Ans. to the ques. no-1:

(A) The problem overfitting occurs when a model performs well on the training data, but performs bad on validation and test data.

Overfitting can occur for two reasons:

① If the model is too complex

② If the dataset is too large.

If the model is too complex, but dataset is not, The model ~~starts~~ shows a good performance ~~on~~ training period.

Also, the the dataset is too large, but the model is not complex enough, the model starts memorizing and so it ~~sh~~ shows good performance on training period.

~~However~~ for this reason, we also need to check the performance of validation, ~~data~~

Because in validation, ~~the~~ if the model is not good enough, it shows bad performance.

For solving this issue, if the model is too complex, we need to make our dataset also complex, or we can make our ~~dataset~~ model simple.
Also, if the ~~data~~ dataset is too complex, we need to implement a complex model to solve overfitting problem.

(B) $N = 10,000$

$D_{shallow} = 1,000$ $\quad d_{shallow} = 2$

$D_{but} = 9,000$ $\quad d_{but} = 7$.

Now, for shallow,

$idf(w,d) = \log_{10}(1 + f(w,d))$

$= \log_{10}(1+2)$

$= 0.48$

$idf(w,D) = \log_{10}\left(\frac{N}{f(w,D)}\right)$

$= \log_{10}\left(\frac{10,000}{1000}\right)$

$= 1.$

So, tf-idf $= 0.48 \times 1$.

$= 0.48$.

for but,

$tf(w,d) = \log_{10}(1 + f(w,d))$.

$= \log_{10}(1+7)$.

$= 0.903$

$idf(w,D) = \log_{10}\left(\frac{N}{f(w,D)}\right)$.

$= \log_{10}\left(\frac{10,000}{9000}\right)$

$= 0.045$.

So, tf-idf $= 0.903 \times 0.045$.

$= 0.041$.

So, the word "shallow" is more importance as the

$tfidf_{shallow}(0.48) > tf\text{-}idf_{but}(0.041)$

(c) Accuracy is ~~not~~ a bad performance metric. Because, accuracy of a model depends on various factors. The ~~dataset~~ size of each class of a dataset can manipulate the accuracy of a model. For example: ~~For~~ In a spam-detection model, if the "spam" classes data amount is ~~to~~ much larger than the size of the "not-spam" class, this model may find spam models with good accuracy. But, it may also identify "not-spam" ~~also~~ as "spam" as it may not be able to identify 'not-spam' mails. So, this biased model's accuracy does not ~~defi~~ make it a good spam detection model.

Ans. to the ques. no- 2:

(A). While performing tokenization the problems that encounters:

① NER ambiguities: ~~I may not be~~ model may not ~~understand~~ discriminate between a person's or place's name.

② Coordination ambiguity: (He and she) left$.

He and (she left).

③ Attachment ambiguity:

I saw a monkey in my dress.

↳ Who is in the dress?

④ Syntactic representation: Tokenization might not capture syntactic representation.

(B) Paris is to France as Oslo is to ____.

$C = A - B + X$.

A = Paris $= [3, 4, 0, 1]$.

B = France $= [3, 2, 1, 0]$.

C = Oslo $= [1, 2, 1, 1]$.

$X = \begin{cases} \text{Sweden} = [3, 1, 1, 1] \\ \text{Denmark} = [0, 0, 1, 2] \\ \text{Norway} = [3, 2, 3, 2] \\ \text{Finland} = [1, 1, 1, 0] \end{cases}$

For Sweden,

$= [3, 4, 0, 1] - [3, 2, 1, 0] + [3, 1, 1, 1]$

$= [3, 3, 0, 2]$.

For Denmark,

$= [3, 4, 0, 1] - [3, 2, 1, 0] + [0, 0, 1, 2]$.

$= [0, 2, 0, 3]$

For Norway,

$= [3, 4, 0, 1] - [3, 2, 1, 0] + [3, 2, 3, 2]$.

$= [3, 4, 2, 3]$

For Finland,

$= [3, 4, 0, 1] - [3, 2, 1, 0] + [3, 2, 3, 2]$.

$= [1, 3, 0, 1]$.

Now,

$$\text{Cosine(Oslo, Sweden)} = \frac{[1,2,1,1]\,[3,3,0,2]}{\sqrt{1+4+1+1}\ \sqrt{9+9+0+4}}$$

$$= \frac{3+6+0+2}{\sqrt{7}\ \sqrt{22}}$$

$$= 0.886.$$

$$\text{Cosine(Oslo, denmark)} = \frac{[1,2,1,1]\,[0,2,0,3]}{\sqrt{1+4+1+1}\ \sqrt{4+9}}$$

$$= \frac{0+4+0+3}{\sqrt{7}\ \sqrt{13}}$$

$$= 0.73.$$

$$\text{Cosine(Oslo, Norway)} = \frac{[1,2,1,1]\,[3,4,2,3]}{\sqrt{1+4+1+1}\ \sqrt{9+16+4+9}}$$

$$= \frac{3+8+2+3}{\sqrt{7}\ \sqrt{38}}$$

$$= 0.98.$$

$$\text{Cosine}(\text{Oslo}, \text{Finland}) = \frac{[1,2,1,1]\,[1,3,0,1]}{\sqrt{1+4+1+1}\ \sqrt{1+9+1}}$$

$$= \frac{1+6+0+1}{\sqrt{7}\ \sqrt{11}}$$

$$= 0.91.$$

As, cosine similarity of Oslo and Norway is the highest;

Paris is to France as Oslo is to Norway.

Ans. to the ques. no-3:

(A) $W_A = [1 \quad 1 \quad 3]^T$. $b_A = 0$.

$W_B = [1 \quad 2 \quad 0]^T$ $b_B = -1$.

$X = [1 \quad 1 \quad 0]^T$.

For classifier A, $y_A = \hat{y} = \sigma(W_A^T X + b_A)$.

$$= \sigma\left([1 \quad 1 \quad 3] \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + 0 \right).$$

$$= \sigma(1+1+0+0)$$

$$= \sigma(2).$$

$$= \frac{1}{1+e^{-(2)}}$$

$$= 0.8807.$$

For classifier B, $\hat{y}_B = \sigma(W_B^T X + b_B)$.

$$= \sigma\left[\left(\begin{bmatrix}1 & 2 & 0\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix}\right) + (-1)\right]$$

$$= \sigma\left[1+2+0-1\right]$$

$$= \sigma(2)$$

$$= \frac{1}{1+e^{-(2)}}$$

$$= 0.8807.$$

Now, for y_A.

To minimize $= -[y \log y_A + (1-y)\log(1-y_A)]$

$$= -[1 \log(1-y_A)]$$

$$= -\log(1-0.8807)$$

$$= 0.923.$$

Also, for y_B. $= -[y \log y_B + (1-y)\log(1-y_B)$.

$$= -[1 \log(1-y_B)]$$

= 0.923.

Here, both classifier ~~sho~~ incurs same cross entropy loss.

(B) · Dataset = 10,000.

Unlabeled.

team of 3.

~~Here,~~ ~~The~~ I have a dataset of 10,000 review but as no star rank is associated with it, creating a review is hard.

~~Because, Because,~~

However, I can create a classifier from it.

First: ① Taking the words "good", "mind-blowing", "bad" from the model.

② I can apply TF-IDF to see ~~how~~

which ~~revi~~ values can actually ~~us~~ help us to do classification.

So, "good" word may increase the good review weight.

For ~~which~~ "big weights" we provide 5 stars.

and decrease as the weight decreases