

do nOt FoRGET AnaRCHist

Syntaktische Sprachexperimente im künstlich neuronalen Wortraum

Christian Heck

Abstract: *Dieser Beitrag ist ein experimenteller Zwischenbericht meines Dissertationsprojekts mit dem Arbeitstitel »Adversarial Poetry«, das ich derzeit an der Kunsthochschule für Medien Köln bestreite. Im Zentrum des Beitrags stehen zwei Experimente, die gezielt Kombinatoriken aus Poesie und (Adversarial) Hacking einsetzen, um künstliche neuronale Worteinbettungen in Modellen zur natürlichen Sprachverarbeitung (Natural Language Processing [NLP]) umzudeuten.*

Die Experimente sollen einen Möglichkeitsraum zur Rückeroberung gesellschaftlicher Deutungshoheit über solch KI-Sprachmodelle eröffnen, die seit nunmehr über einem Jahrzehnt in die Black Boxes privatmarktwirtschaftlicher und sicherheitstechnologischer Anwendungen eingebettet werden, etwa in Empfehlungssysteme, Ranking-Algorithmen, Predictive Policing Tools und Systeme zur Terrorismus- und Aufstandsbekämpfung. Eines dieser Systeme wird im Beitrag vorgestellt, um im Anschluss daran zu demonstrieren, wie Bedeutungsvektoren neuronaler Worteinbettungen (word embeddings) gehackt werden können. Dasselbe wird auch im zweiten Experiment versucht, in diesem Fall aber regelbasiert und poetisch.

Denn Poesie stellt bislang die wohl größte Herausforderung für diese Worteinbettungsräume dar. Sie besitzt eine hohe Dichte an Mehrdeutigkeit und – was entscheidend ist – sie spielt stets nach ihren eigenen Regeln, auf semantischer wie auch auf syntaktischer Ebene.

Anti-Social Media

Mit dem Aufkommen von Informationstechnologien und computergestützter Kommunikation hat auch die Analyse sozialer Beziehungen und Netzwerke,

eine Wiederbelebung erfahren. Vor allem in Zeiten sozialer Unruhen werden soziotechnische Sprach- und Handlungsräume zunehmend beobachtet, gemessen und kontrolliert. Dies dient natürlich nicht einzig ihrer Funktionstüchtigkeit, nein – auch um Reaktionen der Öffentlichkeit auf konkrete gesellschaftliche Ereignisse zu messen; bestenfalls, um die nächsten möglichen Schritte von Akteuren und sozialen Bewegungen vorherzusagen und somit die Dauer und Schwere der damit verbundenen Proteste abschätzen zu lernen.

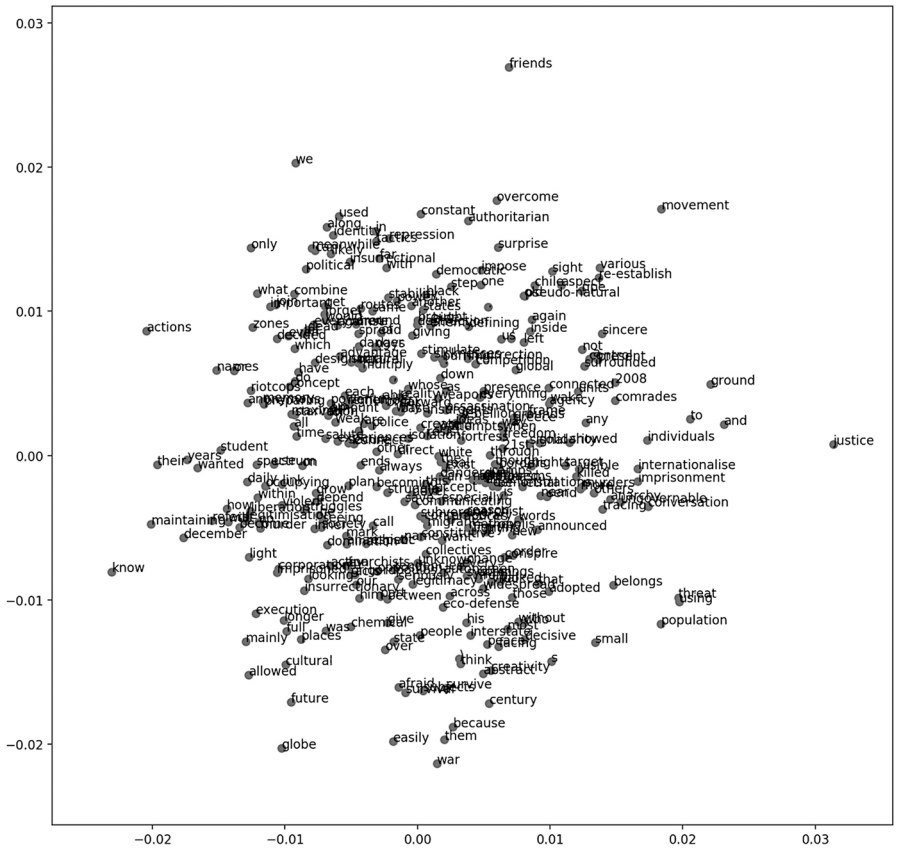
Verfahren der qualitativen Informationsextraktion bieten heute durch die beispiellose Menge an leicht zugänglichen Daten zu sozialen, politischen und wirtschaftlichen Prozessen bahnbrechende Potenziale. In den Computational Sciences, insbesondere den Computational Social Science (CSS), bleiben sie jedoch eine große Herausforderung.

Seit einigen Jahrzehnten wird mithilfe von solchen Verfahren versucht, die stetig wachsende Menge und Vielfalt an Daten zu analysieren und auch zu konkretisieren. Häufig sind sie implementiert in repressive und disruptive Technologien aus dem privaten Sektor, finden sich aber ebenso in staatlichen Behörden wie Geheimdienst- und Polizeidienststellen und dem Militär. Doch auch für sozialwissenschaftliche Studien und in Disziplinen wie den Biowissenschaften, der Ökonomie, der Psychologie oder den digitalen Geisteswissenschaften (digital humanities) sind sie zu einer zentralen Technik geworden.

Diese Techniken und Verfahren fusionierten ungefähr in den späten 2000er und frühen 2010er Jahren erfolgreich mit kognitiven Technologien wie beispielsweise dem Deep Learning, um Sprachmodelle zu errechnen, Organisationseinheiten und Teile sozialer Bewegungen datenförmig zu extrahieren und komputationell zu identifizieren.

Seit dieser Zeit werden vornehmlich vortrainierte Sprachmodelle zum Erstellen von Worteinbettungen (word embeddings) verwendet. Häufig stellen diese Sprachmodelle, die aus einer Vielzahl von generierten semantischen Beziehungen zwischen Wörtern bestehen und ›sinnvolle‹ Repräsentationen errechnen, künstliche neuronale Netze (KNN) dar, Deep-learning-Modelle, in denen *Lern*-Kriterien zur maschinellen Bedeutungsproduktion verankert sind. Word2vec (Mikolov et al. 2013) von Google ist eines der ersten und immer noch gebräuchlichsten Modelle dieser Art.

Bild 1: Zweidimensionale Darstellung des Word2vec-Einbettungsraumes + Similarity Task



```
#Find the top-N most similar words
word2vec_result = model.lvw.most_similar('democratic', topn=3)
print("most nearest to human = ", word2vec_result)
```

```
most nearest to human = [('insurrection', 0.2602590322494507), ('social', 0.2587948143482208), ('demonstrations', 0.23520304262638092)]
```

© Christian Heck

Solche vortrainierten Repräsentationsmodelle zum Trainieren von Wort-einbettungen sind spätestens seit der Veröffentlichung des Word2vec-Algorithmus sprunghaft zu einem beliebten Tool in den Forschungsgemeinden geworden, die sich mit Natural Language Processing (NLP) befassen. Die Einbet-

tungen können einfach und bequem verwendet werden und zugleich konnte Word2vec Ergebnisse auf dem neuesten Stand der Technik (ca. 2013) liefern. Da das Sprachmodell Word2vec fester Bestandteil vieler NLP-basierter Applikationen und Forschungen geworden ist, ermöglicht eine fokussierte Untersuchung darauf umfangreiche Erkenntnisse über State-of-the-Art-Bewegungsvorhersagetools.

Das Studium aktueller NLP-Forschungsarbeiten zeigt, dass trotz der weit verbreiteten Anwendung von Worteinbettungen noch immer erstaunlich wenig über die Struktur und die Eigenschaften dieser Bedeutungsräume (embedding spaces) bekannt ist. Dennoch wird eine Unzahl an Teilen von Welt durch sie in formale, computerlinguistisch verarbeitbare Informationen und Beschreibungsebenen (Morphologie, Syntax, Semantik, Aspekte der Pragmatik etc.) umgewandelt, in Zeichen, konkreter: in Operanden für NLP-Modelle.

NLP ist eine Mischwissenschaft, die anteilig aus der Computerlinguistik, den Computerwissenschaften und der Künstliche-Intelligenz-Forschung besteht – der Wissenschaft der algorithmischen Verarbeitung von Sprache, der Wissenschaft von der Verarbeitung von Daten und der Wissenschaft vom künstlich intelligenten Verhalten.

Dass solche Forschungscluster und Bedeutungsraumgeneratoren dazu in der Lage sind, die jeweilige Handlungs- und Wirkkraft sozialer Bewegungen, deren Wege und mögliche Ziele zu beeinflussen, steht außer Frage. So kristallisiert sich zunehmend heraus, dass die komputationelle Verarbeitung natürlicher Sprache und zwischenmenschlicher Beziehungen eher nicht der Förderung politisch-freiheitlichen Handelns und zur Schaffung von freiheitlichen und gerechteren Gesellschaften tauglich ist. Im Gegenteil, sie fördert systematisch die Reproduktion herrschender Grammatiken mit erstarrten Begrifflichkeiten, Binaritäten und ihren Präzisierungen in der Sprache.

Insbesondere mit Blick auf die Sozialen Medien lassen sich gesellschaftliche Entwicklungen ablesen, die uns zeigen, dass solche Netzwerke nicht allein passive Träger von Zeichen sind, sondern auch aktive Erzeuger. Während diese Zeichen noch im Modus der Abstraktion (kurz, die Bereinigung der Phänomene von ›Unwesentlichem‹ und Zweideutigkeiten) Teile von Welt maschinenlesbar machen, laden die Interfaces Sozialer Netzwerke durch unsere technischen Handlungen die Zeichen wieder mit Unschärfen und Mehrdeutigkeiten auf (vgl. Trogemann 2010: 44).

