Megi Dervishi
megi.dervishi@ens.fr

Question 1

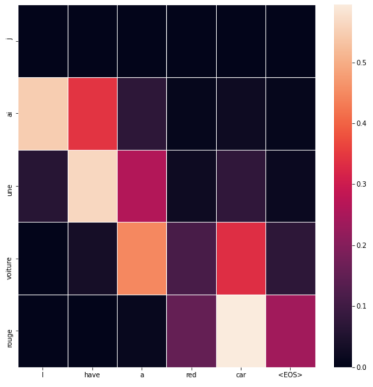
When translating a sentence greedy decoding will guess one word at a time from left to right, guessing at each step the 'most probable' word given the ones that occurred before it. This is a very simple method to implement, however it is heavily sub-optimal. At every step, the model will simply guess a word without thinking of how it could impact the general meaning of the full sentence, which leads to sub-optimal sentences. Furthermore, the model will be biased by the frequency of occurrences of words in our data-sets. However an exhaustive search of the best sentence is intractable. Indeed in our data-set the average sentence length is 8.3 and the vocab size is 7456 for the target set which means that an average sentence has approximately 1.39×10^{32} possibilities.
One alternative is to use beam search. Beam search will simultaneously explore K different hypothesis, expand them and keep the best K, and iteratively repeat this process. In the asymptotic limit $K \to \infty$ this yields the exact solution. However practically the increase in performance coming from the choice of K is not linear and the higher K is the more costly the beam search becomes. It is also harder to implement, hence for a simple application such as this one the improvement might not be worth the effort.

Question 2

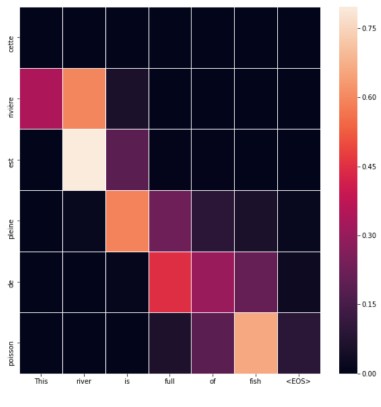
Our model is relatively simple and can usually understand sentences with a simple grammatical structures such as 'I am a student'. However as soon as the sentence becomes more complex, for example introducing a negation 'I did not mean to hurt you' or building a sentence with two clauses 'The cat fell asleep in front of the fireplace' the model gets confused and starts repeating words or degenerates into meaningless gibberish. This is due to the fact that our model has no awareness of the attention history.
Papers [4] and [6] propose two different methods to remedy this problem and make the model more contextually aware. Paper [6] introduces a coverage vector which is sequentially updated during the decoding process and hence keeps track of the attention history. On the other hand, paper [4] introduce 'input-feeding' meaning that they concatenate the previous attentional vector with the input in order to give the model awareness of the attention history at the next step.

Question 3

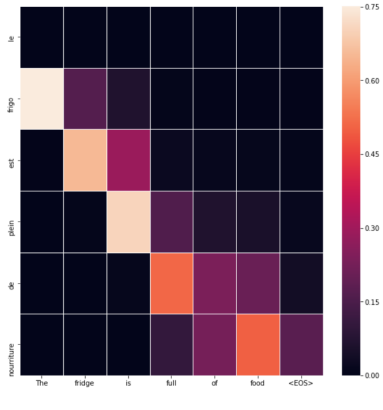
Three different example of the attention vectors are displayed in Figure 1. In the 1a we see that the model correctly detects the adjective-noun inversion 'red car' $\to$ 'voiture rouge'. Indeed we see that 'rouge' is mostly related to 'red' and 'car'. Secondly 'voiture' is mostly related to the verb 'a', 'car' and 'red'. In the other two plots we can also see the noun-adjective, verb-subject relations.



(a) 'I have a red car.' which was translated to 'J'ai une voiture rouge.'



(b) 'This river is full of fish.' which was translated to 'Cette rivière est pleine de poisson.'



(c) 'The fridge is full of food.' which was translated to 'Le frigo est plein de nourriture.'

Figure 1: Attention heatmaps for three different translation pairs.

Question 4

The translations given by the model of the two following sentences are as follows

- '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 .'

However notice that if we add a fullstop then the model correctly understands the sentence. The translations are then given as follows

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

The fact that the model now understands that the sentence is finished shows how an open-ended proposition can lead the model to confusion since it can have multiple possible interpretations. The translation however still is not completely correct especially for the first sentence. Indeed the model fails to understand the meaning of 'mean' in this given context. Papers [3] and [5] adress this problem of polysemy i.e. how a word can have different uses according to its context. Paper [5] introduces 'deep contextualized representations of words' which allow to model both the word and its use in a given context. This is achieved by using the whole input sequence to embed each word. On the other hand, paper [3] introduces another attention mechanism. Multi-headed attention allows the model to leverage information from the whole sentence and hence can help the model to understand better the use of a word in its context.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[2] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

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

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

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

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

[1, 2, 3, 4, 5, 6]