

## Contents

1.	2
g)	2
h)	2
i)	2
<i>i)</i>	2
<i>ii)</i>	3
2.	3
a)	3
b)	3
<i>i)</i>	3
<i>ii)</i>	3
<i>iii)</i>	4
<i>iv)</i>	4
c)	4
<i>i)</i>	5
<i>ii)</i>	7
<i>iii)</i>	9
<i>iv)</i>	9

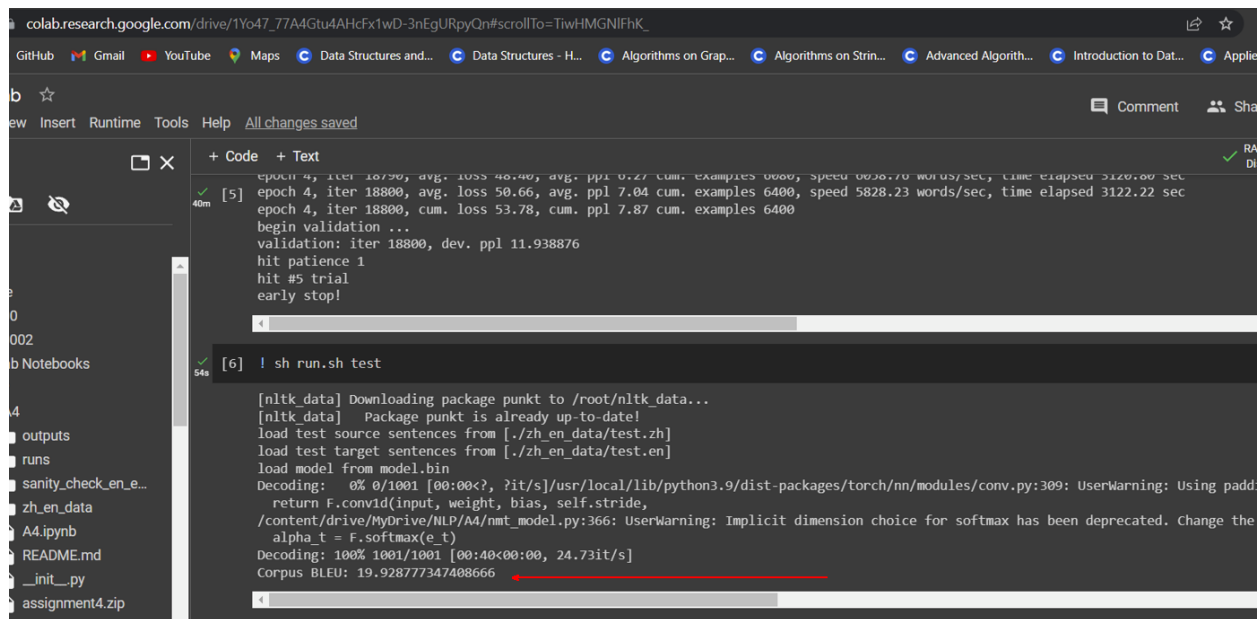
1.

g)

**First part)** as in the question said in *generate\_sent\_mask* function, padded indexes are “1” and non padded are “0” then in *step* function values which are “1” get  $-\infty$ . It means that padding values are  $-\infty$ . It has effect on our softmax function after  $e_t$  getting  $-\infty$  then  $\alpha_t$  after applying softmax function get  $e^{-\infty} = 0$ . And finally  $a_t$  will only effect on those words which are not paddings because those are multiplied by number larger than 0 but padding index are multiplied by zero.

