

# Spas mit Transformer Modellen

[desa@zhaw.ch](mailto:desa@zhaw.ch)

## Transformer

Ein Transformer hat die Fähigkeit, Eingaben zu vervollständigen. Ein trainiertes Modell kann die Eingabe eines Textes Wort für Wort entgegennehmen. In jedem Schritt bestimmt es das wahrscheinlichste Wort, welches auf die bisher entgegengenommenen Wörter folgt.

Zur Erzeugung beliebig langer Texte kann dem Modell ein Satz vorgelegt werden. Darauf bestimmt es das wahrscheinlichste Wort, dass auf diesen Satz folgt. Dieses Wort kann nun als nächstes eingegeben werden, worauf es wieder das wahrscheinlichste Wort bestimmt. Mit wiederholter Eingabe der vorherigen Ausgabe kann ein beliebig langer Text erzeugt werden.

## Fine-Tuning

Es gibt eine einfache Möglichkeit, ein GPT-2 Modell mit eigenen Trainingsdaten so anzupassen, dass es Eingaben spezifisch (gemäss den Trainingsdaten) vervollständigt. Die Anpassung eines vortrainierten Modells wird "Fine-Tuning" genannt und mit den Wörtern "Transformer Fine-Tune" lässt sich mit Google einiges darüber finden.

Das faszinierende an der Idee, ein Transformer "Fine-zu-Tunen": es liefert auf jeden Fall erfreuliche Resultate. Zum Beispiel haben wir eines mit einem Ärzte-Handbuch (Merck Manual) trainiert und danach lieferte es auf die Frage

**What is Diabetes?**

die Antwort

**Diabetes mellitus is a chronic, progressive and usually irreversible condition of the pancreas.**

Ohne Fine-Tuning kommt die Antwort

**Diabetes is a disease of the pancreas that occurs when the blood sugar level rises.**

Kann darin ein Unterschied erkannt werden? Z.B., dass das eine eher in Fach-Jargon antwortet (mit Merck Manual angepasst) und dass das andere einfacher zu verstehen ist? Uns hat es neugierig gemacht. Wir haben noch ein Modell mit einer Home-Edition vom Ärztehandbuch (Merck Home Edition) trainiert und dieselbe Frage gab folgende Antwort.

**Diabetes is a disorder of the pancreas, small intestine, and most of the blood vessels. When the blood levels of insulin and sugars in the blood drop too low, the pancreas produces and secretes too little insulin.**

Der einfachere Stil von der Home-Edition ist im Vergleich zur Ärzteversion ganz gut zu erkennen.

# Vorgehensweise

In diesem Dokument wird beschrieben, wie Fine-Tuning gemacht werden kann.

Das wird benötigt

- Es gibt ein Google Collaboratory Notebook, mit fertigem Code.  
<https://colab.research.google.com/drive/1VLG8e7YSEwypxU-noRNhsv5dW4NfTGce?usp=sharing>
- Ihr kopiert dieses Notebook auf den eigenen Google-Drive und könnt das Notebook in Eurem Google-Account laufen lassen.
- Das Notebook beschreibt in kleinen Schritten was gemacht werden muss. Weil es ein Notebook ist, kann der Code im Dokument angepasst und ausgeführt werden.
- Das Notebook ist als Tutorial gestaltet, welches Euch zum Fine-Tunen eines GPT-2 Modells anleitet.
- Es kann ein beliebiger, eigener Trainingsdatensatz verwendet werden.

Aufgabe

1. **Sucht nach irgendeinem Text (in ENGLISCH)**

Am Ende braucht eine Text-Datei, diese können z.B. aus PDF-Dateien erstellt werden: Google nach "Free Online PDF to Text Converter" und probiere einfach aus welcher geht. Manchmal können PDFs nicht grösser als 50 MB sein ;-). Zum Beispiel ein Buch (euer Lieblingsbuch in PDF suchen, Papers (100 Publikationen mit Google Scholar), oder Theaterstücke oder was auch immer Euch einfällt. Weitere Ideen werden in diesem Dokument weiter unten angedeutet.

