1 Question 1

The greedy decoding method in NMT involves feeding the model an input sentence and then selecting the word with the **highest probability** from its output distribution for each position in the target translation. This continues until an end-of-sequence token (EOS) appears or a set maximum length is attained.

$$\tilde{x} = \arg\max_{x} \log p(x|x < t, Y) \tag{1}$$

This strategy is illustrated in Fig 1.

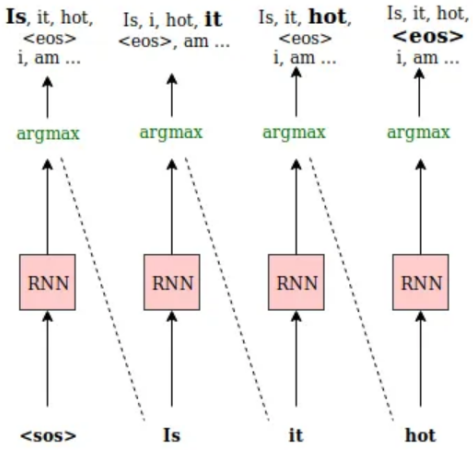

Figure 1: Decoder part of a NTM, greedy strategy. Adapted from [7].

There are both advantages and disadvantages for this greedy decoding strategy [9]. The advantages are that it is **fast** as it is computationally efficient as it only involves choosing the best word at each time step without considering all possible word sequences. In addition, it is **simple** as the implementation is straightforward and does not require any special algorithms or data structures.

The disadvantages are **suboptimal results**, it might not yield the best translation as it selects the word with the highest probability at each step, the model might produce a sequence of words that is not the most coherent or accurate translation. In addition, there is a **lack of exploration**, that means that greedy decoding does not explore different possible sequences of words. This lack of exploration can lead to translations that miss nuances or are less fluent [10].

Some alternatives might be found. One of them is **Beam Search** decoding. It starts with an Empty Beam. For each sequence in the beam, the model predicts possible next words.The sequences are ranked by their probabilities, and only the top sequences are kept. At the end of the process, the sequence in the beam with the highest probability is chosen as the translation. It is illustrated in Fig 2.

In summary, beam search is a balance between the exhaustive search and greedy decoding [4] (which only considers the most probable next word, but might miss better overall translations). The size of the beam determines this balance: a larger beam size results in better translations but requires more computation.

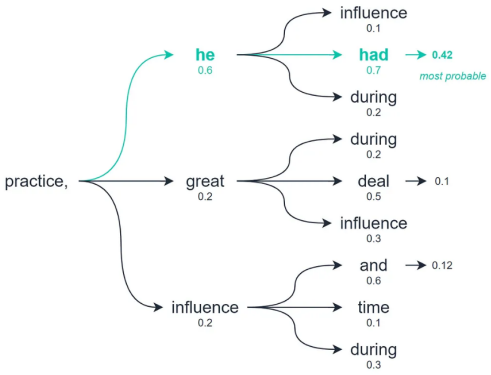

Figure 2: Decoder part of a NTM, beam strategy. Adapted from [1].

2 Question 2

The translations that are achieved with the translations are the one presented in Fig 3.

```
max source index 5281
source vocab size 5278
max target index 7459
target vocab size 7456
= = = = =
 I am a student. -> je suis étudiant . . . . . .
= = = = =
 I have a red car. -> j ai une voiture rouge . . . . . .
= = = = =
 I love playing video games. -> j adore jouer à jeux jeux jeux vidéo . . . . . .
= = = = =
 This river is full of fish. -> cette rivière est pleine de poisson . . . . . .
= = = = =
 The fridge is full of food. -> le frigo est plein de nourriture . . . . . .
= = = = =
 The cat fell asleep on the mat. -> le chat s est endormi sur le tapis . . . . . .
= = = = =
 my brother likes pizza. -> mon frère aime la pizza . . . . . .
= = = = =
 I did not mean to hurt you -> je n ai pas voulu intention de 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 fumer urgence urgence urgence urgence urgence urgence . urgenc
= = = = =
 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 caché caché dentifr
= = = = =
 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
```

Figure 3: Translation after our NMT.

