Khushi K. Shah
60004180045
BE - A

NLP
Assignment - 1

Q1] Differentiate between Interpolation and Backoff.

Ans: Backoff

- Backoff N-gram modelling is a non-linear method
- We build on N-gram model based on (N-1) gram model.
- The difference is that, in backoff, if we non-zero trigram counts we solely rely on trigram counts and don't interpolate the bigram and unigram counts at all.
- Backoff model in trigram format:

$$P(w_i \mid w_{i-2} w_{i-1}) = \begin{cases} \tilde{P}(w_i \mid w_{i-2} w_{i-1}) & \text{if } C(w_{i-2} w_{i-1} w_i) > 0 \\ \alpha(w_{n-2}^{n-1}) \cdot \tilde{P}(w_i \mid w_{i-1}) & \text{if } C(w_{i-2} w_{i-1} w_i) = 0 \text{ and } C(w_{i-1} w_i) > 0 \\ \alpha(w_{n-1}) \cdot \tilde{P}(w_i) & \text{otherwise} \end{cases}$$

- Doesn't yield valid probability distribution
- Works shockingly well for huge datasets.

Interpolation

- This method combines different N grams by linearly interpolating all 3 models whenever we are computing any trigram.
- Here, we don't train 3 λ's as trigram grammar. Instead we make each λ a function of the context
- λ terms are used to decide how much to smooth.
- $\sum_i \lambda_i = 1$
- Mathematically,

$$\tilde{P}(w_0 \mid w_{-2} w_{-1}) = \lambda_3 \cdot P(w_0 \mid w_{-2} w_{-1}) + \lambda_2 \cdot P(w_0 \mid w_{-1}) + \lambda_1 \cdot P(w_0)$$

- Can interpolate 'customised' models with 'general' model.

Khushi K. Shah
60004180045
BE - A

Q2] Viterbi algorithm.

Ans: Viterbi algorithm is a variation of the forward algorithm which considers all words simultaneously in order to compute the most likely path.

Algorithm:
Input: observations of length T, state-graph of length N
Output: best path

for each state s from 1 to N do
$$q[1, s] \leftarrow P(s \mid s_0) \cdot P(o_1 \mid s)$$
$$backpointers[1, s] \leftarrow 0$$

for each time step t from 2 to T do
for each state s from 1 to N do
$$q[t, s] \leftarrow \max_{s'=1}^{N} q[t-1, s'] \cdot p(s \mid s') \cdot p(o_t \mid s)$$

$$backpointers[t, s] \leftarrow \operatorname{argmax}_{s'=1}^{N} q[t-1, s'] \cdot p(s \mid s')$$

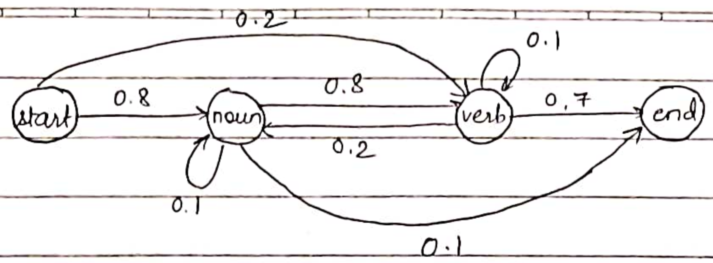
$s \leftarrow \operatorname{argmax}_{s'=1}^{N} q[T, s']$
return the backtrace path from backpointers [T, s]

Example:
Consider a two word language: 'fish' and 'sleep'
Suppose in our training corpus,
'fish' appears 8 times as a noun and 5 times as a verb.
'sleep' appears twice as a noun and 5 times as a verb

∴ Emission probabilities:

- Noun
  - P (fish | noun) : 0.8
  - P (sleep | noun) : 0.2
- Verb
  - P (fish | verb) : 0.5
  - P (sleep | verb) : 0.5

Khushi K. Shah
60004180045
BE - A



Token 1 : fish

	0	1	2	3
~~start~~ start	1	0		
verb	0	0.2 x 0.5		
noun	0	0.8 x 0.8		
end	0	0		

Token 2 : sleep

	0	1	2	3
start	1	0	0	
verb	0	0.1	0.64 x 0.8 x 0.5 ← max ✓ 0.1 x 0.1 x 0.5	
noun	0	0.64	0.64 x 0.1 x 0.2 ← max ✓ 0.1 x 0.2 x 0.2	
end	0	0	–	

Token 3 : end

	0	1	2	3
start	1	0	0	0
verb	0	0.1	0.256	–
noun	0	0.64	0.0128	–
end	0	0	–	0.256 x 0.7 ← max ✓ 0.0128 x 0.1

∴ Now we can backtrack the most likely path.

