Recherche “The Weekly Output”

Ein integraler Teil unseres Projekts ist es Texte verschiedener Arten, sowohl grammatikalisch als auch kontextual möglichst korrekt zu generieren. Der Umsetzung voraus schreitend möchte ich hier näher auf verschiedene Dinge rund um dieses Thema eingehen.

Heutige Programme, welche Texte generieren, sind komplex und ressourcenaufwendig. Sie sind auf Chats wie “Slack”, “Telegram”, “IRC” und vielen mehr zu sehen. Mittlerweile auch ein immer nötigerer Teil bei der Kundenbetreuung (Xu, Liu, Guo + 2017, 1) und finden sogar in Zeitungen (Finley, https://www.wired.com, 2015) schon einen Einsatz. Die Geschichte dieser Art Software ist zwar wie die der Rest der Informationstechnologie kurz aber interessant und wichtig, um zu verstehen was sich, weshalb und in welcher Weise entwickelt hat.

1. Die Geschichte der künstlichen Intelligenz

Noch während der Zeit, in der große Magnetbänder als Speichermedien verwendet und Konstruktionen in Kastengröße als Heimcomputer verkauft wurden, veröffentlicht Alan Turing 1950 “Computing Machinery and Intelligence”. Er schlägt den Turing Test vor und redet sogar bereits über Neuronale Netzwerke. Vorausgestellt veröffentlicht Warren McCulloch “A Logical Calculus Of The Ideas Immanent In Nervous Activity”. Asimov schreibt “I, Robot” und bietet seine “three laws of robotics”.

Kurz nach der Geburt der "Künstlichen Intelligenz" in den späten 1950er Jahren, waren die ersten Bots bereits entworfen und implementiert. Ein paar Jahre später war das Feld schon weiter gewachsen:

> *"One of the first attempts to simulate the behavior of a psychotherapist was in 1966 with a program called Eliza (Weizenbaum, 1976). Eliza acted as a Rogerian therapist, asking the user to explain his/her feelings."* (Epstein, Klinkenberg 2001, 2)

In diesen Chatbots kamen relativ einfache, vorprogrammierte Muster zur Verwendung. Diese waren aber in erstaunlichem Ausmaß effektiv:

> *"Although the goal of the program was to 'demonstrate that the communication between man and machine was superficial' (Nadelson, 1987), Weizenbaum (1976) was surprised to find that people enjoyed using Eliza and actually attributed human-like feelings to the program."* (Epstein, Klinkenberg 2001, 2)

Zu ungefähr der gleichen Zeit konnte Daniel G. Bobrow mit STUDENT einfache Probleme aus Kinderschulbüchern lösen (Bobrow 1964). Zwei große Spieler in diesem Forschungsfeld, Hubert Dreyfus und Joseph Weizenbaum, nahmen zu den Neuerungen Stellung, auch wenn ihre Ansichten mehr akademischer Diskurs waren als breit akzeptiert.

> *"I don't think it distorts their respective stands to say that Dreyfus argued strongly that AI could not be done, but even if it could be, the field was going about it in entirely the wrong way; whereas Weizenbaum argued that AI probably could be done, but should not be, owing to the ways a totalitarian government, such as Nazi Germany's, might abuse it."* (McCorduck 2004, 443)

In den darauf folgenden Jahren fehlten Rechenkraft und Arbeitsspeicher (Crevier 1993, 146-148), "the vodka is good but the meat is rotten" (Russel, Norvig, 2003, 21). Aus einer Innovation wurde ein Spielzeug, der erste "AI Winter" begann. (Russel, Norvig 2003, 24)

Erst in den 80er Jahren wurden "expert systems" in der Industrie eingesetzt.

> *"By 1988, DEC's AI group had 40 expert systems deployed, with more on the way. DuPont had 100 in use and 500 in development, saving an estimated $10 million a year. Nearly every major U.S. corporation had its own AI group and was either using or investigating expert systems."* (Russel, Norvig 2003, 24)

Am Ende der achtziger Jahre forderte DARPA langsam (Defense Advanced Research Projects Agency), die größte Förderagentur der Vereinigten Staaten für die Forschung an Künstlicher Intelligenz, genauere und messbarere Ziele. Somit ging es in den USA nur schrittweise voran, während Forscher im Ausland größere Errungenschaften genossen. (McCorduck 2004, 442)

