1 Question 1

In this lab we used the **Greedy search** decoding strategy for translations, which is a strategy that selects the most probable word(i.e. argmax) from the model's vocabulary at each decoding time-step t as the candidate to output sequence as we can see in the Figure 1 as example. The index of this word with the highest probability after the GRU decoding layer at time step t is used as input for the decoding layer at time t+1, until the EOS token:

$$\tilde{x_t} = \arg\min_x \log p(x|x_{<t}, Y)$$

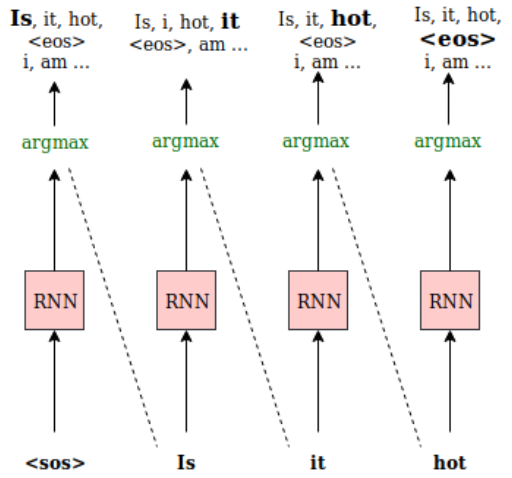


Figure 1: Decoder Segment Example.

This strategy is super efficient in term of both computation and memory, but it does a heavily suboptimal search and it achieves a translation without taking into consideration the quality of the overall sentence.
The problem with this approach is that once the output is chosen at any time-step t, we don't get the flexibility to go back and change our choice. It is seen in practice that greedy decoding strategy is prone to have grammatical errors in the generated text as we can see in the following examples generated by the model:

- I love playing video games. → j adore jouer à jeux jeux jeux vidéo
- I did not mean to hurt you → je n ai pas voulu intention de blesser blesser blesser blesser blesser blesser

It will result in choosing the best at any time-step t but that might not necessarily give the best when considering the full sentence to be grammatically correct and sensible.

To overcome greedy search limitations, we can use the **Beam Search strategy** mentioned in the this presentation in the slides 87-95, which tries to find an output sequence having a maximum likelihood. It does this by extending the greedy sampling to Top-k sampling strategy. At any time-step t, it considers top-k most probable words as the candidate words for that step. Here, k is called the beam size.
Here, we search for a high scoring output sequence by keeping track of top k vocabulary outputs at each time-step while decoding. We usually stop our search when end-of-sentence token (¡eos¿) is reached or till time-step t for all or at-least some n output sequences.
This technique is computationally expensive and not easy to parallelize, but gives much better quality according to the ALC tutorial.

Another technique that could be employed is the **ancestral sampling** also mentioned in this presentation and in the ALC tutorial, which samples a word from the probability distribution given by the softmax layer used to predict the decoder's current output. This technique can generate more natural sequences.

2 Question 2

The following examples are English sentences translated to french using our model after training.

- I did not mean to hurt you → je n ai pas voulu intention de blesser blesser blesser blesser blesser blesser . blesser . blesser . . . . . .
- She is so mean → elle est tellement méchant méchant . ¡EOS¿
- Help me pick out a tie to go with this suit! → aidez moi à chercher une cravate pour aller avec ceci ! ! ! ! ! ! ! ! ! ! ! ! ! ¡EOS¿
- I can't help but smoking weed → je ne peux pas empêcher de de fumer fumer fumer fumer fumer fumer fumer fumer fumer urgence urgence urgence urgence urgence urgence . urgence urgence . urgence urgence .
- The kids were playing hide and seek → les enfants jouent cache cache cache cache caché caché caché caché caché caché caché caché caché caché caché caché caché caché dentifrice perdre caché risques rapide caché risques éveillés
- The cat fell asleep in front of the fireplace → le chat s est en du du pression peigne peigne cheminée portail portail portail portail portail portail portail portail indépendant oiseaux oiseaux oiseaux oiseaux oiseaux oiseaux oiseaux oiseaux oiseaux oiseaux

This translated sentences are not of great quality and presents a major problem which is repetition, especially end of sentence words. We can notice that the sentences with the most repeated words at the end are the ones that do not have a '.' full stop token or a punctuation mark at the end, and if it's the case the network repeats the punctuation mark until the end of the sequence.