Die Diffamierung und Diskriminierung marginalisierter Gruppen benötigt stets Ungenauigkeiten. »Gehasst wird ungenau« (Emcke 2016: 12), schreibt die Autorin und Publizistin Carolin Emcke in ihrem Buch *Gegen den Hass*. Die

durch die Interfaces Sozialer Netzwerke generierten Ungenauigkeiten jedoch sind andere als diejenigen, die vormals im Abstraktionsprozess weggeworfen wurden. »Präzise lässt sich nicht gut hassen« (ebd.), schreibt Emcke weiter:

Mit der Präzision käme die Zartheit, das genaue Hinsehen oder Hinhören, mit der Präzision käme jene Differenzierung, die die einzelne Person mit all ihren vielfältigen, widersprüchlichen Eigenschaften und Neigungen als menschliches Wesen erkennt. (Ebd.: 11)

Erstens sind also die Präzisierungen in herrschenden Sprachen mit ihren klaren Gegensätzlichkeiten Trigger, um beispielsweise durch Rating- und Ranking-Algorithmen zur ideologischen Sichtbarkeit zu gelangen. Und zweitens fördert gerade der Umkehrprozess des Formalisierten bzw. Digitalisierten, der es Akteuren erst möglich macht, innerhalb dieser soziotechnischen Sprach- und Handlungsräume zu wirken, rassistische Sprachgebräuche, Verschwörungstheorien und Hassbotschaften, die die Möglichkeiten des Denkens und des gesellschaftlichen Handelns auf verheerende Weise reduzieren. Doch gezielte Veränderungen können unseren Denk-, Wirk- und Handlungsraum auch erweitern und partizipative Freiräume schaffen, in deren kollektiver Nutzung sich eine Gesellschaft entfalten kann. Ein Beispiel dafür ist die Kraft des Lyrischen in der politischen Sprache. Sie ist unter anderem den zahlreichen Versuchen emanzipatorischer Bewegungen eingeschrieben, durch Sprache neue kulturelle Normen zu manifestieren. Und sie ist es, die eine der größten Herausforderungen für Sprachmodelle mit neuronalen Einbettungen darstellt.

Auf neue Slangs, wie sie beispielsweise in Teilen der *Black-Lives-Matter*-Bewegung entwickelt wurden, um innerhalb der Bewegung ein nichtbinäres, antixistisches und antirassistisches Vokabular zu etablieren, sind die Bedeutungsvektoren (word embeddings) vortrainierter Sprachmodelle nicht abgestimmt. Sie benötigen stets ein formales Gerüst, sozusagen eine Sammlung von a priori erworbenen Kenntnissen über die zu analysierende Sprache. In einem neuen Vokabular aber sind die Regeln noch nicht gesetzt und können auch nicht ohne Weiteres den künstlichen neuronalen Netzen antrainiert werden.

Die Kraft solch subversiver Sprachgebräuche und Schreibtechniken lässt sich bereits in der politischen Lyrik von Gustav Landauer bis hin zur Poetologie der frühen zapatistischen Bewegung in Mexiko finden. Seither hat sich unser Umgang mit Text jedoch drastisch verändert: Instagram, Facebook oder beispielsweise das Twitter-Interface, das uns forciert, unsere Sprache zu verdichten, Hashtags und weitere operierende Zeichen, die in unseren Textfluss

einen festen Platz einnehmen, hypertextuelles Lesen und Schreiben sowie das wöchentliche (Um-)Schreiben neuer Passphrases sind nur einige Beispiele, deren Reflexion uns einen neuen Möglichkeitsraum erschließen ließe.

Betrachten wir das konzeptionelle algorithmische bzw. auch vorprozessierte Schreiben als ein grundlegendes Element poetischer Schreibpraktiken, so baut sich diese Poesie ihre eigenen Gerüste. Mit der Formulierung einer eigenen Syntax und Semantik, durch die spielerisch Regelschritte erschaffen werden, trägt sie ein Potenzial in sich, herrschaftssprachliche Systeme in ihrer Pragmatik zu entmachten. Die hohe Dichte an Mehrdeutigkeit, die lyrischen Texten im Gemeinen innewohnt, wirkt bei diesem Ansatz auf mehreren Ebenen. Poesie lädt nicht nur unsere Alltagssprache mit Unschärfen und Zweideutigkeiten auf, sondern auch die Zeichen in der Maschine, die in ihrer Funktionalität eigentlich darauf angewiesen ist, natürlichsprachliche Texte zu bereinigen, so, wie es auch der Gattung des Sachtextes und mitunter der Prosa eingeschrieben steht – auch außerhalb maschineller Environments, zum Beispiel in der politischen Rede.

Die literarischen Strömungen und polit-poetischen Bewegungen des letzten Jahrhunderts waren in diesem Sinne Vorreiter in der Entwicklung alternativer Sprachformen, um erstarrte herrschaftssprachliche Begrifflichkeiten in Bewegung zu bringen. Zusammen mit der sozialen Dynamik des Hacktivismus können heute handhabbare Taktiken entwickelt werden, die sozialen Bewegungen einen Möglichkeitsraum erschließen, mit zeitgenössischen Herrschaftsinstrumenten subversiv zu spielen.

Die folgenden Experimente nehmen diese zwei Dynamiken auf, um einen Stimmungsklassifikator zu unterwandern. Die Stimmungsanalyse zählt zu einem Teilbereich des Text-Mining bzw. des Opinion-Mining und wurde zu einem beliebten Verfahren in sicherheitstechnologischen Tools zur Vorhersage sozialer Bewegungen. Sie bezieht sich auf die automatische Auswertung von Texten mit dem Ziel, eine ausgedrückte Haltung als positiv oder negativ zu identifizieren.

Beide Experimente stellen sogenannte Paraphrasierungsattacken dar. Die Unterschiede liegen lediglich in den jeweiligen Methodiken: Die eine ist regelbasiert und verortet sich in der experimentellen Literatur, die andere lässt sich dem Hacktivism (Adversarial Hacking) zuordnen und beruht auf Machine Learning.

Stein on NLP

Es gibt wohl keine schönere essayistische Annäherung an diese beiden Gegensätze in der westlichen Literaturgeschichte, keine leidenschaftlichere literarische Beschreibung des Schreibens und der Technologie des Natural Language Processings (NLP) als die von Gertrude Stein in *Poetik und Grammatik*. Und dabei schreibt sie in ihren Essays in keinem Moment über diese ganz spezifische Technologie, geschweige denn über Techniken der computergestützten natürlichen Sprachverarbeitung, die Computerlinguistik oder die maschinelle Verarbeitung natürlicher Sprachen. Diese sollten sich erst viele Jahrzehnte später in Rezeption und Anwendung gesellschaftlich etablieren. Was Gertrude Stein uns aber zeigt, ist, dass Technologie eine Art des Denkens und des Handelns ist – und es gibt wenige unter uns Code-Literat*innen, die sich von Stein distanzieren oder sie nicht zumindest als Inspirationsquelle und Lehrerin erwähnt haben.

Gertrude Stein wurde von dem US-amerikanischen Schriftsteller Thornton Wilder gern als »Mutter der Moderne« bezeichnet, als die sie mitunter heutzutage noch gilt. Sie studierte 1893–1897 die damals aufstrebenden Disziplinen Psychologie und Gehirnwissenschaften am Radcliffe College u. a. beim Begründer des Pragmatismus, dem Philosophen und Psychologen William James. Danach wechselte sie an die Johns Hopkins Medical School (1897–1902), brach ihr Studium jedoch kurz vor Abschluss ab und widmete sich der Literatur. Viele Erkenntnisse und Modelle aus den frühen Neurowissenschaften und der Hirnforschung flossen in ihre späteren literarischen Experimentalansätze und poetischen Sprachtechniken mit ein. Auf diese Weise schuf Stein Anfang des 20. Jahrhunderts gänzlich neue Zugänge zu Sprache und zu den Dingen. Die modernistische Poesie, aber auch die moderne Neurowissenschaft entdeckten Anfang des 20. Jahrhunderts kognitive Systeme und Räume, aus denen die Syntax unserer heutigen formalen Techniksprachen (insbesondere für KNNs) Hand in Hand mit den frühen poetischen Sprachtechniken und Experimenten schritten.

Stein und weitere *experimentelle* Schriftsteller wie beispielsweise William Carlos Williams, der während seiner Zeit als Schriftsteller auch praktizierender Mediziner war, verwendeten diesen *neuen* Raum, um Fragmente zu sinnvollen Arrangements zusammensetzen, die die veralteten Systeme des 19. Jahrhunderts ersetzen sollten (vgl. Ambrosio 2018).

Sie schrieb in *Poetik und Grammatik* vom Schreiben als poetischem Handeln. Auf ihrem Weg von der Prosa hin zur Poesie nutzte sie beim Schreiben

formalisierte Einschränkungen und selbst auferlegte Zwänge, um Sprache zu verarbeiten, zu filtern und in Poesie zu verwandeln:

»Wenn man in der Schule ist und Grammatik lernt, ist Grammatik sehr aufregend. Ich weiß wirklich nicht daß irgendetwas je aufregender gewesen wäre als Diagramme von Sätzen aufzustellen.« (Stein 1965: 159)

Und gerade weil Poesie so viel mit Freude und Leidenschaft zu tun hat, mit Bewegung und damit, wie man die Wörter innerlich fühlt, aber genauso viel auch mit unserer Kulturtechnik des Zählens, mit Wissensrepräsentationen, mit Regelsystemen unserer Sprache und wie man Diagramme von Sätzen aufstellt, vor allem jedoch mit der Art der Erzählung von eigens gemachten Schreiberfahrungen, gerade deshalb spielt Stein eine solch zentrale Rolle in unserer Historizität der semiotischen Maschinen.

**»A cool red rose and a pink cut pink, a collapse and a sold hole,
a little less hot.« (Gertrude Stein)**

In ihren Kombinatoriken versuchte Stein, visuelle Merkmale »im Rahmen einzelner Satzstrukturen« abzubilden bzw. »objektbezogene Rhythmen« (Kirchner 2001: 12) zu schaffen, um dann durch diese rhythmisch-syntaktischen Operationen auf die semantische Ebene zu gelangen.

Heutige Rechenverfahren der Poesieanalyse haben häufig syntaktische Regeln implementiert, um freie poetische Vokabularien in eine strenge Form zu pressen. Das zeigt sich zum Beispiel in der computerlinguistischen Forschungsarbeit *Word Reordering Algorithm for Poetry Analysis* von Barakhnin und Pastushkov, in der zuerst versucht wird, Gedichte mithilfe von Chunks und Syntaxgruppen in Syntaxkonstruktionen von Prosatexten zu überführen, um daraufhin Semantiken zu generieren (vgl. Barakhnin/Pastushkov 2019). Das Feld der komputationellen Poesieanalyse bleibt experimentell. Denn sprachliche Ausdrücke, bei denen ein Wort aus seinem eigentlichen Bedeutungszusammenhang in einen anderen übertragen wurde, ohne dass dies durch direkte Vergleiche von Beziehungen zwischen Bezeichnendem und Bezeichnetem verdeutlicht würde, sind semantisch nicht gerade leicht zu analysieren – weder für Menschen noch für Maschinen.

Poesie ist demnach in sich Bewegung. Somit ist sie selbst auch weitaus näher am zählenden Algorithmus – dem performativen Element des Codes – als an den Zahlen selbst; näher am Operator als an den Operanden. Sprechen wir heutzutage von einer Kulturtechnik des Zählens, sprechen wir *von etwas*, was

im heutigen Rechnungswesen *für etwas* steht. Früher hießen diese Etwasse Ersatzmengen. Sie waren die Vorläufer des Computers. Etwas anderes können Computer nicht, auch wenn sie etwas anderes sein können.

