1 Question 1

What we are each time doing is that we take the most probable word that our model gives us, it's a greedy decoding strategy and it's easy to implement. But when we take at a time t the most probable word, it might impact the translation of the words after it. Maybe, if it took the second most probable word at time t, the next translated word would have been better. In order to avoid this, it's possible to use beam search : **Beam search** resembles breadth-first search of the tree of possible sequences, but with a limited bandwidth.

- At each time-step, the beam search only keeps track of the k best candidates and expands all the successors of these candidates in the next level.

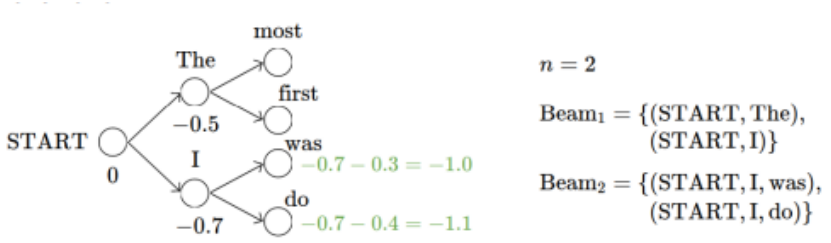

Figure 1: Example of beam Search taken from Mathieu Labeau Classes

- This new method is still greedy, we even augmented the search space but it provides us overall a better translation.

But instead of using a greedy search we can **sample words** : $\hat{w}_i \sim \mathbf{P}_\theta^{(i)}(w'|w_{<i})$. This can avoid the massive computation Beam Search requires. This tends to remove the repetition we can see, but might make some nonsensical outputs. To control this, one can use **Top-k Sampling** : the next token is randomly selected among the top k most likely candidates. Moreover, one can use **temperature** in order to control and scale the distribution of the Top-k sampling and have a more diverse distribution.
But, one can also compute beam search using Top-K sampling and there would have a really reliable output with even more diversity with Top-k sampling but still keeping the explorability of beam search.

2 Question 2

We can see that the model outputs the same token multiple times in a row. For small sentences, it will be a lot of dots at the end, but for longer sentences, it will be words inside the sentence. This phenomenon is called over-translation.
Some solutions are brought forward in the scientific literature. In [4], the idea of coverage is brought up, this would be a vector that would track which word as already been translated and if so how many times, and each word that hasn't been translated. By doing so, one can set a threshold to limit the number of times a word can be translated. Hence this would reduce over-translation and also force the translation of each source word. On the other hand, [1] propose the idea of Input-feeding Approach, in which instead of having a coverage vector, the attention model is adapted : attentional vectors h_t are concatenated with inputs at the next time steps, it tends to make the model more aware of previous alignment choices and the neural network becomes really deep. Moreover, they also change the way the attention model is structured, instead of taking all words of the source side for each target word, they define the local attention model in which only a small subset of source word is used for each target words. This tends to help when the translation is long.

