Assignment 4 - Written: Neural Machine Translation

1. Neural Machine Translation with RNNs

(g) The $generate_{sentmasks}()$ function in $\text{nmt}_{\text{model.py}}$ produces a tensor called enc_{masks}. It has the shape (batch size, max source sentence length) and contains 1s in positions corresponding to 'pad' tokens in the input, and 0s for non-padded tokens. Look at how the masks are used during the attention computation in the 'step()' function. Explain in ~three sentences what effect the masks have on the entire attention computation. Then explain in one or two sentences why it is necessary to use the masks in this way.

Solution: The mask is placed over the attention scores, e_t, and for each individual score masked by a 1, i.e. a boolean *true*, we fill it with $-\infty$ (the smallest possible float). So, in essence, we're saying, for these entries marked as 'pad' tokens, pay them NO mind at all! These tokens get negligible probability weight in the attention distribution and it's corresponding hidden encoded state will be faded away in the attention output.

Now, if we didn't use the masks in this way, padding tokens would somewhat dampen the attention distribution.

(i) Please report the model's corpus BLEU Score.

```
(machine-learning) code [master●●] % sh run.sh test
load test source sentences from [./en_es_data/test.es]
load test target sentences from [./en_es_data/test.en]
load model from model.bin
Decoding: 100%|████████████████████| 8064/8064 [04:04<00:00, 32.96it/s]
Corpus BLEU: 22.637878690123642
```

Figure 1: The model's corpus BLEU Score is roughly 22.64

(j) Provide one possible advantage and disadvantage of each attention, mechanism with respect to either of the other two attention mechanisms. As a reminder:

- Dot Product Attention: $e_{t,i} = s_t^T h_i$
- Multiplicative Attention: $e_{t,i} = s_t^T W h_i$
- Additive Attention: $e_{t,i} = v^T(W_1 h_i + W_2 s_t)$

where $\{v, W, W_1, W_2\}$ are trainable parameters.

Solution:

1. Dot Product
    - **Advantage**: Efficient to compute, memory efficient, and easy to interpret state similarities. Any source hidden state vector that is orthogonal-ish to the target hidden vector will be "washed away". The efficiency can be seen as advantage over the additive mechanism.
    - **Disadvantage**: Rigid. Little flexibility compared to the other two.

1. Multiplicative

    - **Advantage**: Efficient to compute. The learnable parameter matrix, W, allows us to transform the source's hidden state vectors in a way that reflects lower loss in the task (depending on how well training goes). This flexible linear transformation can be seen as advantage over the *dot product* mechanism.
    - **Disadvantage**: Still not as flexible as additive attention.

2. Additive

    - **Advantage**: Both the target hidden vector and source hidden vector states get their own learned transformation in W_1 and W_2, respectively. This makes for more flexibility in the scoring.
    - **Disadvantage**: Slower to compute than the other two mechanisms.

2. Analyzing NMT Systems

(a) For each example of a Spanish source sentence, reference (i.e., 'gold') English translation, and NMT (i.e., 'model') English translation, please:

1. Identify the error in the NMT translation.
2. Provide a reason why the model may have made the error (either due to a specific linguistic construct or specific model limitations).
3. Describe one possible way we might alter the NMT system to fix the observed error

Solution:

i.

1. **Error:** "... otro de mis favoritos" → "... another favorite of my favorites"
2. **Reason:** ...?
3. **Fix:** ...?

ii.

1. **Error:** "... probablemente soy el autor para ninos, mas leido" → "... I'm probably the author for children, more reading"
2. **Reason:** The model is unable to recognize the various kinds of word orderings in Spanish. Here we have the adjective *mas leido* (most widely read) come **after** the noun it modifies, *autor para ninos* (children's author). The model translates this word for word and outputs the error without taking into account the word ordering. In this case it'd be proper English to write the adjective before the noun as given in the reference translation.
3. **Fix:** Try using multiple RNN/LSTM layers or reversing the input words to reduce error rate.

iii.

1. **Error:** "... Richard Bolingbroke" → "... Richard <UNK>"
2. **Reason:** The name *Bolingbroke* was not able to be produced (because not in vocabulary).
3. **Fix:** Either add the name to the vocabulary, learn to "copy" from source text (Gulcehre et al), or switch to character-based embeddings as input.

iv.

1. **Error:** "... dar vuelta a la manzana" → "... have to go back to the apple"
2. **Reason:** The model is unable to take Spanish idioms and produce a translation with the correct semantics in English (we just get literal translations).
3. **Fix:** Train on a well labeled dataset with idiom phrases.

v.

1. **Error:** "... bano de la sala de profesores" → "... bathroom in the women's room"
2. **Reason:** Female context suggests women? Gender-bias with *professor* translation?
3. **Fix:** Feed more data?

vi.