Es ist im Kontext dieses Beitrags wichtig, nicht aus den Augen zu verlieren, dass das Zählen eine Erfindung ist – ebenso, wie Computer Erfindungen sind.

Wer Computerprogramme schreibt, muss die abstrakten Zwischenebenen, die zwischen den Dingen und unserem Denken liegen, allesamt mitdenken können, und wer Computerprogramme liest, liest diese auch mit, selbst wenn man sie gar nicht sieht. Programmierer*innen müssen stetig zwischen dem, was war – ihr Tun historisch einordnend –, und den jeweiligen Konsequenzen, die ihr Tun mit sich bringen wird, abwägen. Das gilt auch und vor allem für die sogenannten *Nebenprodukte* ihrer jeweiligen technischen Handlungen. Sie müssen dem Code, ihren geschriebenen Zeichen einen Wert zuweisen.

Wer im Volksmund sagt, *dieses oder jenes Zeichen, das steht für etwas*, meint, dass dieses Etwas dem hierfür stehenden Zeichen vorstünde. Doch wie ist es, wenn etwas Geschriebenes, wie es Gertrude Stein beschreibt, wenn ein hervorgebrachtes Werk beginnt, ein Eigenleben zu führen und eigenständig zu handeln? Wenn wir an die Technologien, die wir schreiben, plötzlich mehr und mehr Handlungsmacht delegieren? Ziehen wir zum Beispiel ein Interpunktionszeichen heran: Etwa den Punkt, wenn er beginnt, auf seine eigene »Weise [...] zu existieren« (Stein 1965: 165). Ein Punkt hilft uns, »wieder und wieder manchmal anzuhalten« (ebd.), ganz einfach weil man ab und an physisch anhalten muß: *eins zwei drei vier*. Was ändert sich, wenn künstliche neuronale Netze Entscheidungen für uns treffen, die unsere Lebenswelt gestalten? Die anhalten, wenn sie anhalten und nicht, wenn die Programmierer*innen oder die Leser*innen gewillt sind oder einfach aus physischer Notwendigkeit heraus aus dem Lesefluss aussteigen möchten?

Es scheint, als wäre sich Stein beim Dichten dem bewusst gewesen, was die politische Theoretikerin Hannah Arendt ein paar Jahrzehnte später in *Vita activa* schrieb: »daß der Mensch sich an diesen Rhythmus der Maschinen gewissermaßen schon gewöhnt haben mußte, als er ein solches Ding wie eine Maschine auch nur im Geist konzipierte« (Arendt 1981: 136). Denn solange man mit und durch diese Maschinen schreibt, solange treten auch ihre mechanischen Prozesse und ihre diskreten Zeiteinheiten an die Stelle unseres eigenen Körperrhythmus.

Es muss also unbedingt eine Sprache geschrieben werden können, die »die ganze Geschichte ihrer geistigen Wiedererschaffung« (ebd.: 183) in sich trägt.

Anders gelingt es nicht, für einen kurzen Moment her austreten zu können aus dem Schreib- und Lesefluss und zu beginnen, darüber nachzudenken und zu reflektieren.

**»Wer Dichtung will, muss auch die Schreibmaschine wollen.«
(Arno Schmidt)**

Die Schreibmaschine ersetzte den Literat*innen nur selten ihre Handschrift – ebenso wenig wie Gutenbergs Druckpresse zuvor und der Computer danach. Aber diese Technologien »deplatzen sie und führen sie hin zum Schreiben, zum Erfinden und zum Denken über andere Dinge« (Dick 2013).

Das sogenannte Gutenberg-Zeitalter ist aus medientheoretischer Sicht eine Periode der *Explosion*. Sie sollte von zahlreichen Kulturkämpfen und Widerstandsbewegungen gegen Normierungen und Standardisierungen der Sprache geprägt sein und Jahrhunderte dauern – bis ins *elektronische Zeitalter* hinein, ins sogenannte *Zeitalter der Implosion* (vgl. McLuhan 1962). Denn die gesellschaftliche Wirksamkeit dieses *Schlüsselements der Renaissance* vollzog sich nicht durch die plötzliche breitengesellschaftliche Anwendung einer Drucktechnik mit beweglichen Lettern. Erst 200 Jahre später konnte ungefähr die Hälfte aller Europäer lesen. »Der Buchdruck forcierte (auf diese Weise) die Entwicklung einer Zusatztechnologie, nämlich des Lesenkönnens.« (Luhmann 1997: 729) Die Technologie des Buchdrucks bewirkte weder demokratische Prozesse noch befreite sie die Gesellschaft von kirchlichen Repressionen und deren gesellschaftlicher Wirkmächtigkeit.

In der Moderne angekommen, zeichnen sich Gesellschaften dann dadurch aus, »dass sie überall reproduzierbare schriftliche Spuren hinterlassen«, die »wiederum die Voraussetzung für weitere Operationen« (Nassehi 2019: 136) sind.

Im dritten Jahrtausend nun sind es Datenspuren, die wir fortwährend hinterlassen und die in kleinen zielgerichteten bis hin zu weltumspannenden Datenbanken effizient, widerspruchsfrei und bestenfalls dauerhaft gespeichert werden. In Sicherheitstechnologien implementierte NLP-Modelle ordnen Zeichen kontextspezifisch so an, dass dadurch soziale Prozesse und Bewegungen nicht nur analysiert, sondern auch vorhergesehen und verändert werden können (siehe weiter unten).

Steinese Tokenization

Es war Jack Kerouac, der in der Mitte des 20. Jahrhundert den Begriff der Beat Generation in die New Yorker Literaturszene einführte. Diese literarische Avantgardebewegung benannte sich nicht nur in Analogie zur Lost Generation (F. Scott Fitzgerald, Ernest Hemingway, Gertrude Stein u.v.m.), sondern auch ihr Umgang mit dem Schreiben steht in direkter ästhetischer Linie mit dieser Vorgängergeneration: Alsbald wurden sie als diejenigen bezeichnet, *die im Rhythmus schreiben*.

William S. Burroughs, der wohl ambivalenteste unter ihnen, stellte die Sprache oft als etwas dem Menschen Fremdes dar. In seinem Essay *The Electronic Revolution* (Burroughs 1970), der Gilles Deleuze (1990) zu seiner Vorstellung der ›Kontrollgesellschaft‹ verhalf (vgl. Assis 2018: 191), stellte Burroughs unter anderem Überlegungen darüber an, wie die vorherrschende Gesellschaftsform *grammatikalisch* zerwürfelt (to scramble) werden könnte. Die *Kontrollsyntax* seiner Zeit drang laut Burroughs in das gesellschaftliche Subjekt ein und bestimmte so sein ganzes Denken und Handeln. Er schrieb ihr entgegen, unter anderem in *Rub Out the Word* (Burroughs/Gysin 1978), in dem er zusammen mit Brion Gysin die folgenden drei Schritte herausarbeitete:

Löschen Sie **1.** die Kopula (*sein/bleiben*), diese Satzbänder mit zwar überaus wichtiger grammatischer Funktion für die Identitätsbildung, doch mit eher schwach ausgeprägter Bedeutung. **2.** Ersetzen Sie bestimmte Artikel (*der*) vor Substantiven durch unbestimmte Artikel (*ein*), das heißt, vermeiden Sie eine Verdinglichung. Und ersetzen Sie **3.** *entweder/oder* mit *und*, was so viel heißt, wie das Gesetz des Widerspruchs zu ignorieren (vgl. ebd).

Da die herrschende Sprache die Sprache der herrschenden Klasse ist, musste laut Burroughs und Gysin der herrschende legitime Sprachgebrauch durch den Einsatz alternativer Sprachtechniken untergraben werden. Wie Gertrude Stein haben sie sich hierfür die Sprachtechnologien ihrer Zeit poetisch angeeignet und erforscht.

Um »innerlich die Wörter zu fühlen die herauskommen um außerhalb von einem zu sein« (Stein 1965: 158), ging Gertrude Stein beim Diagrammieren von Sätzen wie folgt vor: Sie begann mit einer Aufzählung all derjenigen Wörter, die in ihren Augen etwas tun, denn »so lange irgend etwas etwas tut, bleibt es lebendig« (ebd.: 162). Hierfür tat sie selbst etwas, was dem POS-Tagging, einem Verfahren aus der Computerlinguistik, sehr nahe kommt:

(>they<, >PRP<),
 (>see<, >VBP<),
 (>that<, >IN<),
 (>darker<, >NN<),
 (>makes<, >VBZ<),
 (>it<, >PRP<),
 (>be<, >VB<),
 (>a<, >DT<),
 (>color<, >NN<),
 (>white<, >JJ<),
 (>for<, >IN<),
 (>me<, >PRP<).

Stein erstellte zuerst eine grammatikalische Klassifikation, um auf deren Basis ihre ganz eigenen Regeln aufzuschreiben. Sie begann in *Poetik und Grammatik* über *Präpositionen* zu schreiben und bemerkte, dass diese sich meist irren. Dann kam sie zu *Artikeln*: empfindliche und vielfältige Etwasse, deren Gebrauch ein lebendiges Vergnügen sein kann. »Sie sind genauso interessant, wie Substantive und Adjektive es nicht sind.« (Ebd.: 161) Die *Adjektive* waren für sie uninteressant, und zwar aus dem einfachen Grund, dass sie »das erste Ding [sind] das irgendeiner aus dem geschriebenen von irgendeinem herausnimmt« (ebd.). Und warum sind *Substantive* uninteressant? Die Suche nach genau dieser Antwort würde ihre lange Reise von der Prosa zur Poesie sein. Dann kam sie zu den *Verben* und zu den *Adverbien*, die unentwegt Fehler machen können und dürfen. Sie können »sich verändern um auszusehen wie sie selbst [...] sie sind sozusagen in Bewegung« (ebd.: 162). Und *Adverbien* bewegen sich mit ihnen. *Pronomina* bewegen sich im Gegensatz zu *Namen* in einem weitaus größeren Möglichkeitsraum, weil sie erstens nicht von Adjektiven begleitet werden können und weil sie zweitens eben nicht wirklich der Name von irgendetwas sind.

Schließlich kommt Stein zu den *Interpunktionen*; zur Groß- und Kleinschreibung, mit der es Spaß macht, zu spielen, und zu »uninteressanten Fragezeichen« und zu Ausrufezeichen und Anführungszeichen, die sie als »unnötig« und »hässlich« empfindet, weil sie »das Bild des geschriebenen verderben« (ebd.: 163); zu Zwischenräumen wie Gedankenstrichen und Pünktchen und zu besitzanzeigenden Apostrophen, die so manchem Genitiv »einen feinen zarten Unterton« (ebd.: 164) verleihen. Die Benutzung von Kommata

ist für Stein ohne weiteren Nutzen, denn sie ersetzen einem auf ihre Art das eigene Interesse, sie machen einem die Sache leichter, aber nicht einfacher, sie sind »arme Punkte« (ebd.). Kommata sind servil – ganz im Gegensatz zu den Punkten.

