NLP

End

Sem

Learning

Parkee

NLP — 2022

Sol 1(a.) Tokenization is the process of breaking down a text into smaller units called tokens.
These tokens can be words, characters or subwords depending on the chosen tokenization scheme.
Tokenization is a fundamental step in NLP tasks, as it allows the model to process and understand the text at a more granular level.

BPE (Byte-Pair Encoding) is a popular tokenization algorithm that uses a statistical approach to build a vocabulary of subword units.
It works by iteratively merging the most frequent pairs of characters or character sequences in a corpus.

Advantages of BPE :-
→ Subword representation for handling OOV words
→ Improved generalization by recognizing shared subwords
→ Vocabulary compression for reduced memory usage.
→ Captures morphological information in languages.

Training the BPE algo. using the below mentioned table:-

dictionary	
low_	5
lowest_	2
newer_	6
wider_	3
new_	2

Vocabulary (11 letters)
_, d, e, i, l, n, o, r, s, t, w

Note: Each word is represented as a sequence of characters plus a special end-of-word symbol here i.e. _ (underscore).

Now we count all pairs of letters. (right to left)

lo → 7
ow → 7
we → 8
er → 2
st → 2
w_ → 5
t_ → 2
ne → 8
ew → 8
r_ →

w_ → 7
ow → 7
lo → 7
t_ → 2
st → 2
es → 2
we → 8
r_ → 9 (1st)
er → 9 (2nd)
ew → 8 (3rd)
ne → 8
de → 3
id → 3
wi → 3
ow_ → 5
low → 7
st_ → 2
est → 2
wes → 2
owe → 2 (2nd)
er_ → 9
wer → 6
ewe → 6
new → 8
der → 3
ide → 3
wid → 3
ew_ → 2

Vocabulary
_, d, e, i, l, n, o, r, s, t, w, r_, er_

∴ our system has learned that there should be a token for word-final er represented as er_

Vocabulary
_, d, e, i, l, n, o, r, s, t, w, r_, er_, ew, new, lo, low, newer_

when we check for same freq. we see their alphabetical order for picking them

Aniket
Date
Page

Correct order of iterations are :- (Remember, in each iteration we have to count pairs from dictionary again.)

Initially

	dictionary	vocabulary (11 letters)
5	l o w _	_, d, e, i, l, n, o, r, s, t
2	l o w e s t _	, w
6	n e w e r _	
3	w i d e r _	
2	n e w _	

⇒ We first count all pairs of symbols :- the most freq. is r_ (9 times)

merge r & _ treating r_ as one symbol

next iteration

	dictionary	vocabulary
5	l o w _	_, d, e, i, l, n, o, r, s
2	l o w e s t _	, t, w, r_
6	n e w e r_	
3	w i d e r_	
2	n e w _	

Now, most frequent pair is e & r_ i.e., make them er_

next iteration

	dictionary	vocabulary
5	l o w _	_, d, e, i, l, n, o, r
2	l o w e s t _	, s, t, w, r_, er_
6	n e w er_	
3	w i d er_	
2	n e w _	

⇒ Now, mfp is ew.

Aniket
Date
Page

next iteration

Dictionary	Vocabulary
5 l o w _	_, d, e, i, l, n, o, r, s, t, w, r_, er_, ew
2 l o w e s t _	
6 n ew er_	
3 w i d er_	
2 n ew _	

⇒ now, mfp is new.

If we continue, the next merges are :-

Merge	Current Vocabulary
(n, ew)	_, d, e, i, l, n, o, r, s, t, w, r_, er_, ew, new
(l, o)	, lo
(lo, w)	, low
(new, er_)	, newer_
(low, _)	, low_

When we need to tokenize a test sentence, we just run the merges we have learned, greedily, in the order we learned them, on the test data.
(Thus, the freq.'s in test data don't play a role.)
Just, freq.'s in training data are vital.

Now, Testing :- the final tokenization of the word "lower"

lower_ ⇒ "l" + "o" + "w" + "e" + "r" + "_"

(added special end-of-word symbol)

"lo" = "l" + "o"

"low" = "lo" + "w"

"er_" = "r" + "_"

i.e., lower_ = "low" + "er_"

∴, A new (unknown) testing word "lower_" would be merged into two tokens i.e., "low" + "er_".

Sol 1(b.) Bigram counts :- (given)

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

I am assuming the whole corpus has V unique words.
i.e., $\boxed{V = 1446}$ out of which only 8 words bigram counts are given to us.

Unigram counts :- (assumed)

i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

$$P(w_n | w_{n-1}) = \frac{C(w_{n-1} w_n)}{C(w_{n-1})}$$

$$P(i|i) = \frac{C(ii)}{C(i)} = \frac{5}{2533} = 0.00197 \approx 0.002$$

, $$P(i|want) = \frac{C(want\ i)}{C(want)} = \frac{2}{927} = 0.002157 \approx 0.0022$$

, -- similarly for others.

