SoekiaGPT- Handreichung für Lehrpersonen

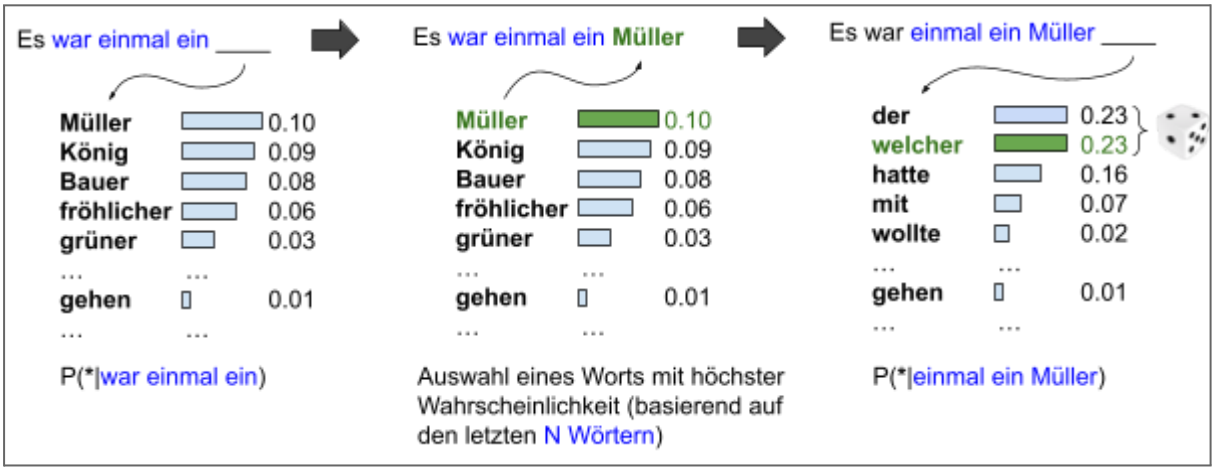
Die Behandlung von künstlicher Intelligenz im Unterricht umfasst ganz viele Aspekte: Was versteht man unter künstlicher Intelligenz? Wie funktioniert künstliche Intelligenz, z.B. Machine Learning, Neuronale Netze, statistische Sprachmodelle? Wie und wo kann künstliche Intelligenz genutzt werden? Welche Auswirkungen hat künstliche Intelligenz auf den Arbeitsmarkt und die Gesellschaft? Die Lernumgebung SoekiaGPT fokussiert auf die Funktionsweise von Sprachmodellen wie etwa ChatGPT und ermöglicht einen Blick hinter die Kulissen solcher mächtiger, textbasierter Sprachmodelle. Damit lassen sich exemplarisch einige Prinzipien der Funktionsweise von Chatbots kennenlernen.

Der Erfolg textbasierter Sprachmodelle wie ChatGPT beruht erstens auf riesigen Dokumentenkollektionen als Trainingsdaten, zweitens auf mächtigen neuronalen Netzen für das maschinelle Lernen und drittens auf grosser Rechenleistung. Alle drei Voraussetzungen sind im Unterricht nicht gegeben. Zudem ist das Verständnis solch komplexer Systeme für Schülerinnen und Schüler schwierig. SoekiaGPT nimmt deshalb eine ganze Reihe von Vereinfachungen vor und ist eine didaktische Lernumgebung für den Unterricht. SoekiaGPT hat damit eine vergleichbare Zielsetzung wie die didaktische Suchmaschine Soekia und ist auch sehr ähnlich strukturiert. Mit den Schülerinnen und Schülern kann auch diskutiert werden, inwieweit sich Sprachmodelle und Suchmaschinen gleichen. In beiden Systemen wird eine grosse Dokumentenkollektion durchsucht und analysiert, in einem effizienten Speicher aufbereitet und anschliessend mit einem Algorithmus basierend auf der Eingabe eine möglichst passende Ausgabe generiert. Im Unterschied zu Suchmaschinen liefert das Sprachmodell als Ausgabe jedoch nicht eine Liste mit Suchtreffern, sondern einen einzigen Text auf Basis aller analysierten Dokumente.

Was ist ein Sprachmodell?

Textbasierte Sprachmodelle nutzen typischerweise eine riesige Dokumentenkollektion, z.B. Texte aus der Wikipedia, aus Büchern und Zeitschriften, aus News-Artikeln, aus Social Media Plattformen und vieles mehr. Diese Texte dienen als Trainingsdaten und das Sprachmodell berechnet basierend aufgrund der mathematisch-statistischen Analyse zum Beispiel, welches Zeichen oder welches Wort in einem Text mit grosser Wahrscheinlichkeit als nächstes kommt und ergänzt so einen Text fortlaufend.

Die folgende Abbildung zeigt vereinfacht die Funktionsweise eines textbasierten Sprachmodells:



Im obigen Beispiel schaut das Sprachmodell immer auf die letzten drei Worte des bereits geschriebenen Textes zurück und fügt ein Wort an, welches mit der grössten Wahrscheinlichkeit auf die drei letzten Worte folgt. Stehen mehrere gleichwahrscheinliche Wörter zur Auswahl, entscheidet der Zufall.

