CS 224n Assignment #4

Yoshihiro Kumazawa

June 23, 2020

1. Neural Machine Translation with RNNs

(a) See `utils.py`.

(b) See `model_embeddings.py`.

(c) See `nmt_model.py`.

(d) See `nmt_model.py`.

(e) See `nmt_model.py`.

(f) See `nmt_model.py`.

(g) The masked logits are made $-\infty$ and hence do not affect the softmax calculation of the other logits. Those masks are put on the hidden states from the padded words, which no attention should be paid to.

(h) (Missing)

(i) The model's corpus BLEU Score was 35.83.

(j) One advantage of dot product attention compared to multiplicative attention is that it is less prone to overfitting since it does not have weight parameters. One disadvantage is its lower expressivity. One advantage of additive attention compared to multiplicative attention is that it can learn not to be affected by the hidden state very much by letting W_1 small. One disadvantage is that it is more prone to overfitting.

2. Analyzing NMT Systems

(a) i. 1. "Aquí" is translated to "Here's" instead of "So". "favorite" is redundant.

2. "Aquí" could be "Here", but the verb "is" is unnecessary anyway. A possible reason is that sentences with no verb were seldom fed during training time. The reason the NMT translated "otro" into "another favorite" instead of "another one" might be that NMT thought it could improve translation accuracy by understanding what such demonstrative words actually mean.

3. For the first error, feeding more data without verbs might work. For the second error, penalizing redundancy might be a good idea.

ii. 1. The latter half of the translation does not make sense.

2. Probably the model tried to directly translate the Spanish sentence word-by-word, which ended up producing the wrong sentence. Chances are the source sentence was inherently hard to translate due to some linguistic difference between Spanish and English.

3. To encourage the network to avoid unnatural direct translations, feeding more data where the reference translations are very different from possible direct translations might work.

iii. 1. "Bolingbroke" is interpreted as unknown.
2. This is simply because the model did not encounter the word "Bolingbroke" during training time.
3. One can either add such proper nouns to the vocabulary or let the network treat unseen words as they are (i.e. do not translate them). To implement the latter idea, imagine you have unknown words in the source sentence and you have got $<unk>$tokens during translation. You can use the attention vector to guess which word in the source language each of those unknown words corresponds to.

iv. 1. The idiom "dar vuelta a la manzana" is translated to "go back to the apple" unlike "go around the block" in the reference translation.
2. The model directly translated the idiom to "go back to the apple", which is less meaningful in English.
3. One possible way to get around this is pre-translate such idioms in the training set and do not let the machine learn the direct translation. More precisely, during translation we can replace the idiom by some random word which is not in the vocabulary, run the NMT and replace the possible unknown token in the target sentence which we expect has a high attention score on the random word representing the idiom.

v. 1. "la sala de profesores" is translated to "the women's" room" unlike "the teacher's lounge" in the reference translation.
2. It might be because the model encountered so many examples of women's bathrooms that it translated the sentence in that way. Another reason might be that the subject of the sentence is "she", which encouraged the machine to predict the inappropriate "women's room".
3. Feeding more data might possibly solve the problem.

vi. 1. "100,000 hect´areas" is translated to "100,000 acres" unlike "250 thousand acres" in the reference translation.
2. This is a unit conversion error.
3. One possible way is to pre-convert such quantities with units which are rarely used in English. My solution for (iv) with an easy calculation might work for that.

(b) The following (i) and (ii) are presented as examples.

i. 1. Este es un tmpano de Groenlandia, de tamao promedio.
2. This is an average-size Greenlandic iceberg.
3. This is a iceberg iceberg , average size .
4. There are 3 errors. Firstly, it puts "a" right before "iceberg", which has to be "an" instead. Secondly, it places 2 "iceberg"s in a row. Lastly, It totally ignores the word "Groenlandia".

5. I have no idea why those errors have ever been caused.
6. We can easily get around the first error by performing a post-processing after translation. One possible way to deal with the second error is to change the beam search algorithm to lower the score when the model tries to predict 2 identical words in a row. For the 3rd error, since proper nouns are often indispensable, it might be a good idea to encourage the machine to choose a word having a high attention score with a proper noun during beam search. In order to do that, we will need a method to judge whether a given word is a proper noun. One easy way is to look at the first letter and check if it is capital or not.

ii. 1. Si dejaba de moverse lo enterraban.
2. If it stopped twitching, you bury them.
3. If I stopped moving it .
4. A word corresponding to "enterraban" is missing.
5. The reason the model ignored "enterraban" might be that it translated "lo" to "it", which accidentally completed the phrase "I stopped moving it".
6. If we have some way to quantify grammatical correctness, we can take the correctness score into consideration during beam search to discourage the network from producing incomplete sentences.

(c) i. For both $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$, we have $BP = 1$. For $\boldsymbol{c}_1$, we have

$$p_1 = (0 + 1 + 1 + 1 + 0)/5 = 0.6,$$
$$p_2 = (0 + 1 + 1 + 0)/4 = 0.5.$$

For $\boldsymbol{c}_2$, we have

$$p_1 = (1 + 1 + 0 + 1 + 1)/5 = 0.8,$$
$$p_2 = (1 + 0 + 0 + 1)/4 = 0.5.$$

Hence we get

$$BLEU_{\boldsymbol{c}_1} = \exp(0.5 * \log(0.6) + 0.5 * \log(0.5)) = 0.77,$$
$$BLEU_{\boldsymbol{c}_2} = \exp(0.5 * \log(0.8) + 0.5 * \log(0.5)) = 0.82.$$

This result matches my intuition that $\boldsymbol{c}_2$ is the better translation.

ii. For both $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$, we still have $BP = 1$. For $\boldsymbol{c}_1$, both p_1 and p_2 remain unchanged. Hence $BLEU_{\boldsymbol{c}_1} = 0.77$ as in (i). For $\boldsymbol{c}_2$, we have

$$p_1 = (1 + 1 + 0 + 0 + 0)/5 = 0.4,$$
$$p_2 = (1 + 0 + 0 + 0)/4 = 0.25,$$

which gives us

$$BLEU_{\boldsymbol{c}_2} = \exp(0.5 * \log(0.4) + 0.5 * \log(0.25)) = 0.61.$$

Now $\boldsymbol{c}_2$ is worse than $\boldsymbol{c}_1$, which is against my intuition.

iii. It is often the case that good translations look very different from each other. That sometimes makes the $BLEU$ score counter-intuitive like in the previous questions.

iv. Advantages are:

- It is automatic and can be easily computed.
- It is objective. We can compare the score across different machines.

Disadvantages are:

- It sometimes does not match our intuition especially when we have few reference translations.
- It treats essentially the same words (like "make" and "makes" in the previous example) differently.