The translations often repeat words or phrases multiple times. For example, the translation for ”I love playing video games” is ”j adore jouer à jeux jeux jeux vidéo”, where ”jeux” is repeated three times. This problem is even more pronounced in some translations, where unrelated words appear repeatedly, as in ”The kids were playing hide and seek” translated to ”les enfants jouent cache cache cache cache [...] dentifrice perdre caché risques rapide caché risques éveillés”.

This could be due to the decoder in the NMT system getting stuck in a **loop**, emitting the same word or phrase repeatedly.

One way to solve this is maybe to enhance the attention mechanism [6]. Indeed, attention allows the model to focus on different part of the input sentence when producing each word in the output, potentially reducing the likelihood of repeated words.To get rid of this over-translation, one were looking at maintaining the attention history via coverage.

One important thing that is discussed in [12] is to incorporate attention mechanism with coverage. Coverage in NMT keeps track of which source words have been attended in the past. This can prevent the model from attending to the same source words repeatedly, avoiding repeated translations. The paper proposes a coverage vector, which is accumulated attention weights over all previous decoder time steps. This process is illustrated in Fig 4. A sum-up of different techniques can be viewed in 1.

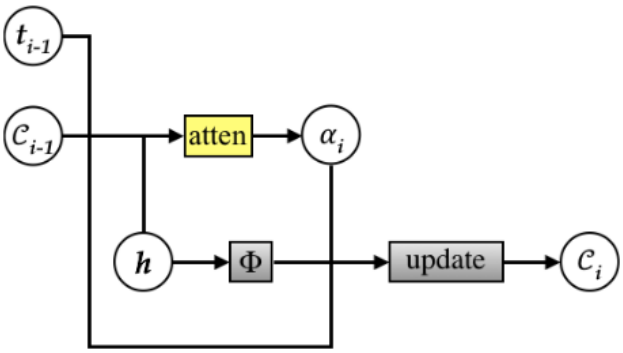

Figure 4: Architecture of coverage-based attention model. C_{i-1} keeps track of which words where translated before time i. Adapted from [12].

Model	**Description**	**Key Benefit/Functionality**
NMT (Basic)	Encoder-decoder architecture compressing source sentence into a fixed-size context vector.	Handles translation without dynamic source context consideration.
NMT with Attention	Encoder-decoder with attention mechanism that computes attention distribution over source for each target word.	Dynamically focuses on relevant source parts during translation, enhancing quality.
NMT with Coverage Attention	Builds on NMT with attention by tracking attention history using a ”coverage” vector.	Ensures balanced attention distribution, reducing over-translation and under-translation.

Table 1: Comparison of NMT Models

3 Question 3

The source / target alignments described in [2] or [6] provides a visualization of how much attention the model pays to each word in the source sentence when producing each word in the target sentence. It is represented in Fig 5.