Heute werden in der Praxis meist neuronale Sprachmodelle wie GPT eingesetzt, welche auf riesigen Dokumentenkollektionen als Trainingsdaten basieren. Sie nutzen neuronale Netzwerke und maschinelles Lernen. Die Funktionsweise von neuronalen Netzen lässt sich zwar ebenfalls vermitteln, ist aber ein eigenes komplexes Thema, welches für das Verständnis von Sprachmodellen nicht zwingend notwendig ist. Wir verwenden deshalb stattdessen klassische statistische Sprachmodelle, die einfacher erklärt werden können. Für detaillierte Ausführungen zu neuronalen und statistischen Sprachmodellen wird auf die Fachliteratur verwiesen. Zur Vertiefung sei der folgende Beitrag empfohlen: https://lena-voita.github.io/nlp_course/language_modeling.html

Was kann man mit dem Sprachmodell von SoekiaGPT zeigen?

Mit klassischen statistischen Sprachmodellen wie in SoekiaGPT kann man u.a. die folgenden Aspekte im Unterricht aufzeigen:

- Sprachmodelle basieren auf einer grossen Dokumentenkollektion und “lernen” aus diesen Dokumenten.
- Die von Sprachmodellen erzeugten Ausgaben sind stark von der Dokumentenkollektion abhängig.
- Sprachmodelle nutzen Statistik, genauer die absolute und relative Häufigkeit von Wortfolgen in der Dokumentenkollektion.
- Bei der Texterzeugung nutzen Sprachmodelle den Zufall, damit ihre Ausgabe nicht immer identisch ist.
- Sprachmodelle können neue, in der Dokumentenkollektion nicht vorhandene Texte erzeugen.
- Sprachmodelle können “Fakten” erfinden, die durchaus glaubwürdig klingen.
- Sowohl der Aufbau als auch die Ausgabe eines Sprachmodells kann durch verschiedene Parameter beeinflusst werden.
- Über den “Prompt” werden die zuvor analysierten statistischen Daten gewichtet und so Ausgaben erzeugt, die tendenziell besser zum eingegebenen Prompt passen.

Um im Unterricht einen handlungsorientierten Einblick in die grundlegende Funktionsweise textbasierter Sprachmodelle zu vermitteln, wurden bei SoekiaGPT eine ganze Reihe von Vereinfachungen vorgenommen:

- Die Trainingsdaten bestehen aus maximal 100 Dokumenten mit je maximal 20’000 Zeichen. Da die Trainingsdaten aber von den Lernenden selbst vorgegeben und angepasst werden können, eröffnen sich verschiedene Experimentiermöglichkeiten.
- SoekiaGPT verwendet ganze Wörter, d.h. beim Schreiben von Texten werden bestehende Wortfolgen durch Worte erweitert.
- SoekiaGPT kann bei seinen Antworten maximal auf die letzten fünf Wörter zurückschauen und hat damit einen stark begrenzten “Kontext”. Ein fortlaufender Dialog wie bei ChatGPT ist somit nicht möglich.
- SoekiaGPT nutzt kein neuronales Netz und lernt die statistischen Zusammenhänge nicht in einem iterativen Prozess.

Eine Besonderheit von SoekiaGPT ist die Möglichkeit, die Quelle von erzeugten Textfragmenten zu visualisieren. Durch farbliche Hervorhebungen wird deutlich, aus welchem Quelldokument ein durch SoekiaGPT gewähltes Wort stammt. Diese didaktische Möglichkeit besteht nur, da eine sehr geringe Anzahl von Dokumenten verwendet wird. Bei grossen Sprachmodellen müssten für jedes Wort schnell tausende Dokumente als mögliche Quelle angegeben werden, womit dies nicht mehr praktikabel wäre. Dies ist ein Grund, warum Sprachmodelle wie ChatGPT nicht einfach Quellenangaben mitliefern können. Unter

dem Stichwort “Explainable AI” forscht man vielfältig an der Herausforderung, wie KI-Systeme besser erklären können, wie genau ihre Ausgaben entstanden sind.

Eine weitere Besonderheit in SoekiaGPT ist die Möglichkeit, die Auswahl des nächsten Wortes nicht nur automatisch zu treffen (vergleichbar mit ChatGPT), sondern auch schrittweise manuell mittels Auswahl von Wortvorschlägen durch die Lernenden selbst. Die manuelle Auswahl des nächsten Wortes ist dem naiven Auswahlalgorithmus (wahrscheinlichster Vorschlag wird gewählt) überlegen und die Lernenden können so eher sinnvolle Sätze bilden. Damit lässt sich die begrenzte Datenbasis durch das Sprachwissen der Schülerinnen und Schüler teilweise kompensieren und dem Sprachmodell “helfen”, eine gute Auswahl zu treffen.

Wie funktioniert das Sprachmodell von SoekiaGPT?

In klassischen statistischen Sprachmodellen werden aus allen Dokumenten Daten extrahiert und gespeichert. Das können einzelne Buchstaben, Silben oder ganze Wörter sein. In SoekiaGPT verwenden wir ganze Wörter und Satzzeichen. Man bezeichnet das kleinste betrachtete Element auch als sogenannte Token.

Bei der Analyse der Dokumente werden N-Gramme gebildet. Das sind Wortfolgen (oder Tokenfolgen) der Länge N, welche im Dokument angetroffen wurden.

Betrachten wir als Beispiel das folgende kurze Dokument:
Der Schmied im Dorf hatte einen Sohn. Der Müller hatte eine Tochter.

