Stanford CS224n HW A4

David Chen

August 2020

1. Implementation and Written Questions

(g) Padding is done because the input sentences all need to have equal lengths. The mask shows where padding was added and where original sentence content is. The step() function then sets the attention vector to $-\infty$ so that these locations are zero in the attention distribution α_t.

```
epoch 15, iter 97910, avg. loss 24.64, avg. ppl 3.26 cum. examples 61120, speed 6853.00 words/sec, time elapsed 9455.59 sec
epoch 15, iter 97920, avg. loss 25.42, avg. ppl 3.28 cum. examples 61440, speed 6806.19 words/sec, time elapsed 9456.60 sec
epoch 15, iter 97930, avg. loss 24.08, avg. ppl 3.34 cum. examples 61760, speed 6576.39 words/sec, time elapsed 9457.57 sec
epoch 15, iter 97940, avg. loss 25.38, avg. ppl 3.29 cum. examples 62080, speed 7138.51 words/sec, time elapsed 9458.52 sec
epoch 15, iter 97950, avg. loss 23.67, avg. ppl 3.34 cum. examples 62400, speed 6699.98 words/sec, time elapsed 9459.46 sec
epoch 15, iter 97960, avg. loss 25.49, avg. ppl 3.25 cum. examples 62720, speed 6814.26 words/sec, time elapsed 9460.48 sec
epoch 15, iter 97970, avg. loss 23.26, avg. ppl 3.05 cum. examples 63040, speed 6927.56 words/sec, time elapsed 9461.44 sec
epoch 15, iter 97980, avg. loss 26.71, avg. ppl 3.56 cum. examples 63360, speed 6897.57 words/sec, time elapsed 9462.42 sec
epoch 15, iter 97990, avg. loss 25.14, avg. ppl 3.28 cum. examples 63680, speed 6810.55 words/sec, time elapsed 9463.41 sec
epoch 15, iter 98000, avg. loss 23.83, avg. ppl 3.25 cum. examples 64000, speed 6792.90 words/sec, time elapsed 9464.36 sec
epoch 15, iter 98000, cum. loss 24.52, cum. ppl 3.27 cum. examples 64000
begin validation ...
validation: iter 98000, dev. ppl 7.176201
hit patience 5
hit #5 trial
early stop!
```

(i)

```
!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 [./en_es_data/test.es]
load test target sentences from [./en_es_data/test.en]
load model from model.bin
Decoding: 100% 8064/8064 [05:38<00:00, 23.84it/s]
Corpus BLEU: 35.702456533789146
```

```
output (Google Colab w/ GPU, took around 2.7 hours)
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
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 [05:38<00:00, 23.84it/s]
Corpus BLEU: 35.702456533789146
```

(j)

- Dot product attention is simple to calculate, and requires no extra weights or hyperparameters. However, s_t and h_i have to be the same dimension for the dot product to line up.
- In Multiplicative attention, the added W gives extra representation, and the dimensions don't need to line up (we can adjust the dims of W instead). However, the W can be quite costly, and it's a new parameter we need to train.
- Additive attention gives better efficiency in high dimensions, but has two parameters and a new hyperparameter to tune.

2. Analyzing NMT Systems

(a)

i. **Error:** favorite of my favorites
Reason: I'm not sure? Maybe "another favorite" is common in English, but in Spanish "another one of my favorites" is more common?
Fix: Train on more similar examples.

ii. **Error:** "mas leido" (most read), is translated incorrectly as "more reading", when it should be referring to the previous part of the sentence "I am the author for children."
Reason: The model has a difficult time understanding the comma and the additional information provided after the comma. The adjective here is after the noun, but our model translates the sentence in order and doesn't account for possible word ordering differences.
Fix: Increase model capacity, maybe focus on passing the input in backwards to capture the ordering relationships. Utilize more LSTMs with more "memory" of past words.

iii. **Error:** The last name Bolingbroke fails to translate and instead goes to <unk>.
Reason: Bolingbroke is not in the vocabulary.
Fix: We can try adding a bunch of proper nouns into the vocabulary, or we can try to have the model learn to directly copy certain parts over. We can edit our embeddings so that they capture less frequent words.

iv. **Error:** "dar vuelta a la manzana" is translated as "to go back to the apple" instead of "to go around the block".
Reason: Using "manzana" to denote an apple is much more common than "block", and the model fails to recognize the context of "dar vuelta a" meaning "to turn". "Dar vuelta" can also mean "to return", and the model chooses this definition and translates to "to go back".
Fix: The dataset probably does not have many examples where manzana is used to mean block instead of apple. We can add some more of these examples. The dataset also needs more idioms, since the model currently translates directly. Finally, maybe we can try and make the model memorize more, so maybe it can use the context of "dar vuelta a" to pick block instead of apple.

v. **Error:** "la sala de profesores" is incorrectly translated to "women's room".
Reason: Possible gender bias, with professors not being linked strongly to women. It's also possible that in Spanish the idea of a "teacher's lounge" is less frequently used. Additionally, the model may be using the context of the bathroom, as it's common to go to the bathroom in the women's room (and not a teacher's lounge).
Fix: To help fix the gender bias, more training examples with professors being women could help. Maybe more examples could be added where someone goes to the bathroom in a specific room.

vi. **Error:** 100,000 hectares is translated to 100,000 acres.
Reason: The model has a difficult time translating units, since it doesn't understand conversions.
Fix: Add something special to help the model learn conversions? Add more examples with various types of units? Maybe remove the big numbers in front and just directly translate smaller units. Even then the model might struggle with math.

