CS224-N

ASSIGNMENT-4.

1. NEURAL MACHINE TRANSLATION WITH RNN's

g) In attention computation we projected all hidden states of encoder to required dimension. Then attention weights is computed for each hidden time step.

Since the inputs are padded to max-length, hidden states and attention weights are computed also for this pad inputs in source. By using mask we make the attention weights corresponding to hidden states of pad input as $-\infty$. So this ensures that pad tokens aren't used in the computation of attention vector at each step, which also makes sure that each step of decoder will attend only the words in sentences not the pad tokens.

j) DOT-PRODUCT ATTENTION:

$e_{t,i} = S_t^T h_i$.

It is the simplest of all attention. It doesn't require any extra weight matrix and can be executed using parallel multiplication. But the disadvantage is raw weights of decoder and encoder states directly take part in attention. So, both these states should also learn the attention configuration in addition to packing of other sequential information.

MULTIPLICATIVE ATTENTION:

$e_{t,i} = S_t^T W h_i$.

It is both faster as it can be implemented using parallel matrix computation and has a weight matrix which can convert the hidden state information into one required for attention. So, the weight matrix can be learned appropriately

Unlike dot product attention where the hidden state itself has to be learned for attention. Disadvantage is attention weights can sometimes explode if the dimensions of decoder state is large as it is not scaled in the computation. One way to solve this is to scale the attention value by $1/\sqrt{d_h}$

ADDITIVE ATTENTION:

$$e_{t,i} = V^T(W_1 h_i + W_2 s_t).$$

Advantage is we have a linear combination of decoder and encoder hidden states by using two different matrices W_1 and W_2. So we don't need to worry about large dimensions of d_h or e_h as it will be scaled down necessarily by W_1 and W_2. Disadvantage is it requires more space (because of W_1 and W_2) and is slower compared to other two attention as it involves three different matrix multiplication and one matrix addition.

2) Analysing NMT Systems

a) i) 1. NMT error: Here's another favorite of my favorites, "The starry Night!"

"favorite" shouldn't be there or right words like "one" should be there

2. Reason: After decoding "Here's another" the decoder might be attending more to the word 'favoritos' than 'de', which results in predicting 'favorite'. After decoding "of my" it is again attending more to 'favoritos'

3. Solution: We can use beam search to search for more translations than predicting one possibility using greedy decoding

ii) 1. NMT error: You know what I do is write for children, and in fact, I'm probably the author for children, more reading in the US

The phrase isn't ~~[illegible]~~ grammatically right where "author for children" and "more reading in the US" could have been combined.

2. Reason: The decoder is just translating in alignment with source sentence when decoded the source sentence. Each part of source sentence separately gives the separate parts in translated sentence. To give the correct sentence in english the decoder have to attend differently to source sentence rather than attending sequentially.

3. Solution: We can collect more parallel data like this and train the MT system on it so that it is able to attend differently when such content appears.

iii) <u>NMT error:</u> A friend of mine did that - Richard <u><unk></u>

Here we aren't able to translate the name Richard Bolingbroke completely from spanish to english

<u>Reason:</u> This is because the word 'bolingbroke' might not be in the vocabulary. So the MT system is unable to find a word for 'Bolingbroke' which would have the maximum attention weight at final step.

<u>Solution:</u> We can use copying mechanism where we can copy the source words directly into the translation sentence if we cannot find a word in the target vocabulary.

iv). <u>NMT error:</u> You just have to go <u>back to the apple to</u> see it as a epiphany.

'back to the apple' isn't the right word when compared with reference translation.

<u>Reason:</u> There are lot of instances of "just have to" in the training data which might have influenced it in the translation. And the phrase "around the block" in reference translation is something intuitive which isn't present in source sentence. (Source sentence has 'manzana' which translates to apple).

<u>Solution:</u> Training with more such instances might improve the results.

v) i) NMT error: She saved my life by letting me go to the bathroom in the women's room

The translation says it's a women's room while there is only the spanish word for "teacher" ('Professores') in the source language.

ii) Reason: This might be because of the gender bias in the training data. where "professors" are highly correlated with "women".

iii) Solution: Should try to remove such biases in the data so that professions aren't automatically tagged with particular gender.

vi) i) NMT error: That's over 100,000 acres

The source sentence mentions "100,000 hectareas" which means the translation should contain "100,000 hectares" or "250 thousand acres", but the actual translated one says "100,000 acres"

Reason: This error might be because of the examples in the training set where "acres" are mostly pre~~ced~~ceded by large numerical values like 10,000 whereas "hectares" are mostly preceded by "million" or "billion". This ~~[illegible]~~ may force the network to prefer "acres" on seeing 10000 in previous step though it may be attending to "hectareas" in source