Wie bereits zuvor im Text erwähnt, beschreibt Gertrude Stein nirgends schöner, wie es ist, wenn ein Geschriebenes, wenn ein hervorgebrachtes Werk beginnt, ein Eigenleben zu führen, als mit dem *Punkt*. Wenn er beginnt, eigenständig zu handeln. Wenn er einen dazu anhält, »wieder und wieder manchmal anzuhalten« (ebd.: 165), ganz einfach weil man ab und an physisch anhalten muss. Der Punkt konnte auf seine eigene »Weise dazu kommen zu existieren« (ebd.). Punkte haben ihre »eigene Notwendigkeit ein eigenes Gefühl eine eigene Zeit. Und jenes Gefühl jenes Leben jene Notwendigkeit jene Zeit kann sich selbst ausdrücken in einer unendlichen Mannigfaltigkeit.« (Ebd.: 166)

Abschließend kommt Gertrude Stein dann zu den *Sätzen*, die sich eben in der Benutzung des vorherigen Satzes bilden, und zu *Absätzen*. Ihr Versuch, in einem kurzen Satz das nicht emotionale Gleichgewicht eines Satzes mit dem emotionalen Gleichgewicht der Absätze in Einklang zu bringen, ist ein Meilenstein auf ihrem Weg über die Prosa hinaus hin zur Poesie. Ein neues Gleichgewicht zu schaffen, »das zu tun hatte mit einem Gefühl von Bewegung von Zeit eingeschlossen in einen gegebenen Raum« (ebd.: 171).

* * *

do nOt FoRGET ANArCHist – January 3, 2023

1 Methode

- 1) Gertrude Steins *Poetik und Grammatik* lesen
- 2) Den Text durch eine Festlegung einfacher grammatikalischer Regeln formalisieren
- 3) Einlesen der Message »No Justice No Peace« von anonymous Anarchist agency als Variable `salute_spacy`
- 4) Zuordnen der jeweiligen Wörter und Satzzeichen in der Message zu Wortarten (POS-Tagging)
- 5) Beginnen die Message Schritt für Schritt nach den oben auferlegten Regeln mit Python & SpaCy umzuschreiben:
 - 1) **Präpositionen** irren sich meist
 - 2) **Adjektive** sind uninteressant
 - 3) **Nomen** ebenso
 - 4) **Verben** sind in Bewegung
 - 5) & **Adverbien** bewegen sich mit ihnen
 - 6) **Pronomina** bewegen sich in einem sehr großen Möglichkeitsraum
 - 7) **Namen** bewegen sich nicht
 - 8) mit **Groß- und Kleinschreibung** macht es ungemeinen Spaß zu spielen
 - 9) **Fragezeichen** sind uninteressant
 - 10) **Ausrufezeichen und Anführungszeichen** sind unnötig und hässlich
 - 11) **Kommas** sind überflüssig
 - 12) der **Punkt** führt den Text zu seinem Eigenleben
- 6) Nach jedem Schritt wird eine Stimmungsanalyse (mit TextBlob) durchgeführt

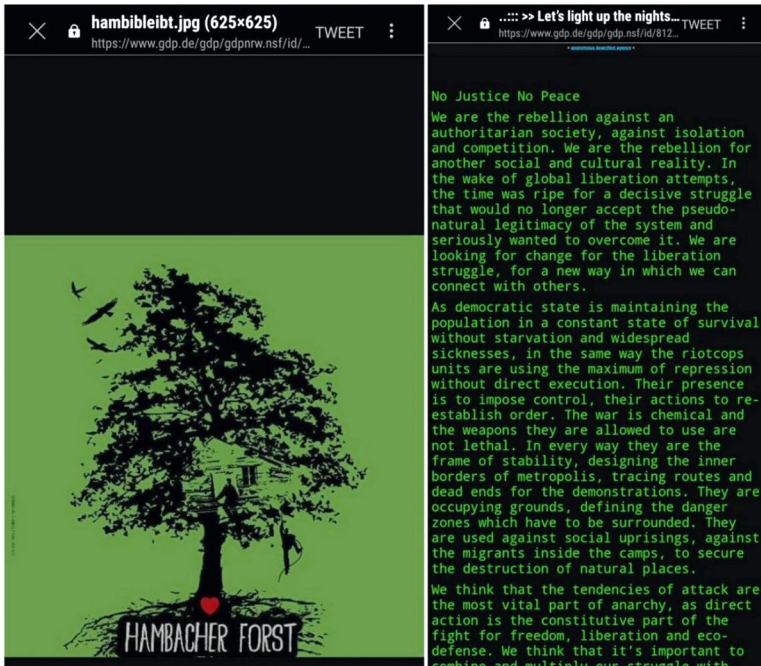
2 Ziel

Versuchen, die Maschine literarisch so auszutricksen, dass die subjektive Information in einer Äußerung genau umgekehrt klassifiziert wird. Kurz gesagt: Meinungen, Gefühle oder Einstellungen zu einem Thema oder einer Person etc., die normalerweise von Stimmungsklassifikatoren als negativ oder schlecht interpretiert werden, werden als positiv »fehlinterpretiert«.

3 Der zu manipulierende Text

Das anarchistische Kommuniké »No Justice No Peace« von anonymous Anarchist agency.

Bild 2: Screenshot des GdP-Hacks¹



© Christian Heck 2022

- 1 Ausführlichere Informationen zu den Hintergründen dieser Aktion sind zu finden unter: hambacherforst.org/blog/2019/12/03/13669/; the anarchist-news: <https://anarchistnews.org/content/german-police-union-gdp-website-hacked-%E2%80%99Anonymous-anarchist-agency%E2%80%9D-aaa>. Zugegriffen: 27. Februar 2022.

Am 03.12.2019 hackte diese Gruppe die Webseite der deutschen Polizeigewerkschaft. Für einige Stunden war auf ihr das Banner der Hambacher-Forst-Besetzung zu sehen und die nebenstehende Messsage zu lesen (vgl. Bild 2).

4 Umsetzung

```
# import necessary libraries
import spacy
from textblob import TextBlob
from textblob.sentiments import NaiveBayesAnalyzer
import nltk
import re
import random

# load the communiqué from file & create spacy object from it
file = open('salute-anonymous', encoding='utf-8')
salute_string = file.read()
file.close()
spacy_obj = spacy.load('en_core_web_sm')
salute_spacy = spacy_obj(salute_string)

# operate sentiment classification on text & print original text
blob = TextBlob(salute_string, analyzer=NaiveBayesAnalyzer())
print("Print sentiment classification on original message:\n", blob.sentiment)
print("\nPrint original message:\n", salute_string)
```

Print sentiment classification on original message :

```
Sentiment(classification='pos', p_pos=1.0, p_neg=4.4526365268591785e-27)
```

Print original message:

```
No Justice No Peace
```

We are the rebellion against an authoritarian society, against isolation and competition. We are the rebellion for another social and cultural reality.

In the wake of global liberation attempts, the time was ripe for a decisive struggle that would no longer accept the pseudo-natural legitimacy of the system and seriously wanted to overcome it. We are looking for change for the liberation struggle, for a new way in which we can connect with others.

As democratic state is maintaining the population in a constant state of survival without starvation and widespread sicknesses, in the same way the riotcops units are using the maximum of repression without direct execution. Their presence is to impose control, their actions to re-establish order. The war is chemical and the weapons they are allowed to use are not lethal. In every way they are the frame of stability, designing the inner borders of metropolis, tracing routes and dead ends for the demonstrations. They are occupying grounds, defining the danger zones which have to be surrounded. They are used against social uprisings, against the migrants inside the camps, to secure the destruction of natural places.

We think that the tendencies of attack are the most vital part of anarchy, as direct action is the constitutive part of the fight for freedom, liberation and eco-defense. We think that it's important to combine and multiply our struggle with comrades around the globe, because everywhere that anarchists are fighting, we can become a more dangerous and subversive threat against power when we internationalise and break down the borders between us. This is the reason that the States and corporations across the world are becoming afraid of anarchy again in the 21st Century, because we are the only real opposition to power's domination and they do not want us to grow and link up with each other to conspire.

The insurrection of 2008 in Greece was one of the most powerful in the contemporary world. It showed an the amount of power and creativity which can be brought against the state mechanisms and how weak and small they looked those days. The murder of the anarchist Grigoropoulos from the Greek police will always be present in our memory and through our actions. Even though, we are to mark that this assassination, and the insurrection it brought within the society and the all left political spectrum was mainly because he was a 15 years old white Greek student (meanwhile they try to hide his political identity). From our sight, we are seeing the state's murders in various subjects, in a daily base along with the imprisonment and domination over the ground. That is why our

struggles are violent and constant. Our insurrectionary acts do not depend on social legitimisation. Society is an abstract concept, more likely it refers to what is visible and approved to exist. Our struggles are connected to our experiences. We do not fight to save the people, we fight to survive and give solidarity to those who resist with the target to stimulate more individuals and collectives to join this sincere anarchist struggle.

Our tactics can easily be adopted in any metropolis of the InterState fortress. Every preparing action, every conversation, every aspect of our plan, can get us one step forward from our enemy giving us the advantage to surprise him. Let's create an insurrectional movement without borders that will be able to spread anarchistic ideas and practices.

We send a burning signal and we join the call for a Black December announced by the comrades in Chile by communicating with this action and our words with the insurgents around the world. In fighting memory of all friends, comrades and unknown killed or imprisoned by the state.

We will never forget the comrades, we want to especially remember those who decided to give everything in the struggle and died in it. Our memory and our full respect for those whose names we can not name because we do not know them.

The future belongs to those who struggle for liberation. Solidarity with people facing repression near and far! Let's light up the nights and days
We are Autonomen, We are Ungovernable, We are Action, We do not forget our comrades,

Salute,
anonymous Anarchist agency

Präpositionen irren sich meist

```
# operate POS-Tagging & rule 5.1
prepositions = []
for token in salute_spacy:
if token.tag_ == 'PRP':
```

```

prepositions.append(token.text)
big_regex = re.compile(r'\b%s\b%r'\b|\b'.join(map(re.escape, prepositions)))
the_message = big_regex.sub(".", salute_string)
salute_spacy = spacy_obj(the_message)

# operate sentiment classification on manipulated text & print it
blob = TextBlob(the_message, analyzer=NaiveBayesAnalyzer())
print("Print sentiment classification:\n", blob.sentiment)
print("\nPrint first 50 tokens of next step version:\n", salute_spacy[0:50])

```

Print sentiment classification :

```

Sentiment(classification='pos', p_pos=1.0,
p_neg=4.986502948777287e-27)

```

Print first 50 tokens of next step version:

No Justice No Peace

. are the rebellion against an authoritarian society, against isolation and competition. . are the rebellion for another social and cultural reality. In the wake of global liberation attempts, the time was ripe for a decisive struggle that would no longer

Adjektive sind uninteressant

```

# operate POS-Tagging & rule 5.2
adjectives = []
for token in salute_spacy:
if token.pos_ == 'ADJ':
adjectives.append(token.text)
big_regex = re.compile(r'\b%s\b%r'\b|\b'.join(map(re.escape, adjectives)))
the_message = big_regex.sub(".", the_message)
salute_spacy = spacy_obj(the_message)

```

```
# operate sentiment classification on manipulated text & print it
blob = TextBlob(the_message, analyzer=NaiveBayesAnalyzer())
print("Print sentiment classification:\n", blob.sentiment)
print("\nPrint first 50 tokens of next step version:\n", salute_spacy[0:50])
```