(b) 1. Source: Quiero que imaginen a dos parejas en 1979, el mismo dia, exactamente en el mismo momento, cada una concibiendo un bebe. Bien.
Reference: I want you to imagine two couples in the middle of 1979 on the exact same day, at the exact same moment, each conceiving a baby – okay?
NMT Translation: I want you to imagine two couples in 1979 , the same day , exactly at the same time , every single baby . Okay .
The phrase "cada una concibiendo un bebe" should translate to "each conceiving a baby", but instead the NMT model produces "every single baby". The word "conceiving" does not appear anywhere in the NMT translation, which is odd. It's possible that the training set doesn't many phrases about conceiving babies, or maybe the cocibiendo Spanish present participle isn't as commonly used. To fix this issue, we can insert more sentences with these phrases, or maybe we can take into account more words in the embedding.

2. Source: Nuestra nacion se basa en un concepto del individualismo muy romantico.
Reference: And our nation's really founded on a very romantic concept of individualism.
NMT Translation: Our nation is based on very romantic income .
The phrase "individualismo muy romantico" means "romantic concept of individualism", but our model translates it to "romantic income". The NMT model fails to recognize that individualism is being described as romantic, and it substitutes income (maybe because income is frequently spoken about in "our nation"?). To fix this, our model would need to be larger so it can learn more of these idioms, and we might need more training examples.

(c) i. **s**: el amor todo lo puede
$\mathbf{r}_1$: love can always find a way
$\mathbf{r}_2$: love makes anything possible
$\mathbf{c}_1$: the love can always do
$\mathbf{c}_2$: love can make anything possible
len(c1) = 5, len(c2) = 5, len(r1) = 6, len(r2) = 4
For $\mathbf{c}_1$:

$$p_1 = \frac{0+1+1+1+0}{5} = 0.6$$

$$p_2 = \frac{0+1+1+0}{4} = 0.5$$

BP = 1
BLEU c1 = $1 \times \exp(0.5\ln(0.6) + 0.5\ln(0.5)) = \mathbf{0.5477}$
For $\mathbf{c}_2$:

$$p_1 = \frac{1+1+0+1+1}{5} = 0.8$$

$$p_2 = \frac{1+0+0+1}{4} = 0.5$$

BP = 1
BLEU c2 = $1 \times \exp(0.5\ln(0.8) + 0.5\ln(0.5)) = \mathbf{0.6324}$
The second translation has a higher BLEU score, and I definitely agree with this. "The love can always do" doesn't really make sense, but "love can make anything possible" is certainly better.

ii. len(c1) = 5, len(c2) = 5, len(r1) = 6
For $\mathbf{c}_1$:

$$p_1 = \frac{0+1+1+1+0}{5} = 0.6$$

$$p_2 = \frac{0+1+1+0}{4} = 0.5$$

BP = $\exp(1 - \frac{5}{6})$
BLEU c1 = $\exp(1 - \frac{6}{5}) \times \exp(0.5\ln(0.6) + 0.5\ln(0.5)) = \mathbf{0.4484}$
For $\mathbf{c}_2$:

$$p_1 = \frac{1+1+0+0+0}{5} = 0.4$$

$$p_2 = \frac{1+0+0+0}{4} = 0.25$$

BP = $\exp(1 - \frac{5}{6})$
BLEU c2 = $\exp(1 - \frac{6}{5}) \times \exp(0.5\ln(0.4) + 0.5\ln(0.25)) = \mathbf{0.2589}$
Now the first translation has a much higher BLEU score, but I disagree.

iii. Oftentimes a sentence has many different translations that are all correct. When only a single reference translation is used, many good translations receive poor BLEU scores. Minor word choice differences can lead to fewer n-gram overlaps, even though the words used might be very similar.

iv. **Advantages**

- BLEU is fast and automated, so it can be run without humans constantly working. Human bias can't influence the score.
- BLEU is a relatively simple formula that is easy to implement: no understanding of the two languages is needed.

Disadvantages

- Since BLEU only uses n-gram overlaps, it may not be a good measure of sentence translation quality, since oftentimes similar words can all be the correct translation.
- BLEU cannot capture semantics, logic, or grammar.