MT Dominik Martinez
Übung 4

Datenset
Als Datenset habe ich einige Werke von Goethe verwendet (siehe Auflistung). Die Texte habe ich aus dem Projekt Gutenberg kopiert und dabei gleich alle Gutenberg-Informationen (Lizenz etc.) nicht mitkopiert. Ich habe mich für Goethe-Texte entschieden, da ich ein Modell analog dem bekannten Shakespeare-Beispiel erstellen wollte.

Briefe aus der Schweiz
Faust I
Faust II
Die Leiden des jungen Werther I
Die Leiden des jungen Werther II
Wilhelm Meisters Lehrjahre Band I

Preprocessing
Zum Preprocessing könnte man als erstes das bereits genannte Ignorieren von Gutenberg-Lizenzen zählen. Danach folgte die Entfernung von newlines, das Splitten der Sätze, die Worttokenisierung, eine Normalisierung der Gross-/Kleinschreibung (auf Kleinschreibung) sowie das Aufteilen des Korpus auf ein Training- und ein Dev-Set. Dies habe ich alles in mit einem eigenen Python-Skript (preprocessing.py im romanesco-Ordner) gemacht. Zum Postprocessing: Auf recasing habe ich bewusst verzichtet.

Veränderungen am Code:
Zuerst habe ich mich auf Veränderungen am Preprocessing beschränkt, darunter auch Durchläufe mit einer unterschiedlichen Aufteilung von Train- und Dev-Set: immer 9:1, aber anders gemischt. Sowohl das Mischen als auch das Weglassen von Case-Normalizing hat aber keine grossen Veränderungen gezeigt. Dass unterschiedliches Mischen keinen Einfluss hat, zeigt, dass das Korpus gross genug ist, um statistische Fehler zu vermeiden.
Desweiteren habe ich versucht, statt einem LSTM-Modell ein GRU-Modell zu verwenden (kleine Änderung in compgraph.py). Dieser Versuch hat, entgegen den Resultaten bei anderen Studenten, eine schlechtere Perplexität gezeigt (ausser bei hidden_size = 2000, siehe Tabelle unten), weshalb ich die Änderung rückgängig gemacht habe.
Ebenso habe ich versucht, verschiedene Grössen für die Hidden Layers zu wählen (1000, 1500 und 2000). Auch die Perplexitäten aus dieser Variation kann man unten aus der Tabelle herauslesen. Die Verkleinerung auf 1000 hat, wie erwartet, schlechtere Resultate geliefert, während die Erhöhung auf 2000 der Erwartung entsprechend bessere Resultate lieferte. Ob sich der zusätzliche Ressourcen-Aufwand aber lohnt, resp. ob die leichte Verbesserung diesen Aufwand rechtfertigt, ist eine Frage der Anwendung.
Als letztes habe ich versucht, mit der Vokabulargrösse rumzuspielen. Die Versuche in der untenstehenden Tabelle haben keine eingeschränkte Grösse. Ich habe verschiedene explizite Werte getestet (z.B. 500 oder 1000), die Perplexität wurde dann allerdings mit «nan» angegeben. Dass der Fehler aber bei mir liegt und ich die Grösse nicht richtig angepasst habe, kann ich natürlich nicht ausschliessen.

Als optimale Kombination von Aufwand und Ertrag habe ich schlussendlich die unten hervorgehobene Kombination gewählt (also so wie ursprünglich im Code implementiert).

Perplexities:

	Hidden_size = 1000	Hidden_size = 1500	Hidden_size = 2000
BasicLSMTCell	-	**152.97**	147.64
GRUCell	213.34	183.43	146.92