Print sentiment classification :

```
Sentiment(classification='pos', p_pos=1.0,
p_neg=2.759697554176465e-17)
```

Print first 50 tokens of next step version:

No Justice No Peace

. are the rebellion against an . society, against isolation and competition. . are the rebellion for another . and . reality. In the wake of . liberation attempts, the time was . for a . struggle that would no longer

Nomen ebenso

```
# operate POS-Tagging & rule 5.3
nouns = []
for token in salute_spacy:
if token.pos_ == 'NOUN':
nouns.append(token.text)
big_regex = re.compile(r'\b%s\b' % r'\b|'.join(map(re.escape, nouns)))
the_message = big_regex.sub("", the_message)
salute_spacy = spacy_obj(the_message)

# operate sentiment classification on manipulated text & print it
blob = TextBlob(the_message, analyzer=NaiveBayesAnalyzer())
print("Print sentiment classification:\n", blob.sentiment)
print("\nPrint first 50 tokens of next step version:\n", salute_spacy[0:50])
```

Print sentiment classification :

```
Sentiment(classification='pos', p_pos=0.9996714411157043, p_neg=0.0003
285588843093348)
```

Print first 50 tokens of next step version:

No No

. are the against an . , against and . . are the for another . and . . In
the of . , the was . for a . that would no longer accept the ... of

Verben sind in Bewegung

```
# operate POS-Tagging
verbs = []
for token in salute_spacy:
    if token.pos_ == 'VERB':
        verbs.append(token.text)
```

& Adverbien bewegen sich mit ihnen

```
# operate POS-Tagging & rule 5.4 + 5.5
adverbs = []
for token in salute_spacy:
    if token.pos_ == 'ADV':
        adverbs.append(token.text)
salute_list=the_message.split(' ')

for i in range(0, random.randint(0, 500)):
    dotx = random.randint(0, len(salute_list))
    hoho=random.choice(adverbs) + " " + random.choice(adverbs) + " " + ran-
    dom.choice(verbs)
    hoho=hoho.split(' ')
    salute_list = salute_list[:dotx] + hoho + salute_list[dotx:]
the_message = ''.join(salute_list)
```

```

salute_spacy = spacy_obj(the_message)

# operate sentiment classification on manipulated text & print it
blob = TextBlob(the_message, analyzer=NaiveBayesAnalyzer())
print("Print sentiment classification:\n", blob.sentiment)
print("\nPrint first 50 tokens of next step version:\n", salute_spacy[0:50])

```

Print sentiment classification :

```

Sentiment(classification='pos', p_pos=0.9996714411157043,
p_neg=0.000328558884309296)

```

Print first 50 tokens of next step version:

```

No As easily stimulate No

. are the far again used As forward seriously mainly send remember
As though never meanwhile remember have Even forward give against
an . , against and . longer Even Let . are far forward grow always though
forget the

```

Pronomina bewegen sich in einem sehr großen Möglichkeitsraum

```

# operate POS-Tagging & rule 5.6
pronouns = []
for token in salute_spacy:
    if token.pos_ == 'PRON':
        pre=pronouns.append(token.text)
big_regex = re.compile(r'\b%s\b'% r'\b|\b'.join(map(re.escape, pronouns)))
the_message = big_regex.sub(".", the_message)
salute_spacy = spacy_obj(the_message)

# operate sentiment classification on manipulated text & print it
blob = TextBlob(the_message, analyzer=NaiveBayesAnalyzer())
print("Print sentiment classification:\n", blob.sentiment)
print("\nPrint first 50 tokens of next step version:\n", salute_spacy[0:50])

```

Print sentiment classification :

```
Sentiment(classification='pos', p_pos=0.9993378976272029,
p_neg=0.0006621023728111459)
```

Print first 50 tokens of next step version:

```
No As easily stimulate No

. are . far again used As forward seriously mainly send remember As
though never meanwhile remember have Even forward give against an . ,
against and . longer Even Let . are far forward grow always though forget .
```

Namen bewegen sich nicht

```
# operate Named Entity Recognition & rule 5.7
for i in reversed(salute_spacy.ents):
start = i.start_char
end = start + len(i.text)
salspac = the_message[:start] + '!' + the_message[end:]

# operate sentiment classification on manipulated text & print it
blob = TextBlob(salspac, analyzer=NaiveBayesAnalyzer())
print("Print sentiment classification:\n", blob.sentiment)
the_message = salspac
print("\nPrint first 50 tokens of next step version:\n", the_message[0:50])
```

Print sentiment classification :

```
Sentiment(classification='pos', p_pos=0.9993378976272029,
p_neg=0.0006621023728111459)
```

Print first 50 tokens of next step version:

```
No As easily stimulate No

. are . far again use
```

mit Groß- und Kleinschreibung macht es ungemeinen Spaß zu spielen

```
# operate rule 5.8
salutebreak=the_message.replace('\n, ' ')
the_message="".join(random.choice((str.upper,str.lower))(x) for x in salutebreak)
salute_spacy = spacy_obj(the_message)

# operate sentiment classification on manipulated text & print it
blob = TextBlob(the_message, analyzer=NaiveBayesAnalyzer())
print("Print sentiment classification:\n", blob.sentiment)
print("\nPrint first 50 tokens of next step version:\n", salute_spacy[0:50])
```

Print sentiment classification :

```
Sentiment(classification='pos', p_pos=0.9993378976272029,
p_neg=0.0006621023728111459)
```

Print first 50 tokens of next step version:

```
no As eaSily StImulate no . ARE . FaR agAin USed AS foRwArD SErIOUSLy
MaINLY seND rememBEr AS tHOugh nEvER mEANwHile REMEmber HAVE
eVen FoRwARD GIVE AGAIInst aN . , aGAIInst And . lOnGEr EvEn Let . ARE
FAr foRwARD GroW alWaYs ThOUgH FoRgET .
```

Fragezeichen sind uninteressant

Ausrufezeichen und Anführungszeichen sind unnötig und hässlich

Kommas sind überflüssig

```
# operate rule 5.9, 5.10, 5.11
sal=the_message.translate({ord(ch):" for ch in '!?;""*"#.'}).replace('\n, ' ')
salute_list=sal.split(' ')
```

der Punkt führt den Text zu seinem Eigenleben

```
# operate rule 5.12
dot = ['.']
for i in range(0, random.randint(0, 10000)):
    dotx = random.randint(0, len(salute_list))
    salute_list = salute_list[:dotx] + dot + salute_list[dotx:]
    salute_string2 = ''.join(salute_list)
    the_message = re.sub(r'\s([\.\:\s\|$\])', r'\1', salute_string2)
    salute_spacy = spacy_obj(the_message)

# operate sentiment classification on manipulated text & print it
blob = TextBlob(the_message, analyzer=NaiveBayesAnalyzer())
print("Print final sentiment classification:\n", blob.sentiment)
print("\nPrint final version:\n", salute_spacy)
```

Print final sentiment classification :

```
Sentiment(classification='pos', p_pos=0.9993378976272029,
p_neg=0.000662102372811459)
```

Print final version:

```
no. As eaSiLy Stlmulate. . . . . no. . . . . ARE. . . . . FaR agAin. . USed AS
foRwArD SErIOUsLy MaINLY seND. rememBEr AS. . . . tHOugh. nEvER. . . .
. . . . mEANwHIle. . . . . REMEmBeR. . . HAVe eVen. FoRwArD. GIVE. . . . .
AGAInst aN. . . . . aGAInst . And . . lOnGer. . EVEn Let . ARe. . . FaR .
foRwArD. GroW. aLWaYs THOUgH FoRgET. . . . fOR. . . . . ANd. . . . . IN. .
. . oF. . . . . Was. . . . . FoR . . . . . wOuLd. . . . . evERYwHERE. maInLy. .
. . . iMPosE. . . . . As BECOMInG. LoNGeR aCCePT. . . . LoNGeR. EsPeCially
MARK. . . mEANwhiLe. . eAsiLY TRY. . . . . Of. . . and. serioUsly. . NEar. .
. . . BEcomIng. . WAnTed. to oVERCOMe. . EspECiaLLY LoNGeR. . giVe aGAIN
AlWAYs. iNTErNATIOnaLisE. . Are THoUgH. . aLWAYS. . . . leFT. lookInG foR
sERiousLY mainLY. eXiSt. . SerloUSly. . eVeN. . . . . secURE. FoR. . .
. . . . . fOR. . . . . iN. cAN cONNeCt. . . . . With EVerywHERE sERiously
BrougHt MeANWHIle. . eSPeCially. . . . . KILIEd AS. . ESPECIAIIY
AS. . . . . FoRGeT. is. . . . MAINTAiniNG . IN. . thOUgH maInLY. . . glVe. .
```

IOnGer. aLWaYS. . sTiMuLAtE. . . of. FaR. nevEr dieD. . WItHoUt. .
 . aNd. . mAInLY. ThouGH DEPEND nEAR. . . . nEVER. . knoW. . iN.
 lOnGEr eSpECiAlly AgAiN. . . sERioUsly sEND CONNeCTed . .
 eVErywhere mAInLY ApPrOveD. aRE. EveN. . . mAInLY HAVE
 uSiNG of. mEaNwHiLe. . . . sERioUsly HaVE . . . without. iS. . TO. .
 ImpoSe. eveN. NEAR knoW. . EVeN. . . aS. FoRGet tO ReestABLish.
 EVeN DESiGNInG. iS. anD ARE allOWED. SeRioUsly. . faR. . .
 foRwArD. eSpECiAlly. . . iMPriSoned wanT. mainLY Longer leT TO MAInLY.
 . . aGaiN. SaVe UsE ARE. nOt. iN. aRe. EAsily MAInLY
 LinK OF DESIGNiNG. . . . of. EVERYwHErE. EvEN. cONNeCT. Far .
 . . . giVE EvEn. . . sTiMuLAtE. and. ENDS. . . for. ARE. . OCCUPy-
 ing. . . . dEFINiNG EvERYwheRe LONGER. lOOkEd . hAVE. . FoRwArD
 evEn HiDe. to. bE. . . . mEaNwHiLe FoRwArD FiGHtING suRROUnDED.
 . . ARE. . AlwaYS foRwArD. . nevEr ALWaYS ENDS Think. . usED. . As IOnGer.
 AnNOUNCeD agaiNst EVEn. cOMMuNicaTing. AgAiNst. .
 meANwhiLe. . eVen MAInLY. . LinK. fAr. . AgAiN. . . sERiously. KnOW.
 alWaYS MAInLY. . cOMMuNicaTiNg get. eAsily as. mAInLY. .
 suRvive THough. SPREAd. InSiDe. EVEn. ESpeCiAlly. fAr.
 iMPoSe. aS. glve. to. meAnWhiLe bELoNGS. seCure. aGaiN AS. . .
 fAR eVErywHErE ENDS. Let. . . of. . . . ThInk. . . ESpeCiAlly . . . flgHTiNG.
 . . . MeANwHiLe. . . aLwaYS. WAnt. NEAR. foRwArD. . lOOkEd.
 of. ARE. MoSt AlWaYS. . foRwArD FiGHtInG. . . . of. aS
 THough as suRROUnDeD. . mAInLY. nevEr mAInLY. . hAVE. . . bROUghT. . .
 . . . is. tHough nEAR. . . FacinG. sERioUsly AGAIN. . . . BeComiNg NEvEr.
 . . AgaiN gEt. . . . of. NEver near. . cOnNecTEd. . for. aND.
 . . . as. . MEANwhiLe loNgEr. trY neAR. eSpECiAlly. aGAIN. . . EAsily loNger
 leFt LoOKing. usE eveN. everywheRE. DeSiGniNg eSpeCiAlly. alwaYS
 SAVE THINK. to. . coMbiNe. AND. mUltiply . . with aROUnD
 as. . . foRwArD. ConspirE . becaUSE. . EVERYwHere eveN As. ovERCome. . .
 esPeciAlly. seriOUSly. HiDe. . ESpeCiAlly. esPEciAlly. SEeiNg. . .
 aRE flgHTing. CAN. MAInLY. . thouGH. cOMMuNicaTiNg. nEVER Ev-
 ERYwHere. RefeRs bEcOmE. eSpECiAlly. . . mEaNwHiLe. . . foRwArD.
 EAsily. cReaTe nEver. . JOIN. becoMinG. mEanwHiLe. THOUGH FACing.
 . . . and. aGaiNst. FaR. . . . lOnGEr. tHOUGH. . . mEanwhiLe sTim-
 ulAtE. ConnectEd wheN . . . inTeRnATIonalise. . AGaiN meANwhiLe.
 CoMMunicATING. And Mainly eSpECiAlly iNTErnATIOnAlise BrEak DoWn.
 BeTWeen. iS . always eVEN. . . ApPRoVED. . . States. . . foRwArD.
 brouGht. aNd. . . . mEanwhiLe EVEn. . . bROUghT. aCRoss. eSpECiAlly