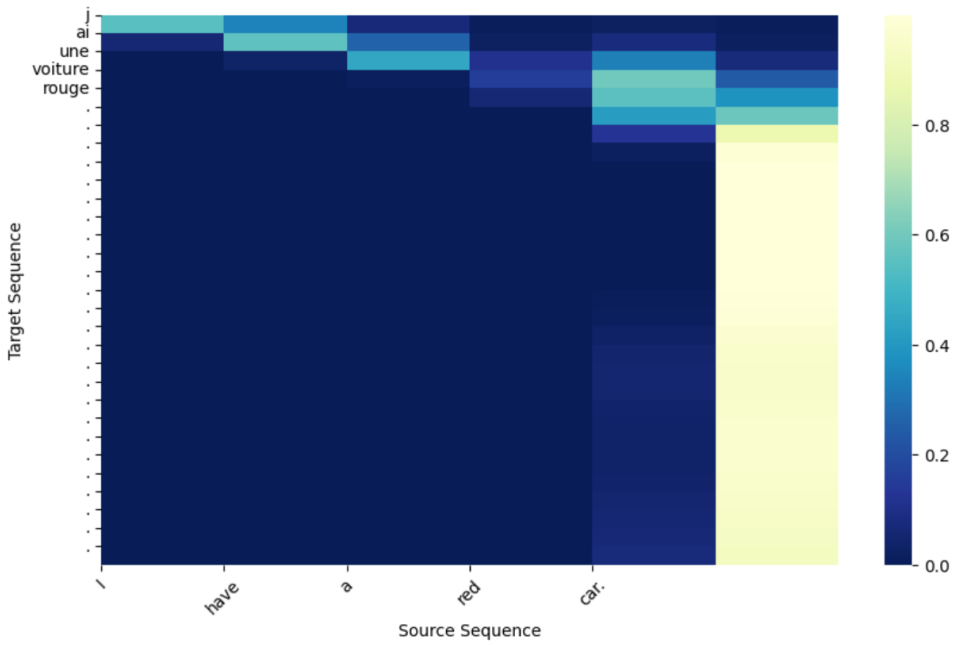

Figure 5: Attention Matrix of the sentence ”J’ai une voiture rouge”

.

From the heatmap, the word ”j” in the target sequence is most closely aligned with ”I” in the source sequence, as seen from the darkest shade in the respective cell. This indicates that ”j” is translated from ”I”. The interesting alignment is between ”red” in English and ”voiture rouge” in French. The model seems to spread attention between ”red” and both ”voiture” and ”rouge”. This is indicative of the adjective-noun inversion typical of French. In English, we say ”red car” (adjective-noun), while in French, it’s ”voiture rouge” (noun-adjective).

This heatmap provides a visual representation of how each word in the English sentence influences the translation of words in the French sentence. It’s evident that the NMT model with attention successfully captures the linguistic structures and nuances, such as the adjective-noun inversion, between the two languages. The same trend is also observed on other attention heatmaps where the highest value is on the word to be translated such as in Fig 6.

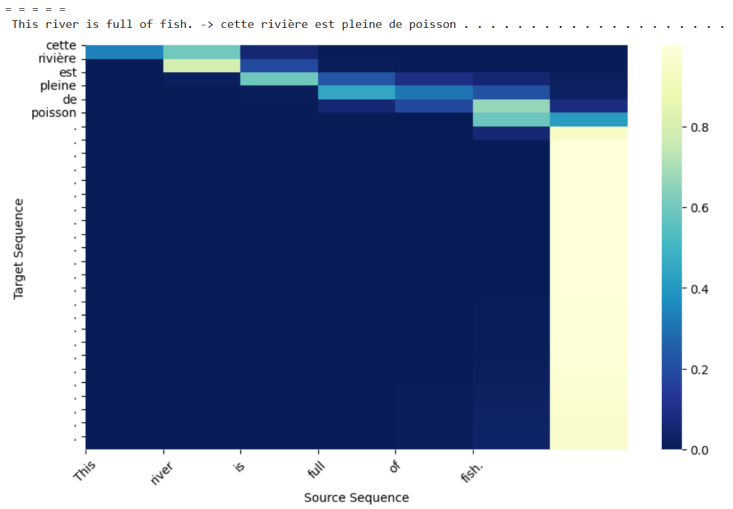

Figure 6: Attention Matrix of the sentence ”Cette rivière est pleine de poisson”

.

4 Question 4

Both translations exhibit repeated words or sequences. For instance, the word ”blesser” is repeatedly used in the translation for ”I did not mean to hurt you”, and ”méchant” is repeated in the translation for ”She is so mean”.

Traditional NMT models, during training, have their previous correct tokens as input. But during inference, they feed their own predictions back. This discrepancy can lead to the model producing sequences it hasn’t seen before, like repeated words. This phenomenon is known as **”exposure bias”**.

Traditional seq2seq models sometimes have difficulty in grasping the broader context, which can lead to repetitive outputs. [5] emphasizes the importance of deep contextualized word representations, highlighting that understanding the context can lead to better translations.

From [3], BERT uses a **deep bidirectional Transformer**, which allows it to consider the context from both the left and the right side of a token. Incorporating such bidirectional context could help in understanding the sentence as a whole, potentially reducing issues like repeated words. Thus process is illustrated in Fig 7.

The [5] paper introduces ELMo representations, which capture syntactic and semantic information at various levels of granularity. By integrating such deep contextualized representations, the translation model can better understand nuances and produce more accurate translations without unnecessary repetitions.

