Benjamin Lapostolle
benjamin.jean.lapostolle@gmail.com

1 Question 1

Our greedy approach, which involves selecting the word most likely to appear in our translated sentence based on the source sentence and previously predicted words, has the advantage of being computationally lightweight. Opting for just one candidate at each step may be optimal at the current point in the sequence. However, as we progress through the rest of the full sentence, it could turn out to be less ideal than initially thought, especially since we couldn't anticipate the words predicted later on. Another approach, known as Beam Search, is to retain, at each step of the algorithm, the N most probable sequences. This method eliminates predicted sentences where one word seemed highly likely considering the previous words, but subsequent predicted words were improbable. Admittedly, this method requires more computational resources, but has the potential to yield superior results.

2 Question 2

Our algorithm seems to predict well short sentences, however when the number of words starts to increase, at around 4 or 5 words, the quality of the prediction decreases. In fact, our algorithm tends to repeat multiple times a word, which is often the translation of a word located in the end of our source sentence. This can be explained by the fact that in such attention mechanism algorithm tends to ignore past alignment information which often leads to over-translation and under-translation

We can find possible solutions to overcome this issue. In the original paper of this code[2], the author talk about the Input-feeding approach. In this approach, the information of the context vector from the previous time step is also incorporated into the computation of the decoder's hidden state. This means that the context vector at time step t-1 is fed as additional input to the computation of the hidden state at time step t. This modification improves the model's ability to consider previously attended source sentence information when generating translations, leading to better translation quality because it now focus its attention on words less used in pervious predictions, reducing repetition and the omission of words.

Another solution is proposed in the paper title 'Modeling Coverage for Neural Machine Translation'[4]. The authors introduce the concept of a "coverage vector" as a technique to prevent the model from over-translating or under-translating. The coverage vector keeps track of which source words have been attended to during the decoding process. It is updated at each step, increase if a source word has been used to predict a new target words. Hence it encourages the model to distribute its attention across different parts of the source sentence.

3 Question 3

These graphics 1 demonstrate that the algorithm appears to have learned translation mechanisms that a human would naturally use. For instance, in English, the sole information from the article preceding a noun is insufficient because it does not indicate whether to use the feminine or masculine gender in French. We need to focus our attention on the words around the article. In our graphics, It is consistently observed that for the words 'a' and 'the,' attention is focused on the source word from the article, but also on the associated noun. Thus, 'the mat' is translated to 'le tapis,' 'a red car' is translated to 'une voiture rouge,' and not 'un voiture rouge,' as attention is also directed towards 'car.' However, for other words like 'chat,' we can see that attention is mainly localized on the source word 'cat' since, for this word, the context will often not affect its translation. Furthermore, in the example of the inversion of noun and adjective, the preceding word 'une' and the attention vector focused on the source word 'car' enable the algorithm to effectively reverse the adjective and noun during translation.

4 Question 4

The word "mean" has two different meanings, and in each translation it's the right meaning. This clearly demonstrates that the attention mechanism, which takes into account the words before and after, is relevant.

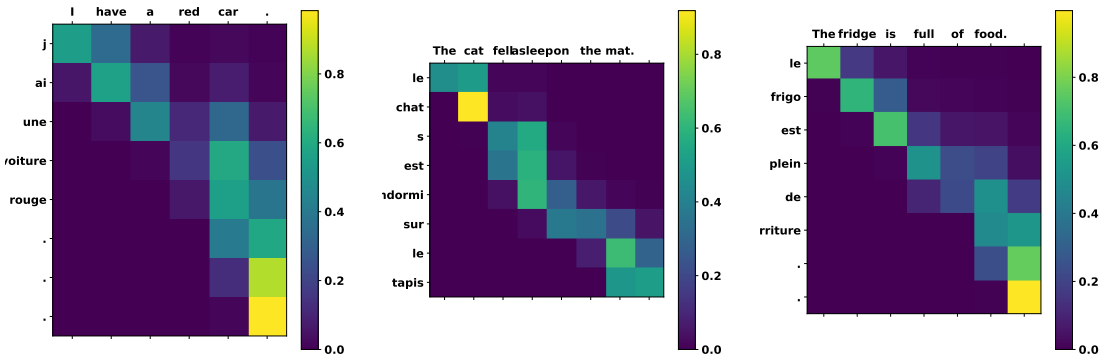

Figure 1: Attention Map of Translated Sentences

In the bidirectional paper[1], the authors' idea is to process the sentence in both directions to have the context before and after the word. During one phase of their training, they use sentences in which some words are replaced by mask tokens, and their algorithm uses the context on the left and on the right to make predictions. Similarly, in reference[3], the authors combine two architectures that predict from left to right and from right to left to improve predictions.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[2] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[3] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

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