung lautet 'Lässt sich die Parteizugehörigkeit von Parlamentariern anhand ihres legislativen Verhaltens bestimmen?'.

Die gelernten Vektoren zeigen aufschlussreiche Muster der Nähe und Distanz zwischen den Parteien, wie in Tabelle 2.3 dargestellt. Diese Muster spiegeln die politischen Ähnlichkeiten wider, die zwischen den Parteien bestehen. Beispielsweise weisen die Vektoren der FDP-Liberalen und der Mitte eine relativ hohe Ähnlichkeit auf, was auf gemeinsame politische Positionen hindeuten könnte. Im Gegensatz dazu zeigen die negativen Ähnlichkeitswerte zwischen den GRÜNEN und der SVP eine deutliche politische Distanz an.

	FDP-Liberale	Grüne	Die Mitte	SP	SVP
FDP-Liberale	-	0.41	0.75	0.42	0.43
Grüne	0.41	-	0.37	0.63	0.26
Die Mitte	0.75	0.37	-	0.5	0.49
SP	0.42	0.63	0.5	-	0.31
SVP	0.43	0.26	0.49	0.31	-
glp	0.46	0.52	0.41	0.27	0.22

Tabelle 2.3. Die berechneten Ähnlichkeiten in Prozent zwischen den Parteien. Grössere Werte weisen auf eine höhere Ähnlichkeit hin.

Die Vektorrepräsentationen der Parteien bildeten die Grundlage für einen Vergleich mit den Vektorrepräsentationen der Parlamentarier, um Einblicke in die Beziehungen zwischen den Parlamentarier / Parlamentarierinnen und ihren Parteien zu gewinnen. Durch die Berechnung der durchschnittlichen Cosinus-Ähnlichkeiten zwischen den Parlamentarierinnen / Parlamentarier und den Parteien konnten diese Beziehungen quantifiziert werden. Die resultierenden durchschnittlichen Ähnlichkeitswerte sind visuell in den Grafiken 2.3 und 2.4 dargestellt. So ist pro Partei (X-Achse) jeweils eine Gruppe von sechs Säulen dargestellt. Eine Säule repräsentiert die durchschnittliche Ähnlichkeit (Y-Achse) der Parlamentarierinnen und Parlamentariern einer Partei zur auf der X-Achse aufgelisteten Partei.

