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
Domain 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 Model

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 Sei 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?

Antworten für Aufgabe 1:

```
ngram-count -order 1 -unk -kndiscount -interpolate -text train.de-en.1-99.en -lm small.1gram.lm
 …
ngram-count -order 4 -unk -kndiscount -interpolate -text train.de-en.1-99.en -lm small.4gram.lm
ngram-count -order 4 -unk -kndiscount -interpolate -text train.de-en.1-99.en -gt3min 1 -gt4min 1 -lm small.4gram.nocutoff.lm
```

ARPA format for N-gram backoff models:
probability(logarithm) w1 w2 (n-gram) [back-off-weight]

Default cutoff für 3- und 4-grams ist 2, d.h. alle 3- und 4-grams, die nur einmal vorkommen, werden nicht ins Modell aufgenommen, so lässt sich im Vergleich der Zahlen aus den beiden Modellen die Anzahl aller 3- und 4-grams, die nur einmal vorkommen, direkt ablesen.
Das Deutsche hat wegen der Morphologie ein höheres Vokabular und somit auch mehr ngramme.

ngram order	EN with default cutoff	EN no cutoffs	DE
ngram 1	12 830	12 830	19 227
ngram 2	107 378	107 378	145 793
ngram 3	67 102	253 472	308 864
ngram 4	78 289	362 002	411 693

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 könne 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?

Antworten für Aufgabe 2:

```
ngram -lm small.4gram.noCutoff.lm -order 4 -unk -ppl test.btec.en.Final.sc
```

LM	ppl
1-gram LM	230.71
2-gram LM	36.25
3-gram LM	23.75
4-gram LM	22.20
4-gram LM without cutoffs	20.14

Komplexere Modelle modellieren die Daten besser, daher sinkt ihre Perplexität.

PPL-DE (no cut off): 33.56
Die Perplexität sinkt für Modelle höherer Ordnung, da das größere Modell die Daten
besser modelliert. Anschaulich: es ist einfacher das nächste Wort zu „raten“ wenn man
mehr Wörter aus der history zur Verfügung hat.
Das deutsche Modell hat wegen des größeren Vokabulars einen deutlich höheren
Verzweigungsgrad.

2.c)

Technik	Ppl
No discounting	23.85
Witten bell	21.81
Kneser ney	22.28

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?

Antworten für Aufgabe 3:

```
ngram –lm small.4gram.noCutoff.lm –order 4 –unk –ppl dev/lecture.en.Final.sc -debug 2 >
small.4gram.noCutoff.lm.ppl.log

ngram –lm big.4.en.unk.srilm –order 4 –unk –ppl dev/lecture.en.Final.sc -debug 2 > big.4.en.unk.srilm.ppl.log

compute-best-mix small.4gram.noCutoff.lm.ppl.log big.4.en.unk.srilm.ppl.log

ngram –order 4 –unk –lm small.4gram.noCutoff.lm -mix-lm big.4.en.unk.srilm –lambda 0.5 -write-lm
Interpol.uniform.4gram.srilm

ngram –order 4 –unk –lm small.4gram.noCutoff.lm -mix-lm big.4.en.unk.srilm –lambda 0.32 -write-lm
Interpol.uniform.4gram.srilm
```

LM	ppl dev	ppl test
small	505.71	553.96
big	296.75	224.01
interpol-uniform	261.31	215.14
interpol-tuned	254.42	203.97
One model (all data) (`all.srilm`)	291.47	218.88

Die Testdaten aus der lecture-domain getestet mit dem Tourismus-LM haben eine deutlich
höhere Perplexität als die in-domain Tourismus Daten.
Die Perplexität sinkt mit optimierten Gewichten.
Lambda(big)= 0.67685; Lambda(small)=0.32315
Die Perplexität sinkt normalerweise für ungesehene Daten weniger stark.

Antworten zu Fragen, die Studenten gestellt hatten:
Q: Was bedeutet ppl1 in der Ausgabe von ngram –ppl?
A: Das ist die Perplexität des Dokuments, wenn man das Satzendesymbol </s> nicht als
implizit gegeben ansieht.

Q: Warum haben nicht alle 1-grams bzw. 2-grams ein backoff weight?
A: Backoff weights werden nur im Modell angegeben, wenn sie als Präfix eines längeren ngrams vorkommen. Für alle anderen ist das backoff weight implizit 1 (oder 0 in der
Logarithmusrepräsentation). Diese Konvention verkleinert die Dateigröße erheblich.

Q: Warum ist die Anzahl von 3-grammen im 3gram LM und im 4gram LM unterschiedlich, obwohl beide Male cutoff 2 für 3-gramme gilt?
A: Beim KN-Discounting wird das Discounting auf die counts der n-gramme angewendet, bevor der cutoff angewendet wird. Dadurch fallen im 4gram LM mehr 3-gramme heraus. Im 3gram LM sind die 3-gramme die n-gramme höchster Ordnung, d.h. die counts werden hier nicht durch das Discounting verändert.

4. Verwendung des Language Model für Hypothesis Ranking

Hyp1: 14.47, rank 2

hyp2: 11.96, rank 1

hyp3: 19.66, rank 4

hyp4: 14.63, rank 3

hyp5: 51.96, rank 5