Exercices - NLP Trad. Automatique

Exercice 1
Distinguez trois types de modèles dans le NLP.

Exercice 2
Expliquez simplement ce qu'est un Réseau de Neurones Récurrents.

Exercice 3
Distinguez **position absolue** et **position relative** d'un token dans une séquence.

Exercice 4
Qu'est-ce que le *Byte Pair Encoding* (BPE) ?

Exercice 5
Dans le contexte des datasets pour l'entraînement de modèles NLP de traduction automatique, comment distingue-t-on **corpus parallèles/alignés** et **corpus comparables** ? Dans quelle catégorie range-t-on *Wikipedia* ?

Exercice 6
Comment distingue-t-on l'**auto-attention** (self-attention) de la **cross-attention** ?

Exercice 7
a) Quelle est la formule permettant de calculer le score de précision BLEU entre une traduction candidate et une traduction de référence ?

b) Prenez les deux phrases **« le chat mange »** et **« le chat dort »** et exprimez sous la forme d'une fraction le score de précision BLEU en décomposant les phrases en mots.

c) Prenez les deux mêmes phrases et exprimez le score de précision en décomposant les phrases en bigrammes (paires de mots).

Corrigé ex. 1

Encodeur seul (encoder-only).
Exemple : BERT.
Objectif d'entraînement : **MLM – Masked Language Modeling** (prédire des tokens masqués).
Usage typique : compréhension du texte (classification, NER, recherche sémantique).

Décodeur seul (decoder-only).
Exemples : EuroLLM, GPT, LLaMA, Mistral.
Objectif d'entraînement : **autoregressif** (prédire le prochain jeton).
Usage typique : génération de texte, complétion, agents conversationnels.

Encodeur–décodeur (seq2seq).
Exemple : T5.
Objectif d'entraînement : **modélisation conditionnelle** (apprendre $p(y \mid x)$), souvent via des tâches de débruitage / *denoising*.
Usage typique : traduction, résumé, réécriture, question-réponse extractive / générative.

Corrigé ex. 2

L'idée d'un RNN, c'est de prédire, de proche en proche, le mot suivant, en s'appuyant sur un état intermédiaire caché. La prédiction se fait donc de manière séquentielle, ce qui a pour effet de réduire la capacité de ces modèles à encapsuler du contexte. Cet inconvénient est compensé en partie par de la rétropropagation.

Corrigé ex. 3

La **position absolue** correspond à la position du token dans la séquence (par exemple, c'est le i-ème token de la séquence). La **position relative** correspond à la position d'un token par rapport à un autre. Typiquement, si p est la position absolue d'un token et q celle d'un autre, la position relative est $|p - q|$.

Corrigé ex. 4

Le **Byte Pair Encoding (BPE)** est une idée simple utilisée à l'origine pour la **compression** et devenue très populaire en **traitement du langage naturel** pour créer des **sous-mots (subwords)** et pour tokeniser.

Principe (apprentissage du vocabulaire).

1. On part d'une séquence de symboles (au départ, des **caractères**).
2. On **compte** les paires de symboles adjacentes les plus fréquentes.

3. On **fusionne** la paire la plus fréquente en un **nouveau symbole**.
4. On **répète** ces fusions un certain nombre de fois (ou jusqu'à ce qu'il n'y ait plus de gains).

En NLP, on applique ce procédé sur un corpus pour apprendre un **vocabulaire** de sous-mots. Ensuite, pour tokeniser un texte, on refait les mêmes fusions dans le même ordre : les mots rares sont découpés en morceaux fréquents.

Corrigé ex. 5

— **Corpus parallèle / aligné**
Deux (ou plus) collections de textes qui sont **des traductions l'une de l'autre**, alignées au **moins au niveau phrase** (souvent segment ou mot via alignements).
Exemples : Europarl, OpenSubtitles (après alignement).
Usage typique : entraînement/évaluation directe de modèles de traduction (supervisée), extraction de lexiques bilingues.

— **Corpus comparable**
Collections de textes dans différentes langues qui **traitent des mêmes thèmes/domaines/époques** mais **ne sont pas des traductions** entre elles (pas d'alignement phrase-à-phrase garanti).
Exemples : articles de presse du même jour dans plusieurs langues, articles *Wikipedia*.
Usage typique : pré-entraînement, adaptation domaine/registre, induction lexicale, apprentissage semi/auto-supervisé.

Wikipedia est majoritairement un corpus comparable.

Corrigé ex. 6

	Auto-attention	**Attention croisée**
Ce qui fait attention à quoi	Les éléments d'une séquence font attention à eux-mêmes (chaque token regarde les autres tokens de la même séquence).	Les éléments de la séquence A font attention à la séquence B (les tokens d'une séquence cible regardent les tokens d'une séquence source).
Q, K, V proviennent de	Q, K, V du même input X.	Q de la cible Y ; K, V de la source X.
Utilisation typique	Construction de représentations au sein d'une entrée : couches d'encodeur dans BERT ; auto-attention causale (masquée) dans les décodeurs	Conditionnement sur un contexte externe : décodeur faisant attention aux sorties de l'encodeur (traduction)
Masquage	Peut être bidirectionnel (encodeurs) ou causal/masqué par anticipation (décodeurs).	Généralement pas de masque causal ; peut masquer le padding ou la structure—la visibilité concerne quelles positions sources sont vues, pas les tokens futurs.
Complexité	$O(n^2)$ pour une longueur de séquence n.	$O(n_{\text{cible}} \cdot n_{\text{source}})$; utile quand un côté est court/fixe.

Corrigé ex. 7

a) Formule générale :

$$\text{Précis°}(\text{Candidate}, \text{Target}) = \frac{\text{nombre de } n\text{-grammes concordants entre Candidate et Target}}{\text{nombre total de } n\text{-grammes dans Target}}$$

b) Avec des 1-grammes (unigrammes)
Soit $n = 1$.

- **Candidate** : « le chat mange »
- **Target** : « le chat dort »

1-grammes dans *Candidate* : {« le », « chat », « mange »}
1-grammes dans *Target* : {« le », « chat », « dort »}
1-grammes concordants : {« le », « chat »} $\Rightarrow$ 2 mots en commun
Nombre total de 1-grammes dans *Target* : 3

$$\text{Précision} = \frac{2}{3} \approx 0{,}67 = 67\,\%$$

c) Avec des 2-grammes (bigrammes)
Un bigramme est une paire de mots consécutifs. Ici, (*le*, *chat*) est un seul bigramme. Appliquons la formule avec $n = 2$.

- **Candidate** : « le chat mange la souris »
- **Target** : « le chat dort »

2-grammes dans *Candidate* : {« le chat », « chat mange », « mange la », « la souris »}
2-grammes dans *Target* : {« le chat », « chat dort »}
2-grammes concordants : {« le chat »} $\Rightarrow$ 1 bigramme en commun
Nombre total de 2-grammes dans *Target* : 2

$$\text{Précision} = \frac{1}{2} = 0{,}5 = 50\,\%$$