1 Question 1

The greedy search strategy for decoding presents the advantage of being computationally efficient. It is based on the principle of selecting at each time the next word as the one with the highest probability of coming after the incomplete sentence.

But because of this choice of only selecting the highest probable one, it becomes suboptimal. In fact, it can get stuck in the repetitive prediction because it will only look at the penultimate predicted word. For example, if after the prediction **The Cat**, it choses **The** as being the highest probable word, it will keep on predicting **The Cat The Cat The Cat The Cat...**.

In that sense it presents an inconvenient that for example is taken care off my decoding strategy of beam search, which will give the **k** most probable words to predict, but this method if highly inefficient computationally although it is asymptotically exact.

2 Question 2

The main problem that we observe on our translations is **Repetition**. It is caused by the architecture of the greedy search which chooses the highest probable word as a prediction from the previously predicted words. This can be solved by using a beam search with beam width **k**, which will give the prediction of the k next best words instead of only one.

One other approach proposed by [2] is the NMT-COVERAGE. It is presented as a solution for the problem of over-translation which happens when a word is unnecessarily translated multiple times.
The idea of the NMT-coverage is to introduce a coverage set, which keeps track of the words that are being translated. It starts with a coverage set which looks like this $\{0,0,0,0\}$ for a sentence of 4 words for example, and aims at reaching a coverage vector of $\{1,1,1,1\}$ meaning that all the words have been translated.

3 Question 3

I created a function to visualize the source/target alignment scores, inspired from the tutorial in [1]. Down below we can see the results for 2 examples.

In the first one I explored the adjective/noun inversion with the case of 'white horse' which should be translated into 'cheval blanc' in french. We can see that the model does a good job at giving the right order for the french language. We can still notice the repetition issue for both examples.

4 Question 4

The examples :

- I did not mean to hurt you $\rightarrow$ je n ai pas voulu intention de blesser blesser blesser blesser blesser blesser . blesser . blesser
- She is so mean $\rightarrow$ elle est tellement méchant méchant .

raise some challenges concerning machine translation and translation in general. Especially polysemy and the absence of genre in the english language.

In fact, we can observe that the word "mean" is used for two different meanings and our model seems to have picked that up. However, we can see that it didn't identify the fact that "mean" in the second sentence has to be translated to "méchante" rather than "méchant".

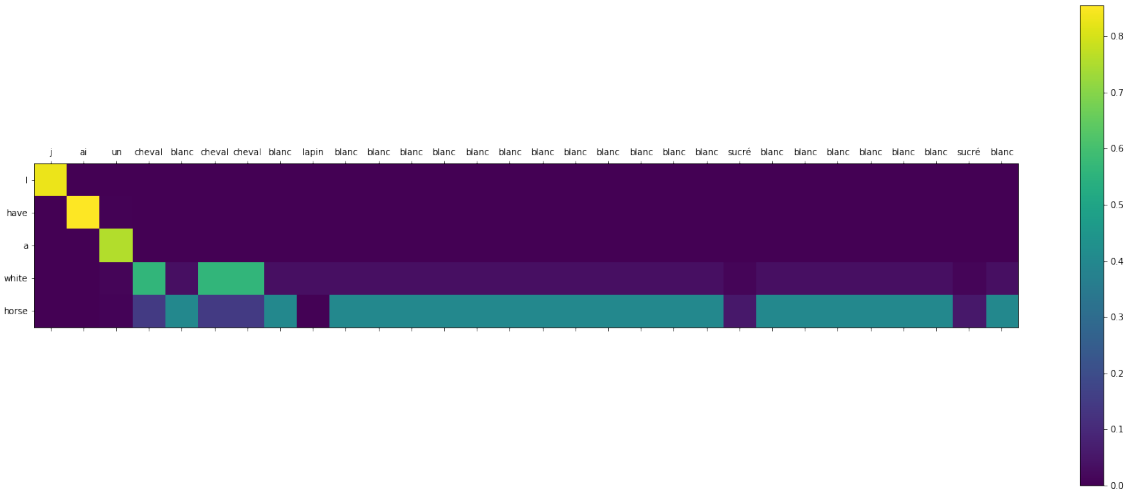

Figure 1: Source/target alignement

Figure 2: Source/target alignment

References

[1] https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html.

[2] Yang Liu Xiaohua Liu Zhaopeng Tu, Zhengdong Lu and Hang Li. Modeling coverage for neural machine translation. page arXiv preprint arXiv:1601.04811, 2016.