Es werden für N≤4 die folgenden N-Gramme erzeugt:

1-Gramme	**2-Gramme**	**3-Gramme**	**4-Gramme**
Der	Der Schmied	Der Schmied im	Der Schmied im Dorf
Schmied	Schmied im	Schmied im Dorf	Schmied im Dorf hatte
im	im Dorf	im Dorf hatte	im Dorf hatte einen
Dorf	Dorf hatte	Dorf hatte einen	Dorf hatte einen Sohn
hatte	hatte einen	hatte einen Sohn	hatte einen Sohn .
einen	einen Sohn	einen Sohn .	einen Sohn . Der
Sohn	Sohn .	Sohn . Der	Sohn . Der Müller
.	. Der	. Der Müller	. Der Müller hatte

Der	Der Müller	Der Müller hatte	Der Müller hatte eine
Müller	Müller hatte	Müller hatte eine	Müller hatte eine Tochter
hatte	hatte eine	hatte eine Tochter	hatte eine Tochter .
eine	eine Tochter	eine Tochter .	
Tochter	Tochter .		
.			

In der ersten Spalte wurden zur besseren Lesbarkeit die Wörter “Der” und “hatte“ zweimal aufgeführt. In der praktischen Umsetzung werden sie aber nur einmal in der Liste gespeichert und gezählt, wie häufig sie vorkamen. Die Häufigkeit eines N-Gramms wird später für die Wortvorschläge genutzt.

Diese Wortfolgen (N-Gramme) können nun auf die bisherige Eingabe angewendet werden, um einen Vorschlag für das nächste Wort zu finden. Nehmen wir an, es wurde „Der“ als Eingabe (Prompt) gegeben. Wir können alle N-Gramme markieren, welche sich für die Fortsetzung eignen würden:

1-Gramme	**2-Gramme**	**3-Gramme**	**4-Gramme**
Der	Der Schmied	Der Schmied im	Der Schmied im Dorf
Schmied	Schmied im	Schmied im Dorf	Schmied im Dorf hatte
im	im Dorf	im Dorf hatte	im Dorf hatte einen
Dorf	Dorf hatte	Dorf hatte einen	Dorf hatte einen Sohn
hatte	hatte einen	hatte einen Sohn	hatte einen Sohn .
einen	einen Sohn	einen Sohn .	einen Sohn . Der
Sohn	Sohn .	Sohn . Der	Sohn . Der Müller
.	. Der	. Der Müller	. Der Müller hatte
Der	Der Müller	Der Müller hatte	Der Müller hatte eine
Müller	Müller hatte	Müller hatte eine	Müller hatte eine Tochter
hatte	hatte eine	hatte eine Tochter	hatte eine Tochter .
eine	eine Tochter	eine Tochter .	
Tochter	Tochter .		
.			

Das letzte Wort des N-Gramms (grün markiert) nutzen wir als Vorschlag, wenn alle vorherigen Wörter des N-Gramms mit der Eingabe übereinstimmen.

Das Sprachmodell würde nun „Schmied“ und „Müller“ als passende nächste Wörter vorschlagen. Da beide gleich häufig im Dokument vorkommen (jeweils ein Mal), wird zufällig ausgewählt. Wir wählen „Müller“ aus, verlängern die Eingabe auf „Der Müller“ und prüfen erneut auf mögliche Fortsetzungen (Tabelle gekürzt auf passende N-Gramme):

1-Gramme	2-Gramme	3-Gramme	4-Gramme
.	. Der	. Der Müller	. Der Müller hatte
Der	Der Müller	Der Müller hatte	Der Müller hatte eine
Müller	Müller hatte	Müller hatte eine	Müller hatte eine Tochter

Da die Eingabe bereits zwei Wörter enthält, können wir auf zwei Wörter “zurückschauen” und sowohl 2-Gramme als auch 3-Gramme verwenden. Je länger das N-Gramm, desto mehr Kontext zum bereits eingegebenen Satz besteht. Wir vereinbaren deshalb, dass längere N-Gramme einen höhere Passung haben und damit bei Wortvorschlägen priorisiert werden. Im konkreten Beispiel liefern beide passenden N-Gramme „hatte“.

Wir verlängern die Eingabe auf „Der Müller hatte“ und prüfen erneut auf mögliche Fortsetzungen:

1-Gramme	2-Gramme	3-Gramme	4-Gramme
hatte	hatte einen	hatte einen Sohn	hatte einen Sohn .
Der	Der Müller	Der Müller hatte	Der Müller hatte eine
Müller	Müller hatte	Müller hatte eine	Müller hatte eine Tochter
hatte	hatte eine	hatte eine Tochter	hatte eine Tochter .

An dieser Stelle wird nun „eine“ und „einen“ vorgeschlagen. Das längste N-Gramm würde priorisiert und “eine” ausgewählt. Weiter fortgesetzt entsteht nicht überraschend der Satz: „Der Müller hatte eine Tochter.“