**Second part)** because in fact in attention, which means focus on word which have more impact on our translation, we want to focus on true words in source language which means we’re translating this word right now, but padding values doesn’t make any sense to get involved in our attention computation because they don’t have any meanings.

h)



```
epoch 4, iter 18750, avg. loss 46.40, avg. ppl 0.27 cum. examples 6000, speed 6036.70 words/sec, time elapsed 3120.00 sec
epoch 4, iter 18800, avg. loss 50.66, avg. ppl 7.04 cum. examples 6400, speed 5828.23 words/sec, time elapsed 3122.22 sec
epoch 4, iter 18800, cum. loss 53.78, cum. ppl 7.87 cum. examples 6400
begin validation ...
validation: iter 18800, dev. ppl 11.938876
hit patience 1
hit #5 trial
early stop!

[6] ! sh run.sh test

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
load test source sentences from [./zh_en_data/test.zh]
load test target sentences from [./zh_en_data/test.en]
load model from model.bin
Decoding: 0% 0/1001 [00:00<?, ?it/s] /usr/local/lib/python3.9/dist-packages/torch/nn/modules/conv.py:309: UserWarning: Using padd
return F.conv1d(input, weight, bias, self.stride,
/content/drive/MyDrive/NLP/A4/nmt_model.py:366: UserWarning: Implicit dimension choice for softmax has been deprecated. Change the
alpha_t = F.softmax(e_t)
Decoding: 100% 1001/1001 [00:40<00:00, 24.73it/s]
Corpus BLEU: 19.928777347408666
```

i)

j)

**Advantage :** it’s easier and faster to compute because it doesn’t have additional parameters to learn. Because in *multiplicative attention*  $W$  must be learned.

**Disadvantage :** it’s simple and don’t know in what part it should pay attention more, but in *multiplicative attention*  $W$ , parameters know how to handle focus on which part by making important features to have larger weights. Another disadvantage is that in simple dot product  $s$  and  $h$  must be same dimension.

ii)

**Advantage :** it has separate learnable weight matrix which can capture feature of each decoder and encoder individually and focus more. And it also adds non linearity to our attention model with *tanh* activation function which can handle complexity more and learn better with more interesting result.

**Disadvantage :** because it has two separate learnable matrix and non linearity activation it cost much more than *multiplicative attention*.

2.

a)

in RNN, it just handles token one point at a time and it doesn't care about the length of sentences and position of each token. But in 1D convolution cares about spatial and position matters for it. And this means that if 2 characters 电, 脑 (mentioned in the document) happens next to each other RNN itself can't recognize whether they're related to have one meanings or not. Because if they happen separately they have different meaning but next to each other combine one word "computer". And here 1D convolution helps us which care about positioning and learn local patterns. So in language such as Chinese, Japanese and etc., there are character tokenized by sentence piece instead of word tokenization, and here CNN will help us to capture spatial features when character besides each other have different meaning

b)

i)

**error:** the model didn't understand the plurality of word in source language

**possible reason:** possible reason for that can be, when model use subword embedding it splits the plural part from the main part and then model just translate main part, or model knows all formation of a word as a single meaning and don't care it's plural or not.

**Possible way to fix:** we can add 1D convolution to capture spatial data in subword embedding to group them as single word with plural meaning, another way can be to add the effect of plural part more in *attention* so it has higher probability and our decoder model will translate truly.

ii)

**error:** the model repeat *resources have been exhausted* two times

**possible reason:** it may be for our attention model doesn't work well and when we're translating at decoder part model still has higher probability to the previous ones and doesn't assign more probabilities to the new one.

**Possible way to fix:** when translating model just relies on attention result and the output from previous step in decoder doesn't have enough effect in current time step. So we can add more attention to the decoder part or just use self attention (transformers).

iii)

**error:** the model didn't recognize *national mourning today* and translate differently

**possible reason:** it may have some possible reason. One could be that our attention model doesn't have effect on *morning today* and their probabilities are too low. Another reason is that our model didn't understand the meaning of national mourning and consider it as one word. And final reason could be that subword tokenizer didn't work well in *morning today* phrase.

**Possible way to fix:** adding extra training data and more hidden layers maybe help. Attention model should use additive attention, more phrase like *morning today* should be added.

iv)

**error:** the model didn't understand the meaning of *act not err not*.

**possible reason:** it may be for our sub word tokenizing and translating one by one which lead to this result single meaning for each sub word at encoder level. Or our model didn't see enough expression like this before and translate something similar to this.

