Stanford CS 224n Assignment 4

Shivanshu Shekhar

December 2021

1 Neural Machine Translation with RNNs (45 points)

(a) CODE

(b) CODE

(c) CODE

(d) CODE

(e) CODE

(f) CODE

(g) The mask makes the attention scores of "¡pad¿" tokens negative infinity so that the weighting coefficient of the encoder hidden vector corresponding to these tokens becomes zero i.e. they are given zero attention.

It is necessary to use mask as not all the sentences are of the same length and to make them of the same length we add "¡pad¿" tokens, so these tokens are just to fill spaces **they doesn't contain any useful information about anything** so it is meaning less to use there hidden states for decoding.

(h) CODE

(i) The training ended on - epoch with a BLEU score of - on the test set.

(j)
- Dot-product attention is very efficient as compared to multiplicative attention, for the dot product attention we need to have same dimensions of two vectors.
- Multiplicative attention is faster and more space-efficient in practice but additive attentions performs better for large dimensions.

2 Analyzing NMT Systems (30 points)

(a) (i)
- **Error:** favorite of my favorites
- **Reason:**
- **Solution:**

(ii)
- **Error:** Didn't capture the relationship between punctuation.
- **Reason:** The model failed to capture sentence structure and directly translated word by word which may be caused by insufficient representation of the context.
- **Solution:** Increase the hidden layer size and increase the training data accordingly as overfitting might become a problem.

(iii)
- **Error:** The model was unable to recognize Bolingbroke.
- **Reason:** Bolingbroke wasn't present at the train time that is it was out of vocabulary for the model.
- **Solution:** We can leverage the text in the test set itself as Bolingbroke is a noun with same representation in both the language or simply expand our vocabulary but it might get slower to train

(iv)
- **Error:** block replaced by apple
- **Reason:** The word "manzana" has two meaning apple and block so the model got confused between those two.
- **Solution:** Include more training data with "manzana" as block.

(v)
- **Error:** Teachers lounge got translated as women's room.
- **Reason:** There is a gender bias in the training data as men can be teacher's too but since the training data has mostly woman as a teacher the line gets translated as women's room.
- **Solution:** Include mode unbiased data.

(vi)
- **Error:** 100,000 hect areas got converted to 250 thousand areas.
- **Reason:** The model failed to keep the number constant between translations
- **Solution:** The model needs to trained for number representation to learn better about numbers and also convert all numbers to digit or text training data shouldn't have mixtures of both as this could lead to inconsistency.

(b) I didn't have enough GPU resources to train and neither do I have Colab pro to train the model for large amounts of time so I couldn't get the sample =(, the code ran fine for local tests so I just need more GPU resources, I will update this section once I have all my hardware needs satisfied.

(c) (i) For c_1:
$p_1 = 0.6$, $p_2 = 0.5$, $\text{len}(c_1) = 5$, $\text{len}(r_{smallest}) = 4$
Hence BP = 1
$BLEU = \exp 0.5 * (\ln 0.6 + \ln 0.5 = 0.5477$
For c_2:
$p_1 = 0.8$, $p_2 = 0.5$, $\text{len}(c_2) = 5$, $\text{len}(r_smallest) = 4$
Hence BP = 1
$BLEU = \exp 0.5 * (\ln 0.8 + \ln 0.5) = 0.6324$
According to BLEU c_2 is better as it should be.

Unigram	**Numerator**
the	0
love	1
can	1
always	1
do	0

Bigram	**Numerator**
the love	0
love can	1
can always	1
always do	0

Table 1: c_1

Unigram	**Numerator**
love	1
can	1
make	0
anything	1
possible	1

Bigram	**Numberator**
love can	1
can make	0
make anything	0
anything possible	1

Table 2: c_2

(ii) For c_1: $p_1 = 0.6$, $p_2 = 0.5$, len$(c_1) = 5$, len$(r_{smallest}) = 6$
Hence BP = $\exp 1 - 6/5$
$BLEU = BP * \exp 0.5 * (\ln 0.6 + \ln 0.5 = 0.4484$
For c_2
$p_1 = 0.4$, $p_2 = 0.25$, len$(c_2) = 5$, len$(r_s mallest) = 4$
Hence BP = $\exp 1 - 6/5$
$BLEU = BP * \exp 0.5 * (\ln 0.4 + \ln 0.25) = 0.2589$
According to BLEU c_1 is better which is not correct.

Unigram	**Numerator**
the	0
love	1
can	1
always	1
do	0

Bigram	**Numerator**
the love	0
love can	1
can always	1
always do	0

Table 3: c_1

Unigram	**Numerator**
love	1
can	1
make	0
anything	0
possible	0

Bigram	**Numberator**
love can	1
can make	0
make anything	0
anything possible	0

Table 4: c_2

(iii) With only one reference the good translation may be evaluated as bad as there are many ways to successfully translate a sentence. With more references the n-gram space increases and gives a more accurate result.

(iv) **Advantages**

- It is easy to implemeant and provides a single number to compare different translations.
- It is fully automated and cheap to compute and is a very fast way to evaluate a model.

Disadvantages

- This is not reliable if we have very less number of reference sentence.
- This measures only n-gram overlap and there is no measure that directly measures the grammatical accuracy and semantics.