∴ Bigram Probability matrix

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33						
want	0.0022	0						
to	0.00083	0						
eat	0	0						
chinese		0						
food		0						
lunch		0						
spend		0						

Similarly, you can make table (matrix) for Bigram Probability with add-1 smoothing, using below mentioned formula.

$$P^*_{\text{Laplace}}(w_n | w_{n-1}) = \frac{C(w_{n-1} w_n) + 1}{C(w_{n-1}) + V}$$

Sol 1(c.)

(1.) Segmentation :-

It is the process of dividing a sentence or a document into smaller units like words or phrases or sentences for further analysis.

This is a common technique used in natural language processing and text analysis.

Ex:- "I love chatting with friends"
⇒ ["I", "love", "chatting", "with", "friends"].

(2.) Lemmatization :-

It is the process of reducing words to their base or root or dictionary form - known as the lemma.

This is done to simplify the analysis of text by grouping together words that have the same base form.

Ex:- "am", "is", "are", "was", "were" gets reduced in Lemmatization to their base form "be".

(3.) Stemming :-

It is the process of reducing words to their root form by removing the suffixes.

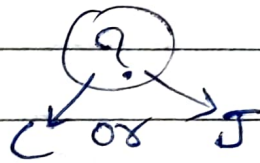
This is a simpler approach than lemmatization because it does not consider the context of the word.

ex:- "Jumping" gets reduced to "Jump".

Sol 2(a.)

	Doc(d)	Words	class
Training	1	Chinese, Beijing, Chinese	C
	2	Chinese, Chinese, Shanghai	C
	3	Chinese, Macao	C
	4	Tokyo, Japan, Chinese	J
Test	5	Chinese, Chinese, Tokyo, Japan	?

To compute :- most likely class for Doc 5.



Assume, a multinomial naive bayes classifier and use add-α La Place smoothing for the likelihoods. say $\alpha = 1$.

$$P(\hat{X}) = \frac{\text{count of } X}{N}, \quad N = \text{total count of classes.}$$

($\hat{X}$: a class)

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

- count(w, c): total no. of w words in the c class.
- 1: α
- count(c): total no. of words in c class
- $|V|$: vocabulary size (size of unique words dictionary from training dataset)

Priors :-

$$P(C) = \frac{3}{4} \quad \text{(Probability of class C)}$$

$$P(J) = \frac{1}{4} \quad (\text{" " " J})$$

Conditional Probabilities :-

$P(\text{Chinese}|C) = (5+1)/(8+6) = 6/14$

$P(\text{Tokyo}|C) = (0+1)/(8+6) = 1/14$

$P(\text{Japan}|C) = (0+1)/(8+6) = 1/14$

$P(\text{Chinese}|J) = (1+1)/(3+6) = 2/9$

$P(\text{Tokyo}|J) = (1+1)/(3+6) = 2/9$

$P(\text{Japan}|J) = (1+1)/(3+6) = 2/9$

Predicting class for Doc 5. :-

Probability of DOC5 to come in C class

$$P(C|d5) = P(C) \cdot (P(\text{Chinese}|C))^2 \cdot (P(\text{Tokyo}|C))^1 \cdot (P(\text{Japan}|C))^1$$

$$= \frac{3}{4} \cdot \left(\frac{6}{14}\right)^2 \cdot \frac{1}{14} \cdot \frac{1}{14} = 0.0007028$$

Probability of DOC5 to come in J class

$$P(J|d5) = \frac{1}{4} \cdot \left(\frac{2}{9}\right)^2 \cdot \frac{2}{9} \cdot \frac{2}{9} = 0.0006097$$

$\therefore$, $P(C|d5) > P(J|d5)$

$\therefore$, Doc5 will come into class 'C'.

Sol 2(b.)

Training data

S. No.	Document	Class
1	Natural Language Processing	A
2	Language Model Learning	A
3	Ngram Language Model	A
4	Text Classification Model	A
5	Text Processing Model	A
6	Computer Vision	B
7	Image Classification Model	B
8	Object Segmentation	B
9	Image Processing	B
10.	Object Recognition	B

Testing data

S. No,	Document	Class
1	Object Recognition Model	?
2	Text Recognition Model	?

Using Multinomial Naive Bayes (Unigram Model)

$P(A) = \frac{5}{10} = \frac{1}{2}$, $P(B) = \frac{5}{10} = \frac{1}{2}$, $|V|$ = Unique Words = 14

Now, test data :- Object Recognition Model.

