CS224n Winter 2019 Homework 4

SUNet ID: 05794739
Name: Luis Perez
Collaborators:

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

Problem 1

(a) In "utils.py".

(b) In "model_embeddings.py".

(c) In "nmt_model.py".

(d) In "nmt_model.py".

(e) In "nmt_model.py".

(f) In "nmt_model.py".

(g) As per the code, the masks end-up setting the $e_{t,i} = -\infty$ at all positions corresponding to 'pad' tokens. This corresponds to assigning $-\infty$ energy to the "annotations" (encoder hidden states) corresponding to our padded sequences. After running through a Softmax, this leads to a zero-probability for these states, which means they don't contribute to our overall attention vector a_t since this is the result of a weighed (by the softmax probability) average of the encoder hidden states.

It is necessary to use the masks in this way since the 'pad' tokens are artificial additions (mainly for performance) which should not be used by our decoder to generate translations. This information is not only useless, but could harm the overall ability of the model to translate sentences since many tranlations could be mis-matched in length, leading to 'pad' tokens being alinged with real worlds.

(h) Trained on Azure VM.

(i) The trained model's BLUE score on the provided test-set is 22.608129364779256.

(j) i Dot Product Attention

- **Advantage**: The biggest advantage is computational, as well as the intepretability of this mechanism. The computational complexity of dot product is $O(n)$. Furthermore, this mechanism is very intuitive, where values with high cosine-similarity to the query will receive the most attention.

- **Disadvantage**: One big disadvantage of this method is a practical one. In order to use this attention mechanism, the values and the query must be of the same dimension. Generally speaking, this is an unnecessary restriction. Additionally, there are not explicit parameters to learn for this attention step, therefore the model is more restricted.

ii Multiplicative Attention:

- **Advantage**: With the introduction of the weight matrix, there is no longer a restriction that the dimension of the query and the values must match. Furthermore, the weight matrix itself can be learned, allowing for more expressivity in the model (ie, a value vector that's close to the query vector can be given a low-score due to the linear transformation it first undergoes).
- **Disadvantage**: Computational more complex than dot product attention, and some level of interpretability is lost since now there is an additional linear transformation of the value vectors.

iii Additive Attention

- **Advantage**: Explicitly allows for tuning the "attention dimensionality" thereby allow for techniques which try to bottleneck or expand the capacitor of the attention mechanism.
- **Disadvantage**: The most complex of all of the mechanisms, with the least level of interpretability.

Problem 2

(a) i 1 The error in the translation consists mainly of mis-translating into "favorite of my favorites" rather than "one of my favorites." This is a dependency error, where the system was not able to maintain the dependency between "one" and "favorites".

2 The model likely made this error due to limitations in our decoder. While the model has attention over the entire input sentence, it has no knowledge of what words it will generate in the future. Therefore, I expect the model found "favorite of my" to be a good translation of "otro de mis favoritos", since the encoder has no knowlede of the future "favorites" translation which will be produced.

3 One possible mechanism for correcting this would be to allow the encoder to be bi-directional, similar the encoder. In this way, the encoder states will have knowledge of not only what has been translated so-far, but also what it proposes as a translation for future words.

ii 1 The error in the translation involves the loss of the the association between "most read" and "children's author". The NMT system mistakenly associates

"most read" with the US, rather than "author". This is an example of a word aligment error.

2 The model likely made this error due to the complexity of the input sentence, especially when it comes to the ordering it's given in. This is likely a limitation of the attention mechanism in it's ability to select annotations and context from far-away words and phrases.

3 One possible improvent would be to use additive attention instead of multiplicate attention. This form of attention is better suited at maintaining long-range dependencies given that the input and annotations have their own weight functions, which can be seperately learned, especially for translation systems. Other alternatives to the attention mechanism we used can be tried to.

iii 1 The error in the translation occurs when " Bolingbroke." is mistranslated to "¡unk¿".

2 The reason for this error is a model limitation since it has encoutered an out-of-vocabulary word. Our model maps all out-of-vobaculary words to a special token – as such, they cannot be translated.