To solve this repetition issue, one of the solutions would be to add a system to keep track of the translated words and untranslated words in order to avoid the translation repetition of the same word, as mentioned in article [4] that proposes an approach C to maintain and utilize a coverage vector to indicate whether each source word is translated or not, to encourage NMT to pay less attention to translated words and more attention to untranslated words. This can help us avoid the over-translation and under-translation issue faced by our model.
For instance for a set $x = \{x_1, x_2, x_3, x_4\}$ as input sentence. The initial coverage set is {C={0,0,0,0}} which denotes that no source word is yet translated; When a translation rule $bp = (x_2x_3, y_my_{m+1})$ is applied, the produced coverage is $C = \{0, 1, 1, 0\}$. It means that the second and third source words are translated.

Another way is to use the "local attention" technique mentioned in article [2] instead of "global attention" that has several limitations. It requires to learn attention weights and the context vector not on the entire set of previous encoder outputs, but on a window of size D and select a position pt to pay attention to words in the windows $[p_tD, p_t + D]$ only. Since French and English languages are different in structure we can't really assume that a position t of a word in the input is the same as the in the output. We might thus use "predictive alignment" where we have:

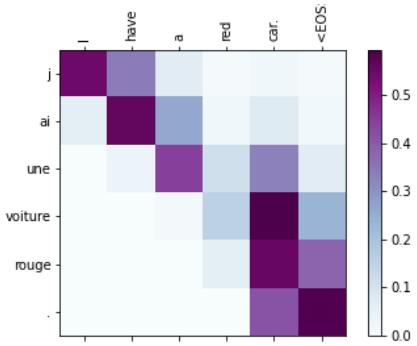
$$p_t = T_x * sigmoid(v_p^T * \tanh(W_p h_t)$$

where T_x is the size of the input sequence, and the value in the sigmoid is the score we calculated for the alignment vector $\alpha_{t,i}$.

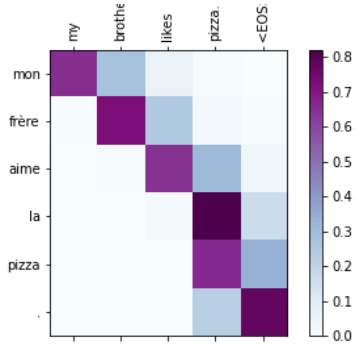
3 Question 3

In order to add the generation of the alignments visualizations, we modified several parts in the existing code. First, in the 'seq2seqAtt' class we returned the norm scores, which represents the attention scores. Then, in the seq2seqModel we changed the forward function by initializing the overall weights, assigning the attention weights to it with some modifications and returning it at the end as a tensor of the overall attention scores. We then implemented a $visualizeattention(inputsentence, outputsentence, attentionweights)$ method in the seq2seqModel. All attention weights are cut to their limits in input ('EOS' token) and output sentences ('.' token).
The figures 2a, 2b and 2c shows the results of the visualization of some examples:

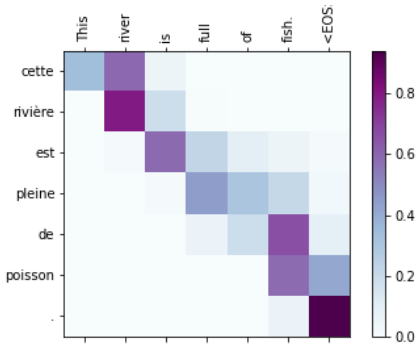
Relying for instance of the first example in figure 2a, where our model assigned more attention to "voiture" than "rouge", meaning that the model can distinguish between the noun car and the adjective that describe it red. Therefore, we can say that our model successfully implemented the adjective-noun inversion.



