MT Übung 4

Thema: RNNs

Anastassia Shaitarova, Anna Fertig

Für die Aufgabe haben wir die öffentlichen Untertiteln auf Englisch (OpenSubtitles von der Opus-Seite mit 2,2 MB) gewählt. Für uns war es interessant, mit diesem Datenset zu arbeiten, da der Inhalt der Texte sehr vielfältig ist, und die Texte aus ganz verschiedenen Bereichen kommen können. Und das haben wir sehr gut bemerkt, als wir Sampling gemacht haben. Ausserdem waren alle Sätze im Text von der Opus-Seite schon auf den einzelnen Zeilen, was uns die Arbeit beim Preprocessing sehr erleichtert hat.

Preprocessing

In dieser Übung haben wir uns am meisten auf das Preprocessing des Datensets konzentriert. Zuerst haben wir die Normalisierung des Textes gemacht. Das bedeutet, dass alle Sonderzeichen im Text normalisiert wurden, z.B. die gerichteten Anführungs- und Schlusszeichen vereinheitlicht wurden. Das haben wir auf dem Server mit dem folgenden Befehl gemacht:

perl $mosesdecoder/scripts/tokenizer/normalize-punctuation.perl < OpenSubtitles.txt > open_subtitles.norm.txt

Dann haben wir die Tokenisierung des Textes durchgeführt.

perl $mosesdecoder/scripts/tokenizer/tokenizer.perl -l en -q < open_subtitles.norm.txt > open_subtitles.tok.en.txt

Als nächstes haben wir das Truecasing auf dem Google Cloud Server gemacht. Dafür wurde zuerst der Moserdecoder auf dem Server eingerichtet:

git clone https://github.com/moses-smt/mosesdecoder

Dann wurde ein neues Truecase-Modell auf unseren Daten trainiert:

perl mosesdecoder/scripts/recaser/train-truecaser.perl --model truecase-model.en –corpus open_subtitles.tok.en.txt

Und als letztes haben wir in diesem Schritt den Text mit dem neuen Model truecast.

perl mosesdecoder/scripts/recaser/truecase.perl –model truecase-model.en < open_subtitles.tok.en.txt > open_subtitles.tc.txt

Als einen zusätzlichen Schritt haben wir noch BPE für den Text gemacht. Zuerst haben wir probiert, dass auf dem Server zu machen, leider hat es nicht richtig funktioniert, deshalb haben wir das Modell von Rico Sennrich auf unseren Computer lokal kopiert mit

git clone https://github.com/rsennrich/subword-nmt

Dann hat das Modell (learn_bpe.py) auf unserem Text gelernt und wir haben eine Datei mit den BPE-Zeichen (codes.bpe) erhalten.

python learn_bpe.py -i open_subtitles.tc.txt -o codes.bpe -s 10000

Hier haben wir die Vokabulargrösse von 10000 benutzt. Für Englisch nimmt man meistens die Vokabulargrösse zwischen 10000 und 100000, damit man mit diesen Parameter den BPE-Effekt nicht deaktiviert. Die Größe vom Vokabular entspricht im Allgemeinen der Anzahl der Regeln in BPE. Wir haben den Parameter 10000 für BPE ausgewählt und unser Vokabular hat die entsprechende Grösse. Die Ergebnisse haben gezeigt, dass diese Vokabulargrösse ausreicht.

wc codes.bpe
10001 20003 98732 codes.bpe

Und danach haben wir dieses Model auf unseren Text angewendet:

python apply_bpe.py -i open_subtitles.tc.txt -o open_subtitles.bpe.txt -c codes.bpe

Trainingsets und Training

So haben wir zwei Datensets für das Training benutzt, einmal den tokenisierten, normalisierten und getruecaseten Text und einmal den Text noch zusätzlich mit BPE. Mit diesen Experimenten wollten wir prüfen, ob BPE dann eine grosse oder eher kleine Verbesserung für das Sprachmodell bringt.

Weiter haben wir unsere Datensets auf das Trainingsset (mit 63471 Zeilen) und das Dev-Set (mit 7053 Zeilen) aufgeteilt und das Trainingsset trainiert.

Scoring und Sampling

Beim Scoring haben für das erste Dev-Set (ohne BPE) die Perplexität von **48.51** und für das zweite Dev-Set (mit BPE) die Perplexität von **69.31** bekommen. Diese Perplexität zeigt uns, wie überrascht unser Modell von dem Dev-Set ist. Je höher die Perplexität ist, desto schlechter ist unser Modell. Also wir hier bemerken konnten, ist das erste Modell, wo wir kein BPE verwendet haben, besser.

Beim Sampling haben wir neue Texte mit 200 Zeichen generiert. Wenn wir die Ergebnisse vergleichen, sehen wir, dass beide Modelle ihre Stärken und Schwächen haben. Im ersten Beispiel (ohne BPE) finden wir solche komischen Sätze wie **„I am a record of no one, their revolution of law I couldn, like if the Devil has the grownup, Houston, Jim Steel“**, aber auch gute Sätze: **„I advise that price“** oder **„I will stay home“**. Es gibt auch ein paar sehr lustige Sätze: **„See your bra!“** oder **„I curse about a couple of things“**. Es scheint, dass dieses System besser für die kürzeren Sätze funktioniert.

Auch im zweiten Beispiel (mit BPE) können wir einige sinnvolle Sätze finden: **„Hey, Jim, it ' s true...“**, **„You don ' t wanna do“**. Aber die meisten Sätze sind sehr komisch: **„I smell it extreme tw@@ se from me out of his hand emotional models all the for@@ our@@ irs traffic as purpose as records mate rial in my ro@@ ge ...“**. Das Zeichen @ hat etwas gestört, den Text zu verstehen. Wahrscheinlich mussten wir noch vor dem Training die Wörter mit diesem Zeichen wieder normalisieren.

Wie unser Experiment gezeigt hat, hat BPE keine wesentlichen Verbesserungen für das Sprachmodell gebracht.