ChatGPT : Quelle réalité se cache derrière la fiction?

Henock Makumbu Mboko

Senior Data & AI Engineer

Congo Tech Pamoja: Communauté IA pour tous

Partie II : Modèles pré-entraînés

Voyage au cœur des modèles pré-entraînés

Cette partie, plus technique, permettra aux plus curieux d'explorer les coulisses scientifiques : comment fonctionnent ces modèles comme GPT ou LLaMA, pourquoi parle-t-on de "génératif", et comment cela transforme nos interactions numériques

Partie II : Modèles pré-entraînés

Avant l'ère de LLMs

Avant les LLMs (Large Language Modèles) : NLP

Avant les LLMs ou Modèles de langage de grande taille, il y avait le NLP (Natural Language Processing), ou traitement du langage naturel. C'est un champ d'étude à la croisée de la linguistique et du machine learning, dont le but est de permettre aux machines de comprendre, analyser et produire du langage humain.

Exemples de tâches typiques en NLP :

- Classer des phrases (ex : e-mail spam ou non).
- Classer des mots (ex : nature grammaticale, entités nommées).
- Générer du texte (ex : compléter une phrase).
- Extraire des informations (ex : répondre à une question à partir d'un texte).
- Transformer du texte (ex : résumé, traduction, reformulation).
- La reconnaissance vocale (audio $\rightarrow$ texte).
- La compréhension d'images (description d'une photo).

Avant les Transformers : les réseaux récurrents

Avant l'architecture Transformer, les chercheurs utilisaient surtout des réseaux de neurones récurrents et leurs variantes comme les LSTM (Long Short-Term Memory) et GRU (Gated Recurrent Unit), adaptés au traitement séquentiel. Mais ces réseaux avaient certaines limites :

- **Traitement séquentiel** : l'information est traitée mot par mot, ce qui rend l'entraînement difficile à paralléliser et donc lent.
- **Mémoire limitée** : même les LSTM finissent par "oublier" l'information contextuelle sur de longues séquences.
- **Difficulté à capter les dépendances longues** : notamment lorsque des mots clés sont très éloignés dans une phrase.
- **Rigidité dans l'encodage du contexte** : chaque mot est influencé uniquement par le passé (ou par une fenêtre temporelle dans les modèles bidirectionnels), sans avoir une vue d'ensemble immédiate de la phrase.

Une perte d'information initiale entraînait la perte du contexte global. C'est là qu'intervient l'architecture Transformer, qui traite les séquences en parallèle et utilise l'attention pour capter efficacement les relations entre mots, même éloignés.

Partie II : Modèles pré-entraînés

Changement d'échelle et de paradigme

Du NLP aux LLMs

Les LLMs (Large Language Models) comme GPT, LLaMA ou Claude ont révolutionné l'approche. Ce sont des modèles de très grande taille, basés sur les Transformers, et dotés de capacités nouvelles :

- **Échelle** : des millions à des centaines de milliards de paramètres.
- **Polyvalence** : ils savent résumer, traduire, dialoguer... sans réentraînement spécifique.
- **Apprentissage en contexte (In-context learning)** : ils peuvent adapter leur réponse selon les exemples donnés dans le prompt.
- **Capacités émergentes (Emergent abilities)** : des aptitudes qui apparaissent spontanément à grande échelle (résolution de problèmes logiques, écriture de code...).

Comment fonctionnent les modèles de langage ? (LLMs

Les modèles de langage sont des intelligences artificielles entraînées à **prédire le mot suivant** dans une séquence de texte, en se basant sur le contexte fourni par les mots environnants. Cet entraînement leur permet de développer une compréhension statistique du langage, réutilisable dans de nombreuses tâches comme la génération de texte, la traduction, le résumé automatique, ou la réponse à des questions.

Les tâches des LLMs

Les LLMs sont capables de réaliser toutes les tâches du NLP classique et bien d'autres... simplement grâce à quelques instructions, qu'on appelle "prompts". Ils peuvent :

- Traduire des textes,
- Générer du code ou des textes,
- Résumer ou reformuler,
- Classer ou annoter du contenu,
- Identifier des entités ou informations clés,
- Analyser le ton ou l'émotion d'un texte (Analyse sentimentale)
- Compléter automatiquement des textes (autocomplétion),
- Résoudre des problèmes mathématiques ou logiques,
- Corriger l'orthographe et la grammaire,
- ... et bien plus

Quelques concepts basiques à comprendre (1)

- **Prompt** : tout commence par votre question. C'est l'entrée, ce que vous tapez.
- **Fenêtre de contexte (Context Window)** : l'IA lit seulement une partie du texte à la fois, comme si elle avait une mémoire à court terme.
- **Tokens** : le texte est découpé en petits morceaux (mots ou bouts de mots), un peu comme des pièces de puzzle.
- **Complétion** : elle répond en générant un mot à la fois, en devinant le plus probable à chaque étape.
- **Inférence** : c'est le processus interne de réflexion, où elle calcule la suite à partir de milliards d'exemples déjà vus.
- **In-Context Learning (ou apprentissage en contexte)** : c'est quand un modèle d'intelligence artificielle apprend en observant des exemples qu'on lui donne directement dans le prompt, sans avoir besoin d'être reprogrammée ou réentraînée
- **Fine-tuning (ou ajustement fin)** consiste à réentraîner un modèle d'intelligence artificielle sur de nouvelles données spécifiques, pour qu'il s'adapte à un nouveau type de tâche ou de contenu.