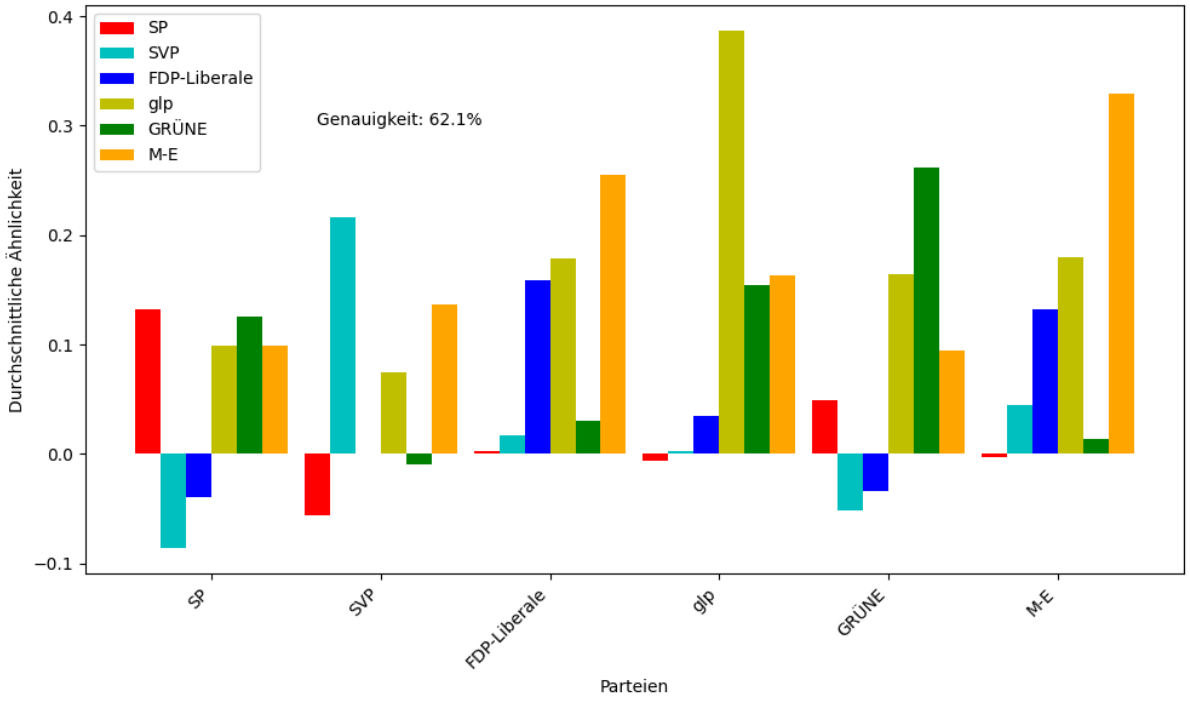

Abbildung 2.3. Ähnlichkeiten zwischen Parlamentarierinnen / Parlamentarier und den Parteien

Eine signifikante Erhöhung der Genauigkeit (prozentualer Anteil der korrekt berechneten Zugehörigkeit) konnte mit der Eingrenzung auf Daten ab der 51. Legislaturperiode (ab Dezember 2019) erzielt werden (Grafik 2.4). Dies ist kurz vor der Coronakrise und markiert eine politische Wende.

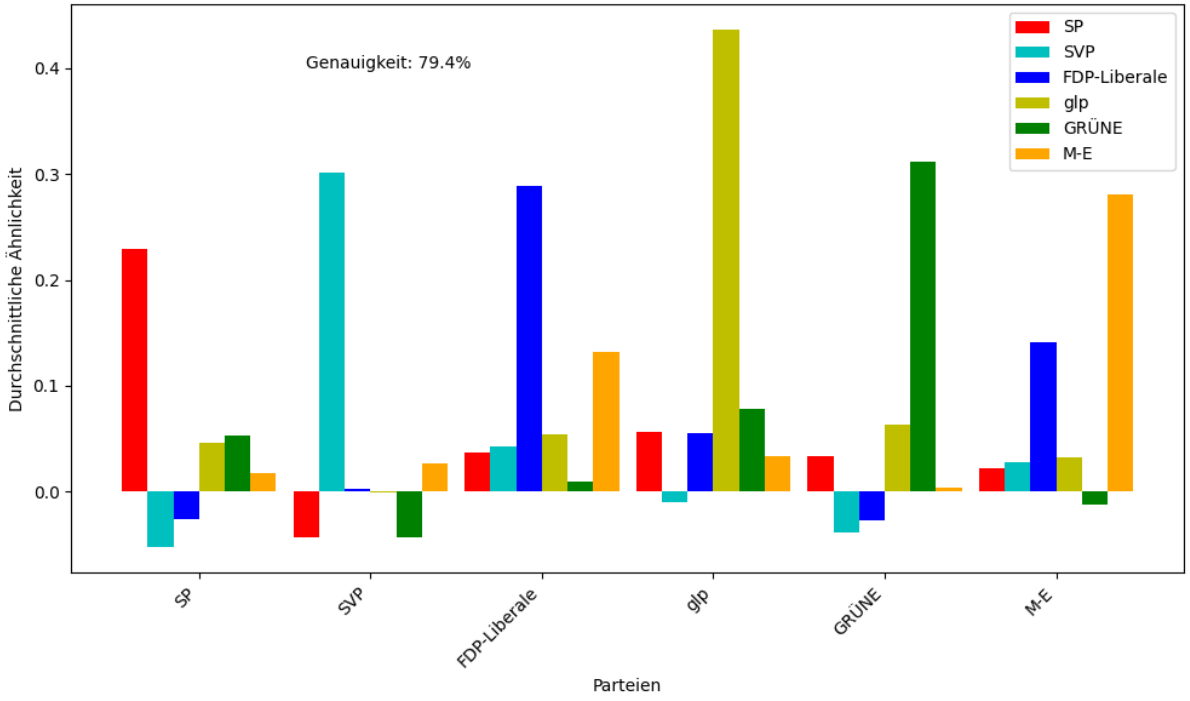

Abbildung 2.4. Ähnlichkeiten zwischen Parlamentarierinnen / Parlamentarier und den Parteien ab Legislaturperiode 51

Zweite Fragestellung

Im Rahmen der Untersuchung der zweiten Forschungsfrage wurden drei populistische Aussagen formuliert, basierend auf der Definition von Rooduijn und Pauwels, 2011.