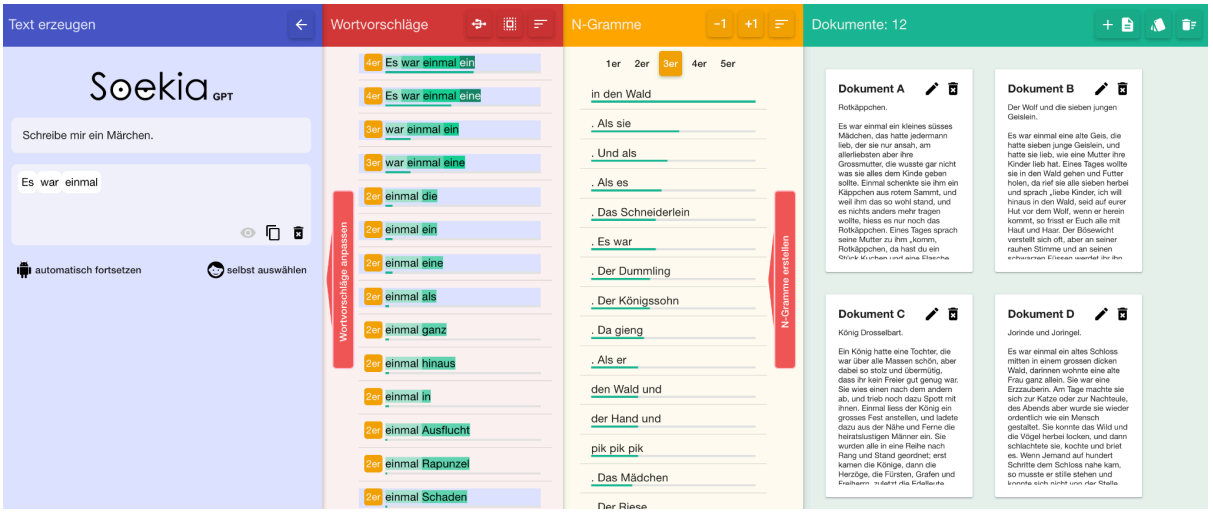
Durch Einstellungen am Sprachmodell lässt sich das Auswahlverhalten verändern. Würde etwa „einen“ ausgewählt, würde der Satz auf „Der Müller hatte einen Sohn.“ vervollständigt. Eine Aussage, die so im Dokument nicht vorkam.

Im Gegensatz zu klassischen statistischen Sprachmodellen werden bei GPT keine langen Listen mit N-Grammen gebildet, sondern ein neuronales Netzwerk mit Wortfolgen als Eingabe trainiert. Die Stärke eines neuronalen Netzwerks ist seine Fähigkeit zur Verallgemeinerung bzw. Abstraktion. Grosse Sprachmodelle können aus grossen Datenmengen lernen, dass bestimmte Wörter eine ähnliche Bedeutung haben, weil sie immer in vergleichbaren Kombinationen mit anderen Wörtern vorkommen.
Dies betrifft den syntaktischen Aufbau eines Wortes: also dass “Baum” sehr ähnlich ist wie "Bäume", da mehrheitlich die gleichen Buchstaben in ähnlicher Reihenfolge vorkommen, wie auch die semantischen Bedeutung eines Wortes: etwa dass “Pkw” und “Auto” häufig in einem vergleichbaren Kontext vorkommen. Ein neuronales Sprachmodell kann Milliarden von Wortfolgen erlernen, für die wir sonst gigantische N-Gramm-Listen benötigen würden, welche die Grenzen nur schon des Arbeitsspeichers sprengen würden. Systeme wie GPT können zudem auf mehrere tausend Token “zurückschauen”, indem diese als Eingabe für das neuronale Netzwerk verwendet werden. Die Systeme haben somit deutlich mehr “Kontext”.

Dennoch sind die Prinzipien und Grenzen von neuronalen Sprachmodellen, wie die Analyse und Aufbereitung einer Dokumentenkollektion und die Sprachsynthese basierend auf statistischen Daten, mit jenen einfacher statistischer Modelle vergleichbar.

Aufbau von SoekiaGPT

SoekiaGPT ist in vier farbige Bereiche eingeteilt, welche die verschiedenen Verarbeitungsschritte verdeutlichen.



Ganz links (blau) findet die Texterzeugung statt, also die für den Benutzer üblicherweise sichtbare Ebene bei der Nutzung eines Sprachmodelles. Es kann ein Textanfang (Prompt) eingegeben und eine Ausgabe automatisch oder manuell generiert werden.

Ganz rechts (grün) wird die Dokumentenkollektion mit maximal 100 Dokumenten dargestellt, aus welchen das Sprachmodell aufgebaut wird. In SoekiaGPT können eigene Dokumentenkollektionen erstellt und verwaltet werden. Dazu werden Texte mit maximal 20’000 Zeichen per Copy&Paste in den Dokumenten abgespeichert. Eine so erstellte Dokumentenkollektion kann als JSON-Datei heruntergeladen und in einem späteren Zeitpunkt erneut verwendet oder durch die Lehrpersonen zur Verfügung gestellt werden. Zudem kann eine “Gemeinsame Kollektion” erstellt werden, die über einen Code von allen Schülerinnen und Schülern kollaborativ bearbeitet werden kann.
Einige Sonderzeichen werden bei der Verarbeitung der Dokumente entfernt und damit ignoriert.