Khushi K. Shah
60004180045
BE-A

Q3] Corpus:

<S> I am from DJ </S>
<S> I am a teacher </S>
<S> All students are good and intelligent </S>
<S> Students from DJ score high marks </S>

Test data

<S> students are from DJ </S>

Ans: Unigram

	<S>	students	are	from	DJ	</S>
	4	2	1	2	2	4

Bigram

First we find occurrence count

	<S>	students	are	from	DJ	</S>
<S>	0	1	0	0	0	0
students	0	0	1	1	0	0
are	0	0	0	0	0	0
from	0	0	0	0	2	0
DJ	0	0	0	0	0	1
</S>	0	0	0	0	0	0

Bigram

	<S>	students	are	from	DJ	</S>
<S>	0	1/4	0	0	0	0
students	0	0	1/2	1/2	0	0
are	0	0	0	0	0	0
from	0	0	0	0	2/2 = 1	0
DJ	0	0	0	0	0	1/2
</S>	0	0	0	0	0	0

Khushi K. Shah
60004180045
BE - A

Using MLE to estimate probability for test data.

$$P = P(\text{students} / s) \times P(\text{are} / \text{students}) \times P(\text{from} / \text{are}) \times P(\text{DJ} / \text{from}) \times P(</s> / \text{DJ})$$

$$= \frac{1}{4} \times \frac{1}{2} \times 0 \times 1 \times \frac{1}{2}$$

↑ hence we need to apply laplace smoothening.

before applying laplace, we find
V = count of unique vocabulary in corpus
= count ({ <s>, </s>, I, am, from, DJ, a, teacher, all, students, are, good, and, intelligent, score, high, marks })
= 17

$$\therefore P = \left(\frac{1+1}{4+17}\right) \times \left(\frac{1+1}{2+17}\right) \times \left(\frac{0+1}{1+17}\right) \times \left(\frac{2+1}{2+17}\right) \times \left(\frac{1+1}{2+17}\right)$$

$$= \frac{2}{21} \times \frac{2}{19} \times \frac{1}{\cancel{18}\ \cancel{6}\ 3} \times \frac{\cancel{3}}{19} \times \frac{\cancel{2}}{19} = \boxed{9.257 \times 10^{-6}}$$

Khushi K. Shah
60004180045
BE-A

Q4] Corpus:

<S> I am Sam </S>

<S> Sam I am </S>

<S> I do not like green eggs and ham </S>

a) Calculate bigram probability for ① P(am / Sam) ② P(do / I) ③ P(am / I)

soln:
$$P(w_n \mid w_{n-1}) = \frac{C(w_{n-1} w_n)}{C(w_{n-1})}$$

① $P(am / Sam) = \frac{P(Sam\ am)}{P(Sam)} = \frac{0}{2} = 0$

② $P(do / I) = \frac{P(I\ do)}{P(I)} = \frac{1}{3}$

③ $P(am / I) = \frac{P(I\ am)}{P(I)} = \frac{2}{3}$

b) Calculate the trigram probability for 'I am Sam'

soln:
$$P(w_n \mid w_{n-2} w_{n-1}) = \frac{C(w_{n-2} w_{n-1} w_n)}{C(w_{n-2} w_{n-1})}$$

$$P(Sam / I\ am) = \frac{C(I\ am\ Sam)}{C(I\ am)} = \frac{1}{2}$$

c) Calculate MLE for 'I am Sam' using bigram

soln: (<S>, I), (I, am), (am, Sam), (Sam, </S>)

MLE = P(I / <S>) × P(am / I) × P(Sam / am) × P(</S> / Sam)

$$= \frac{2}{3} \times \frac{2}{3} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{9}$$

Khushi K. Shah
60004180045
BE - A

Q5] Corpus:

<S> John read Moby Dick </S>
<S> Mary read a different book </S>
<S> She read a book by Cher </S>

a) Calculate MLE for 'John read a book' using bigram
b) Calculate MLE for 'Cher read a book' using bigram

Soln: MLE for 'John read a book'

(<S>, John), (John, read), (read, a), (a, book), (book, </S>)

$$MLE = P(John/\langle S\rangle) \times P(read/John) \times P(a/read) \times P(book/a) \times P(\langle /S\rangle/book)$$

$$= \frac{1}{3} \times \frac{1}{1} \times \frac{2}{3} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{18} = \boxed{0.056}$$

MLE for 'Cher read a book'

(<S>, Cher), (Cher, read), (read, a), (a, book) (book, </S>)

$$MLE = P(Cher/\langle S\rangle) \times P(read/cher) \times P(a/read) \times P(book/a) \times P(\langle /S\rangle/book)$$

$$= \frac{0}{3} \times \frac{0}{1} \times \frac{2}{3} \times \frac{1}{2} \times \frac{1}{2}$$

using add-one smoothing (Laplace) $\frac{count+1}{count+V}$

total no. of unique tokens = V = 11

$$= \frac{0+1}{3+11} \times \frac{0+1}{1+11} \times \frac{2+1}{3+11} \times \frac{1+1}{2+11} \times \frac{1+1}{2+11}$$

$$= \frac{1}{14} \times \frac{1}{\cancel{12}\,\cancel{4}\,2} \times \frac{\cancel{3}}{14} \times \frac{\cancel{2}}{13} \times \frac{\cancel{2}}{13} = \boxed{3.019 \times 10^{-5}}$$