Die Forschung an künstlicher Intelligenz etablierte sich trotzdem in der Wissenschaft immer mehr. Es war möglich auf einer Variation von Theorien aufzubauen und neue Ergebnisse unterlagen Experimenten sowie Analysen zu ihrer Validität oder Relevanz. Gleichzeitig wurden Jahre an Einsichten in der Mathematik miteinbezogen und implementiert. Spracherkennung, Übersetzung und viele andere Felder der künstlichen Intelligenz profitierten von diesen Entwicklungen. (Russel, Norvig 2003, 25)

Die Verbreitung des Internets brachte die Verfügbarkeit gigantischer Datensätze mit sich. Es konnten Internetseiten zu jeglichen Themen nach Information durchsucht und analysiert werden. Dies ließ jene Formen der künstlichen Intelligenz gedeihen, welche von vorhandenen Daten lernten und oder statistische Analysen durchführten. (Amant, Cohen 1996, 1)

2. Natural Language Processing

2.1 Worttypen und Morphologien

Einige linguistische Prinzipien wurden schon sehr früh in der Verarbeitung natürlicher Sprache verwendet. Verben, Nomen, und Adjektive können beispielsweise in vorgefertigte Vorlagen eingesetzt werden. Unter Berücksichtigung der Morphologien eines Wortes erhält man hier bessere Ergebnisse: Das Lemma, die Stammform eines Wortes, kann mit verschiedenen Attributen modifiziert werden. So können zum Beispiel Wörter in der Mehrzahl, verschiedenen Zeitformen, mit verschiedenen Geschlechtern und anderen Besonderheiten modelliert werden. (Nadeau, Sekine 2017, 8)

2.1 Python

Nach der langen und fast exklusiv von Wissenschaftlern und Philosophen geprägten Geschichte der künstlichen Intelligenz kommt es durch immer stärkere und billigere Computer, dem Internet und dessen Bergen von Information sowie Dokumentation, der Open-Source-Community und immer besseren Cloud-Computing Lösungen zu einem "Boom" an "Hobby-Data-Scientists". Social Media Plattformen werden von

“Bots” überrannt und die neuen User, mit ihren Spenden, geben alten Projekten neue Kraft.

Die Skriptsprache Python, eigentlich entworfen, um sowohl funktionales als auch objektorientiertes Programmieren zu lehren, wird in der Wissenschaft zahlreich verwendet, da es einfach aufgebaut wie auch erweiterbar ist. Gleichzeitig macht es dies für Amateure zugänglich. Ein Ökosystem an Programmierern, Software und wissenschaftlichen Arbeiten entsteht.

2005 wird “NLTK” (Natural Language Toolkit), eines der am weitesten verbreiteten Softwarepakete zur Verarbeitung natürlicher Sprache, nachdem es schon 2001 als Teil eines Computerlinguistik Kurses am “Department of Computer and Information Science” an der “University of Pennsylvania” geschrieben wurde, für Python 2.4 veröffentlicht (Bird, Kline, Loper 2017, XIV Preface). Zusammen mit “NumPy”, einem Tool für schnelle lineare Algebra mit Matrizen, “NetworkX”, für Bäume und Graphen, und verschiedensten Machine-Learning Frameworks wie “scikit-learn” oder “Tensor Flow”, wird es zum Epizentrum des Natural Language Processing in Python.
Zahlreiche dieser Module sind durch Subsets der Sprache wie “Cython” um vielfaches effizienter als normaler Python Code. Seit Python 3 werden diese auch standardmäßig verwendet.

