1 Question 1 : What do you think about our greedy decoding strategy? Base your answer on slides 87-95 from this presentation (taken from this ACL tutorial).

According to the ACL tutorial, Greedy Search is a decoding strategy that is computationally efficient but not provide a good quality. The goal of this strategy is to choose among the most probable word that has been implemented in the model's vocabulary. This selection is reiterated at each step of the decoder with respect to the previous word of the same sentence. However, in practice, we find that the greedy decoding strategy is likely to contain grammatical errors in the generated text as it is not possible to go back and modify chosen word given as output at any time step t. If this strategy outputs the best result at each step t, it however does not necessarily give the best result with respect to the whole sentence. The main critic of this method refers to the fact that the output is conditionned with respect to the previous word only. This fact is also way it allows to be computationally efficient, but in cases where we want to improve the quality of translation, other solutions are needed to be implemented.
An other type of decoding strategy presented in this presentation, i.e the beam decoding strategy can be a good alternative to deal with this limitation. At each step, the beam strategy select the K words having the maximum likelihood of being well aligned with the target word, as opposed to a unique one as in Greed strategy. We note that when K=1, is it the same as performing Greedy search. If beam search is present much better quality results i.e providing a more coherent translation (especially in longer sentences), it is as contrary computationally expensive when K (the beam size) is large, and not easy to parallelize.

2 Question 2 : What major problem do you observe with our translations? How could we remediate this issue?

To make a bridge with question 1, we have seen that a recurrent problem is obtain a good trade-off between the quality of the translation, requiring to obtain a meaning of the word in the sentence (good representations), and the computational efficiency. Indeed, with translation problem, the vocabulary furnishes to the model have to be quite big to obtain good results, so do the length of training steps.
We recall that Neural Machine Translation work by learning to align source sentences with target sentences to generate an appropriate translation. However, they tends to ignore past alignment information, leading to a sub-optimal translation as the global context is not conditioning the output. Note that sub-optimal translation can either refer to under-translation - when words are erroneously translated - , or to over-translation - when words are translated multiple times - . A cause of over-translation can be the lack of information about which word has been translated, exposing source words to undergo multiple unnecessary translations. To remediate this problem, a solution such as proposed by Zhaopeng et al., (2016) [2] is to keep track of the translation history (attention history). This can be implemented by a so-called "coverage vector" that will be fed back to the attention model to tag source words that have not been translated yet. By being informed of which words has already been translated, the model can directly work on new source words.

3 Question 3 : Write some code to visualize source/target alignments in the style of Fig. 3 in [1] or Fig. 7 in [4] . Provide and interpret your figures for some relevant examples (e.g. to illustrate adjective-noun inversion).

Here we plotted the alignment visualizations, showing images of the attention weights learned by our model given an input vector of words (source). To visualize source/target alignments in the style of Fig. 3, the abscisses of the plot correspond to the words in the source sentence (original version - English) and the ordinates to correspond to the words in the target (generated version - French) translated sentence. Each pixel shows the weight ij of the annotation of the j-th source word for the i-th target word.

4 Question 4 : What do you observe in the translations of the sentences below? What properties of language models does that illustrate? Read [3, 5] to get some ideas.

- ”I did not mean to hurt you”
- ”She is so mean”

First, let’s note that in both sentences , ”mean” is used but is not conveying the same meaning. If the first phrase, it is used as a verb whereas it is used as an adjective in the second one. The importance of this word is not the same in both phrase, and if we look for a synonym for each phrase, it would for sure not be the same depending of the phrases. Consequently, a good translation should be able to translate a same word differently according to the context. For example, this is particularly important when dealing with sentiment analysis challenges, as illustrated by the two sentences above, or in cases of polysemous words.
This characteristic of a good model refers to the need of the good understanding of the semantic meaning of the word in a given sentence, i.e the contextualisation of word representations.
To provide a translation with respect to the context and therefore improve the understanding of the word in a specific sentence, both past words and future words have to be considered when generating the target sequence. As we saw in class, this can be achieve by implementing a bidirectional language model pre-trained on large text data set [1].

References

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

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