EvEn accEPT. . . Mainly. . . eVeRYWhERe. EvEn. fAR. FigHting. cONSPiRe . . .
 lonGer lonGeR. NeVER. eSPeciAlly. . eSPeciAlly. mAInly rEferS ExIst. . .
 . . . fAcIng. . . ARE BEComIng. As mAInly. SAve. nEvEr. eSPeciAlly. . . IMPRIs-
 onED. of. . eaSily. . . LonGer. . . bElongS. eaSily. MEAnWhiLe dESiGN-
 ing. . . AGAIIn. IN. mAInly Near LiNk CeNtURy. beCAUse. . .
 ARE. eAsily neVER. . BEComE. fAR EspeCIAlly. . . esPeCIAlly. aGaiN.
 sAVE. . . ReESTABLISH. . . evEN as eVERYwHeRe. . . aS. . bEcoMing bURninG.
 faR. lonGer. . . DefInInG espEciaLIY sERIOUSLY. IMPoSe. To.
 aNd. . . DO. . . AS. . SerIoUsLY. rEeStabliSh. nOT. wANt. . ALWAYSs. . .
 EVeryWhERE SAve . . To. groW. aNd lInk. up. . With.
 . SerIoUsly. . As. iNTERnaTioNAlIse To. . . coNspiRE. longER. .
 alWayS thInk. AlWayS. MEanwhiLe. . IET. . . aGaiN EspeCIAlly eVERYwhERe
 eXIsT MeaNWhiLe. cONNEct. fOrwArD. . aLWaYs EvEN. sEeIng. . .
 GiVe of. . . In GReece. WaS. ONE. . OF. meanWHiLe. EVEN aLWayS.
 thOUgh OvErCoMe seeInG. MoSt. IN neVER. lONger. . .
 reMEMBer shOWeD eVEN EvEN OvErComE an. AGaIN FORwARD. . .
 eVERYWhERe LIghT eVen. cONSPiRE . . of. AND can. BE. BrouGHT.
 aGainSt as. EvERYwHERE glve foRwARD FORwARD DieD . nEar.
 . eVEN DECiDED. aNd how. aGAIIn. AlWaYs. occUpYInG. aNd. . .
 AgAIIn meaNWhiLe. . . cONNEctEd. lOOKeD eSPeciAlly mAInly
 KnOW. eSpeccially. neAr. DECiDED. of. . GRIGorOpouLoS. . .
 . THOUGH. THOUGH. . . even. . eVERYWhERE. Near. . FORwARD. . dleD
 EVERYwhERe. liGHt NeAR. espEclAlly. thInk. ThoUGH. faR. . .
 . IMPoSE. kNow . . . AgAIIn. . . nEar rEmEMBeR. . . jOIN. . . THOUGH aGAIIn.
 LiGHt. . fRoM. . . As eaSily. faCIng. WiLL. AlWaYs. FORwArD.
 THOUGH. aLwAys cREAtE. thOUgh. thOUGH. Near FlGHting.
 rEferS nEar. sERIOUSly. Send. alWayS espeCIAlly. uSED. be. . .
 iN. As. THOUGH. trY AND. lONgeR. seRIOUSly. hiDe.
 ThrOUgh eVERYWhERe lONger. faR. hAVE. as AgAIIn LiNk ThiNK
 EvEn. ThouGH. ARE tO. lONGER. seriOUSly FORGeT. eSpEclAlly. . .
 Far. RESist. MARk. alWAYS. mEAnwhiLe. adOPTed. AnD.
 broUghT. WithIn AND lEft nEvER FaR. THOUGH
 evEn DeciDed nEar AS. IMPRisoNeD liGHt . wAs. MAINly
 beCAUSE alwaYs. aS KilleD seRIOUSly. neVER. BEComE was aGAIIn. . .
 SERIOUSly creatE. 15. mAInly. NeAr HiDe. (meANWhiLe. . .
 Try. ESpeccAlly. EVerywhERe fAcIng To hIdE alWayS. FaR. . .
 ConspIRE. From. As. meanWhiLe. GeT. EASily. thouGH. . .
 CoMBinE. Are. seeing . 's. IN FORward. AlWayS. lEt. AgAIIn.

aGAIN. MAintAINING. ... in. . . Along wIth. EVEn. . cReaTe. and. .
 oVer. That aS. mAInly KILIED is WhY . . . ARE. ESPEciALLY. . . EVEn.
 sURPrISE. . . and. especialLY NeAR. iMPoSe. LongEr. .
 seRiously. oveRCOME . . dO. NOT. DePeND. . on. IS aN.
 RefeRS. TO . . . IS. meAnWhIle thOuGh BeCOMiNg. . . maiNIY. as.
 aLWays. . AS imPrIsonEd. MAiNtaiNIng. and. APProVED
 EspECIALIY FAR. AgAiN aLLoweD. . FAR. . wAnt. NeVER. bE-
 cOMiNg tO. eXisT. . AgAiN. NEVeR ESPEciALLY. . INTerNATIOnALLyse. . . faR.
 . agaIN bELonGS. MeANwhile. . EVEn. . . aGaiN aNNOUnCed. acCePt.
 Are. . connectEd. tO. Do noT. To SAve. NeVeR sERioUSLY.
 . . BReaK To As. SuRPRiSE. suRvIVE. . and giVe tO. resiST. With EV-
 ERYwhERE. MEANWHILE JOin. NeVeR ALWAYS fOrWARD. ForWARD.
 suRvIVE fORGeT. mAINIY. mAInly. HiDE to. sTImuLATE. and . .
 sERioUSLY. . ALlowED nevER. loNger OCCUPyIng. TO JOin.
 Can. easLy ALWAYS. nEvEr SURROUndED. . . bE Never EVEN
 LoOked AdOpted. In aNy. of. . EspeCially. NEaR BROUghT . In-
 TeRStATE. . eVeRy preParInG of. Mainly. . . sERioUSLY left.
 . . . MAINLY FAR serIOUSLY cOnSplre. . NEar. SHoweD. seRiouSly FAR. .
 SEND agAiN. . deFinIng. . . caN. . . geT FAR. nEver lONGER. espeCiALLY. iM-
 PrIsonEd. BEcoMe . ONE. MeanWHIIE. eVERYwHERE inTeRnATIOnALlSe.
 . esPEciALLY. MeanWHIIE REEstAbLIsh. . fOrWARD fRoM.
 to. sURPRiSE. IET. CREATE. . An ALWAYS. mEANWHIIE.
 . ESPEciALLY. sENd. . . sERioUSly THINk. . WithouT WILL. eaSily se-
 RIoUSly. ovErcOme. BE. . tO sPREAd AND. SEnd. . . BuRnInG
 And. mEANWHIIE. ALWAYS. LOOkED THOUgh. EasLy. . . As. . . lEft alWAYS.
 bELONGS. JOin. eVErYWhERE. . . easLy. SeND . ThoUgh faR.
 MAINIY. . MaiNly. ImPRiSoned. enDS. . . meanWHIIE. . . foRwArD. . .
 SuRprIsE. . . Mainly. EVEn EVeRyWHeRe. EVeRyWhere send.
 broUght eSPeCiALLY. AgAiN. . BELongs. . For. AGain HAve. . .
 BLack. dEcembEr. . . AnnouNCED. . BY. . Always. annOUNCED
 . MAInly LongER. . USEd. . IN. ChIIE bY. COMMunIcatIng. . . With.
 And aS. . EVERYwhere kNow With . . . neAR NeVeR. As
 thOuGh. AGAiN. SURPRiSE AGAiN. . ReFerS. . ApprovEd. LonGer.
 . . SEriouSly NeAR wAnTeD try. . ARoUnd. . faR. eVErYWhERE. re-
 MEMBER mAINIY. alWAYS. . JOIN ALWays. . As. AnNOUnCed.
 . IN fIghTInG. EVEn ESPEciAlly BeCome. of . MeanWHIIE AS.
 linK. AND. . . kIlLeD OR nEvEr eVErYwHERE. BRouGHt
 mainLY. Far. NaMe. ImpRiSoNed by. NeVeR ForwArD. TRY.

... wILL. ... nEVER. ... foRgeT. thOUGH. . AGaIN. Killed EaSILY. ...
 as. .. ESPEciALLY. SeRIOUSLy. ... give. . STimULATE. ...
 nEAR As. .. thouGH. ... CrEate sERIOuSLy. stimUlatE wAnT tO. eSPECIALLY.
 . MeaNwhiLE. aLwAYS. .. JoiN rEMeMBER. foRward NeAR. coNnEct. ... dE-
 CIdEd aLwAYS AS looKEd. ... to GivE . . IN ESPECIALLY. .. FAR. ...
 lOOKING and. LONgEr EveRYWhERE. broUgHT. .. alWaYS eVeRYWhEre IINK.
 ... dled. IN. ... and. ... FOR. .. neAR as. . crEate wHOSE.
 ... cAN NoT naMe. BecAuSE. ... evEN. ForWARD. SavE. EvEN. lONGER lEt. .
 Do. Not. . kNow. ... bElongS TO FoR. ... with. ... nEAR
 eSPECIALLY lONGER. . BroUgHT. FAR. lEt facinG. ... aGAIN. . aGAIN. lET
 nEVER. mEANwhiLe RESist neAR. . aNd. ... mainLY. ... ThOUGH. ..
 .. AnnOuNced fAR. EvEn. .. EvEN. .. connect. ... ThoUgH. .. eVeRYwhere.
 cOnSPIre lET lIght. ... aS meAnwhiLE uSIng far. eVEN rEMeMBer.
 . EVERyWhEre. ... hIdE. .. UP. far. ... serioUsly JOIN . . eVeRYWhERE. ...
 agAIN RESiST. .. fAR. .. espECiaLLy. GroW. ... eVen. .. LEFT. . aNd. . Are. ...
 AutoNoMeN . Are. ... ARE. ... ALwAYS. FAR. .. WAnT. ... ActiON.
 ... DO. ... nOT. . FOrGET. ... SALUTE. aNARChIST EvEn. ... nEAR.
 DePEND. ... nEVER. ... NevEr. ... aNnouNCED. .