1. **Error:** "100,000 hectareas" → "100,000 acres"
2. **Reason:** The model does not understand unit conversions from metric system (*hectare*) to English based (US & Imperial) systems (*acre*).
3. **Fix:** Not sure... Teach the model conversions!?

(b) Now it is time to explore the outputs of the model that you have trained! The test-set translations your model produced in question 1-i should be located in outputs/test outputs.txt. Please identify 2 examples of errors that your model produced.2 The two examples you find should be different error types from one another and different error types than the examples provided in the previous question.

Solution:

i.

1. **English-Source**: "**We have this bucket list**, we have these things we want to do in life, and I thought about all the people I wanted to reach out to that I didn't, all the fences I wanted to mend, all the experiences I wanted to have and I never did."
2. **Spanish-Source**: "**Tenemos esta lista de cosas para hacer antes de morir**, estas cosas que queremos hacer en vida, y pens en toda la gente a las que quera llegar y no lo hice, todas las cercas que quera reparar, todas las experiencias que he querido tener y nunca tuve."
3. **Model-Translation**: "**We have this list of things to do before they die**, these things that we want to do in life, and I thought about all the people I wanted to get and I did all the fences that I wanted to <unk> all the experiences I've wanted to have and never <unk>"
4. **Error**: "Tenemos esta lista de cosas para hacer antes de morir" → "We have this list of things to do before they die"
5. **Reason**: The model is unable to take Spanish sentences and map them to corresponding idioms in English, when possible. Here "bucket list" is the common English idiom for a list of things to do before death. The model can't deduce this from the given source. Note this is different from problem **(a) iv** in that we want a translation from *phrase* → *idiom*, instead of *idiom* → *phrase*.
6. **Fix**: If the model can work with the right context window size and we train it on well labeled data having Spanish phrases mapped to common English idioms, things should work better.

ii.

1. **English-Source**: "Pink is my favorite color."
2. **Spanish-Source**: "El rosa es mi color favorito."
3. **Model-Translation**: "The rose is my favorite color."
4. **Error**: "El rosa..." $\rightarrow$ "The rose"
5. **Reason**: The model has trouble distinguishing between a word with semantic differences depending on the specified gender of a noun. Here *el rosa* is masculine which suggests **pink** (color), where as **la rosa** (feminine) translates to **rose** (botany).
6. **Fix**: Ensure data is labeled properly to handle grammatical gender usage. By context the model should know the difference. It should correlate color with pink and not rose. Maybe it'd work if it had to translate "Mi color favorito es el rosa"?

(c)
i.
Source Sentence **s: el amor todo lo puede**
Reference Translation r_1: love can always find a way
Reference Translation r_2: love makes anything possible
NMT Translation c_1: the love can always do
NMT Translation c_2: love can make anything possible

Please compute the BLEU scores for c_1 and c_2. Let $\lambda_i = 0.5$ for $i \in \{1, 2\}$ and $\lambda_i = 0$ for $i \in \{3, 4\}$ (this means we ignore 3-grams and 4-grams, i.e., don't compute p_3 or p_4). When computing BLEU scores, show your working (i.e., show your computed values for p_1, p_2, c, r^*, and BP.

Solution:

$\mathbf{c_1}$: *the love can always do*

$$
\begin{aligned}
p_1 &= \frac{\Sigma_{g \in \{\text{the, love, can, always, do}\}} min(max_{i=1,2} Count_{r_i}(g), Count_{c_1}(g))}{\Sigma_{g \in \{\text{the, love, can, always, do}\}} Count_{c_1}(g)} \\
&= \frac{0+1+1+1+0}{5} \\
&= \frac{3}{5} \\
p_2 &= \frac{\Sigma_{g \in \{\text{the love, love can, can always, always do}\}} min(max_{i=1,2} Count_{r_i}(g), Count_{c_1}(g))}{\Sigma_{g \in \{\text{the love, love can, can always, always do}\}} Count_{c_1}(g)} \\
&= \frac{0+1+1+0}{4} \\
&= \frac{1}{2}
\end{aligned}
$$

Brevity Penalty: $c = len(c_1) = 5$, $r^* = 5$. Because $c \geq r^*$

$$BP = 1$$

BLEU: $\lambda_1 = \lambda_2 = 0.5$

$$\begin{aligned}\text{BLEU}_{c_1} &= BP * exp(\lambda_1 log(p_1) + \lambda_2 log(p_2)) \\ &= 1 * exp(0.5 * log(\frac{3}{5}) + 0.5 * log(\frac{2}{4})) \\ &\approx 0.76994\end{aligned}$$

c_2: *love can make anything possible*

$$\begin{aligned}p_1 &= \frac{\Sigma_{g \in \{\text{love, can, make, anything, possible}\}} min(max_{i=1,2} Count_{r_i}(g), Count_{c_2}(g))}{\Sigma_{g \in \{\text{love, can, make, anything, possible}\}} Count_{c_2}(g)} \\ &= \frac{1+1+0+1+1}{5} \\ &= \frac{4}{5} \\ p_2 &= \frac{\Sigma_{g \in \{\text{love can, can make, make anything, anything possible}\}} min(max_{i=1,2} Count_{r_i}(g), Count_{c_2}(g))}{\Sigma_{g \in \{\text{love can, can make, make anything, anything possible}\}} Count_{c_2}(g)} \\ &= \frac{1+0+0+1}{4} \\ &= \frac{1}{2}\end{aligned}$$

