1 What do you think about our greedy decoding strategy?

In sequence-to-sequence models, the decoder constructs the output sequence incrementally, generating one token at each step. In our implementation, we employ the greedy decoding strategy. Specifically, at the end of each decoding step, the decoder chooses the next token in the output sequence based on the highest probability among the possible tokens.

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

This approach is straightforward, widely adopted, and efficient. The efficiency arises from the method's computational simplicity and speed, as the decoding process occurs only once for each input sequence. Furthermore, in terms of memory usage, the strategy retains only the token with the highest probability instead of maintaining an entire set of tokens and their respective probabilities.
However, a drawback of the greedy decoding algorithm is its potential suboptimality in terms of output quality. This is because it makes locally optimal choices at each step, hoping that these decisions will collectively lead to a globally optimal solution. Unfortunately, this assumption does not always hold, and the locally optimal choices may result in a subpar global solution. Another limitation is the absence of backtracking in the algorithm. Once a decision is made at a particular step, it cannot be undone in subsequent steps. Additionally, the algorithm does not consider the long-term impact of the current decision on future steps.

There some other decoding strategies such as the Beam decoding strategy which introduces a more sophisticated strategy. Instead of solely selecting the token with the highest probability, beam search explores multiple potential sequences in parallel. At the end of each decoding step, the model evaluates and scores all candidate sequences based on their probabilities. The top-k sequences with the highest cumulative scores are then selected to proceed to the next decoding step. The beam width represents the number of candidate sequences that the decoder considers at each decoding step. So if the beam width is equal to 1, the decoding strategy essentially becomes equivalent to greedy decoding. The beam decoding strategy is computationally expensive, not easy to parallelize but is of much better quality. Also we have the exhaustive search, the ancestral search that are generally not optimal or better than the greedy decoding search

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

The problems encountered in these translations are essentially:

- **Repetition of words:** In several translations, words are unnecessarily repeated in multiple times, such as "jeux" in "j adore jouer à jeux jeux jeux vidéo", "blesser" in "je n ai pas voulu intention de blesser blesser blesser blesser blesser blesser", and "fumer" in "je ne peux pas empêcher de de fumer fumer fumer fumer fumer fumer fumer fumer fumer".
- **Incorrect translations:** Some sentences are not translated correctly. For example, "The cat fell asleep in front of the fireplace" is translated as "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", which is not a correct translation.
- **Hallucinations:** Certain translations include irrelevant or unrelated terms. For instance, in the translation of "The kids were playing hide and seek," words like "dentifrice" (toothpaste) and "risques" (risks) unexpectedly appear. This phenomenon, known as hallucination, is a common challenge in machine translation where the model generates words or phrases that are not present in the source sentence

The problems of the repetition of words or untranslated words are referred to as over-translation and under-translation in the litterature. To address the issues of over-translation and under-translation in Neural Machine Translation (NMT), Tu. et al. [2] propose the implementation of a coverage mechanism. This mechanism is designed to keep track of which source words have been translated or "covered" during the decoding process. The coverage mechanism works by appending a coverage vector to the intermediate representations of an NMT model. This vector is initialized as a zero vector but is updated after every attentive read during the decoding

process. The purpose of this vector is to keep track of the attention history. When this coverage vector enters into the attention model, it can help adjust future attention. This adjustment allows the NMT system to focus more on untranslated source words, thereby alleviating issues of over-translation and under-translation.
Also the quality of our translations can be improved by improving our attention mechanism implementation.

3 Write some code to visualize source/target alignments

The source code can be found in the notebook. The alignment vector is represented by the normed scores as in the implementation in [1]. In the translation of the sentence 'I have a red car', we observe an interesting

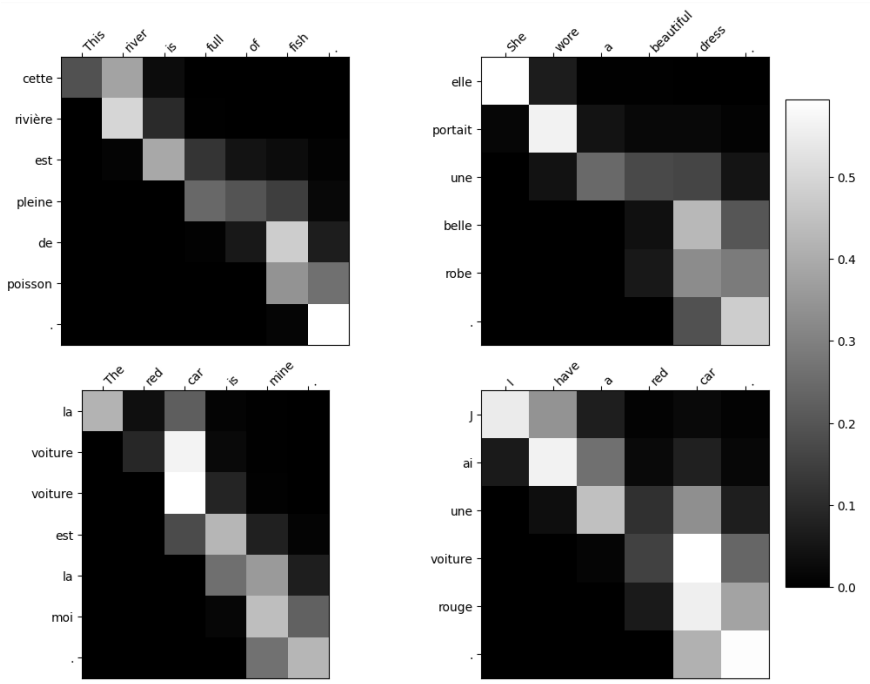

Figure 1: Source/target alignment plot for 4 different examples. The repeated punctuations at the end of the sentences have been removed from the plots.

pattern in the alignment plot. The word 'car' in English aligns with 'voiture' in French before 'red' aligns with 'rouge'. This suggests that the model is correctly learning the adjective-noun inversion rule in French, a key grammatical difference between English and French.
However, an unexpected alignment is observed where 'red' in English aligns more with 'car' rather than 'rouge'. This could indicate that while the model is learning some aspects of the grammatical structures, it may not be perfectly capturing all nuances, such as the alignment between adjectives and nouns across languages.

4 What do you observe in the translations of the sentences below? What properties of language models does that illustrate?

• I did not mean to hurt you • She is so mean
In both translations, the word 'mean' is employed in varying contexts, demonstrating its multiple interpretations. This shows the concept of polysemy, where a single word can possess several meanings. It appears that the model has accurately translated these words in accordance with their respective contexts

References

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

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