In conclusion, the observed translation issues underline the challenges faced by traditional NMT models in handling broader contexts and avoiding repetitive patterns. Incorporating insights from the provided papers can potentially lead to improved translation quality.

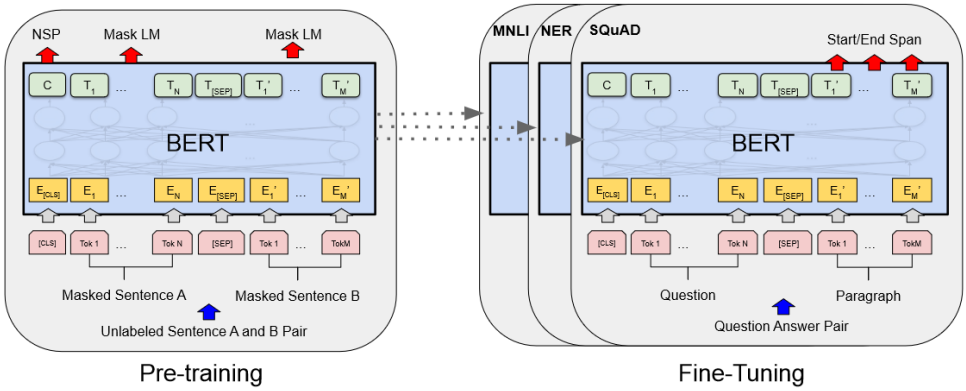

Figure 7: Architecture of BERT model. Adapted from [3].

5 Question 5 - Teacher Forcing

Teacher forcing is a training method. When training such models to predict sequences, the typical approach is to use the model's own previous predictions as input for subsequent time steps [8]. It is presented in Fig 8. Without Teacher Forcing the model reads an input sequence, makes a prediction for the next token, and then uses its own prediction as the input for the subsequent step.

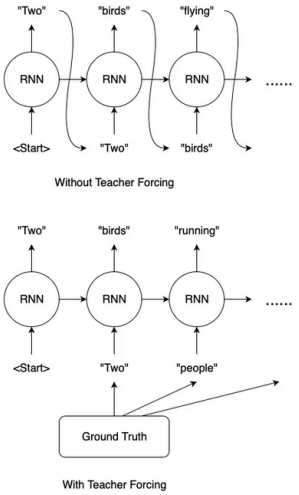

Figure 8: Teacher forcing training principle. Adapted from [11].

The interesting aspect is our test loss is way better than the NMT training without teacher forcing as it is shown in Fig 9 vs Fig 10. And in our translation, even if it sometimes feel less correct, there are no extra words or loop as it stops as soon as it encounter the token EOS (Fig 11

```
Epoch : 0/19: 100%|| 2.13k/2.13k [01:49<00:00, 19.4it/s, loss=4.66, test loss=3.73]
Epoch : 1/19: 100%|| 2.13k/2.13k [01:48<00:00, 19.6it/s, loss=3.36, test loss=3.04]
Epoch : 2/19: 100%|| 2.13k/2.13k [01:48<00:00, 19.8it/s, loss=2.87, test loss=2.72]
Epoch : 3/19: 100%|| 2.13k/2.13k [01:50<00:00, 19.3it/s, loss=2.6, test loss=2.52]
Epoch : 4/19: 100%|| 2.13k/2.13k [01:49<00:00, 19.5it/s, loss=2.41, test loss=2.38]
Epoch : 5/19: 100%|| 2.13k/2.13k [01:48<00:00, 19.7it/s, loss=2.27, test loss=2.27]
Epoch : 6/19: 100%|| 2.13k/2.13k [01:49<00:00, 19.6it/s, loss=2.16, test loss=2.18]
Epoch : 7/19: 100%|| 2.13k/2.13k [01:49<00:00, 19.4it/s, loss=2.07, test loss=2.11]
Epoch : 8/19: 100%|| 2.13k/2.13k [01:48<00:00, 19.7it/s, loss=1.99, test loss=2.05]
Epoch : 9/19: 100%|| 2.13k/2.13k [01:49<00:00, 19.6it/s, loss=1.92, test loss=2]
Epoch : 10/19: 100%|| 2.13k/2.13k [01:49<00:00, 19.5it/s, loss=1.86, test loss=1.95]
Epoch : 11/19: 100%|| 2.13k/2.13k [01:48<00:00, 19.6it/s, loss=1.82, test loss=1.92]
Epoch : 12/19: 100%|| 2.13k/2.13k [01:47<00:00, 19.9it/s, loss=1.77, test loss=1.89]
Epoch : 13/19: 100%|| 2.13k/2.13k [01:46<00:00, 20.1it/s, loss=1.74, test loss=1.86]
Epoch : 14/19: 100%|| 2.13k/2.13k [01:45<00:00, 20.2it/s, loss=1.7, test loss=1.86]
Epoch : 15/19: 100%|| 2.13k/2.13k [01:46<00:00, 20.1it/s, loss=1.67, test loss=1.81]
Epoch : 16/19: 100%|| 2.13k/2.13k [01:47<00:00, 19.9it/s, loss=1.64, test loss=1.79]
Epoch : 17/19: 100%|| 2.13k/2.13k [01:47<00:00, 19.9it/s, loss=1.61, test loss=1.78]
Epoch : 18/19: 100%|| 2.13k/2.13k [01:46<00:00, 20.0it/s, loss=1.59, test loss=1.76]
Epoch : 19/19: 100%|| 2.13k/2.13k [01:46<00:00, 20.1it/s, loss=1.57, test loss=1.75]
```