3 Question 3

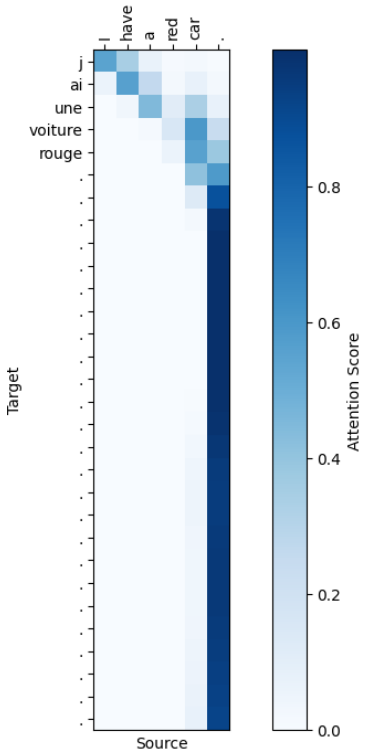

Figure 2: Attention weights

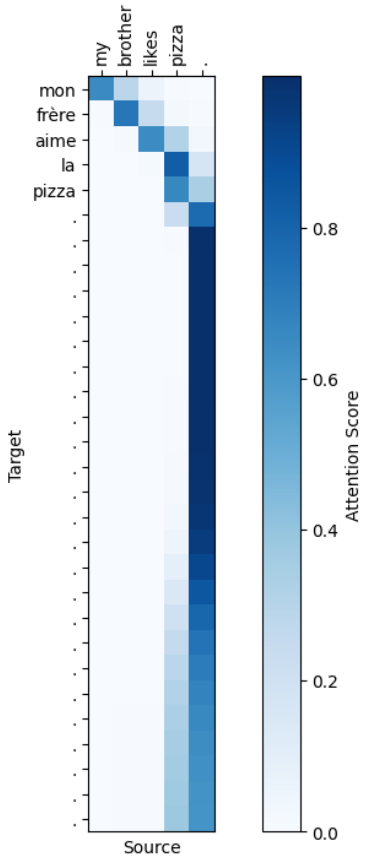

Figure 3: Attention weights

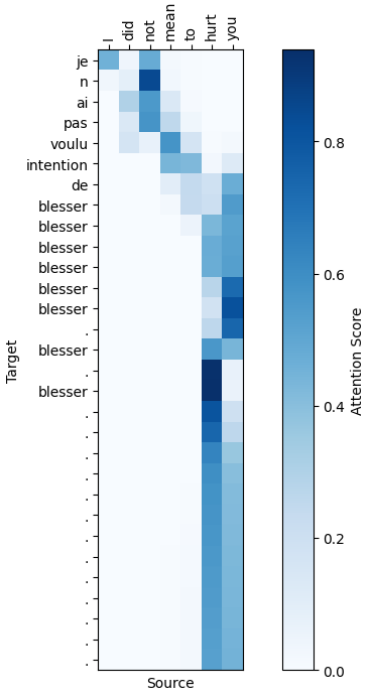

Figure 4: Attention weights

Figure 2 (2): We can see here that in the first translation we have an inversion : 'Red car' becomes 'voiture rouge'. This is an adjective noun inversion, and we can see how it's done with the weights, 'voiture' mainly take information from 'car' and a bit from 'red', and 'rouge' takes info from both in order to have the good gender.

Figure 3 (3): We can see in the second translation that in English there is no need for pronoun for Pizza, but in french we need to have a la before. As we can see by the weights, 'la' is added and the weights are mainly on 'pizza' in order to take the proper gender. It's the same for 'my' which is ungendered and 'mon' that is, we can see that the attention weight for brother which infers the gender is not zero.

Figure 4 (4): Lastly, we can see an important mechanism : there is the translation of mean. It's a polysemous word, hence traducing it can be tricky. It needs context in order to get the good translation. We see here that the attention weights for this traduction are spread between 'mean' and 'to' : having 'mean to' tells our model to not translate it as the mathematical mean.

4 Question 4

We see that in those two sentences, the word mean is well translated. This could have been an issue because mean is polysemous, hence the translation must be made in context in order to be right. It tells us that the models can translate a word by taking in account their context.
In a way to have an even better translation, [2] introduce a new architecture : a deep bidirectional language model (BiLM), by doing so it improves a lot on the disambiguation of polysemous words by having left and right context. The vectors representing the words are derived from a bidirectional LSTM trained on multiple language model. Those representation are deep : they are a linear combination of the vectors stacked above

each input word for each end task. They show how better it is by showing what are the closest word to play using a Glove embedding : only sports vocabulary. But when they use the biLM architectures context embeddings in order to find the closest sentence, it can decide whether it's a sports use or maybe a theatrical one.
An other approach to word context is brought forward in [3] as they have a totaly new bidirectional architecture modeled around Transformers. Those rely around self attention that is close to what we have as attention mechanism in our encoder-decoder model. And this self attention is especially made to have context based task.
Hence we see that all of the new methods that are presented in those article emphasize a lot on the **left and right context** of word in sentence in order to be performing in their tasks.

References

[1] Minh-Thang Luong Hieu Pham Christopher D. Manning. Effective approaches to attention-based neural machine translation. 2015.

[2] Mohit Iyyer Matt Gardner Christopher Clark Kenton Lee Luke Zettlemoyer Matthew E. Peters, Mark Neumann. Deep contextualized word representations. 2018.

[3] Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2019.

[4] Yang Liu Xiaohua Liu† Hang Li Zhaopeng Tu, Zhengdong Lu. Modeling coverage for neural machine translation. 2016.