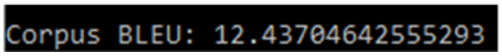
NLP Assignment 4 Sara Kodeiri 96521443

1. Neural Machine Translation with RNNs

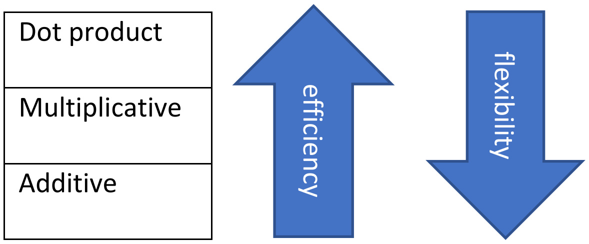
g) In the step function, what is masked gets set to -infinity. We don’t want to have the padding characters affecting the final output because it is not a word in the vocabulary. But in the data, we have multiple sentences which have the padding character in them to equalize the sentence lengths. Wherever the value is 1 in the mask matrix, the corresponding position in e_t is set to be masked and ignored and wherever the value is 0 in the mask matrix, the corresponding position in e_t remains transparent, letting through its value for calculating the weight alpha_t. Masking retains the matrix dimension without counting '<pad>' and weighing it, leaving it out of representation in Attention.

h)



```
Corpus BLEU: 12.43704642555293
```

i) Dot product vs. multiplicative: Dot product is less flexible but more efficient to compute because of ignoring the vectors that are [even almost] orthogonal. Multiplicative is more flexible and also efficient, but not as efficient as dot product. Dot product also captures information from the encoder hidden layer without their weights and relevant importance.
Additive vs. multiplicative: Additive is more flexible, but slower. It also has the advantage of capturing the underlying implications for every input word embedding.



2. Analyzing NMT Systems

a) “In linguistic typology, polysynthetic languages are highly synthetic languages, i.e., languages in which words are composed of many morphemes (word parts that have independent meaning but may or may not be able to stand alone).” (Wikipedia)
Since our source language contains many morphemes, using a subword-level NMT probably grasps a better understanding of it and takes individual morphemes into consideration. We might have a lot of unique words but also a lot of common morphemes. This way, the morphemes get their own weights and we can have a more accurate translation.

b) Word embeddings have more unique samples in them. For example, “revolution” “imagination” “imaginary” “situation” “revolutionary” need 5 different word embeddings but only 4 sub-word embeddings.

c) Multilingual machine translation processes multiple languages using a single translation model. Though data skew across language-pairs is a great challenge in NMT, it also creates an ideal scenario in which to study transfer, where insights gained through training on one language can be applied to the translation of other languages. On one end of the distribution, there are high-resource languages like French, German and Spanish where there are billions of parallel examples, while on the other end, supervised data for low-resource languages such as Yoruba, Sindhi and Hawaiian, is limited to a few tens of thousands. This effect is already known, but surprisingly encouraging, considering the comparison is between bilingual baselines (i.e., models trained only on specific language pairs) and a single multilingual model with representational capacity similar to a single bilingual model. This finding, hints that massively multilingual models are effective at generalization, and capable of capturing the representational similarity across a large body of languages. (Link in question) By my understanding, it's similar to how for polyglots, learning a third language is easier than learning the second one, and how the learning curve flattens after learning each one. The knowledge of the base language strengthens, and the translation rules become more evident over time.

d) i) There might have been instances where "hair" and "crown" have been used beside each other and have a similar embedding; since they both are on the top of one's head. Daisies has not been translated and I think it's probably because it's an uncommon word, so, it probably hasn't appeared a lot in the training data. If these are the issues, they can be fixed by having more uses of "crown" in the training data where "hair" and possible close words like "head" aren't mentioned.

ii) My guess is that in Cherokee, there is one pronoun for both things and people. A linguistic approach would be to have a normalized dataset for this cause, where "it", "he" and "she" are used almost the same number of times to be the Cherokee pronoun's translation. For a model fix, one thing I can think of is having a bidirectional decoder as well, so it sees "joy" sooner than the pronoun. This way, it might be able to realize that having joy is used to describe people, not things.

iii) "Littlefish" is a known word (since it's capitalized) and the most probable reason for the error is that it wasn't in the target language training data. The fix would be to have it in the data. If it already is there, having models who pay attention to name entities (like character-based models) could give a more accurate result.

e) i) Line 981: "and he saith unto him, Follow me. And he arose," is the same in both files.