(a) 'I have a red car'



(b) 'my brother likes pizza'



(c) 'This river is full of fish'

Figure 2: Three simple Examples

We can see for example illustrated in figure 2b that the attention weights for "la" and "pizza" were very high, meaning the network understood the semantic relation between the pronoun and the noun.
It's the same phenomenon for example 3 in figure 2c, where there is more attention for the pronoun referring to "fish" rather than to "poisson" itself.

4 Question 4

In the following the visualizations and translations of attentions for both of the sentences in both cases.

Case1: Without the '.' token

- I did not mean to hurt you → je n ai pas voulu intention de blesser blesser blesser blesser blesser blesser . blesser . blesser . . . . . .
- She is so mean → elle est tellement méchant méchant . ¡EOS¿

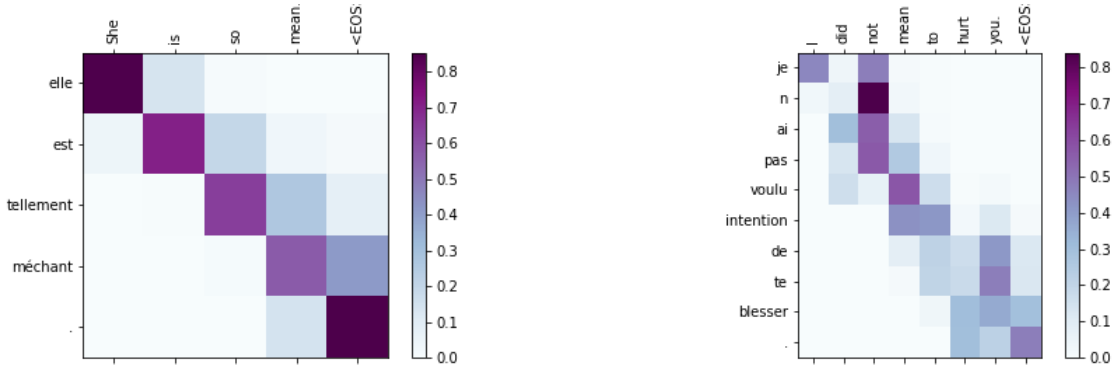
Case2: With the '.' toke

- I did not mean to hurt you. → je n ai pas voulu intention de te blesser . . . . . .
- She is so mean. → elle est tellement méchant . ¡EOS¿

As we can see in the two figures the model translated the word "mean" differently according if it's a verb or an adjective, even that there is a difference between the English and the French syntax. Therefore, we can say that our model succeeded to distinguish between adjective/verb alignment of "mean" according to the context.

However, for just this two simple examples the sentence was not of high quality espiecilly the second example "I did not mean to hurt you" that became "je n ai pas voulu intention de te blesser", which shows that for more complex semantics and contexts of sentences, our model might have limitations to identify the difference between adjective/verbs.

To solve this issue, one option to consider would be to use an architecture based on [1], with transformers, that would train to obtain "deep bidirectional representations from unlabeled text by jointly conditioning on



(a) 'She is so mean'

(b) 'I did not mean to hurt you'

Figure 3: Examples

both left and right context in all layers".
Another option would be to keep the sequential-like architecture (RNN, LSTM or GRU) but have a bidirectional language model [3] so that we have both:

$$p(t_1, t_2, ..., t_N) = \prod_{k=1}^{N} p(t_k | t_1, t_2, ..., t_{k-1}$$

$$p(t_1, t_2, ..., t_N) = \prod_{k=1}^{N} p(t_k | t_{k+1}, t_{k+2}, ..., t_N)$$

This way, the network would be able to learn and infer from both directions and it would greatly help if the semantics and the context of certain entities are more difficult to dissect.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

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

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

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