Die gelbe Spalte “N-Gramme” zeigt alle aus der Dokumentenkollektion erzeugten N-Gramme und die Häufigkeit ihres Vorkommens in der Dokumentenkollektion an. Über 1er, 2er … 6er kann zwischen den verschiedenen N-Gramm-Listen gewechselt werden. In SoekiaGPT lässt sich über +1 und -1 das maximale N wählen, bis zu welchem die N-Gramme erzeugt werden sollen. Je höher das N, desto längere Wortfolgen werden gespeichert und für die Vorschläge genutzt. Allgemein gilt, je höher das N, desto mehr Kontext wird genutzt und desto eher werden grammatikalisch korrekte Sätze entstehen. Gleichzeitig benötigt ein höheres N auch mehr Arbeitsspeicher und Rechenzeit, da immer mehr und längere Listen mit Wortfolgen vorbereitet und durchsucht werden müssen. Daraus ergeben sich praktische Grenzen. In

SoekiaGPT kann deshalb maximal N=6 gewählt werden. Über den vertikalen roten Button “N-Gramme erstellen” kann die Analyse der Dokumente animiert werden, um den Verarbeitungsprozess zu visualisieren.

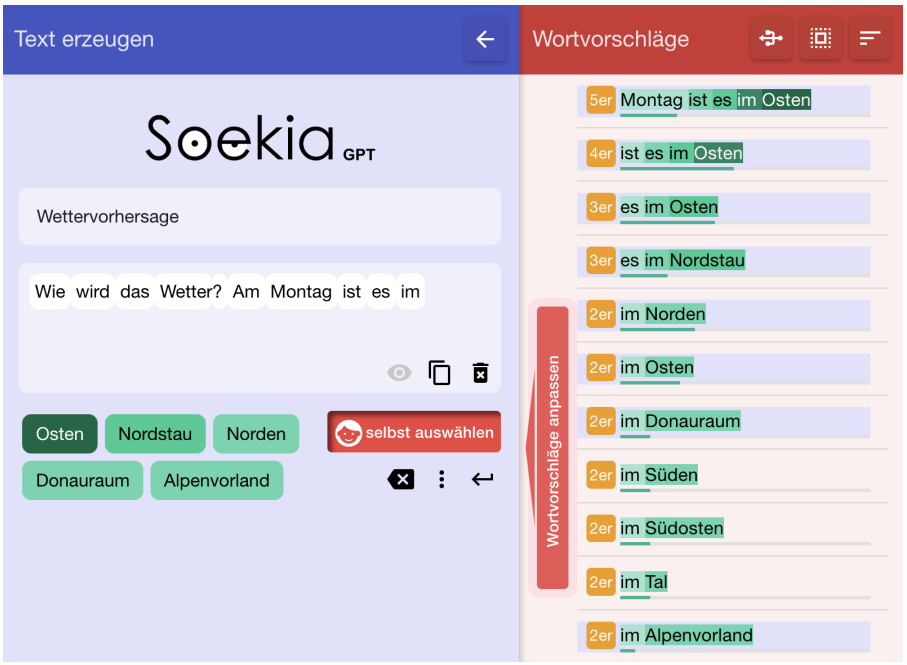
Die rote Spalte “Wortvorschläge” zeigt in jedem Verarbeitungsschritt zum bereits erzeugten Text passende nächste N-Gramme an. Passend sind jene N-Gramme, bei denen die letzten N-1 Wörter des bereits erzeugten Textes mit den ersten N-1 Worten des N-Gramms übereinstimmen. Die Liste wird nach den berechneten Wahrscheinlichkeiten der N-Gramme geordnet dargestellt, lässt sich aber auch auf alphabetisch umstellen. Die Wahrscheinlichkeit der potentiell passenden N-Gramme wird aus dem N (je höher, desto höher die Wahrscheinlichkeit), der Häufigkeit des N-Gramms in der Dokumentenkollektion (je häufiger, desto höher die Wahrscheinlichkeit) und einem Kontext-Wert (wie gut passt der eingegebene Prompt zu einem Dokument in der Kollektion) bestimmt. Es ist nicht relevant, die Berechnung im Detail nachzuvollziehen, sie ist an das Verhalten grosser Sprachmodelle angelehnt, wo ebenfalls der Prompt massgeblich die Wahrscheinlichkeit der Ausgaben beeinflusst.

Wird ein Textanfang vom Sprachmodell fortgesetzt, findet ein Auswahlprozess aus allen passenden N-Grammen statt. Über den vertikalen Button “Auswahl anpassen” lässt sich wählen, wie viele Wortvorschläge aus der Menge aller passender N-Gramme ausgewählt werden sollen.

Der Auswahl-Algorithmus wird über die sog. Temperatur gesteuert. Eine niedrige Temperatur priorisiert die statistisch wahrscheinlichsten N-Gramme (nach höchstem N und nach Häufigkeit in den Quelldokumenten und in Abhängigkeit des eingegebenen Prompts). Je höher die Temperatur, desto zufälliger erfolgt die Auswahl der Wortvorschläge. Eine höhere Temperatur führt quasi zu einem “kreativeren” Sprachmodell, aber auch zu mehr unsinnigen und grammatikalisch falschen Sätzen. In SoekiaGPT kann die Temperatur