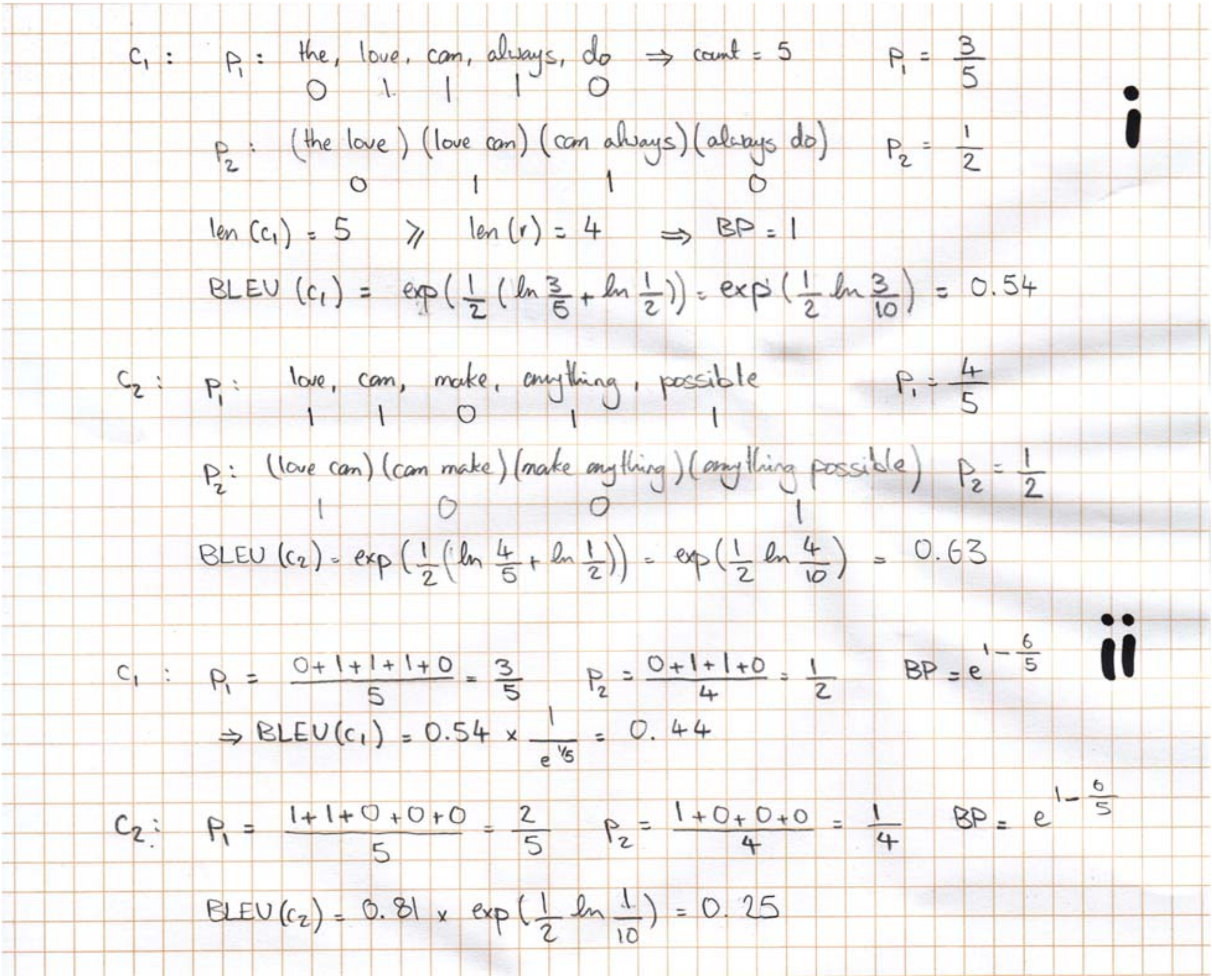
In train data, these examples and much more were found:

1- he saith unto him, Follow me.
2- and said unto him, Follow me.
3- And Jesus arose, and followed him,

The MT system has accurately translated the words it had frequently seen beside each other. It has fully learned something it has frequently seen.

ii) test.en : Jesus answered and said, Ye know not what ye ask.
test_outputs: Jesus answered and said, Ye cannot know that I know not what ye know that ye know not of the cup of the cup of the cup, and to do it.
The model decoding behavior seems to somehow neglect the attention layer outputs, not not take them into account as much as it should.

f) i) ii)



i

c_1 : P_1 : the, love, can, always, do $\Rightarrow$ count $= 5$ $P_1 = \frac{3}{5}$

0 1 1 1 0

P_2 : (the love) (love can) (can always) (always do) $P_2 = \frac{1}{2}$

0 1 1 0

$\text{len}(c_1) = 5 \gg \text{len}(r) = 4 \Rightarrow BP = 1$

$\text{BLEU}(c_1) = \exp\left(\frac{1}{2}\left(\ln\frac{3}{5} + \ln\frac{1}{2}\right)\right) = \exp\left(\frac{1}{2}\ln\frac{3}{10}\right) = 0.54$

c_2 : P_1 : love, can, make, anything, possible $P_1 = \frac{4}{5}$

1 1 0 1 1

P_2 : (love can) (can make) (make anything) (anything possible) $P_2 = \frac{1}{2}$

1 0 0 1

$\text{BLEU}(c_2) = \exp\left(\frac{1}{2}\left(\ln\frac{4}{5} + \ln\frac{1}{2}\right)\right) = \exp\left(\frac{1}{2}\ln\frac{4}{10}\right) = 0.63$

ii

c_1 : $P_1 = \frac{0+1+1+1+0}{5} = \frac{3}{5}$ $P_2 = \frac{0+1+1+0}{4} = \frac{1}{2}$ $BP = e^{1-\frac{6}{5}}$

$\Rightarrow \text{BLEU}(c_1) = 0.54 \times \frac{1}{e^{1/5}} = 0.44$

c_2 : $P_1 = \frac{1+1+0+0+0}{5} = \frac{2}{5}$ $P_2 = \frac{1+0+0+0}{4} = \frac{1}{4}$ $BP = e^{1-\frac{6}{5}}$

$\text{BLEU}(c_2) = 0.81 \times \exp\left(\frac{1}{2}\ln\frac{1}{10}\right) = 0.25$

When we had two references, c_2 turned to be the better translation with a higher BLEU score. I agree with this conclusion. In part ii, with only one reference, c_1 has scored higher, which I don't think it's the better translation.

iii) The example done above shows perfectly how this might affect the final result. the BLEU score is highly sensitive to word placement in the reference. A single reference translation can't possibly take into account all the instances of the word usage and all its possible neighbors.

iv) Advantages:

1. Quantitative and fast compared to human-evaluation being qualitative and slow.
2. Does not need too much memory or other resources

Disadvantages:

1. Not always accurate especially when it comes to semantics.
2. Highly sensitive to references which makes the process of choosing a good reference more important as opposed to having human evaluators use their common sense.