1. Das Volk muss sich gegen die korrupten Eliten erheben und für die wahre Freiheit kämpfen.
2. Die politische Klasse verrät das Volk, indem sie sich den Interessen der Banker und Geschäftsleute unterwirft.
3. Wir, das Volk, müssen die Wahrheit über die Machenschaften der politischen Propaganda aufdecken.

Um diese Aussagen mit den Positionen der Parteien zu vergleichen, war es notwendig, sie in derselben Weise zu verarbeiten und zu vektorisieren, wie in Kapitel 2.3.1 beschrieben wird.

Daraufhin wurde die Cosinus-Ähnlichkeit zwischen den Vektoren der Aussagen und derjenigen der Parteien berechnet. Die Ergebnisse sind in der Abbildung 2.5 dargestellt. Die Säulen wurden pro Aussage Gruppiert und eine Säule zeigt die Ähnlichkeit zwischen der Partei und der jeweiligen Aussage.

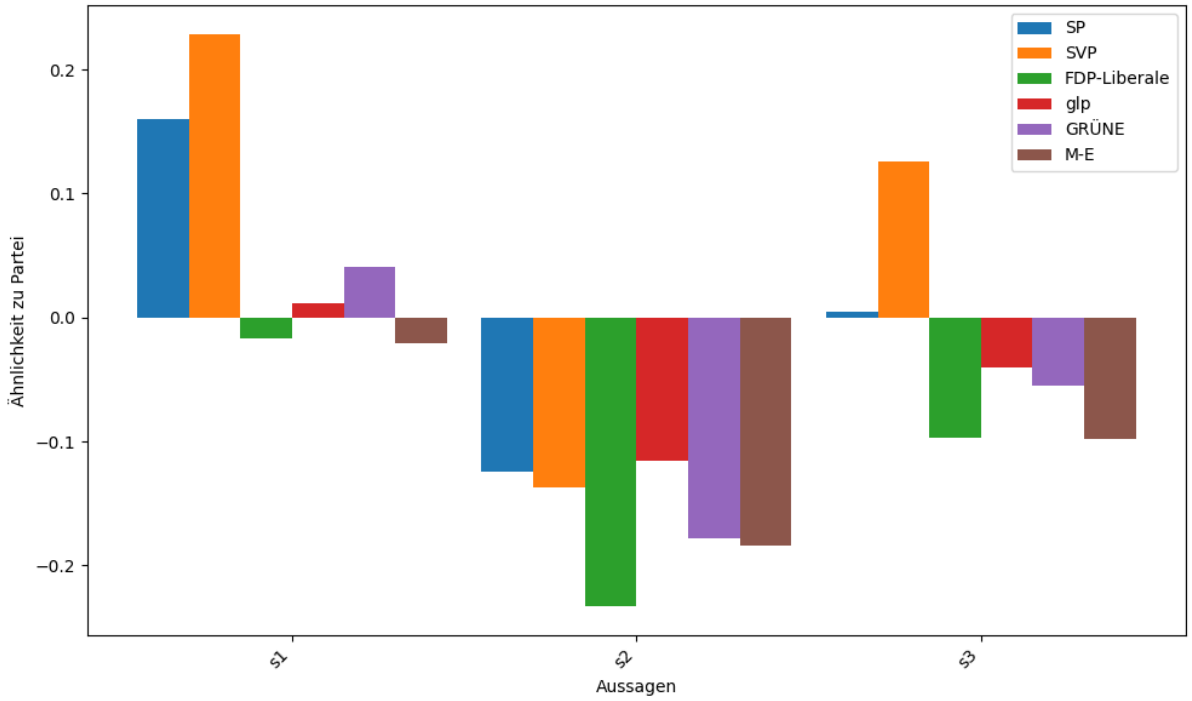

Abbildung 2.5. Ähnlichkeiten der Aussagen zu den Parteien

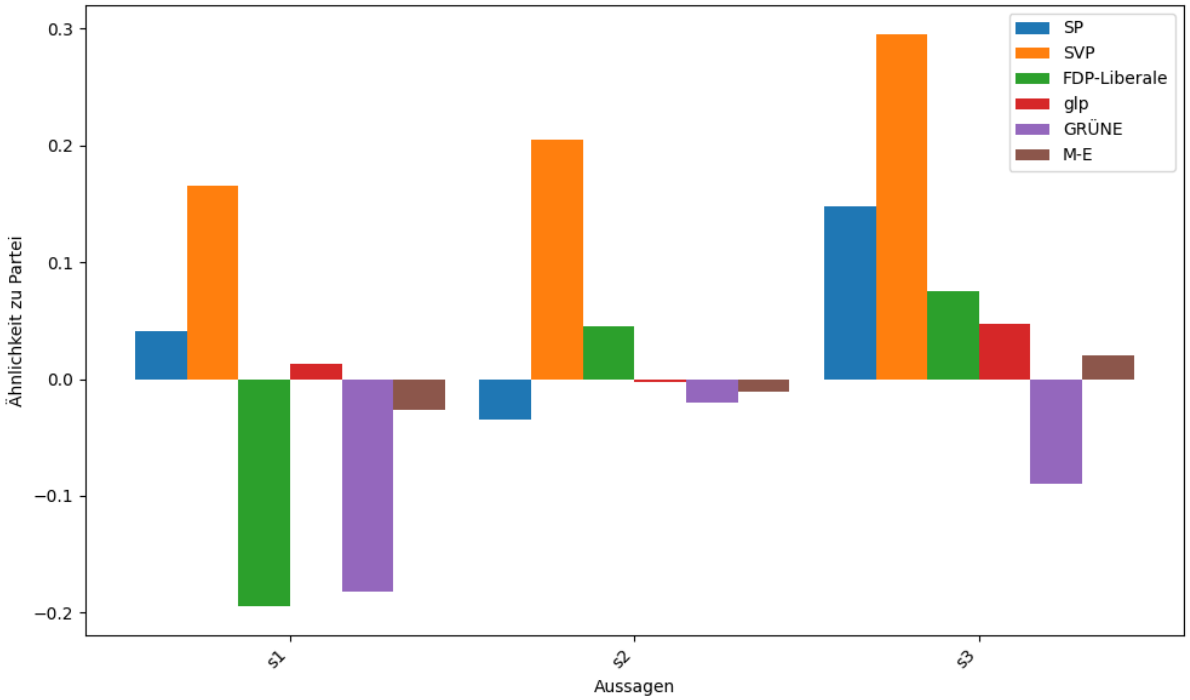

Abbildung 2.6. Ähnlichkeiten der Aussagen zu den Parteien ab Legislaturperiode 51

Besonders auffällig sind die Veränderungen ab der 51. Legislaturperiode. Insbesondere zeigt die SVP eine gesteigerte Übereinstimmung mit der zweiten Aussage, wie in Abbildung 2.6 ersichtlich ist. Zudem ist eine Zunahme der Ähnlichkeit bei den Pol-Parteien feststellbar.

3 Ergebnisse

3.1 Diskussion