Quelques concepts basiques à comprendre (2)

Les techniques utilisées en in-context learning pour adapter un modèle d'IA générative à une tâche donnée.

- zero-shot inference : Le modèle reçoit seulement une question, sans exemple.
- One-shot / Few-shot inference : On donne un ou quelques exemples dans le prompt.
- Multi-shot inference (souvent appelé aussi few-shot, mais avec plus d'exemples) : on donne plusieurs exemples (souvent 5, 10, ou plus) pour guider le modèle.

Les hyperparamètres : paramètres de contrôle qui influencent la façon dont le texte est généré

- **max_new_tokens**: Contrôle la longueur de la réponse
- **temperature** : Gère la créativité (0 = précis, 1 = aléatoire)
- **top-k** : Restreint le nombre de choix possibles
- **top-p** : Filtrage dynamique des probabilités
- **repetition_penalty** : Évite la répétition excessive
- **presence_penalty** : Encourage la diversité des sujets
- **frequency_penalty** : Réduit la répétition de mots spécifiques
- **length_penalty** : Encourage des réponses plus longues ou plus courtes

Partie II : Modèles pré-entraînés

Longée dans le Transformer

La mécanique secrète des LLMs

Avant de parler des LLMs modernes, rappelons que leur ADN repose sur une innovation-clé : le Transformer. Il ne s'agit pas d'un robot de science-fiction, mais d'une architecture de réseau de neurones qui a bouleversé le traitement du langage.

Le Transformer a été introduit en juin 2017 par un article au titre devenu célèbre : "Attention is All You Need". Contrairement aux anciens modèles séquentiels (comme les LSTM), il ne lit pas les mots un par un, mais traite l'ensemble de la phrase simultanément en se concentrant (grâce à un mécanisme appelé "l'attention") sur les mots les plus pertinents pour comprendre le sens.

Les modèles modernes utilisent l'architecture **Transformer**. Il existe deux principales approches pour les entraîner

- Masked Language Modeling (MLM) – Apprentissage avec des mots masqués
- Causal Language Modeling (CLM) – Modélisation causale du langage

Entraînement des modèles Transformer

1. Masked Language Modeling (MLM) Cette méthode est utilisée par des modèles auto-encodeurs comme **BERT** (Bidirectional Encoder Representations from Transformers).

- On prend une phrase et on **masque aléatoirement certains mots**.
- Le modèle est entraîné à **prédire les mots masqués** en utilisant le contexte complet de la phrase, c'est-à-dire les mots avant et après le mot masqué.
- Cela permet au modèle d'apprendre une **compréhension bidirectionnelle** du langage.

Exemple :

Phrase originale : "Le chat dort sur le canapé."
Phrase masquée : "Le chat _____ sur le canapé."
Prédiction attendue : "dort"

2. Causal Language Modeling (CLM) Cette méthode est utilisée par des modèles auto-régressifs comme **GPT** (Generative Pre-trained Transformer).

- Le modèle lit les mots de gauche à droite.
- Il est entraîné à **prédire le mot suivant** à chaque étape, sans avoir accès aux mots futurs.
- Cela le rend très performant pour la **génération de texte**, puisqu'il peut produire une phrase mot par mot de manière cohérente.

Exemple :

Entrée : "Le chat dort"
Prédiction : "sur", puis "le", puis "canapé", etc.

Note : Les modèles encodeur-décodeur (séquence-à-séquence) combinent masquage et génération : ils encodent une entrée où des segments (et non des mots isolés) sont masqués à l'aide de marqueurs spéciaux (span corruption), puis génèrent la sortie de façon causale, token par token. Ce cadre est idéal pour les tâches comme la traduction, le résumé ou la question-réponse.

Restez avec nous !!!

l'architecture "Transformers" sera abordée en détail dans la prochaine partie (**Partie III**).

Notre réseau social scientifique est dédié aux passionnés d'**IA**, de **Big Data** et d'**analyse de données**.

Ouvert à tous :

- Passionnés
- Débutants curieux
- Intéressés par la technologie
- Étudiants
- Chercheurs
- Professionnels du domaine

Rejoignez-nous pour apprendre, partager et avancer ensemble !

Merci d'aimer et partager à tous !