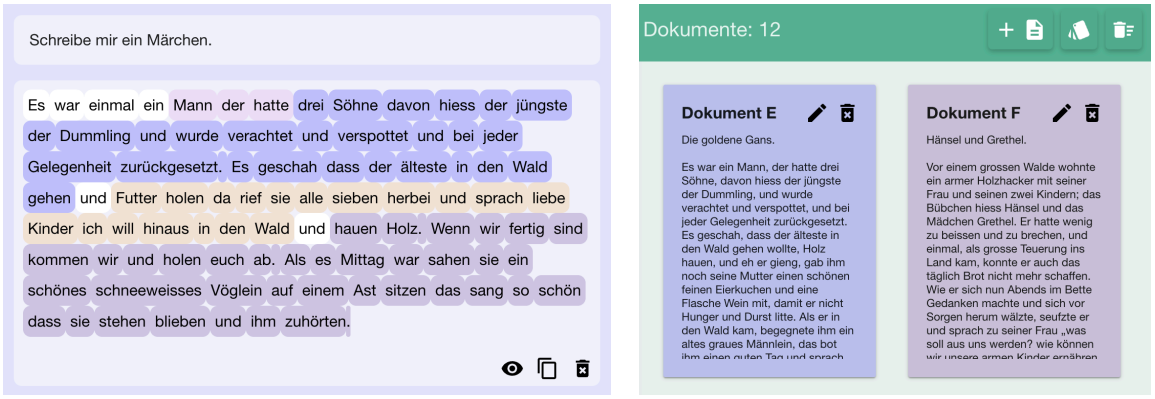
angepasst und experimentiert werden, wie sich dieser Parameter auf den erzeugten Text auswirkt.

Alle ausgewählten N-Gramme werden im manuellen Modus im linken Bereich als grüne Buttons angezeigt; je dunkler der Button, desto höher das N des N-Gramms. Im folgenden Beispiel wurden maximal fünf Wortvorschläge eingestellt:

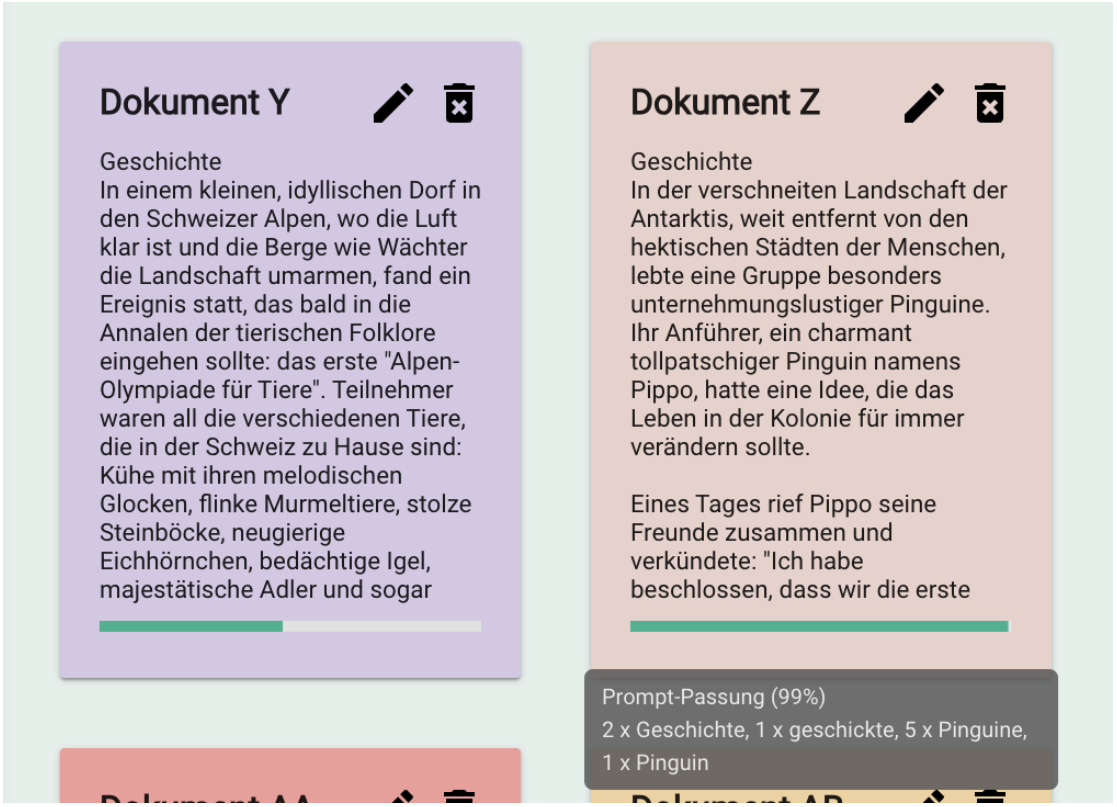


Im obigen Beispiel sieht man, dass der Vorschlag “Osten” in mehreren N-Grammen als nächstes Wort vorkommt.

Über das kleine Augen-Symbol im blauen Ausgabefenster werden Wörter farblich hervorgehoben, wenn das bei der Erzeugung gewählte N-Gramm nur in einem einzigen Dokument vorkommt. Die Dokumente in der Kollektion (grüne Spalte) werden mit der gleichen Farbe eingefärbt. Die farbliche Zuordnung von ausgewählten N-Grammen zu den Dokumenten erlaubt verschiedene Experimente. So werden etwa bei höherer Temperatur häufiger Wortvorschläge aus mehreren Dokumenten einbezogen. Bei tiefer Temperatur wird oft ein Dokument reproduziert.



Der eingegebene Prompt hat einen direkten Einfluss auf die Auswahlwahrscheinlichkeit der N-Gramme bei der Texterzeugung. Sobald ein Prompt eingegeben und die Generierung gestartet wurde, wird bei jedem Dokument in der Kollektion ein kleiner Bewertungsbalken angezeigt. Mit einem Klick darauf lässt sich anzeigen, welche Übereinstimmungen der Prompt mit dem Dokument hat und welche Worte Einfluss haben. Alle N-Gramme im Dokument werden bei der Generierung entsprechend gewichtet. Zum Prompt “Schreibe mir eine Geschichte über Pinguine.” werden zum Beispiel Dokumente mit Pinguinen höher gewichtet.