Brevity Penalty: $c = len(c_2) = 5$, and $r^* = 5$. Since $c \geq r^*$, we have that

$$BP = 1$$

BLEU:

$$\begin{aligned}\text{BLEU}_{c_2} &= BP * exp(\lambda_1 log(p_1) + \lambda_2 log(p_2)) \\ &= 1 * exp(0.5 * log(\frac{4}{5}) + 0.5 * log(\frac{1}{2})) \\ &\approx 0.81957\end{aligned}$$

c_2 is considered a better translation according to BLEU. I agree that it is a better translation as c_1 doesn't even convey the same meaning as the references.

ii.) Our hard drive was corrupted and we lost Reference Translation r2. Please recompute BLEU scores for c1 and c2, this time with respect to r1 only. Which of the two NMT translations now receives the higher BLEU score? Do you agree that it is the better translation

Solution:

c_1: *the love can always do*

$$
\begin{aligned}
p_1 &= \frac{\Sigma_{g \in \{\text{the, love, can, always, do}\}} min(max_{i=1,2} Count_{r_i}(g), Count_{c_1}(g))}{\Sigma_{g \in \{\text{the, love, can, always, do}\}} Count_{c_1}(g)} \\
&= \frac{0+1+1+1+0}{5} \\
&= \frac{3}{5} \\
p_2 &= \frac{\Sigma_{g \in \{\text{the love, love can, can always, always do}\}} min(max_{i=1,2} Count_{r_i}(g), Count_{c_1}(g))}{\Sigma_{g \in \{\text{the love, love can, can always, always do}\}} Count_{c_1}(g)} \\
&= \frac{0+1+1+0}{4} \\
&= \frac{1}{2}
\end{aligned}
$$

Brevity Penalty: $c = len(c_1) = 5$, $r^* = 5$. Because $c \geq r^*$

$$BP = 1$$

BLEU: $\lambda_1 = \lambda_2 = 0.5$

$$
\begin{aligned}
\text{BLEU}_{c_1} &= BP * exp(\lambda_1 log(p_1) + \lambda_2 log(p_2)) \\
&= 1 * exp(0.5 * log(\frac{3}{5}) + 0.5 * log(\frac{2}{4})) \\
&\approx 0.76994
\end{aligned}
$$

c_2: *love can make anything possible*

$$
\begin{aligned}
p_1 &= \frac{\Sigma_{g \in \{\text{love, can, make, anything, possible}\}} min(max_{i=1,2} Count_{r_i}(g), Count_{c_2}(g))}{\Sigma_{g \in \{\text{love, can, make, anything, possible}\}} Count_{c_2}(g)} \\
&= \frac{1+1+0+0+0}{5} \\
&= \frac{2}{5} \\
p_2 &= \frac{\Sigma_{g \in \{\text{love can, can make, make anything, anything possible}\}} min(max_{i=1,2} Count_{r_i}(g), Count_{c_2}(g))}{\Sigma_{g \in \{\text{love can, can make, make anything, anything possible}\}} Count_{c_2}(g)} \\
&= \frac{1+0+0+0}{4} \\
&= \frac{1}{4}
\end{aligned}
$$

Brevity Penalty: $c = len(c_2) = 5$, and $r^* = 5$. Since $c \geq r^*$, we have that

$$BP = 1$$

BLEU:

$$\begin{aligned}\text{BLEU}_{c_2} &= BP * exp(\lambda_1 log(p_1) + \lambda_2 log(p_2)) \\ &= 1 * exp(0.5 * log(\frac{2}{5}) + 0.5 * log(\frac{1}{4})) \\ &\approx 0.60653\end{aligned}$$

Now $\mathbf{c_1}$ receives a higher score. I do not agree c_1 is a better translation.

iii. Due to data availability, NMT systems are often evaluated with respect to only a single reference translation. Please explain (in a few sentences) why this may be problematic.

Solution: The references are somewhat subjective translations. No translator can be perfect, so certain things can be interpreted differently. Someone translating to or from a non-native tongue may miss idioms and resort to literal translations.

iv. List two advantages and two disadvantages of BLEU, compared to human evaluation, as an evaluation metric for Machine Translation.

Solution:

Advantges:

1. It is a single simple quantitative score that removes subjectiveness from the evaluation process.
2. Fast to compute as opposed to forcing humans to evaluate translations.

Disadvantage:

1. It lacks the ability to evaluate sentence/corpus semantics.
2. It lacks the ability to evaluate structure/positions.