Im Kapitel 2.3 wurden zwei zentrale Forschungsfragen formuliert. Die erste Frage zielte darauf ab, zu untersuchen, ob die Parteizugehörigkeit von Parlamentariern durch ihr legislatives Verhalten bestimmt werden kann. Die zweite Frage beschäftigte sich mit den Parallelen zwischen den Parteien und populistischen Aussagen. Dabei wurde die Populismusdefinition von Rooduijn und Pauwels (2011) verwendet. Die angewandte Methodik, die in Kapitel 2.1 beschrieben wird, ermöglichte es, Antworten auf diese Fragen zu finden, allerdings nur bis zu einem bestimmten Grad.

Die Analyse der vorhandenen Daten erlaubte es, politische Gruppierungen zu identifizieren und deren Beziehungen zu quantifizieren. Eine Präzisierung dieser Beziehungen wurde durch die Eingrenzung der Daten auf die Periode ab der 51. Legislatur erreicht. Es ist jedoch anzumerken, dass die Verwendung moderner neuronaler Netzwerke für die Erstellung von Wortvektoren möglicherweise zu robusteren Ergebnissen führen könnte. Solche Modelle könnten in denselben Datensätzen feinere Unterscheidungen treffen, was beispielsweise die Differenzierung zwischen der FDP und der Mitte verbessern würde.

Darüber hinaus lieferte die Untersuchung der zweiten Forschungsfrage Ergebnisse, die Parallelen zu politikwissenschaftlichen Studien aufweisen, wie sie Bernhard (2017) in Bezug auf Populismus durchgeführt hat. Insbesondere die Analyse der populistischen Aussage, dass die politische Klasse das Volk verrät, indem sie sich den Interessen der Banker und Geschäftsleute unterwirft, führte zu abweichenden Ergebnissen, wenn die Daten auf die 51. Legislatur eingegrenzt wurden.