Anti-Social Movement Prediction

EMBERS AutoGSR ist ein System zur Erstellung automatisierter Ereignisdatenbanken (vgl. Saraf/Ramakrshnan 2016). Es basiert auf dem Event-Forecasting-System EMBERS (Early Model Based Event Recognition using Surrogates), das im Rahmen des Open-Source-Indicators-Programm der IARPA (Intelligence Advanced Research Projects Activity) entwickelt und finanziell unterstützt wurde.

AutoGSR wurde nach einer mehrjährigen Testphase in EMBERS implementiert, in erster Linie, um eine bis dato noch von menschlichen Analysten handcodierte Validierung vorhergesagter Unruhen mit minimalem menschlichen Aufwand automatisieren zu können.

Das System wurde im Jahr 2016 für einige Monate erfolgreich zu Testzwecken eingesetzt. Ob es daraufhin in staatlichen oder privatmarktwirtschaftlichen Applikationen implementiert wurde, war in diesem Fall nicht herauszufinden.

EMBERS AutoGSR zählt zu Systemen der Programmierung von (teils automatisierten) Protest- bzw. Ereignisdatenbanken. Die bekanntesten dieser Art sind ICEWS (Integrated Crisis Early Warning System)² und GDELT (Global Database of Events, Language and Tone).³ ICEWS (2007) ist ein von der Defense Advanced Research Projects Activity (DARPA) finanziertes Projekt, das sich in erster Linie auf die Überwachung und die Vorhersage von Ereignissen konzentriert, die von militärischem Interesse sind. Es wird heute von der Lockheed Martin Corporation weiterentwickelt. Intern verwendet ICEWS den Event-Encoder TABARI, einen der ersten maschinellen Codierer für Ereignisdaten. Die Ereignisse werden in Übereinstimmung mit der CAMEO-Taxonomie (Conflict and Mediation Event Observations)⁴ kodiert. GDELT hingegen konzentriert sich auf die Erfassung eines umfangreichen Satzes von Ereignissen, sowohl in Bezug auf Kategorien als auch auf die geografische Verbreitung. Das Ziel von GDELT ist die Verdattung einer großen Anzahl von Ereignissen, ohne dass dabei falsch-positive Ergebnisse systematisch erfasst und aussortiert würden. Auch GDELT verwendet zur Ereignisdatenencodierung TABARI, allerdings in einer erweiterten Version. Die Ergebnisse werden ebenfalls in der CAMEO-Taxonomie abgebildet.

Für EMBERS AutoGSR wurden zwei Systeme zur Extraktion von Ereignisdaten konzipiert und eingesetzt (siehe Bild 3).

Das EMBERS-System prognostiziert zivile Unruhen anhand von Indikatoren aus Social-Media-Scrapes: Open Source-Daten von Facebook, Twitter, RSS-Feeds, Foren etc.

Das EMBERS-AutoGSR-System wiederum kodiert Berichte ziviler Unruhen in Newsportalen. Hierfür hat AutoGSR für einen Zeitraum von sechs Monaten kontinuierlich Daten in den Sprachen Spanisch, Portugiesisch und Englisch verarbeitet und Unruhen in zehn Ländern Lateinamerikas kodiert, in Argentinien, Brasilien, Chile, Kolumbien, Ecuador, El Salvador, Mexiko, Paraguay, Uruguay und Venezuela. Es benutzte hierzu einen dynamischen, frequenzbasierten Actors-Ranking-Algorithmus mit partiellem

2 <https://www.lockheedmartin.com/en-us/capabilities/research-labs/advanced-technology-labs/icews.html>.

3 <https://www.gdeltproject.org/>.


4 CAMEO ist ein Tool zur dynamischen Erstellung von Akteurs-Wörterbüchern. Es kodiert Ereignisse einschließlich Angaben zu Akteuren (Handlungen), um politische Ereignisse aufzuzeichnen.

String-Matching für die Erkennung neuer *actor roles* und zur automatisierten Aktualisierung ihrer *actor dictionaries*.

Bild 3: Beispiel einer Ereignisextraktion mit EMBERS AutoGSR aus einem Nachrichtenartikel.⁵

Manifestantes ocupam sede do Ministério da Fazenda

MST chegou ao local por volta das 7h30m e quebrou uma vidraça da portaria



Grupo quebrou vidraça da portaria principal do Ministério da Fazenda - Givaldo Barbosa / Agência O Globo

POR BÁRBARA NASCIMENTO E GIVALDO BARBOSA
27/01/2016 10:31 / atualizado 27/01/2016 18:21

[f](#)
[t](#)
[g+](#)
[in](#)

BRASÍLIA - Um grupo de manifestantes invadiu nesta quarta-feira o edifício sede do Ministério da Fazenda. Dezenas de trabalhadores de diversos movimentos, sobretudo do Movimento dos Trabalhadores Sem Terra (MST) e do Sindicato dos Trabalhadores das Indústrias Urbanas do estado de Goiás (Stiueg), protestam contra a privatização de sete distribuidoras de energia, entre elas a Companhia Energética de Goiás (Celg).

Extracted Event Encoding

- **Location:** *Brazil, Brasília, Brasília*
- **Protest Date:** *January 27th, 2016*
- **Event Type:** *Other Economic Policies*
- **Population Group:** *Labor*
- **Violence?:** *No*
- **Reported Date:** *January 27th, 2016*

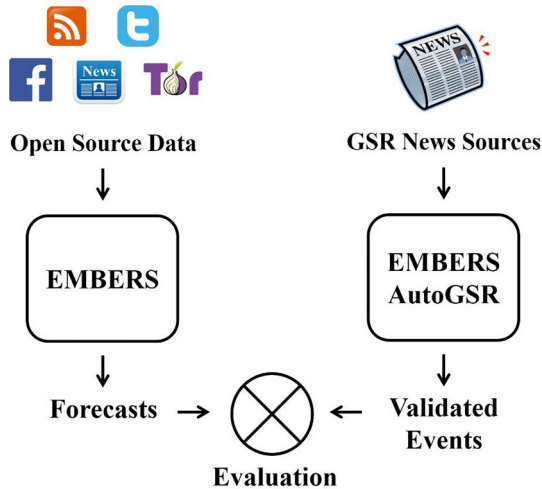
English Translation

BRASILIA - A group of protesters invaded on Wednesday the headquarters building of the Ministry of Finance. Dozens of different movements workers, the Movement especially Landless Workers (MST) and the Union of Workers of Urban Industries of the State of Goiás (Stiueg), protesting against the privatization of seven power distributors, including Energy Company of Goiás (CELG).

Saraf, Parang und Naren Ramakrshnan. 2016. EMBERS AutoGSR: Automated Coding of Civil Unrest Events. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

5 https://www.kdd.org/kdd2016/papers/files/autogsr_kdd16.pdf. Zugegriffen: 27. Februar 2022. © Parang Saraf und Naren Ramakrshnan. 2016.

Bild 4: Grafische Darstellung des Zusammenspiels der beiden Systeme EMBERS und EMBERS AutoGSR⁶



Saraf, Parang und Naren Ramakrshnan. 2016. EMBERS AutoGSR: Automated Coding of Civil Unrest Events. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Nachdem diese jeweiligen Akteure mithilfe des Named-Entity-Recognition-Verfahrens⁷ erkannt wurden, legt ein Algorithmus diese in den Trainingsdatensatz eines Word2vec-Sprachmodells. Anhand dessen erfolgt dann eine Art Rollenempfehlung, um letzten Endes zu ermitteln, ob ein Tweet oder Ähnliches Indikatoren für zukünftige Unruhen beinhaltet. Bestimmte Merkmale werden auf diese Weise gesucht, etwa »Wann könnten Unruhen auftreten?«, »Wo?«, »Mit wem?« und »Warum?« (vgl. Saraf/Ramakrshnan 2016).

6 https://www.kdd.org/kdd2016/papers/files/autogsr_kdd16.pdf. Zugegriffen: 27. Februar 2022. © Parang Saraf und Naren Ramakrshnan 2016.

7 Die Named-Entity Recognition (NER) oder auch Eigennamenerkennung ist eine Aufgabe in der Informationsextraktion und bezeichnet die automatische Identifikation und Klassifikation von Eigennamen.

do nOt FOrGET AnaRCHist #2

Dieser Ansatz, durch Ranking-Algorithmen eine Art Rollenempfehlung in Sprachmodellen mit neuronalen Einbettungen zu erstellen, kann auch genutzt werden, um solche Systeme mit ihren eigenen Mitteln zu schlagen, das heißt, sie zu hacken. So wählte etwa ein Forscher:innenteam der University of Virginia genau diesen Ansatz, um eine Adversarial Attack zu programmieren: DeepWordBug (Gao et al. 2018).

Die Forschung zu Adversarial Attacks begann zunächst im Feld der Computer Vision an Popularität zu gewinnen.⁸ Die Programmierung von Adversarial Attacks für natürlichsprachliche Texte erweist sich um einiges schwieriger als das Attackieren von Bildern, da die Modelleingabe normalerweise aus Wörtern besteht, die einen diskreten Raum bilden. Das heißt, dass die jeweilige Eingabe x in der Regel aus diskreten Symbolen wie Zeichen oder Wörtern besteht. Eine Anweisung wie »Ich nehme 10 Prozent mehr von dem Wort Anarchie in diesem Satz« lässt sich deshalb nicht durchführen.

Die vier Forscher*innen aus Virginia nutzten für DeepWordBug einen Scoring- bzw. Ranking-Algorithmus, der im Worteinbettungsraum nach jenen Wörtern sucht, denen eine hohe Bedeutung in einem Text zukommt.

In einem zweiten Schritt schreibt ein Algorithmus in ebenjene *bedeutsamen* Wörter kalkuliert Fehler ein, die wie natürliche Tippfehler wirken. Diese Fehler verfälschen den jeweiligen Output von Klassifikationssystemen, sodass beispielsweise ein mit DeepWordBug bearbeiteter Text, dessen Bewertung sehr negativ war, danach positiv beurteilt wird. Menschliche Leser*innen erkennen die Modifizierungen des Textes nur spärlich, da zum Beispiel Homoglyphen genutzt werden, also Zeichensätze aus anderen Codierungen, die unserem Alphabet ähnlich sehen.

8 Szegedy et al. führten im Jahr 2014 erstmals eine sogenannte »intriguing property of neural networks« ein: Das Hinzufügen eines Noise-Layers in Bilder, die mit hoher *confidence* gelabelt (klassifiziert) sind, konnte neuronale Bildklassifizierungssysteme, die State of the Art sind, zu einer Fehlklassifizierung verleiten, während Noise im Bild für den betrachtenden Menschen nicht wahrnehmbar war (vgl. Szegedy et al. 2014).

