Sarah Kiener
09-110-958

Maschinelle Übersetzung, Übung 4

Bericht

Datenset

Für mein Datenset habe ich alle 6 Romane von Jane Austen von Project Gutenberg heruntergeladen und in einem File aneinander gehängt. Ich habe diese Texte ausgewählt, weil sie sicherlich stilistisch gute, eher komplexe Sätze enthalten, ein grosses Vokabular aufweisen und wahrscheinlich nur wenige (Tipp-)Fehler enthalten.

Preprocessing

Im Proprocessing habe ich den Text zuerst in seine Sätze aufgeteilt und jeden Satz auf eine neue Zeile geschrieben. Anschliessend habe ich die Sätze mit dem NLTK-Tokenizer tokenisiert.

Anpassungen

Ich habe die verschiedenen Hyperparameter angepasst. Die untenstehende Tabelle enthält eine Übersicht über die vier durch geführten Trainings. Ich habe die Hyperparameter jeweils über die Kommandozeile mit den entsprechenden Flags angepasst. Deshalb enthält mein Code überhaupt keine Änderungen. Anstatt die Hyperparameter via Kommandozeile zu verändern, könnte man natürlich auch die entsprechenden Zahlen in der Datei const.py anpassen.

Zunächst einmal erschien es mir sinnvoll, die Anzahl Epochen zu erhöhen, da im ersten Training die Perplexität auch nach 10 Epochen noch kontinuierlich zu sinken schien. Ich erhöhte die Epochenzahl also auf 15 für alle nachfolgenden Trainings.

Im zweiten Training erhöhte ich ausserdem die Batch-Size auf 128. Gleichzeitig erhöhte ich die Vokabulargrösse auf 15‘000. Die gleichzeitige Erhöhung der Batch-Size und der Vokabulargrösse erwies sich als katastrophal: Die Perplexität war sowohl nach der 15. Trainingsepoche auf dem Train-Set wie auch auf dem Dev-Set wesentlich höher als nach dem ersten Training.

Fürs dritte Training halbierte ich die ursprüngliche Batch-Size deshalb auf 32. Die Vokabulargrösse beliess ich beim Standard von 10‘000. Nach 15 Epochen war die Perplexität auf dem Train-Set nun zwar deutlich tiefer. Auf dem Dev-Set dagegen war sie ein wenig höher als nach dem ersten Training. Vielleicht handelt es sich schon fast um ein Overfitting auf den Daten des Train-Sets.

Die Standard-Batch-Size von 64 scheint also bereits eine gute Wahl zu sein. Fürs vierte Training beliess ich die Batch-Size deshalb bei 64. Um nun den Effekt der veränderten Vokabulargrösse zu betrachten, setzte ich sie runter auf 7000. Dies hatte nun endlich eine positive Auswirkung auf die Perplexität, die gegenüber dem ersten Training um 12 Punkte sank. Allerdings kennt das Sprachmodell dadurch auch 3000 Wörter weniger. Es stellt sich also die Frage, ob es durch Herabsetzen der Vokabulargrösse im Endeffekt tatsächlich besser wurde.

Sarah Kiener
09-110-958

Training	Anpassungen	Perplexität	
		Train-Set nach letzter Epoche	Dev-Set
Training 1	Standard-Code	62.89	76.46
Training 2	–e 15 –b 128 –v 15000	102.45	107.82
Training 3	–e 15 –b 32	29.55	77.67
Training 4	–e 15 –v 7000	39.34	64.67

Probleme

Schwierig fand ich die Installation und Inbetriebnahme der verschiedenen Systeme (Google Cloud, Server aufsetzen, Tensorboard etc.). Ausserdem brauchte ich etwas Zeit, mich im romanesco-Code zurecht zu finden und den Code nachzuvollziehen. Nachdem das erste Training aber erfolgreich abgeschlossen war, fielen mir die weiteren Trainings leichter. Die Anpassung der Hyperparameter klappte sehr gut. Mit github hatte ich allerdings einige Probleme. Da ich zum ersten Mal mit github gearbeitet habe, war mir noch nicht so klar, wo sich nun welche Dateien befinden. Im Tutorat konnte ich diese Schwierigkeiten dann klären.

BEMERKUNGEN ZUR ABGABE

Ich habe für alle 4 durchgeführten Trainings die Loss-Übersichten aus Tensorboard sowie die generierten Sample-Texte in mein github-Repository geladen. Die Daten sind jeweils von 1 bis 4 durchnummeriert. Bei den Sample-Texten habe ich leider eine Datei falsch benannt: austen_sample.txt enthält den Text aus dem dritten Trainingsmodell.