Lehrstuhl für Informatik 6
der RWTH Aachen
Prof. Dr.-Ing. H. Ney

Aachen, den 13. April 2021
C. Herold, J. Rosendahl

Software-Projektpraktikum Maschinelle Übersetzung 1. Übung

Thema

Da die menschliche Bewertung von hypothetisierten Übersetzungen aufwendig und teuer ist, wird sie nur selten durchgeführt und, wo es möglich ist, werden automatisch berechenbare Metriken verwendet. Dazu vergleicht man den Ausgabesatz mit einer oder mehreren von Hand generierten Referenzübersetzungen.
Ziel dieses Aufgabenblatts ist es, ein Programm zu schreiben, welches WER, PER und BLEU für eine gegebene Menge von Referenz-Hypothese Paaren berechnet. Unter anderem muss dazu die Levenshtein-Distanz mithilfe von dynamischer Programmierung effizient berechnet werden.
Im Kontext dieses Aufgabenblatts verstehen wir „Wörter“ als die durch Leerzeichen getrennten Einheiten (oft „tokens“ genannt). In den gegebenen Daten sind die durch Leerzeichen getrennten Einheiten zu einem Großteil auch Wörter im klassischen Sinne, aber auch andere Einheiten wie z.B. Satzzeichen, Zahlen, etc.
Auf der Webseite zum Praktikum finden Sie die Dateien `newstest.de` und `newstest.en`, sowie die Hypothesen von drei Deutsch→Englisch Übersetzungssystemen `newstest.{hyp1,hyp2,hyp3}`.
Außer Aufgabe 1 sollen sämtliche Aufgaben in der Programmiersprache Python gelöst werden.

Aufgaben

1. Berechnen Sie folgende Korpus-Statistiken für Quell- und Zielseite der Testdaten (Dateien `newstest.de` und `newstest.en`): Anzahl der laufenden Wörter, Anzahl verschiedener Wörter, durchschnittliche Satzlänge. Setzen Sie dazu Unix-Tools wie `sed` und `wc` ein.
   Anmerkung: Unter „laufenden Wörter“ ist die Gesamtanzahl an Wörtern gemeint, wobei mehrfach Vorkommnisse eines gleichen Wortes auch mehrfach gezählt werden.

2. Schreiben Sie ein Programm, das die Levenshtein-Distanz anhand der in der Vorlesung vorgestellten Methode berechnet:

   - Legen Sie eine Matrix der Länge $J+1$ und der Breite $K+1$ an. In jedem Feld (j,k) soll später die Levenshtein-Distanz für die Strings $\tilde{e}_1^j$ und $\hat{e}_1^k$ stehen.
   - (Initialisierung) Initialisieren Sie die Koordinate $(0,0)$ mit 0.
   - Um zu einem Knoten (j,k) zu gelangen, gibt es drei relevante Übergänge aus vorher berechneten Knoten:
     - eine Löschung von Knoten $(j-1,k)$
     - eine Einfügung von Knoten $(j,k-1)$

– ein Matching/Substitution von Knoten $(j-1, k-1)$

Durchlaufen Sie die Matrix in geeigneter Weise und berechnen Sie zu jedem Knoten die minimalen Kosten aus diesen drei möglichen Vorgängerknoten.

- Lesen Sie aus dem Knoten (J, K) die Levenshtein-Distanz von zwei Sätzen ab.
- Geben Sie aus, welche Einfügungen, Auslassungen und Ersetzungen Ihr Programm in einem beliebig wählbaren Satz vorgenommen hat, um eine minimale Distanz zu erhalten.

3. Schreiben Sie ein Programm, das die BLEU Metrik berechnet:

   - Ein *n-gram* ist eine Sequenz von aufeinanderfolgenden n Wörtern. Schreiben Sie eine Funktion, die bei gegebenem n alle Übereinstimmungen von n-grams in der Hypothese mit denen der Referenz zählt. Wird ein n-gram in der Referenz gefunden, gilt es als verbraucht und wird nicht erneut benutzt.
   - Berechnen Sie die modifizierte n-gram Präzision P_n für eine Menge von L (Referenz, Hypothese) Paaren (r_l, h_l) als

$$P_n = \frac{\sum_{l=1}^{L} \sum_{n\text{-gram} \in h_l} \min\{\#\ n\text{-gram in } r_l, \#\ n\text{-gram in } h_l\}}{\sum_{l=1}^{L} \sum_{n\text{-gram} \in h_l} \#\ n\text{-gram in } h_l}$$

   - Schreiben Sie eine Funktion, die die sogenannte Brevity Penalty (BP) zurückgibt, wenn c die Länge der Hypothese und r die Länge der Referenz ist:

$$BP = \begin{cases} 1 & \text{if c} > \text{r} \\ e^{(1-r/c)} & \text{if c} \leq \text{r} \end{cases} \tag{1}$$

   - Berechnen Sie BLEU für $N = 4$ mittels der akkumulierten P_n als

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} \frac{1}{N} \log P_n\right). \tag{2}$$

4. Berechnen Sie WER, PER und BLEU für die Systeme mit den Hypothesen `newstest.hyp1`, `newstest.hyp2` und `newstest.hyp3`. Was ist die Rangfolge der zugrundelegenden Übersetzungssysteme?

Abnahmetermin: Montag, 26. April, Uhrzeit nach Absprache

Schriftliche Ausarbeitungen werden nicht verlangt. Um die Übung abzugeben legen Sie bitte ein **privates** git-Repository unter `https://git.rwth-aachen.de/` an und laden beide Betreuer ein (Adressen dazu siehe eMail). Bitte achten Sie darauf beiden Betreuer die „pull“ Erlaubnis zu geben. Sie können dieses Repository gerne für Ihre Arbeitsabläufe nutzen. Als finale Abgabe schicken Sie bitte die entsprechende commit-ID **bis zum Sonntag, (25. April, 23:59 Uhr)** an:

mtprak21assi@i6.informatik.rwth-aachen.de

Am Montag erläutern Sie uns dann Ihre Lösungen und demonstrieren Ihre Programme im Rahmen eines online-Anrufs (vmtl. mittels der Software zoom).