Solution: For certain words like hectares, acres we can modify the model to output the actual raw translation if them in source. For instance if a step in decoder is attending maximum to the word "hectareas" in source, we can directly output it's equivalent "hectare" if the probability distribution gives some other outputs like "acre". i.e) we can combine some elements of statistical MT and neural MT

c) i) r_1: love can always find a way

r_2: love makes anything possible

c_1: the love can always do

c_2: love can make anything possible

For c1: P_3, P_4 is not needed as $\lambda_3, \lambda_4 = 0$

$P_1 \Rightarrow$ 1 grams are {the, love, can, always, do}

$$P_1 = \frac{\min(0,1) + \min(1,1) + \min(1,1) + \min(1,1) + \min(0,1)}{1+1+1+1+1}$$

$$= \frac{0+1+1+1+0}{5} = 3/5$$

$c = 5$, $r^* = 4 (r_2)$.

$BP = 1 \quad (c \geq r^*)$

$P_2 \Rightarrow$ 2 grams are {the love, love can, can always, always do}

$$P_2 = \frac{\min(0,1) + \min(1,1) + \min(1,1) + \min(0,1)}{1+1+1+1} = \frac{0+1+0+1}{4} = 2/4$$
$$= 1/2$$

$$BLEU = 1 \times \exp\left(0.5 \times \log 0.6 + 0.5 \times \log 0.5\right)$$

$$= 0.76994$$

For c2:

$P_1 \Rightarrow$ 1 grams are {love, can, make, anything, possible}

$$P_1 = \frac{\min(1,1) + \min(1,1) + \min(0,1) + \min(1,1) + \min(1,1)}{1+1+1+1+1}$$

$$= \frac{1+1+0+1+1}{5} = 4/5$$

$C = 5$

$r^* = 4 (r_2)$

$BP = 1 \ (c \geq r^*)$

$P_2 \Rightarrow$ bigrams are { love can, can make, make anything, anything possible }

$$P_2 = \frac{\min(1,1) + \min(0,1) + \min(0,1) + \min(1,1)}{1+1+1+1}$$

$$= \frac{1+0+0+1}{4} = 2/4 = 1/2$$

$$BLEU = 1 \times \exp\left(~~0.5~~ \ 0.5 \times \log 0.8 + 0.5 \times \log 0.5 \right)$$

$$= 0.8195744$$

Therefore C2 is considered better translation according to BLEU. From the context, C2 looks like a better translation than C1.

ii) r_2 is lost, only with respect to r_1 only

<u>For C1:</u>

$P_1 \Rightarrow$ 1-gram

$$P_1 = \frac{\min(0,1) + \min(1,1) + \min(1,1) + \min(1,1) + \min(0,1)}{1+1+1+1+1} = \frac{3}{5}$$

$C = 5 \quad r^* = 6.$

$$BP = \exp(1 - 6/5) = 0.81873$$

$P_2 \Rightarrow$ { the love, love can, can always, always do }

$$P_2 = \frac{\min(0,1) + \min(1,1) + \min(1,1) + \min(0,1)}{1+1+1+1} = \frac{2}{4} = 1/2$$

$$\text{BLEU} = 0.81873 \times \exp(0.5 \times \log 0.6 + 0.5 \times \log 0.5)$$
$$= 0.630315$$

For C2:

$$P1 = \frac{\min(1,1) + \min(1,1) + \min(0,1) + \min(0,1) + \min(0,1)}{1+1+1+1+1} = \frac{2}{5}$$

P2 ⇒ bigrams are { love can, can make, make anything, anything possible }

$$P2 = \frac{\min(1,1) + \min(0,1) + \min(0,1) + \min(0,1)}{1+1+1+1} = \frac{1}{4}$$

$C = 5$

$r^* = 6$

$$BP = \exp(1 - 6/5) = 0.81873$$

$$\text{BLEU} = 0.81873 \times \exp(0.5 \times \log 0.4 + 0.5 \times \log 0.25)$$

$$= 0.81873 \times 0.60653 = 0.496584$$

Now C1 receives the higher BLEU score. Though the BLEU score looks high for C1, it doesn't ~~[illegible]~~ look like a better translation than C2.

iii) ~~[illegible]~~ Evaluating the MT systems with respect to only one reference isn't good practice because a single source sentence might have multiple good translations. The MT system might produce a good translation but because of the unavailability of similar reference translation, the candidate might get a low BLEU score. So a limited number of different translation should be there for reference for evaluating a single MT. Also using large number of reference text is also not preferable as it will lead to high BLEU scores even for average translations

1 v) Advantages:

→ Bleu scores are quick and inexpensive to calculate compared to human evaluation

→ It is language independant whereas in human evaluation, we need a different person for evaluating each types of MT tasks.

Disadvantages:

→ A single or two reference translation may sometimes fail to cover good translations, while large reference translations might give high BLEU score even for average translation. So, a tradeoff should be attained

→ It is independant of grammar. A small word change gets penalized by BLEU in very small amount. But the small change can affect the entire sentence's meaning