

- **1. Training:**

Für das erste Training habe ich mich auf eine einzige Änderung beschränkt. Ich habe einfach das BPE-Vokabular auf 60000 vergrößert. Damit wollte ich besser unbekannten Wörtern entgegenwirken. Das Preprocessing bestand also aus Tokenisierung (mit Moses-Decoder), Truecasing (mit Moses-Decoder, trainiert auf den Trainingsdaten) und Byte-Pair Encoding (mit subword-nmt, trainiert auf den Trainingsdaten).

Das Training hat mehr als zweieinhalb Tage gedauert (von Freitagabend bis Montagmittag). Allerdings habe ich über 10 Epochen trainiert. Ich konnte leider keine Übersetzung durchführen und das Modell nicht evaluieren, da es mit dem Übernehmen eurer Änderungen am Code Probleme gab und dabei das Modell gelöscht wurde.

- **2. Training:**

Für das zweite Training habe ich mehr Sachen angepasst. Im Preprocessing habe ich ein gemeinsames BPE-Modell verwendet und dessen Vokabular noch einmal vergrößert (auf 89500). Den Rest des Preprocessings habe ich wie oben beschrieben durchgeführt. Bei diesem zweiten Versuch habe ich zusätzlich noch den Code in reader.py verändert. Die Input-Sequenz wird jetzt in umgekehrter Reihenfolge eingelesen. Dadurch sollte die Qualität der Übersetzung steigen, da die Distanz zwischen den ersten Wörtern der Input-Sequenz und den entsprechenden Wörtern in der Output-Sequenz kleiner wird.

Ich habe das Training am Dienstagabend nach dem Server-Neustart gestartet (von Montagmittag bis Dienstagabend hätte es wohl für kein Training gereicht). Es wurde dann am Mittwochabend durch den unerwarteten Neustart noch einmal unterbrochen, worauf ich es erneut laufen liess. Insgesamt dauerte dieses Training auch mehr als 2 Tage.

Leider habe ich am Abgabetag gemerkt, dass ich das Training nach dem Serverabsturz am Mittwoch mit den falschen Trainingsdaten laufen liess. Das Modell wurde ohne das vergrößerte / gemeinsame BPE-Modell trainiert, also nur mit dem reversing der Input-Sequenz.

- **3. Training**

Gerne hätte ich noch ein drittes Training durchgeführt. Da es aber oben genannte Verzögerungen gab und ich die Slots nicht dauernd besetzen wollte, verzichtete ich darauf. Ich hätte versucht, Dropout zu implementieren.

- **Übersetzung**

Das Resultat der Übersetzung des Testsets findet ihr im Abgabe-Ordner in meinem daikon-repository.

- **Evaluation**

Beim Evaluieren meines Modells auf dem Dev-Set hat score.py eine Perplexity von 8.41 ergeben.

Den BLEU-Score auf dem Dev-Set konnte ich aufgrund von Verbindungsproblemen mit dem Server nicht mehr rechtzeitig berechnen (die Übersetzung läuft noch). Ich werde ihn am Montagabend noch nachreichen.