Die Interpretation dieser Ergebnisse und Veränderungen ist jedoch komplex, da die zugrundeliegenden Ursachen vielfältig sein können. Mögliche Erklärungen könnten sein, dass die Daten ab der 51. Legislaturperiode, beginnend im Winter 2019, tatsächlich eine stärkere populistische Tendenz aufweisen. Eine weitere Möglichkeit wäre, dass die Themenvielfalt in dieser Legislaturperiode geringer war, was zu präziseren Gruppierungen führte. Zudem könnte das Fehlen von Begriffen wie 'Geschäftsleute' und 'Banker' im Korpus zu einer Verzerrung der Ergebnisse geführt haben. Diese Aspekte unterstreichen die Bedeutung einer sorgfältigen Auswahl und Aufbereitung der Daten sowie der Wahl der Analysemethoden, um zuverlässige und aussagekräftige Ergebnisse zu erzielen.

3.2 Fazit

Die durchgeführten Analysen und detaillierten Erläuterungen haben es ermöglicht, Unterschiede in den parlamentarischen Daten aufzudecken. Die Anwendung der ausgewählten Methodik wurde dargelegt, wobei die Stärken und Schwächen der Algorithmen Word2Vec und Doc2Vec betrachtet wurden. Es ist anzumerken, dass die Verwendung alternativer Verfahren potenziell zu anderen, möglicherweise präziseren Ergebnissen geführt hätte. Dies unterstreicht die Bedeutung der Methodenauswahl für die Datenanalyse und die Notwendigkeit, verschiedene Ansätze zu berücksichtigen.

3.3 Schlusswort

In den vorangegangenen Abschnitten wurde eine Methodik zur Vektorisierung von Wörtern und Dokumenten vorgestellt, die dann anhand parlamentarischer Daten exemplifiziert wurde. Diese Herangehensweise ermöglichte nicht nur eine detaillierte Erläuterung der Methodik, sondern auch eine kritische Betrachtung ihrer Stärken und Schwächen im praktischen Einsatz.

Es wurde eine bewusste Entscheidung getroffen, die Komplexität der wissenschaftlichen Standards zu reduzieren, um die Methodik anschaulicher zu machen. Diese Vereinfachung war notwendig, da die Zielsetzung der Analysen primär in der Demonstration der Methodik lag und nicht in der Erstellung belastbarer wissenschaftlicher Ergebnisse. Auch die Erläuterung des Word2Vec-Algorithmus wurde auf das Wesentliche beschränkt. Insbesondere wurde der Schritt der Backpropagation nur oberflächlich behandelt, da dieser nicht Berufsmaturastoff gehört.

Dennoch war es überraschend, dass selbst aus einem relativ kleinen Datensatz sinnvolle Analysen abgeleitet werden konnten, was der Arbeit zu einem aussagekräftigeren Gesamtbild verhilft. Darüber hinaus wurde durch die erfolgreichen Analysen das primäre Ziel der Arbeit erreicht: Ein fundiertes Verständnis für Word-Embeddings, basierend auf dem Word2Vec-Algorithmus, zu schaffen. Dieses Wissen bietet einen Einblick in die Möglichkeiten der Textanalyse mittels maschinellen Lernens.

Literatur

Bengio, Y., Ducharme, R., & Vincent, P. (2000). A neural probabilistic language model. *Advances in neural information processing systems*, *13*.

Bernhard, L. (2017). Three Faces of Populism in Current Switzerland: Comparing the Populist Communication of the Swiss People's Party, the Ticino League, and the Geneva Citizens' Movement. *Swiss Political Science Review*, *23*(4), 509–525. https://doi.org/https://doi.org/10.1111/spsr.12279

DataScientest. (2022). *K-means: Fokus auf diesen Clustering Machine Learning Algorithmus* [Zugegriffen: 20.03.2024]. https://datascientest.com/de/was-ist-k-means

Jones, J. (2020). *Word2Vec: Out of the Black Box* [Zugegriffen: 20.03.2024]. https://towardsdatascience.com/word2vec-out-of-the-black-box-a404b4119681

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. *International conference on machine learning*, 1188–1196.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Parlamentsdienste. (2023). *Das Schweizer Parlament in Grafiken* (Techn. Ber.). Das Schweizer Parlament.

Rooduijn, M., & Pauwels, T. (2011). Measuring populism: Comparing two methods of content analysis. *West European Politics*, *34*(6), 1272–1283.

Abbildungsverzeichnis

Tabellenverzeichnis