MT Übung 5 – Encoder-Decoder-Modelle

Preprocessing

Unser Preprocessing hat sich eng am Preprocessing aus Übung 2 orientiert. Dabei haben wir die Texte:

1. Normalisierung:
perl mosesdecoder/scripts/tokenizer/normalize-punctuation.perl < corpus.dev.de > corpus.dev.norm.de
2. Tokenisierung:
perl mosesdecoder/scripts/tokenizer/tokenizer.perl -l de -q < corpus.dev.norm.de > corpus.dev.tok.de
3. Truecasing-Modell trainieren (wie bei Punkt 3 der Aufgabe beschrieben):
perl mosesdecoder/scripts/recaser/train-truecaser.perl --corpus corpus.train.tok.de –model truecase-model.de
4. Truecase-Modell anwenden:
perl mosesdecoder/scripts/recaser/truecase.perl --model truecase-model.de < corpus.tok.de > text.tc.de
5. BPE-Modell trainieren (wie in Schritt 3 der Aufgabe gezeigt), nun allerdings mit 100000 Symbolen:
python subword-nmt/learn_bpe.py -i corpus.train.tc.de -s 100000 -o bpe.codes.de
6. BPE-Modell anwenden
python apply_bpe.py -i text.tc.de -o text.bpe.de -c bpe.codes.de

Alle Preprocessing-Schritte wurden für alle Texte und alle Sprachen durchgeführt, ausser die Trainings der Truecasing-Modelle und des BPE-Modells, dies wurde jeweils nur mit den corpus.train-Files durchgeführt.

Hyperparameter

Wir führten das BPE-Training mit 100'000 Symbolen durch, dafür mussten wir einfach beim Aufruf des BPE-Training das Flag «-s» von 50'000 zu 100'000 ändern. Schlussendlich führte dies zu einer Trainingszeit von ca. 60 Stunden, weshalb weitere Adaptionen nicht realistisch überprüft werden konnten.

Zudem haben wir manuell die Anzahl der Epochen auch 6 beschränkt mit dem Flag «-e 6». Insgesamt waren bis etwas über 86'500 Iterationen pro Epoche nötig und pro 1000 Iterationen brauchte das Training ca. 7 Minuten.

Weitere Überlegungen

Weitere Adaptionen, die wir ausprobieren wollten, aufgrund der GPU-Situation aber nicht möglich war:

Wir hätten eigentlich auch noch gerne ausprobiert, was geschehen würde, wenn wir die Learning Rate erhöhen (bspw. 0.001 statt 0.0001) oder wenn wir ein extra LSTM-Layer in das *compgraph.py*-Script einbauen.