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

Aachen, den 26. Mai 2011
M. Huck, M. Freitag

Software-Projektpraktikum Maschinelle Übersetzung 4. Übung

Thema:

Durch die Wort-für-Wort-Übersetzung entsteht das Problem, dass Kontext-Informationen verloren gehen. In der Praxis wird daher üblicherweise ein phrasenbasiertes Übersetzungsmodell angewandt. Für ein wortaligniertes Satzpaar sind alle Teilstrings gültige Phrasen, wenn auf Quellseite keine anderen Wörter der Zielphrase, und auf Zielseite keine anderen Wörter der Quellphrase zugeordnet sind, d.h. alle Zuordnungen sind innerhalb eines festen Blocks. In dieser Aufgabe schreiben wir ein Programm, das die Extraktion auf Phrasenebene erweitert. Wir führen dann die Suche nach der besten Übersetzung mit Phrasen durch.
Neben dem wortbasierten und dem phrasenbasierten Übersetzungsmodell, die wir bis jetzt kennen gelernt haben, sind viele weitere Wissensquellen denkbar, die zur Verbesserung der Übersetzungsqualität beitragen könnten. Wir integrieren in dieser Aufgabe weitere Modelle in unser statistisches maschinelles Übersetzungssystem. Dazu kombinieren wir eine beliebige Anzahl von Modellen in einem log-linearen Framework:

$$\hat{e}_1^{\hat{I}} = \arg\max_{e_1^I} \left\{ p(e_1^I | f_1^J) \right\} \tag{1}$$

$$= \arg\max_{e_1^I} \left\{ \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J)\right)}{\sum_{\tilde{e}_1^I} \exp\left(\sum_{m=1}^M \lambda_m h_m(\tilde{e}_1^I, f_1^J)\right)} \right\} \tag{2}$$

$$= \arg\max_{e_1^I} \left\{ \sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J) \right\} \tag{3}$$

für Skalierungsfaktoren λ_m und Funktionen $h_m(e_1^I, f_1^J)$.
Da nicht alle Modelle gleich nützlich bzw. zuverlässig sind, sollen die Scores mittels der Skalierungsfaktoren λ_m, $m = 1, \ldots, M$, unterschiedlich gewichtet werden. Die Skalierungsfaktoren stellen wir so ein, dass die Ausgabe des Decoders direkt bezüglich einem der automatischen Fehlermaße optimiert wird. Dazu werden wir später im Praktikum geeignete Optimierungsalgorithmen einsetzen.

Aufgabe:

1. Erweitern Sie Ihr Training, d.h. Extraktion und Berechnung der relativen Häufigkeiten, auf Phrasen. Benutzen Sie (selbstimplementierte) Präfixbäume für das Speichern der Daten. Extrahieren Sie nur Phrasen bis zu einer festgelegten Maximallänge (z.B. höchstens drei Wörter auf Quell- und höchsten vier Wörter auf Zielseite). Schreiben Sie die phrasenbasierte

Übersetzungstabelle in einem Format analog zu Aufgabenblatt 1 in eine komprimierte Datei. Geben Sie negative Logarithmen der Wahrscheinlichkeiten aus. Setzen Sie die bekannte Pruning-Technik ein, um die Größe der Phrasentabelle zu verringern.

2. Erweitern Sie Ihr Übersetzungsprogramm auf Phrasen. Laden Sie dazu die Phrasentabelle wiederum in Ihren Präfixbaum.

3. Übersetzen und evaluieren Sie die Test-Sätze. Vergleichen Sie die Ausgabe mit der einzelwortbasierten Übersetzung. Um wieviel haben sich die Fehlermaße bzw. BLEU verbessert? Welche Fehler werden jetzt offensichtlich weniger gemacht? Geben Sie auch wieder die zehn (oder wahlweise hundert) besten Hypothesen aus.

4. Erweitern Sie Ihre Übersetzungstabelle um Scores mit

   - Phrase Penalty, $= 1$ für alle Einträge
   - Word Penalty, $= |e|$
   - einem Single Count Bit, $= 1$ falls $N(e)$ und $N(f) > 1$, sonst $= 0$
   - Source-Target Ratio, $= \frac{|f|}{|e|}$

   Passen Sie Ihre Phrasenextraktion entsprechend an.

5. Ergänzen Sie die Phrasentabelle um einzelwortbasierte Übersetzungskosten. Berechnen Sie dazu für alle Einträge in der Phrasentabelle zwei weitere Werte mittels der einzelwortbasierten Modelle aus Aufgabenblatt 1:

   - Source-to-Target einzelwortbasierte Übersetzungskosten auf Phrasenebene
   - Target-to-Source einzelwortbasierte Übersetzungskosten auf Phrasenebene

6. Erzeugen Sie ein Unigramm-Sprachmodell basierend auf relativen Häufigkeiten, d.h. $p(e) = \frac{N(e)}{N(\cdot)}$, wobei $N(\cdot)$ die Anzahl der laufenden Wörter auf der Zielseite der Trainingsdaten bezeichnet. Geben Sie negative Logarithmen aus. Die Unigramm-Sprachmodell-Kosten können auf Phrasenebene vorberechnet werden. Erweitern Sie Ihre Phrasentabelle um einen entsprechenden Eintrag.

7. Setzen Sie die zusätzlichen Modelle in Ihrem Decoder ein. Erweitern Sie Ihren Suchalgorithmus um (als Kommandozeilenparameter übergebene) Skalierungsfaktoren, mit denen die verschiedenen Kosten multipliziert werden. Geben Sie in Ihren n-best-Listen die Kosten, die jedes Modell für die jeweilige Hypothese insgesamt liefert, einzeln an, ohne dabei den Skalierungsfaktor mit einzubeziehen. Experimentieren Sie von Hand mit verschiedenen Skalierungsfaktoren. (Setzen Sie z.B. die Skalierungsfaktoren für die insgesamt vier phrasen- und einzelwortbasierten Übersetzungsmodelle und das Unigramm-Sprachmodell auf je 0.2 und alle übrigen auf 0.0.) Evaluieren Sie die Übersetzungsqualität der Hypothesen mittels der bekannten automatischen Metriken.

8. Beurteilen Sie die Software-Architektur einer anderen Gruppe dieses Praktikums. Achten Sie auf Dokumentation, Verständlichkeit, Modularität und allgemeine Sauberkeit des Codes und verfassen Sie dazu eine Zusammenfassung in der Größenordnung von einer maschinell geschriebenen DIN A4 Seite.

Abnahmetermin: Donnerstag, 9. Juni, ab 14:00 Uhr

Schriftliche Ausarbeitungen werden nicht verlangt. Schicken Sie bitte Ihre kommentierten Quelltexte bereits bis Mittwoch Abend (8. Juni, 18:00 Uhr) an

huck@i6.informatik.rwth-aachen.de.

Am Donnerstag erläutern Sie uns dann Ihre Lösungen und demonstrieren Ihre Programme.