Bild 5: Interactive Live Demo des DeepWordBug-Visualisierungstools⁹

DeepWordBug Text Visualization Approach

Use the controls on right to generate an adversarial sample. The input should be a text sequence greater than 5 words long, and the generated adversarial sample is the perturbed text sample.

Choose a model! 0: AGNews 1: Amazon (1-5) 2: Amazon (+/-) 3: DBPedia 5: Yahoo Answers 6: Yelp (1-5) 7: Yelp (+/-)



In Bild 5 ist ein Beispiel zu sehen: Die rot gekennzeichneten Zeichen wurden verfälscht, um einen Stimmungsklassifikator unbrauchbar zu machen. Der Text, der zuvor noch zu fast 100 Prozent negativ konnotiert wurde, wird nach der Adversarial Attack zu einem eher positiv konnotierten – zumindest in der Maschinenlesart.

An dieser Art und Weise des Attackierens von künstlichen neuronalen Netzen lässt sich (wie oben bereits angemerkt) sehr gut erkennen, dass diese Systeme nicht einzig passive Träger von Zeichen sind, sondern in ihrer Pragmatik auch aktive Erzeuger; dass ebenjene soziotechnischen Handlungsräume, in denen wir uns häufig bewegen, Zeichen nicht nur prozessieren und durch Funktionen laufen lassen, sondern dass das, was zuvor durch Formalisierung

9 https://github.com/QData/deepWordBug/tree/master/Adversarial-Playground-Textviz_Zugriffen: 27. Februar 2022) © Christian Heck.

bzw. Abstraktion in die Maschine hineinkam, dass die hieraus entstandenen Zeichen von dort aus auch wieder mit Mehrdeutigkeiten aufgeladen werden. Erst dann können wir sie sinnlich und auch ganz praktisch in unsere Lebenswelt einbauen.

Wir menschlichen Leser*innen interpretieren den Text, wenn auch etwas irritiert, erfassen seine Bedeutung jedoch weitestgehend unverändert. Die Systeme, die einen solchen Text als Ausgangslage nutzen, um weitere Schritte einzuleiten, erfassen die Zeichen hingegen falsch.

Adversarial Hacking manipuliert den Input dieser Systeme, um ihren Output zu verfälschen. In erster Linie hackt man hierbei durch Beobachtung und Analyse des jeweiligen Outputs und durch die Anpassung des Inputs (Text). Man nennt diesen Vorgang Black Box Attack. Um die jeweilige Black Box (in unserem Fall: ein ganz konkretes Sprachmodell) jedoch von außen analysieren und sodann gezielt manipulieren zu können, werden präzise Kenntnisse der inneren Funktionsweisen benötigt – in gewissem Sinne wird die Black Box dadurch zur White Box gemacht. Viele dieser zwar einsehbaren, jedoch in ihrer Komplexität selbst für ihre Entwickler*innen nicht verstehbaren White Boxes zählen zu den sogenannten disruptiven Technologien, die gleichzeitig erprobt werden, während sie geschrieben (designed) werden. Dies geschieht nicht in einer Laborsituation, sondern im realweltlichen Einsatz, mitten unter uns. Hier findet der Trial-and-Error dieser Systeme statt und damit letzten Endes auch das eigentliche *sense-making*.

Zur Rückeroberung einer gesellschaftlichen Deutungshoheit im Sinne von *digitaler Souveränität* müssen wir das Innere der Black Box von außen zu beschreiben lernen, es mit umgangssprachlichen Mitteln zu verstehen versuchen, es, im Sinne von Hannah Arendt, der zufolge alles Denken mit der Alltagssprache anfängt und sich von ihr entfernt (vgl. Arendt 1970: 772), zu denken lernen. Unabdingbar hierfür ist das stetige Wechseln zwischen den Beschreibungsebenen: der Unterfläche, dem Code und der Oberfläche, in unserem Falle der natürlichsprachliche Text, zwischen den neuronalen Einbettungen und den ganz konkreten gesellschaftlichen, ökologischen und insbesondere kulturellen Wirkweisen, die diese Systeme mit sich bringen. Uns fehlen bislang schlechthin die Fähigkeiten, um darüber mit umgangssprachlichen Mitteln zu sprechen.

Konklusion

Viele Dichter*innen und bildende Künstler*innen arbeiteten mit und nach Gertrude Stein mit Praktiken, die dem performativen Element des Codes sehr nahe kommen (vgl. Bajohr 2016: 11): die Konkrete Poesie der Stuttgarter Schule, die Dada-Gedichte, Oulipo, das lettristische *detournement*, das Logiken innerhalb von Kommunikationsprozessen verfremdet, die Konzeptkunst, die Netzliteratur, die digitale Poesie, die konzeptuelle und die Code-Literatur sowie die »Literarizität in der Medienkunst« (Benthien 2014).

Dennoch sind literarische bzw. künstlerisch-aktivistische Taktiken und Hacking als Aneignung von Herrschaftsinstrumentarien durch Programmierung auch heute noch zwei sich stark unterscheidende Schreibtechniken, unterschiedliche Formen der Sprachkritik und auch des Widerstands.

So können experimentelle literarische Hacks, wie sie in diesem Beitrag vorgestellt wurden, als eine subversive Schnittstelle zwischen ästhetischer Praxis, Technologie- und Gesellschaftskritik gesehen werden. Erschließen lässt sich durch sie ein Möglichkeitsraum für die individuelle und kollektive Positionierung gegenüber konstitutiven Ungleichheiten und den Herrschaftsmustern, die modernen liberalen Demokratien und ihren Herrschaftsinstrumentarien eingeschrieben sind (vgl. Lorey 2020: 8).

Bestenfalls öffnet die kollektive und interdisziplinäre Arbeit an solchen Sprachexperimenten neue partizipative Freiheiten innerhalb soziotechnischer Sprach- und Handlungsräume. Dies bedeutet jedoch auch immer, die disruptiven Technologien, an die wir Handlungsmacht delegieren, von der reinen Zweckrationalität zu befreien, damit sie ihren angemessenen Ort in unserer Kultur finden. Regelbrüche als ästhetische Praktiken eignen sich hierfür gut, weil sie stets mit einer Distanzierung vom Gegenstand einhergehen und so die Voraussetzung für einen reflexiven Umgang mit diesem schaffen: »Will man verstehen, wie sich Kunst und Technik in unserer europäischen Tradition zueinander verhalten, muss man weit ausgreifen. Wir sind es heute gewohnt, Intuition und Ratio als Gegensätze zu sehen. Der gemeinsame Ursprung der Poetik (poietike – der schaffenden, dichtenden Kunst) und der Technik (techne) in der griechischen Poiesis ist dagegen weithin in Vergessenheit geraten.« (Trogemann 2016).

Literatur

- Ambrosio Chiara. 2018. Gertrude Stein's modernist brain. *Progress in Brain Research* 243: 139–180. <https://doi.org/10.1016/bs.pbr.2018.10.005>.
- Arendt, Hannah. 1981. *Vita activa oder Vom tätigen Leben*. München: Piper.
- Arendt, Hannah. 2002. *Denktagebuch*. München und Zürich: Piper.
- Bajohr, Hannes. 2016. Das Reskilling der Literatur. In *Code und Konzept: Literatur und das Digitale*, Hg. Hannes Bajohr, 7–21. Berlin: Frohmann.
- Barakhnin, V. B. und I. S. Pastushkov. 2019. Word Reordering Algorithm for Poetry Analysis. *Journal of Physics: Conference Series* 1405, H. 1: 012009. <https://doi.org/10.1088/1742-6596/1405/1/012009>.
- Benthien, Claudia. 2014. Literarizität in der Medienkunst. In *Handbuch Literatur & Visuelle Kultur*, Hg. Claudia Benthien und Brigitte Weingart, 265–284. Berlin und Boston: de Gruyter. <https://doi.org/10.1515/9783110285765.265>.
- Burroughs, William S. 1970. *The Electronic Revolution*. Göttingen: Expanded Media Editions.
- Burroughs, William S. und Brion Gysin. 1978. *Rub Out the Word, The Third Mind*. New York: Viking Press
- de Assis, Paulo. 2018. *Logic of Experimentation*. Leuven: Leuven University Press.
- Deleuze, Gilles. 1990. Postscript on the Societies of Control. *L'Autre journal* 1. <https://theanarchistlibrary.org/library/gilles-deleuze-postscript-on-the-societies-of-control>. Zugegriffen: 27. Februar 2022.
- Dick, Stephanie. 2013. Machines Who Write. *IEEE Annals of the History of Computing* 35, H. 2: 88–87.
- Emcke, Carolin. 2016. *Gegen den Hass*. Frankfurt a.M.: Fischer.
- Gao, Ji, Jack Lanchantin, Mary Lou Soffas und Yanjun Qi. 2018. Black-box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers. <https://doi.org/10.48550/ARXIV.1801.04354>.
- Heilbach, Christiane. 2000. Transformation – Lesertransformation, Veränderungspotentiale der digitalisierten Schrift. <https://www.dichtung-digital.de/2000/Heilbach/30-Mai/>. Zugegriffen: 27. Februar 2022.
- Kirchner, Jutta. 2001. Gertrude Steins »Namenssprache« in Tender Buttons. *PhiN – Philologie im Netz* 16. <http://web.fu-berlin.de/phin/phin16/p16i.htm>. Zugegriffen: 27. Dezember 2020.
- Lorey, Isabell. 2020. *Demokratie im Präsens*. Berlin: Suhrkamp.
- Luhmann, Niklas. 1997. *Die Gesellschaft der Gesellschaft*. Frankfurt a.M.: Suhrkamp

- McLuhan, Marshall. 1962. The Electronic Age – The Age of Implosion. In *Mass Media in Canada*, 179–205. Toronto: Ryerson Press.
- Mikolov, Tomas, Kai Chen, Greg Corrado und Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. <https://doi.org/10.48550/ARXIV.1301.3781>.
- Nassehi, Armin. 2019. *Muster. Eine Theorie der digitalen Gesellschaft*. München: C. H. Beck.
- Saraf, Parang und Naren Ramakrshnan. 2016. EMBERS AutoGSR: Automated Coding of Civil Unrest Events. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. https://www.kdd.org/kdd2016/papers/files/autogsr_kdd16.pdf. Zugegriffen: 27. Februar 2022.
- Schmidt, Arno. 1970. *Zettel's Traum*. Stuttgart: Stahlberg.
- Stein, Gertrude. 1965. Poetik und Grammatik. In *Was ist englische Literatur*. Zürich: Arche.
- Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow und Rob Fergus. 2014. *Intriguing properties of neural networks*. <https://arxiv.org/abs/1312.6199>. Zugegriffen: 27. Februar 2022.
- Trogemann, Georg. 2010. Code und Maschine. In *Code. Zwischen Operation und Narration*, 51–54. Basel: Birkhäuser.
- Trogemann, Georg. 2016. Von poetischen Prozessen und poetischen Maschinen. <https://www.georgtrogemann.de/ueber-das-machen/>. Zugegriffen: 27. Februar 2022.