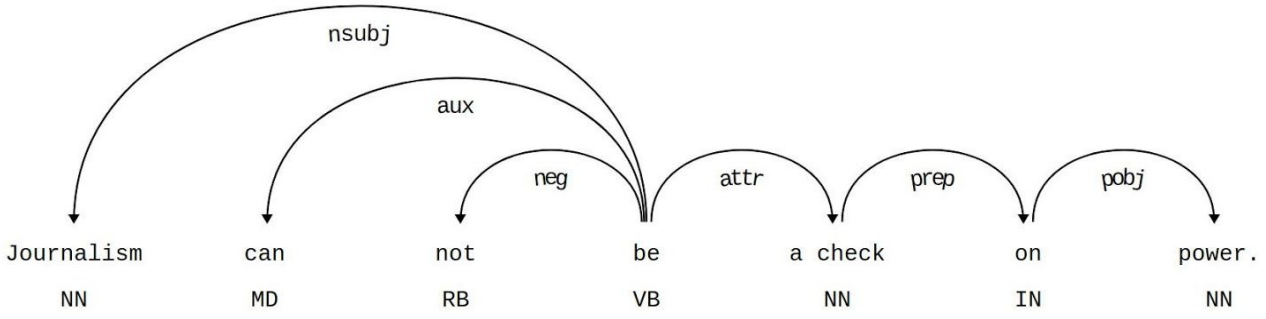
2.2 Entity Recognition

“Entity Recognition” beinhaltet das Erstellen von Metadaten zu Personen, Firmen, Zahlen (Daten, Perzentile, Geld, etc.), Produkten, Orten, manchmal sogar Medikamenten und Chemikalien. (Nadeau, Sekine 2017, 3) Es können: alle Erläuterungen der

Entität Verzeichnet werden (somit kann auch eine Liste an Aliasen erstellt werden), Wikipedia Seiten verlinkt werden und ID Nummern hinzugefügt werden. (Nadeau, Sekine 2017, 11)

2.3 Universal Dependencies

Nachdem 2006 Marie-Catherine de Marneffe die "Stanford Dependencies", eine auf Abhängigkeiten basierende Repräsentation der Englischen Sprache, vorschlägt, werden diese 2014 von einem Team Computerlinguisten (unter anderem ihr selbst) erweitert. Die "Universal Dependencies" (Universelle Abhängigkeiten), ein Modell, das alle Sprachen darstellen können soll, entstehen.



[Abb. 1] Abbildung der Abhängigkeiten in einem Satz.
Die Darstellung beinhaltet den Worttypen und Abhängigkeitstypen.

Das Modell besteht aus Typen von grammatikalischen Abhängigkeiten zwischen Worten. Sie bilden eine Baumstruktur, die von dem "Root Verb" (wörtlich Übersetzt: "Wurzel Verb") ausgeht (siehe Abb. 1). Einzelne Wörter und Phrasen können unter Berücksichtigung der Abhängigkeiten ausgetauscht, entfernt oder erweitert werden um die Aussage des Satzes zu verändern.

2.4 Sentiment Analysis

Bei der "Sentiment Analysis" wird versucht festzustellen, ob etwas mit positiver oder negativer Konnotation erwähnt wird. Meist braucht diese Form der Analyse aber größere Datensätze als sie manuell produzierbar wären. Somit ist diese Analysemethode für Amateure unzugänglich und wird öfter bei Wahlanalysen oder der Marktforschung verwendet.

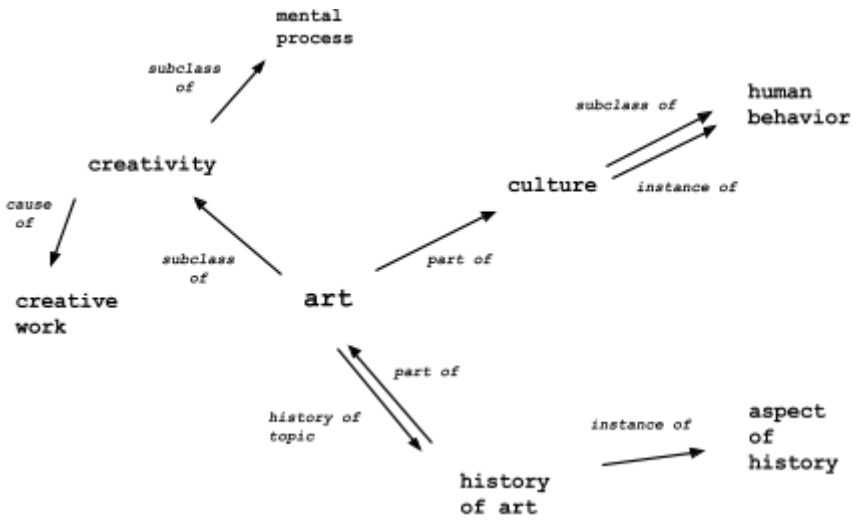
2.5 Derzeitige Softwarelösungen

Neben NLTK (siehe 2.1) sind in den letzten Jahren diverse andere Lösungen aufgesprungen. "Google LLC" führt unter anderem die einzigen kommerzielle Natural Language Processing Cloud-Computing Angebote, welche auch von Einzelpersonen verwendet werden können. SpaCy, ein Python Modul, unterstützt alle oben genannten Analysemethoden, benötigt aber (im Gegensatz zur Lösung von "Google LLC") für die "Sentiment Analysis" eigene Corpora. Es erlaubt eigenen Code in die Analyse einzuschleusen und Neuronale Netzwerke einfach einzubinden.