**Possible way to fix:** add 1D convolution to capture whole word as single word and detect spatial data. Add more training data and use multi-layer hidden layer. Add more expression and idioms to our training data that our model just not translate piece by piece see the meaning of whole sentence

c)

$$p_n = \frac{\sum_{\text{ngram} \in \mathbf{c}} \min \left( \max_{i=1, \dots, k} \text{Count}_{\mathbf{r}_i}(\text{ngram}), \text{Count}_{\mathbf{c}}(\text{ngram}) \right)}{\sum_{\text{ngram} \in \mathbf{c}} \text{Count}_{\mathbf{c}}(\text{ngram})}$$

$$BP = \begin{cases} 1 & \text{if } \text{len}(c) \geq \text{len}(r) \\ \exp \left( 1 - \frac{\text{len}(r)}{\text{len}(c)} \right) & \text{otherwise} \end{cases}$$

$$BLEU = BP \times \exp \left( \sum_{n=1}^4 \lambda_n \log p_n \right)$$

Source Sentence **s**: 需要有充足和可预测的资源。

Reference Translation **r**<sub>1</sub>: *resources have to be sufficient and they have to be predictable*

Reference Translation **r**<sub>2</sub>: *adequate and predictable resources are required*

NMT Translation **c**<sub>1</sub>: there is a need for adequate and predictable resources

NMT Translation **c**<sub>2</sub>: resources be sufficient and predictable to

i)

**FOR C1:**

1-gram: {'there', 'is', 'a', 'need', 'for', 'adequate', 'and', 'predictable', 'resources'}

2-grams: {'there is', 'is a', 'a need', 'need for', 'for adequate', 'adequate and', 'and predictable', 'predictable resources'}

for each word in 1-gram we iterate and see how many same word occurs in other reference translation:

\*\* in table are shown count of occurrence of each *grams* in each NMT and reference r1 and r2 translation. max\_ref means the maximum in formula (max (r1, r2)) and min\_ref\_source means the min in formula (min(max\_ref, c))

	c	r1	r2	max_ref	min_ref_source
adequate	1	0	1	1	1
for	1	0	0	0	0
is	1	0	0	0	0
predictable	1	1	1	1	1
resources	1	1	1	1	1
a	1	0	0	0	0
and	1	1	1	1	1
need	1	0	0	0	0
there	1	0	0	0	0

$$P_1 = \frac{1+0+0+1+1+0+1+0+0}{1+1+1+1+1+1+1+1+1} = \frac{4}{9}$$

$$\text{Len}(c) = 9, \text{len}(r1) = 11, \text{len}(r2) = 6 \rightarrow \text{len}(r) = 11$$

$$BP = e^{1 - 11/9} = e^{-2/9} = 0.8$$

And do this for 2grams:

	c	r1	r2	max_ref	min_ref_source
for adequate	1	0	0	0	0
and predictable	1	0	1	1	1
there is	1	0	0	0	0
need for	1	0	0	0	0
adequate and	1	0	1	1	1
predictable resources	1	0	1	1	1
is a	1	0	0	0	0
a need	1	0	0	0	0

$$P_2 = \frac{0+1+0+0+1+1+0+0}{1+1+1+1+1+1+1+1} = \frac{3}{8}$$

$$BLEU_{c1} = BP \times e^{0.5 \log(4/9) + 0.5 \log(3/8)} = 0.8 \times e^{-0.17-0.21=-0.38} = 0.547$$

FOR C2:

1-gram: {'resources', 'be', 'sufficient', 'and', 'predictable', 'to'}

2-grams: {'resources be', 'be sufficient', 'sufficient and', 'and predictable', 'predictable to'}

for each word in 1-gram we iterate and see how many same word occurs in other reference translation:

	c	r1	r2	max_ref	min_ref_source
be	1	2	0	2	1
predictable	1	1	1	1	1
resources	1	1	1	1	1
and	1	1	1	1	1
to	1	2	0	2	1
sufficient	1	1	0	1	1

$$P_1 = \frac{1+1+1+1+1+1}{1+1+1+1+1+1} = \frac{6}{6} = 1$$

$$\text{Len}(c) = 6, \text{len}(r1) = 11, \text{len}(r2) = 6 \rightarrow \text{len}(r) = 6$$

$$BP = 1$$

And do this for 2grams:

	c	r1	r2	max_ref	min_ref_source
resources be	1	0	0	0	0
and predictable	1	0	1	1	1
sufficient and	1	1	0	1	1
be sufficient	1	1	0	1	1
predictable to	1	0	0	0	0

$$P_2 = \frac{0+1+1+1+0}{1+1+1+1+1} = \frac{3}{5} = 0.6$$

$$BLEU_{c2} = BP \times e^{0.5 \log(1) + 0.5 \log(0.6)} = 1 \times e^{0-0.11=-0.11} = 0.89$$

based on BLEU score the second NMT translation works better, but I don't think it's better because it doesn't have our desired meaning and it also has grammatical mistakes. it just works better Because it has higher similar word rate it receives better result.

ii)

FOR C1:

for each word in 1-gram we iterate and see how many same word occurs in other reference translation:

	c	r2	max_ref	min_ref_source
predictable	1	1	1	1
a	1	0	0	0
need	1	0	0	0
and	1	1	1	1
for	1	0	0	0
adequate	1	1	1	1
resources	1	1	1	1
is	1	0	0	0
there	1	0	0	0

$$P_1 = \frac{1+0+0+1+0+1+1+0+0}{1+1+1+1+1+1+1+1+1} = \frac{4}{9}$$

$$Len(c) = 9, len(r) = 6 \rightarrow BP = 1$$

And do this for 2grams:

	c	r2	max_ref	min_ref_source
is a	1	0	0	0
a need	1	0	0	0
need for	1	0	0	0
for adequate	1	0	0	0
and predictable	1	1	1	1
predictable resources	1	1	1	1
adequate and	1	1	1	1
there is	1	0	0	0

$$P_2 = \frac{0+0+0+0+1+1+1+0}{1+1+1+1+1+1+1+1} = \frac{3}{8}$$

$$BLEU_{c1} = BP \times e^{0.5 \log(4/9) + 0.5 \log(3/8)} = 1 \times e^{-0.17-0.21=-0.38} = 0.68$$

FOR C2:

for each word in 1-gram we iterate and see how many same word occurs in other reference translation:

	c	r2	max_ref	min_ref_source
to	1	0	0	0
sufficient	1	0	0	0
resources	1	1	1	1
and	1	1	1	1
be	1	0	0	0
predictable	1	1	1	1

$$P_1 = \frac{0+0+1+1+0+1}{1+1+1+1+1+1} = \frac{3}{6} = 0.5$$

$$Len(c) = 6, len(r) = 6$$

$$BP = 1$$



And do this for 2grams:

	c	r2	max_ref	min_ref_source
sufficient and	1	0	0	0
be sufficient	1	0	0	0
and predictable	1	1	1	1
predictable to	1	0	0	0
resources be	1	0	0	0

$$P_2 = \frac{0+0+1+0+0}{1+1+1+1+1} = \frac{1}{5} = 0.2$$

$$BLEU_{c2} = BP \times e^{0.5 \log(0.5) + 0.5 \log(0.2)} = 1 \times e^{-0.5} = 0.6$$

Here  $c_1$  have better translation and I agree on that, because it gives us sentence meaning although it doesn't have same word like reference.

iii)

it may cause some problem. Let's explain by formula. if we have multiple reference translation is much better because if in our NMT translated sentence, we have some word which has multiple definition so even if it doesn't appear in some reference translation there is high chance to happen in other reference translation and the operation of getting MAX is helping us in this way that we are not ignoring that word and that word is considered. So even if our model translates well but words are different in single reference translation it will get bad result

In multiple reference translation in fact it will gather the association of all translations so negative point doesn't affect our NMT model.

And on the other hand by using multiple reference, n-grams BLEU score will sum over whole translation and there is a chance that one translation hit our needs. And there are more flexibility for our model

iv)

advantages:

- Human can make mistake, but this has restricted rules and can be relied on
- It doesn't depend on language, in all languages we have same rules
- It's fast

disadvantages:

- Sentence with same meaning or different words are not evaluated correctly and gets low result
- Multiple references needed
- It's so rigid that can't capture n-grams which are distant from each other