2. **Macht damit das Fine-Tuning**

3. **Abgabe:** Gebt 3 Beispiele von Ein- und Ausgaben aus, die Euer Fine-Tuning zum Ausdruck bringen.

Tips & Tricks

- Zum Google Collaboratory Notebook gibt es auch einen Blog-Artikel, vielleicht ist das für die einen ein guter Einstieg:  
<https://minimaxir.com/2019/09/howto-gpt2/>
- Hier hat ein Schriftsteller etwas gemacht und dabei gute Beispiele von Daten-Aufbereitung gegeben:  
<https://towardsdatascience.com/how-to-fine-tune-gpt-2-so-you-can-generate-long-form-creative-writing-7a5ae1314a61>

Das mit <| startoftext |>, WP, [RESPONSE] und <| endoftext |> sind wichtige Details, die Ihr auch in Euren Daten einbauen könnt. Arbeitet mit Text-Editoren, Excel, SQL, Java - alles erlaubt, um die Trainingsdaten zu erzeugen.

- Es gibt Alternative Transformer-Modelle zu GPT2. Es gibt z.B. GPT3 und Bert und mit all diesen können dieselben Sachen gemacht werden.

## Ideen

Es ist möglich, ein GPT so zu trainieren, dass eine UI einer App in Worten beschrieben wird, und dass als Antwort z.B. HTML mit Bootstrap oder VUE.js Code kommt<sup>1</sup>. Weitere Ideen finden sich mit Google und z.B. "GPT Applications".

### Extraktion

Mit NLP<sup>2</sup> kann ein Trainingsdatensatz erzeugt werden, in dem z.B. zu Texten die darin enthaltenen Entities (Named Entity Recognition) und/oder Triples (Subjekt, Objekt, Prädikat) angegeben sind. Ein einzelner solcher Eintrag könnte die folgende Form haben.

```
<item>
<text>Afghanistan: Biden Vows to End Nation's Longest War by 9/11</text>
<facts>country(Afghanistan). person(Biden). misc(Longest).
cause_of_death(War). number(9/11). have(Nation, War). end(Biden, Nation
's Longest War). </facts>
</item>
```

Zum Beispiel kann dies mit Nachrichtentexten gemacht werden, welche in einem RSS-Feed (z.B. <https://www.democracynow.org/democracynow.rss>) gefunden werden können. Das resultierende Dokument für das Fine-Tuning hat die folgende Form.

```
<item><text>...</text><facts>... </facts></item>
<item><text>...</text><facts>... </facts></item>
<item><text>...</text><facts>... </facts></item>
...
```

Das Transformer-Modell kann im Anschluss mit folgender Anfrage zur Vervollständigung zur Extraktion verwendet werden.

```
<item><text>Afghanistan: Biden Vows to End Nation's Longest War by
9/11</text><facts>
```

### Generation

Basierend auf derselben Idee wie bei der Extraktion, kann ein umgekehrter Trainingsdatensatz der folgenden Form erzeugt werden.

```
<item><facts>...</facts><text>... </text></item>
<item><facts>...</facts><text>... </text></item>
<item><facts>...</facts><text>... </text></item>
...
```

Mit folgender Anfrage kann das Transformer-Modell eingesetzt werden, um aus den Extrakten einen Text zu erzeugen.

```
<item><facts>country(Afghanistan). person(Biden). misc(Longest).
cause_of_death(War). number(9/11). have(Nation, War). </facts><text>
```

---

<sup>1</sup> [https://youtu.be/G6Z\\_S6hs29s?t=121](https://youtu.be/G6Z_S6hs29s?t=121)

<sup>2</sup> Code-Base mit Beispiel-Code: <https://github.com/zhaw-iwi/nlp-starter-zero.git>

## Literatur

<https://towardsdatascience.com/transformers-explained-65454c0f3fa7>

<http://jalammar.github.io/illustrated-transformer/>

<https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04>

[https://en.wikipedia.org/wiki/Transformer\\_\(machine\\_learning\\_model\)](https://en.wikipedia.org/wiki/Transformer_(machine_learning_model))