3 Vorhandene strukturierte Wissens-Akkumulationen

Neben proprietären, geschlossenen Sammlungen an verlinkten Daten, wie "Google LLC" Indexierung des World Wide Web oder IBM Watsons Sammlung an Wissen, entsprangen Wikipedia einige "Semantische Netzwerke" wie "Wikidata" und "DBPedia" welche die Texte und Links der Website in einer Computer freundlichen

Weise aufbereiten. Diese Services verwenden meist graphenbasierte Abfragen-Sprachen wie SPARQL (siehe 2.7.2).



[Abb. 2] Darstellung einiger der Verbindungen in dem Semantischen Netzwerk "Wikidata"

Die dort gespeicherten Daten können dann in einfachen Sätzen von einem Computer ausformuliert werden, oder in einem anderen Prozess berücksichtigt werden. Im Falle eines Programms könnten sie dazu dienen, weitere Informationen über die Subjekte und Objekte in einem Text einzubeziehen.

"Yet Another Great Ontology" ("YAGO"), ein offenes "Semantisches Netz" das seinen Daten sowohl eine zeitliche als auch eine räumliche Achse zuteilt, kombiniert Daten von "Wikipedia", "Word Net" und "Geo Names" um eine der größten Sammlungen an semantisch verlinkten Daten zur Verfügung zu stellen. Alle Einträge – über die mehr als 10 Millionen Entitäten – werden manuell überprüft und genehmigt. Unkomprimiert nimmt diese Sammlung 168 Gigabyte Speicherplatz ein und kann als .tsv (ähnlich wie das CONLL-U Format, siehe 4.1) oder .ttl erhalten werden.

4 Datenspeicherung

Die oben genannten Methode zur Analyse und Sammlung von Daten erfordert es früher oder später, diese zu speichern. Zwei Formen können hier beide ihre eigene Anwendung finden:

4.1 CoNLL-U Format

Dieses Format wird von dem "Universal Dependencies" Projekt vorgeschlagen, es ist eine Erweiterung des CoNLL-X Formats (Buchholz, Marsi 2016, 1). Es scheint dafür entworfen zu sein große Texte zu distribuieren aber nicht zu durchsuchen oder zu verwenden, da es einfach aufgebaut ist und erst in eine Datenbank importiert werden muss um Relationen aufzulösen und es schneller durchsuchen zu können.
Attribute werden ähnlich wie im CSV Format, mit einem Tabulator getrennt aufgelistet (siehe Tab. 1).

```
1	Кучето	куче	NOUN	_	Definite=Def|Gender=Neut|Number=Sing	3	nsubj:pass	_	_
2	се	се	PRON	_	Case=Acc|PronType=Prs|Reflex=Yes	3	expl:pass	_	_
3	преследваше	преследвам	VERB	_	Aspect=Imp|Mood=Ind|Number=Sing|Person=3|Tense=Past|VerbForm=Fin	0	root	_	_
4	от	от	ADP	_	_	5	case	_	_
5	котката	котка	NOUN	_	Definite=Def|Gender=Fem|Number=Sing	3	obl	_	_
6	.	.	PUNCT	_	_	3	punct	_	_
```

[Tab. 1] Russischer Satz, Analysiert, im CoNLL-U Format
"Der Hund verfolgt die Katze"

4.2 Datenbanken: SPARQL, ORMs

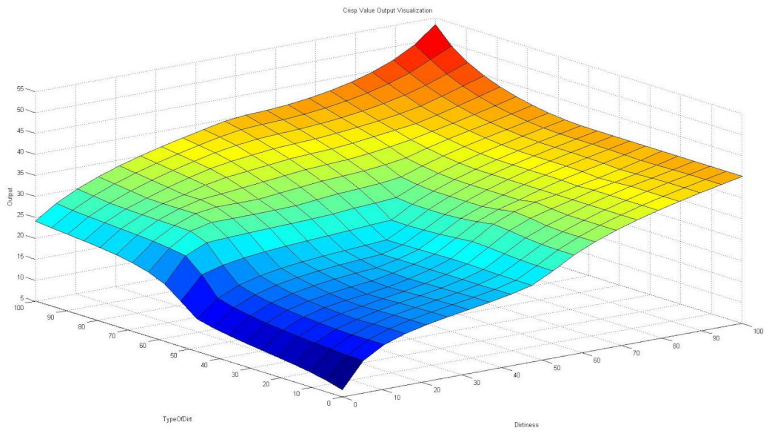
Datenbanken ermöglichen es, diese Strukturen und Daten so zu speichern, dass keine unnötige Redundanz entsteht. Wenn zum Beispiel zweimal das Wort "Revolution" aus zwei verschiedenen Texten in das Vokabular eines Algorithmus übernommen wird, so

wird es im CONLL-U Format zweimal gespeichert wenn die Texte als solche gespeichert werden. Mit einer Datenbank nur einmal, wobei es zweimal in verschiedenen Texten referenziert werden kann.