3 A possible mitigation to this issue is to use a character-based model, rathern than a word-based model. With character-based models, there are no out-of-vocabulary words, and as such, the model will be able to make a somewhat reasonable guess at the translation.

iv 1 The error in the NMT translation is that it's translation occurs when the model translates "manzana" literally, rather than metaphorically as is intended.

2 This is a linguistic limitation, caused by the metaphorical aspect of the source sentence. The source sentence is an idiom, which is not meant to be translated literally. This idiom makes no literal sence, even in the source language. However, the model does not have a way to handle such a translation, and instead performs a literal translation which, while technically correct, is not conveying the same meaning.

v 1 The error in the NMT translation is that it's translating "teacher's lounge" to "women's room". This is not a model limitation per-se, rather a limitation in our training data. It is exposing the bias (women being associated with teacher) in our training data.

2 The model likely made this error because, in its input data, women and teacher would frequently be translated from the same word. This is an example of our model overfitting in our training data (to some extent) and picking up the bias inherint therein.

3 One possible solution for this problem would be to modify the training data so that the bias, especially for protected classes (gender, race, etc.) is reduced. This could be done by simply replacing gendered nouns with ungendered

versions, but this in itself is quite difficult. An alternative is to modify the NMT system so that gendered nouns are translated into all possible genders.

vi 1 The error is in the number of acres. The reference translation is 250,000 while the model translation is 100,000. This can be viewed as a limitation of the model as well as a liminitation of the training data.

2 The error is likely made because, in the training data, the number '100,000' almost always maps to itself (also '100,000') in the very infrequent number of cases in which it is encoutered by the model. As such, the model is incapable of understanding the context of surrounding this value (ie, hectares and acres).

3 One possible improvement to the model would be to have it operate at a subword level, which could help deal with these infrequently encountered words. However, for this problem specifically, it might be possible to remove this sort of conversion from the model entirely (ie, unit conversion), and instead have the model train on and output special "numeric" tokens which are filled in later by a more deterministic system.

(b) • **Example 1**

1 Le encontramos un lugar, la internamos, y la cuidamos y nos encargamos de su familia, porque era necesario, es algo que sabamos cmo hacer.

2 We found her one, we got her there, and we took care of her and watched over her family, because it was necessary.

3 We found a place, ¡unk¿ and ¡unk¿ and ¡unk¿ from their family, because it was necessary, because it was necessary, because it was necessary, because it was necessary, because it was necessary, because it was necessary, because it was necessary, because it was necessary, because it was necessary, because it was necessary, because it was necessary, because it was necessary, because it was necessary, because it was necessary, because it was necessary, because it

4 There are multiple errors in this translation, however, we focus on the fact that the translation is incorrectly repeating the last phrase, many times.

5 The error is likely caused by capacity limitations in our decoder. It appears that we have entered some sort of loop where the decoder loses knowledge of the previously output translation. This is likely caused by the capacity limitation.

6 There is one possible solutions we can consider for this type of error, where the decoder will sometimes get stuck on loops. The first, is to provide an attention mechanism not only over the annotated queries, but also over the previously translated words (in this way, the decoder can choose to focus on previous hidden states, thereby increasing capacity).

• **Example 2**

1 Si se fijan en esta foto... soy de origen italiano, todos los nios en Italia crecen con esta foto en la pared de su dormitorio. Pero la razn por la que les muestro esto es que ha sucedido algo muy interesante en las carreras de Frmula 1 en las ltimas dos dcadas.

2 Now if you take this picture – I'm Italian originally, and every boy in Italy grows up with this picture on the wall of his bedroom – but the reason I'm showing you this is that something very interesting happened in Formula 1 racing over the past couple of decades.

3 If you look at this picture, I'm from Italian ¡unk¿ all the children in Italy growing up with this picture on the wall of his bedroom. But the reason I show you this is that something very interesting is that has happened very interesting in the last two decades.

4 There are multiple errors in this translation, we focus on the fact that the formula completely drops the reference to Formula 1 racing from the translations, which appears to be some sort of model limitation.

5 The error is likely caused by the fact that Formula 1 is a very infrequently occuring word, and as such, during the decoder step, producing such a word as a tranlsation is relatively unlikely, leading the decoder to instead do a loop ("very interesting" is repeated.)

