Praktikum Maschinelle Übersetzung – Language Model

Um die Aufgaben auszuführen, können sie ihre Daten in folgendem Verzeichnis speichern:

/project/smtstud/ss17/systems/USERNAME/

Wir werden verschiedene Sprachmodelle von Daten trainieren, die aus der Tourismus Domäne stammen. Sehen Sie sich die Trainingsdaten in deutsch und englisch einmal an, um einen Eindruck davon zu gewinnen, womit Sie arbeiten:
/project/smtstud/ss17/data/corpus/train.de-en.1-99.de
/project/smtstud/ss17/data/corpus/train.de-en.1-99.en

Später werden wir testen, wie gut/schlecht dieses Modell auf Daten aus einer aufgenommenen Vorlesung
/project/smtstud/ss17/data/dev/lecture.en.Final.sc
im Vergleich zu den Tourismus Daten passt und wie wir dies durch Interpolation mit einem weiteren LM, das auf Parlamentsdebatten trainiert wurde, verbessern können.

Bitte melden Sie sich nach Abschluss jeder Aufgabe, sodass wir ausschließen können, dass Sie mit fehlerhaften Dateien weitermachen.

1. Trainieren eines Language Models

Wir trainieren das Language Model (LM) mittels des SRI Language Model Toolkit (SriLM). Informationen dazu finden Sie unter:

http://www-speech.sri.com/projects/srilm/

1.a) Trainieren Sie zunächst 4 englische Sprachmodelle (1 bis 4-gram) mittels des Programms: /project/smtstud/ss17/bin/**ngram-count**
Information zu den Parametern für das Programm finden Sie unter:
http://www-speech.sri.com/projects/srilm/manpages/ngram-count.1.html

– trainieren Sie ein „open vocabulary“ LM
– benutzen Sie (modified) Kneser-Ney-Discounting (für alle n-gramme)
– aktivieren sie die Interpolation der verschieden n-gram Wahrscheinlichkeiten (für alle n-gramme)
– lesen sie den Trainingstext aus dieser Datei:
/project/smtstud/ss17/data/corpus/train.de-en.1-99.en
– schreiben Sie ein LM des Formats: „backoff N-gram model“ in eine Datei

Schauen Sie sich den Inhalt der Datei für das 4-gram LM an:
(Hinweis: das default LM file format ist plain text)
– Wieviele uni-gramme, bi-gramme, tri-gramme und 4-gramme befinden sich im 4-gram LM?
– Wie sind die Einträge in der Modelldatei zu lesen?
http://www-speech.sri.com/projects/srilm/manpages/ngram-format.5.html

1.b) Trainieren Sie nun ein 4-gram LM mit den zusätzlichen Parametern:
–gt*3*min 1 –gt*4*min 1

- Wie ändern sich die Anzahlen der n-gramme und warum?
- Was lässt sich daraus ablesen?

(Hinweis: Parameter in der manpage: –gt*n*min. Die default Werte für diesen Parameter lassen sich per ngram-count –help herausfinden.)

1.c) Trainieren Sie nun ein 4-gram LM mit den selben Parametern für die deutschen Trainingsdaten:
/project/smtstud/ss17/data/corpus/train.de-en.1-99.de

- Wie ändern sich die Anzahlen der n-gramme und warum?

ngram order	**EN with default cutoff**	**EN no cutoffs**	**DE**
ngram 1			
ngram 2			
ngram 3			
ngram 4			

2. Berechnung der Preplexität eines Language Models

Wie in der Vorlesung besprochen, kann man die Qualität von LMs mittels der Perplexität für ein Testdatendokument vergleichen. Dazu können Sie das Programm
/project/smtstud/ss17/bin/**ngram**
benutzen.

Informationen zu den Parametern zu diesem Programm finden sie unter:
http://www-speech.sri.com/projects/srilm/manpages/ngram.1.html
Hinweis: Benutzen Sie den Parameter **-ppl**
Die Testdaten finden Sie unter:
/project/smtstud/ss17/data/test/test.btec.en.Final.sc

2.a) Berechnen Sie die Perplexität der Testdaten mittels aller englischen LMs, die Sie in Aufgabe 1 trainiert haben.

- Wie verändert sich die Perplexität für LMs aufsteigender Ordnung?
- Welche Schlussfolgerung lässt sich daraus ziehen?