Object Relational Mappers (ORMs) ermöglichen es, mit verschiedenen Programmiersprachen Relationen zwischen Objekten in Datenbanken zu speichern und abzurufen. Ähnlich ist "SPARQL", eine graphenbasierte Abfragen-Sprache für Datenbanken, sie wird verwendet, um verschiedenste Dinge nach Ontologien (Beziehungen zwischen Typen), abzubilden. Beide haben die Vorteile einer Performanten Datenbank und die Möglichkeit komplexe Strukturen abzubilden.

5 Fuzzy Logic

Natürliche Sprachen sind ungenau. Sie wurden entworfen, um Eindrücke zu vermitteln und Gefühle auszudrücken. Wie zum Beispiel definiert ein Computer "Reichtum"? Ist "Reichtum" ein binärer Zustand? Verschiedene Menschen würden Reichtum verschieden auffassen. Deshalb verwenden Softwareentwickler, besonders wenn es um die Interaktion mit Menschen geht, "Fuzzy Logic". Statt mit einem binären Zustand (0 oder 1), wird mit einem Gradienten gearbeitet.



[Abb.3] Visualisation eines 3 Dimensionalen Gradienten

Jemandem, der “eher reich” ist könnte also zum Beispiel ein Wert von 35 zugeordnet werden. Diese Idee kann bei der künstlichen Produktion von Text helfen:

0-10 10-20 20-30 30-40 40-55

<Person> ist [nicht, weniger, mittelmäßig, eher, sehr] reich.

Solche Logik müsste aber kompliziert manuell einprogrammiert, abstrakt definiert oder mit Trainingsdaten trainiert werden. Je nach dem wie dieses Konzept implementiert ist benötigt es auch Steigerungsformen und Antonyme.

Literaturverzeichnis

Amant, Robert St; Cohen, Paul R.; "Massive Data Sets and Artificial Intelligence Planning" – University of Massachusetts, 1996

Bird, Steven; Klein, Ewan; Loper, Edward; "Natural Language Processing with Python" – Sebastopol, CA: O'Reilly
XIV Preface, 2009

Bobrow, Daniel G., "Natural Language Input for a computer problem solving system" – Massachusetts Institute of Technology, 1964

Buchholz, Sabine; Marsi, Erwin; "CoNLL-X shared task on Multilingual Dependency Parsing" – Cambridge Research Lab, Tilburg University, 2016

Crevier, Daniel, "AI: The Tumultuous Search for Artificial Intelligence" – University of Michigan, 1993
pp. 146-148, pp. 203

David Nadeau, Satoshi Sekine, "A Survey of named entity recognition and classification" – National Research Council Canada / New York University
p. 3, pp. 8, p. 11, 2017

Epstein, J.; Klinkenberg, W.D.; "From Eliza to Internet: a brief history of computerized assessment" – University of Missouri School of Medicine, Missouri Institute of Mental Health, 2001

McCorduck, Pamela; "Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence" 2nd Edition – Natick, MA: A. K. Peters
p. 442, p.443, 2004

Marie-Catherine de Marneffe, Dozat, Timothy; Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, Christopher D. Manning, "Universal Dependencies: A cross-linguistic typology" – Ohio State University, Stanford University, University of Turku, Uppsala University, 2014

Russel, Stuart J.; Norvig, Peter; "Artificial Intelligence: A Modern Approach" – Essex, England: Pearson
p. 21, p. 24, pp. 25, 2003

Xu, Anbang; Liu, Zhe; Guo, Yufan; Vibha, Sinha; Akkiraju, Rama; "A New Chatbot for Customer Service on Social Media" – IBM Research, 2017

Abbildungsverzeichnis

Internetverzeichnis

20.10.2015, https://www.wired.com/2015/10/this-news-writing-bot-is-now-free-for-everyone/
Finley, Klint; "This News-Writing Bot Is Now Free for Everyone" – Wired