700770810

Aragonda Gayathri

Q1 1) Test Document: "Predictable no-fun"

Vocabulary Size (V) = 20

Negative class

Total tokens = 14 → Denominator = 14 + 20 = 34

Prior : $P(-) = 3/5 = 0.6$

Word likelihoods using Add-1 smothing

$P(\text{Predictable}|-) = (2+1)/34 = 3/34 \approx 0.0882$

$P(\text{no}|-) = (0+1)/34 = 1/34 \approx 0.0294$

$P(\text{fun}|-) = (0+1)|34 = 1/34 \approx 00294$

Score for Negative class $= 0.6 \times 0.0882 \times 0.0294 \times 0.0294$

$= 0.0000458$

Positive class:

Total tokens = 12 → Denominator = 12 + 20 = 32

Prior $P(+) = 2/5 = 0.4$

All words are unseen in the positive class

$P(w|+) = (0+1)/32 = 1/32 = 0.03125$

Score for positive class $= 0.4 \times 0.03125 \times 0.03125 \times$

0.03125

$= 0.0000122$

2) Compare the scores

$-P(-|d) \approx 0.000458$

$-P(+|d) \approx 0.0000122$

Since $P(-|d) > P(+|d)$, the system should assign the document to the : Negative class

Q2 1) → Representational harm occurs when AI systems or classifiers reinforce societal stereotypes or treat certain groups unfairly in their representation.

→ Kiritchenko + Mohammad (2018) studied sentiment analysis models and found that these systems assigned more negative sentiment scores to names commonly associated with African American individuals - Even when the names where not in negative context

→ This is representational harm because the classifier was biased against names linked to a specific racial group, reflecting and amplifying social biases

2) → A key rick is the over censorship of marginalized Communities

→ Dixon et ali (2018) found that phrases like "I am gay" or "I am muslim" were incorrectly flaged as toxic - not because of the meaning, but due to biases in the data.

→ This leads to silencing of valid identity expressions and can prevent marginalized voices from being heard online

3) → Although African American English (AAE) and Indian English legitimate varieties of English they differ in grammar, vocabulary and usage

→ classifiers often performs worse on these dialects because they are trained mostly on standard American English, which dominates the trainning datasets. This lack of dialectal diversity results in poor accuracy and unfair outcomes for speakers of underrepresented English varieties

Q3 1) ~~6~~

Sentence S_1 : <s> I love NLP </s>

$P(I|<s>) = 2/3 = 0.6667$

$P(love|I) = 2/2 = 1.0$

$P(NLP|love) = 1/2 = 0.5$

$P(</s>/NLP) = 1/1 = 1.0$

Sentence S_2 : <s> I love deep learning </s>

$P(I|<s>) = 2/3 = 0.6667$

$P(love|I) = 2/2 = 1.0$

$P(deep|love) = 1/2 = 0.5$

$P(learning|deep) = 2/2 = 1.0$

$P(</s>| learning = 1/2 = 0.5$

$P(S_2) = 0.6667 \times 1.0 \times 0.5 \times 1.0 \times 0.5 = 0.1667$

Sentence 1 is more probable under the diagram model

2) $P(\text{noodle} \mid \text{ate})$ using MLE;

$c(\text{ate}, \text{noodle}) = 0$

Total after 'ate' $= 12$

$P(\text{noodle} \mid \text{ate}) = 0/12 = 0.0$

Assiging zero probability to an unseen biagram makes the entire sentence probability zero.
This Causes issues when computing sentence probability or perprexity, leading to "unreliable" or answerable models

Vocabulary size $V = 10$

Total count after 'ate' $= 12$

$P(\text{noodle} \mid \text{ate}) = (0+1)/(12+10) = 1/22 = 00455$

Q4) 1) $P(\text{cats} \mid \text{I}, \text{like}) = c(\text{I like cats}) / c(\text{I like})$

Since we don't have trigram count for I like cats, we back off to

bigram

$P(\text{cats} \mid \text{like}) = c(\text{like cats}) / c(\text{like})$

$c(\text{like cats}) = 2$

$c(\text{like}) = c(\text{like cats}) + c(\text{like dogs}) = 2 + 1 = 3$

$p(\text{cats} \mid \text{I}, \text{like}) = 2/3 = 0.6667$

2) Try trigram first

$P(\text{dogs} \mid \text{you}, \text{like}) = c(\text{you like dogs}) / c(\text{you like})$

But C(you like dogs) = 0 (unseen trigram)
So we back off to bigram
P(dogs | like) = C(like dogs) / C(like)
C(like dogs) = 1
C(like) = 3
P(dogs / you, like) = 1/3 = 0.3333

3) → Back off is necessary because some trigrams are not observed in the training corpus. If we only use trigram probability, such cases would load to a probability of zero for entire sentence.
→ Back off helps mitigate this by falling back to lower order n-grams (bigrams or unigrams), enabling more to bust probability estimates and preventing zero probability issues

Q5) i) formulas
precision = Tp / (Tp + FP)
Recall = Tp / (Tp + FN)

class : Cat
Tp = 5, Fp = 15, FN = 15
precision = 5 / (5 + 15) = 0.25
Recall = 5 / (5 + 15) = 0.25

Class : Dog
Tp = 20, Fp = 25, FN = 25

precision = 20/(20+25) ≈ 0.4446
Recall = 20/(20+25) ≈ 0.4444

class: Rabbit
Tp = 10, Fp = 15, FN = 15
precision = 10 / (10+15) = 0.4
Recall = 10 / (10+15) = 0.4

2) 1) From per-class metrics
- precision values: Cat = 0.25, Dog = 0.4444, Rabit = 0.4

- Recall values: Cat = 0.25, Dog = 0.4444, Rabbit = 0.4

Macro Precision = (0.25 + 0.4444 + 0.4)/3 = 1.0944/3 =
≈ 0.3648

Macro recall = (0.25 + 0.4444 + 0.4)/3 = 0.3648

2) Total Correct Predictions (True positives):
5 + 20 + 10 = 35
Total predictions = 90 → Incorrect predictions
90 − 35 = 55
Micro precision = 35/(35+55) = 35/90 = 0.3889
Micro Recall = 35/(35+55) = 35/90 = 0.3889

3) Macro Areraging
- Averages precision and recall acrossall classes Equally

-usefull when each class should be treated with Equal importance especially in imbalanced datasets

Micro Averaging

- Computes precision and recall by aggregating total TP, FP and FN across all classes.
- useful for overall performance measurement and reflects the influence of more fequent classes

Conclusion:

- use macro averaging to Evaluate per-Class fairness
- use micro averaging to Evaluate overall system effectiveness.