2.b) Berechnen Sie die Perplexität dieser deutschen Testdaten:
/project/smtstud/ss17/data/test/test.btec.de.Final.sc
mittels des deutschen LM aus Aufgabe 1.

- Wie verhält sich die Perplexität im Vergleich zum englischen LM und warum?

LM	ppl
1-gram LM	
2-gram LM	
3-gram LM	
4-gram LM	
4-gram LM without cutoffs	

2.c) Trainieren Sie drei weitere 4-gram Language Modele ohne den interpolate parameter. Das erste Model ohne Discounting, und das zweite mit Witten-Bell Discounting, und das letzte mit modified Kneser-Ney Discounting. Dann berechnen Sie die Perplexitäten auf den gleichen Testdaten wie in 2.c. Welche Technik ist die beste?

Technik	ppl
No discounting	
Witten-bell	
Kneser-Ney	

3. Interpolieren von Language Models

Gegeben sind:
Zwei 4-gram Sprachmodelle, die auf Textdaten aus unterschiedlichen Quellen trainiert wurden:
- das englische 4-gram LM aus Aufgabe 1 und
  /project/smtstud/ss17/models/big.4.en.unk.srilm

Testdaten, die aus einer weiteren unterschiedlichen Quelle stammen. Diese Daten wurden in ein development set und ein test set zerlegt:
/project/smtstud/ss17/data/dev/lecture.en.Final.sc
/project/smtstud/ss17/data/test/lecture.en.Final.sc

Zwei LMs können mittels des Programms:
/project/smtstud/ss17/bin/**compute-best-mix**
/project/smtstud/ss17/bin/**ngram**

interpoliert werden.
Informationen zum Benutzung dieses Programms finden sie unter:
http://www.speech.sri.com/projects/srilm/manpages/ppl-scripts.1.html

3.a) Berechnen Sie die optimalen Interpolationsgewichte für die beiden LMs für das gegebene development set.
Interpolieren Sie die beiden LMs gleichgewichtet (0.5 0.5) und mit optimalen Interpolationsgewichten.

Hinweis: Das interpolieren zweier LMs erfordert 3 Schritte:

- Erstellen je einer Zwischendatei mittels ngram -debug 2 –ppl –order … für jedes Modell,
- Berechnen der optimalen Interpolationsgewichte mittels compute-best-mix und
- Erstellen des interpolierten LM mittels ngram –lm … -mix-lm …

    - Welche Perplexität hat das development set für die beiden einzelnen LMs?
      (Hinweis: Steht in der von ngram –debug 2 … ausgegebenen Datei)
    - Wie verändert sich die Perplexität zwischen der gleich gewichteten Anfangsinterpolation und der mit optimalen Gewichen?
    - Was sind die Werte der beiden Lambdas?

3.b) Berechnen Sie die Perplexitäten für die beiden einzelnen LMs und die beiden interpolierten LMs für die Testdaten, die nicht zur Interpolation verwendet wurden.

- Wie verändert sich die Perplexität in diesem Fall?

LM	**ppl dev**	**ppl test**
small		
big		
interpol-uniform		
interpol-tuned		
One model (all data)		

4. Verwendung des Language Model für Hypothesis Ranking

Der folgende Quellsatz wurde mit einem unserer Syteme übersetzt: „gibt es einen freien Tisch für drei Personen um acht Uhr heute Abend ?“

4.a) Benutzen Sie das letzte Language Model (interpol-tuned) um die folgenden Hypothesen, die von unserem System erstellt wurden, zu ranken:
Hypothese 1: „is there a table available for three at eight tonight ?“
Hypothese 2: „is there a table available for three at eight o 'clock this evening ?“
Hypothese 3: „is there a table available for three people at eight o 'clock tonight ?“
Hypothese 4: „there is a table available for three at eight tonight ?“
Hypothese 5: „is there a table available for three at eight p.m. Tonight ?“

4.b) Was können Sie daraus schließen?

Hypothesis	**Mein Rank**	**LM Score/ppl**	**LM Rank**
H1			
H2			
H3			
H4			
H5			

Hinweis:
– Benutzen Sie die Parameter -ppl - -debug 1
– Sie können auch alle Hypothesen in einem echo-Befehl mit „-e“ und „\n“ Zeichen kombinieren.