Um die Fähigkeit von Sprachmodellen zur Verallgemeinerung thematisieren zu können, kann der Algorithmus für die Auswahl passender N-Gramme in SoekiaGPT über den Schalter “ähnliche Wörter einbeziehen" und “Synonyme einbeziehen” angepasst werden.

Die syntaktische Ähnlichkeit von Wörtern lässt sich über verschiedene Verfahren berechnen. Ein bekanntes Mass für die Ähnlichkeit zweier Zeichenketten ist die Levenshtein-Distanz (auch Editierdistanz). Sie beschreibt die minimale Anzahl einfügender, löschender und ersetzender Operationen, um eine Zeichenkette in eine andere Zeichenkette umzuwandeln. In SoekiaGPT wird die Jaro–Winkler-Distanz verwendet, ein schnelleres, aber ähnliches Verfahren. Wird für ein N-Gramm eine Ähnlichkeit von 95% oder mehr berechnet, wird dieses N-Gramm mit in die Liste der Wortvorschläge aufgenommen. Ähnliche Wörter werden in der Liste der Wortvorschläge rot hervorgehoben. So wird etwa “Müllerin” und “Müller” als ähnliches Wort behandelt.

Die semantische Ähnlichkeit von Wörtern lässt sich in einer kleinen Dokumentenkollektion nicht berechnen. Aus diesem Grund greift SoekiaGPT im Hintergrund auf eine umfangreiche Liste deutscher Synonyme von https://www.openthesaurus.de/ zurück. Damit werden N-Gramme mit einer ähnlichen Bedeutung ausgewählt und in der Liste blau hervorgehoben. Wird der Synonym-Modus aktiviert, werden auch die Begriffe im Prompt auf Synonyme untersucht. So wird SoekiaGPT bei der Frage nach dem schnellsten “Auto” auch Dokumente mit “PKW” als passend einstufen.

Hinweise zu möglichen Aufgabenstellungen zu SoekiaGPT

SoekiaGPT kann auf verschiedenen Schulstufen und mit unterschiedlicher Tiefe eingesetzt werden. Die nachfolgenden Aufgaben dienen als Hinweise für Lehrpersonen zur Gestaltung des Unterrichts und müssen nicht zwingend in dieser Reihenfolge oder vollständig bearbeitet werden. Die Aufgaben sind nach den zwei wesentlichen Prozessen in SoekiaGPT - Erzeugen der N-Gramm-Listen aus der Dokumentenkollektion und Auswahl der Wortvorschläge für die Texterzeugung - strukturiert. Die Aufgaben zeigen, welche Aspekte von statistischen Sprachmodellen bearbeitet oder erforscht werden können.

Dokumente analysieren und auswerten

Sprachmodelle basieren auf einer grossen Dokumentenkollektion und lernen aus diesen Dokumenten. SoekiaGPT nutzt nur eine sehr kleine Dokumentenkollektion, erlaubt dafür aber das Experimentieren mit den Dokumenten. So kann gezeigt werden, dass die von Sprachmodellen erzeugten Ausgaben stark von der zugrundeliegenden Dokumentenkollektion abhängen.

- ☐ Für den Einstieg stehen verschiedene Dokumentenkollektionen (Grimm Märchen, Fairy Tales, Wettervorhersagen, Musik in ABC-Notation, Tiergeschichten und -beschreibungen) zur Verfügung. In einem ersten Schritt können die Lernenden ausgehend von einer dieser Dokumentenkollektionen anschauen, wie die Analyse der Dokumente erfolgt, also das Erstellen der N-Gramme.
- ☐ Die Einträge in den N-Gramm-Listen zeigen die relative Häufigkeit (grüner Balken unter dem N-Gramm). Die Liste lässt sich alphabetisch oder nach Häufigkeit ordnen. Ein Klick auf ein N-Gramm zeigt die absolute Häufigkeit an und die Quelldokumente, in denen das N-Gramm vorkommt. Es kann über die Häufigkeit von bestimmten Wörtern und Wortkombinationen in der deutschen Sprache und spezifisch in der Kollektion diskutiert werden.
- ☐ Die Länge der N-Gramm-Listen hängt von der Dokumentenkollektion ab. In SoekiaGPT können die Dokumentenkollektionen sehr einfach bearbeitet werden. Die Lernenden können so den Zusammenhang der Grösse der Dokumentenkollektion und der Länge der N-Gramm-Liste erkunden. Die Häufigkeiten bestimmter Wörter oder Wortfolgen können durch bewusste Anpassung der Dokumente verändert werden.
- ☐ Sprachmodelle können neue, in der Dokumentenkollektion nicht vorhandene Texte erzeugen. Die Lernenden können mit SoekiaGPT selbst solche Texte erzeugen, ihre Qualität diskutieren und die Texte ggf. zur Dokumentenkollektion hinzufügen oder die Kollektion gezielt ergänzen, um ihre Wunschtexte erzeugen zu können.
- ☐ Einzelne Dokumente können gekürzt oder ergänzt werden. Damit lässt sich die Qualität der erzeugten Texte beeinflussen. Bei Wettervorhersagen können die

Dokumente z.B. durch für Wetterprognosen typische Formulierungen wie “Morgen Montag, Morgen Dienstag, ….” oder “Am Montag, am Dienstag, …” ergänzt werden.