Figure 9: Loss of the teacher forcing.

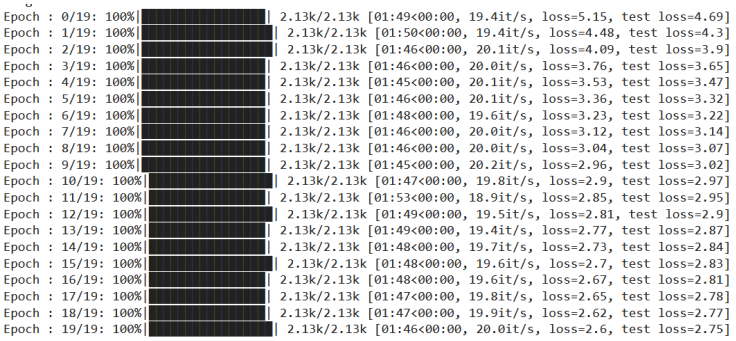

Figure 10: Loss of the training without teacher forcing. The test loss remains higher.

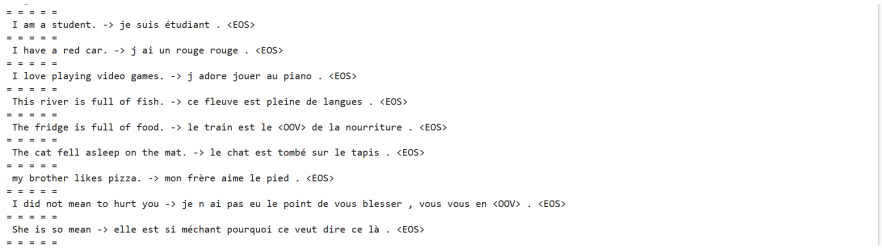

Figure 11: Translation of the teacher forcing.

References

[1] James Briggs. The three decoding methods for nlp. In *Medium / Towards Data Science*, 2021.

[2] Kyunghyun Cho Dzmitry Bahdanau and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *arXiv preprint arXiv:1409.0473*, 2014.

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

[4] Yaser Al-Onaizan Markus Freitag. Beam search strategies for neural machine translation.

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

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

[7] Prakhar Mishra. Word sequence decoding in seq2seq architectures. In *Medium / Towards Data Science*, 2019.

[8] Navdeep Jaitly Noam Shazeer Samy Bengio, Oriol Vinyals. Scheduled sampling for sequence prediction with recurrent neural networks. In *1506.03099*, 2015.

[9] Christopher Manning Thang Luong, Kyunghyun Cho. Neural machine translation, acl tutorial. 2016.

[10] Gian Wiher, Clara Meister, and Ryan Cotterell. On Decoding Strategies for Neural Text Generators. volume 10, pages 997–1012, 09 2022.

[11] Wanshun Wong. What is teacher forcing? In *Medium / Towards Data Science*, 2019.

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