6 One possibly solution for this problem would be to use sub-words, rather than real-words. Similar to using a character based model, it would make sense to split based on frequently occuring sequence of characters.

(c) We first begin by making a few clarifications. Our $n-grams$ are case-insensitive (so, 'Love' and 'love' are the same word). However, we do no other processing (eg, no stemming, etc.). As such, 'make' and 'makes' are distinct words.

i We begin by computing the score for $\boldsymbol{c}_1$.

$$
\begin{aligned}
p_1 &= \frac{0+1+1+1+0}{1+1+1+1+1} && = 0.6 \\
p_2 &= \frac{0+1+1+0}{1+1+1+1} && = 0.5 \\
c &= 5 \\
r^* &= 4 \\
BP &= 1
\end{aligned}
$$

$$BLEU = 1 \times \exp\{0.5 \log 0.6 + 0.5 \log 0.5\} \approxeq 0.547723$$

Next, we compute the score for $\boldsymbol{c}_2$.

$$\begin{aligned} p_1 &= \frac{1+1+0+1+1}{1+1+1+1+1} && = 0.8 \\ p_2 &= \frac{1+0+0+1}{1+1+1+1} && = 0.5 \\ c &= 5 \\ r^* &= 4 \\ BP &= 1 \end{aligned}$$

$$BLEU = 1 \times \exp\{0.5 \log 0.8 + 0.5 \log 0.5\} \approxeq 0.632456$$

According to the above calucations, $\boldsymbol{c}_2$ is considered the better translation. This is in agreement with what I, as a human rater, consider to be the better translation.

ii We re-compute the previous, but using only $\boldsymbol{r}_1$ as a reference. We begin by computing the score for $\boldsymbol{c}_1$.

$$\begin{aligned} p_1 &= \frac{0+1+1+1+0}{1+1+1+1+1} && = 0.6 \\ p_2 &= \frac{0+1+1+0}{1+1+1+1} && = 0.5 \\ c &= 5 \\ r^* &= 6 \\ BP &= \exp(1 - \frac{6}{5}) && \approxeq 0.81873 \end{aligned}$$

$$BLEU = 0.81873 \times \exp\{0.5 \log 0.6 + 0.5 \log 0.5\} \approxeq 0.448438$$

Next, we compute the score for $\boldsymbol{c}_2$.

$$\begin{aligned} p_1 &= \frac{1+1+0+0+0}{1+1+1+1+1} && = 0.4 \\ p_2 &= \frac{1+0+0+0}{1+1+1+1} && = 0.25 \\ c &= 5 \\ r^* &= 6 \\ BP &= \exp(1 - \frac{6}{5}) && \approxeq 0.81873 \end{aligned}$$

$$BLEU = 0.81873 \times \exp\{0.5 \log 0.4 + 0.5 \log 0.25\} \approxeq 0.258905$$

According to the above calculations, we now have $\boldsymbol{c}_1$ as receiving the higher score. This is not the better translation, as per my human-rating abilities.

iii Evaluating on a single-reference translation is problematic because the score won't reflect possibly better translations, since it will only be comparing against a single

reference, which means the ability for the translation system to translate meaning (but not necessarily the exact words) will be penalized. This is clearly demonstrated, albeit in a toy example, by the two example evaluations above where with a single reference translation, the BLEU score is better for the qualitatively worse translation since the correct translation, which is capturing the true meaning, is using synonyms to words in the reference translation.

iv We presen two advantage to BLEU:

- It is cheap to compute, especially when compared to the alternative of asking humans to evaluate the quality of individual translations.
- It is consistent – on the same dataset, with the same translations, the BLEU score will always be the same. This is especially in comparison to human evaluation, which can not only vary from person to person, but also from day to day.

Next, we presen two disadvantages of BLEU:

- It only approximates semantic meaning (using n-gram language models), but does not capture it. Two very dissimilar sentences can be perfect translations, yet they will have low BLEU scores.
- As revealed in the example above, BLEU scores are heavily reliant on the available reference translations – too few, and you can expect poor results.