$P(\text{object}|A) = \frac{0+1}{15+14} = \frac{1}{29}$

$P(\text{recognition}|A) = \frac{0+1}{15+14} = \frac{1}{29}$

$P(\text{Model}|A) = \frac{4+1}{15+14} = \frac{5}{29}$

$P(\text{Object}|B) = \frac{2+1}{11+14} = \frac{3}{25}$

$P(\text{recognition}|B) = \frac{1+1}{11+14} = \frac{2}{25}$

$P(\text{Model}|B) = \frac{1+1}{11+14} = \frac{2}{25}$

(d)

Predicting class for Test doc 1.

$$P(A|d1) = (P(A)) \cdot (P(\text{object}|A)) \cdot (P(\text{recognition}|A)) \cdot (P(\text{Model}|A))$$

$$= \frac{1}{2} \cdot \frac{1}{29} \cdot \frac{1}{29} \cdot \frac{5}{29} = 0.0001$$

$$P(B|d1) = (P(B)) \cdot (P(\text{object}|B)) \cdot (P(\text{recognition}|B)) \cdot (P(\text{Model}|B))$$

$$= \frac{1}{2} \cdot \frac{3}{25} \cdot \frac{2}{25} \cdot \frac{2}{25} = 0.0003$$

∴ d1 :- Object Recognition Model will come into class 'B'

Now test data :- Text Recognition Model

$P(\text{Text}|A) = \frac{2+1}{15+14} = 3/29$	$P(\text{Text}|B) = \frac{0+1}{11+14} = 1/25$
$P(\text{Recognition}|A) = \frac{0+1}{15+14} = 1/29$	$P(\text{Recognition}|B) = \frac{1+1}{11+14} = 2/25$
$P(\text{Model}|A) = \frac{4+1}{15+14} = 5/29$	$P(\text{Model}|B) = \frac{2}{11+14} = 2/25$

Predicting class for Test doc (d) 2

$$P(A|d2) = (P(A)) \cdot (P(\text{Text}|A)) \cdot (P(\text{recognition}|A)) \cdot (P(\text{Model}|A))$$

$$= \frac{1}{2} \cdot \frac{3}{29} \cdot \frac{1}{29} \cdot \frac{5}{29} = 0.0003$$

$$P(B|d2) = \frac{1}{2} \cdot \frac{1}{25} \cdot \frac{2}{25} \cdot \frac{2}{25} = 0.0001$$

∴ d2 :- Text Recognition Model will come into class 'A'.

Using Multinomial Naive Bayes (Bigram Model)

	Class
<s> Natural Language Processing </s>	A
<s> Language Model Learning </s>	A
<s> Ngram Language Model </s>	A
<s> Text Classification Model </s>	A
<s> Text Processing Model </s>	A
<s> Computer Vision </s>	B
<s> Image Classification Model </s>	B
<s> Object Segmentation </s>	B
<s> Image Processing </s>	B
<s> Object Recognition </s>	B

~~We include </s> & <s>, we get~~

~~we won't including </s> if we include </s> this can also appear as second element of a bigram,~~
we get $|V| = 16$ for our vocabulary.
$(14 + 1 + 1)$
(<s>) (</s>)

Test doc 1 :- <s> object recognition Model </s>

$$P(A|d1) = P(A) \cdot P(\text{object}|<s>) \cdot P(\text{recognition}|\text{object}) \cdot P(\text{Model}|\text{recognition}) \cdot P(</s>|\text{Model})$$

$$= P(A) \cdot \frac{(\text{Count}(<s>, \text{object}) + 1}{\text{count}(<s>) + |V|} \quad \text{---} \quad \text{(similarly)}$$

$$= \frac{5}{10} \cdot \left(\frac{0+1}{5+16}\right) \cdot \left(\frac{0+1}{0+16}\right) \left(\frac{0+1}{0+16}\right) \cdot \left(\frac{3+1}{4+16}\right) = 0.000019$$

$$P(B|d1) = \frac{5}{10} \cdot \left(\frac{2+1}{5+16}\right) \cdot \left(\frac{1+1}{2+16}\right) \cdot \left(\frac{0+1}{1+16}\right) \cdot \left(\frac{1+1}{1+16}\right) = 0.000055$$

∴, d1 :- "Object Recognition Model" will come into class 'B'.

Test doc2 :- <s> Text recognition model </s>

$$P(A|d2) = \frac{5}{10} \cdot \left(\frac{2+1}{5+16}\right) \cdot \left(\frac{0+1}{2+16}\right) \cdot \left(\frac{0+1}{0+16}\right) \cdot \left(\frac{3+1}{4+16}\right) = 0.00005$$

$$P(B|d2) = \frac{5}{10} \cdot \left(\frac{0+1}{5+16}\right) \cdot \left(\frac{0+1}{0+16}\right) \cdot \left(\frac{0+1}{1+16}\right) \cdot \left(\frac{1+1}{1+16}\right) = 0.00001$$

∴, d2 :- "Text Recognition Model" will come into class "A".