- ☐ Lernende können eigenen Dokumentenkollektionen erstellen: z.B. News-Meldungen zu einem selbstgewählten Thema, in verschiedenen Sprachen, usw. und die gebildeten N-Gramm-Listen untersuchen.
- ☐ Über eine gemeinsame Kollektion können Texte zu einem vorgegebenen Thema (z.B die eigene Schule / Gemeinde) gesammelt werden und anschliessend Fragen als Prompts formuliert werden. Wie viele und welche Informationen benötigt das Sprachmodell, um gute Ausgaben zu generieren? Welche Daten sind wünschenswert und welche nicht?
- ☐ Um die Mächtigkeit echter neuronaler Sprachmodelle einzuschätzen, können die Lernenden recherchieren, wie gross die zugrundeliegende Dokumentenkollektion bei aktuellen Sprachmodellen sind. In der Fachsprache wird anstelle von den in SoekiaGPT genutzten Wörtern von sog. Tokens gesprochen. Tokens sind Wörter oder Teilwörter und können auch Satzzeichen etc. umfassen. Es kann recherchiert werden, auf wie viele Eingabetoken aktuelle Sprachmodelle zurückschauen können.
  Ebenfalls recherchieren können Lernende, wie viel Speicherplatz und Rechenleistung aktuelle neuronale Sprachmodelle benötigen und welcher Energieverbrauch damit verbunden ist.

Texte aus Wortvorschlägen generieren

Bei der Texterzeugung nutzen statistische Sprachmodelle die Häufigkeit von Wortfolgen in der Dokumentenkollektion. Die Sprachmodelle nutzen dabei auch den Zufall, so dass die erzeugten Texte nicht immer identisch sind. Die Lernenden können mit den Parametern “Anzahl ausgewählter Wortvorschläge”, “Temperatur” experimentieren und so ein Gefühl für die Rolle des Zufalls entwickeln.

- ☐ Zunächst können mehrere Beispiele generiert und deren Qualität verglichen werden. Mit der Option “Quellen anzeigen” im Ausgabefenster lässt sich teilweise nachvollziehen, aus welchen Quelldokumenten das Ergebnis generiert wurde.
- ☐ Die Lernenden können ihre, mit der gleichen Dokumentenkollektion, generierten Texte miteinander vergleichen und Hypothesen aufstellen, welche Wortfolgen zu bestimmten Ergebnissen führen werden.
- ☐ Die Lernenden können mit unterschiedlichen maximalem N und Einstellungen der Anzahl ausgewählter Wortvorschläge und der Temperatur experimentieren und deren Einfluss auf die erzeugten Texte erforschen (z.B. “kreative Texte” versus sprachliche Korrektheit, Vermeidung von zyklischen Texten).

- ☐ Die Lernenden können mit SoekiaGPT durch mehrfaches Hinzufügen bestimmter Worte in den Dokumenten auch erfahren, wie Sprachmodelle manipuliert werden können.
- ☐ Sprachmodelle werden mit wachsender Dokumentenkollektion immer leistungsfähiger. In kleinen Kollektionen gibt es häufig statistisch zu wenig Daten. In SoekiaGPT können die Lernenden deshalb Wortvorschläge auch manuell auswählen und so aufgrund ihres Vorwissens (“Intelligenz”) und des Kontextes bessere Texte erzeugen. Damit können sie quasi umfangreichere Sprachmodelle simulieren.
  Wichtig ist jedoch das Verständnis, dass bei Systemen wie ChatGPT keine Menschen die Antworten formulieren, sondern ein hinreichend grosse Sprachmodell verwendet wird.
- ☐ Bei gleicher Einstellung von maximalem N, der Anzahl ausgewählter Wortvorschläge und gleicher Temperatur können die Lernenden die erzeugten Texte vergleichen und so erkennen, dass sich diese je nach Wahl der beiden Parameter weniger oder mehr unterscheiden.
- ☐ Sprachmodelle können vermeintliche Fakten erfinden, die durchaus glaubwürdig klingen. Lernende können sich diese Effekte zum Beispiel anhand der Dokumentenkollektion Wettervorhersage schnell bewusst machen.
- ☐ Sprachmodelle können lernen zu Verallgemeinern und sowohl ähnliche Wörter, Synonyme und Wörter mit ähnlicher Bedeutung gleich zu behandeln. So könnten etwa “„Geldautomat“ und „Bankautomat“ oder „Auto“ und „Pkw“ jeweils durch die gleichen internen Token abgebildet werden. Über den Modus “ähnliche Wörter einbeziehen” und “Synonyme einbeziehen” lässt sich in SoekiaGPT mit der Fähigkeit von Sprachmodellen zur Verallgemeinerung experimentieren.
- ☐ Sprachmodelle können auch andere Medienformate als Text generieren. Ein Beispiel dafür ist die Musiknotation ABC als Textbeschreibung für Musik.
  Beispiele zur ABC-Notation finden sich etwa auf https://abcnotation.com/examples und ein ABC-Player auf https://abc.rectanglered.com/ .
  Die Auswirkung einer hohen Temperatur lässt sich gut mit der Kollektion ABC Musiksprache zeigen. Dazu können die generierten Musikstücke kopiert und mit einem ABC-Player abgespielt werden.