Natural Language processing

Beesa Vaishnavi
700772902

Home-work-2.

Q.1. Sol 1. priors $P(-) = 3/5$, $P(+) = 2/5$, $|V| = 20$.

$P_{wi/c}$ negative $n_- = 14$

positive $n_+ = 9$

$$P(w_i|c) = \frac{\text{count}(w_i, c) + 1}{(\sum_{w \in V} \text{count}(w, c)) + |V|}$$

$$P(\text{"Predictable"}/-) = \frac{1+1}{14+20}$$

$$P(\text{"no"}/-) = \frac{1+1}{14+20}$$

$$P(\text{"fun"}|-) = \frac{0+1}{14+20}$$

$$P(\text{"Predictable"}/+) = \frac{0+1}{9+20}$$

$$P(\text{"no"}/+) = \frac{0+1}{9+20}$$

$$P(\text{"fun"}/+) = \frac{1+1}{9+20}$$

2. scoring the test set

$$P(-) \cdot P(s|-) = 3/5 \times \frac{2 \times 2 \times 1}{34^3} = 6 * 1 \times 10^{-5}$$

$$P(+) \cdot P(s|+) = 2/5 \times \frac{1 \times 1 \times 2}{29^3} = 3.2 \times 10^{-5}$$

system should assign negative class.

Q3 sol^n → 1. formula: $P(w_i|w_{i-1}) = \frac{Count(w_{i-1}, w_i)}{Count(w_{i-1})}$

from <s>

Total = 2(I) + 1(deep) = 3

P(I | <s>) = 2/3

from I

Total = 2 (love)

P(love | I) = 2/2 = 1

from love

Total = 1(NLP) + 1(deep) = 2

P(NLP | Love) = 1/2

P(deep | love) = 1/2

from deep

Total = 2 (learning)

P(learning | deep) = 2/2 = 1

from Learning

Total = 1(</s>) + 1(is) = 2

P(</s> | Learning) = 1/2

from NLP

Total = 1 (</s>)

P(</s> | NLP) = 1/1 = 1

* Sentence s1

<s> I Love NLP </s>

P = P(I | <s>) × P(love | I) × P(NLP | Love) ×
P(</s> | NLP)

= (2/3) × (1) × (1/2) × (1)

= (2/3) × (1/2) = 2/6 = 1/3

$P(S1) = 1/3 \approx 0.33$

* Sentence S2

<s> I love deep learning </s>

$P = (2/3) \times (1) \times (1/2) \times (1) \times (1/2)$

$= (2/3) \times (1/2) \times (1/2)$

$= 2/12$

$= 1/6$

$P(S2) = 1/6 \approx 0.17$

$\because 1/3 > 1/6$

So,

Sentence S1 is more probable.

2. Zero-probability Problem.

MLE Probability.

from the table after ate:

Total = 6 + 3 + 2 + 1 = 12

"noodle" does not appear

$P(\text{noodle}|\text{ate}) = 0/12 = 0$

Why is this a problem?

If one word in a sentence has probability 0:

Entire sentence probability = 0

This makes:

- sentence probability = 0
- perplexity = infinite. Even if the sentence is reasonable, the model rejects it completely.

3. Laplace (Add-1) smoothing

formula: $p(w|ate) = \frac{Count(w, ate) + 1}{Total + V}$

Given. count = 0

- Total = 12
- Vocabulary size V = 10

$P(noodle|ate) = \frac{0+1}{12+10} = 1/22 \approx 0.045$

Q4 sol^n * Compute P(cats | I, like)

Using trigram MLE:

$P(w_3|w_1, w_2) = \frac{Count(w_1, w_2, w_3)}{Count(w_1, w_2)}$

from corpus:

- Count (I like cats) = 1
- Count (I like) = 2

P(cats | I, like) = 1/2 = 0.5

* Compute P(dogs | You, like) using Backoff

step ①: check trigram

- Count (You like dogs) = 0
- Count (You like) = 1

P(dogs | You, like) = 0

since trigram count is 0, we back off to bigram.

step ②: Use Bigram

$P(w|like) = \frac{Count(like, w)}{Count(like)}$

Total after like:
- like cats = 2
- like dogs = 1

Total = 3

P(dogs | like) = 1/3

P(dogs | You, like) = 1/3.

* Why is Backoff necessary?

Backoff is needed because:
- some trigrams (like You like dogs) never appear in training data.
- MLE gives probability 0.
- Zero probability would make the entire sentence probability zero.

Backoff solves this by:

Using a simpler model (bigram) when the trigram is unseen. This prevents zero probabilities and make the model more robust.

Q5 sol[n]:- *Per-class Precision & Recall:

confusion matrix:

system \ Gold	Cat	Dog	Rabbit
Cat	5	10	5
Dog	15	20	10
Rabbit	0	15	10

Total samples = 90.

→ CAT

- TP = 5
- Predicted Cat (row total) = 5 + 10 + 5 = 20
- Actual Cat (Column total) = 5 + 15 + 0 = 20

Precision(cat): 5/20 = 0.25

Recall (cat): 5/20 = 0.25

→ DOG

- TP = 20
- Predicted Dog = 15 + 20 + 10 = 45
- Actual Dog = 10 + 20 + 15 = 45

Precision (Dog): 20/45 0.444

Recall (Dog): 20/45 0.444

→ RABBIT

- TP = 10
- Predicted Rabbit = 0 + 15 + 10 = 25
- Actual Rabbit = 5 + 10 + 10 = 25

Precision (Rabbit): 10/25 = 0.40.

Recall (Rabbit): 10/25 = 0.40

* Macro vs Micro Averaging

Macro - Averaged

Average across classes:

Precision: (0.25 + 0.444 + 0.40) / 3 0.365

Recall: (0.25 + 0.444 + 0.40) / 3 0.365

Micro-Averaged:-

Micro uses global totals.

Total TP = 5 + 20 + 10 = 35

Micro precision: 35/90 = 0.389

Micro Recall: 35/90 = 0.389

Difference:

⇒ Macro Averaging treats all classes equally.

⇒ Micro Averaging gives more weight to larger classes (uses total counts).

Macro is better when class balance matters.

Micro reflects overall system accuracy.