IREM INFO

IAG et Prompt Engineering

pour la génération de code

Christophe Schlick

FAITS MARQUANTS

1958 : Perceptron = 1er réseau de neurones artificiels
1966 : Eliza = 1er agent conversationnel déterministe
1970 : Sieve = 1er système de composition musicale
1972 : Myrcin = 1er système-expert (diagnostic médical)
1973 : Prolog = 1er langage de programmation pour l'IA
1979 : Gamanoid = 1e victoire vs champion Backgammon
1989 : Alvinn = 1er système de conduite autonome
1992 : Civilization = 1er jeu vidéo avec IA (contrôle PNJ)
1995 : Dragon = 1e dictée vocale avec transcription
1997 : DeepBlue = 1e victoire vs champion d'Echecs
1999 : Aïbo = 1er animal domestique robotisé
2001 : Black & White = 1er jeu avec gestion totale par IA

FAITS MARQUANTS

2004 : RapidMiner = 1er outil d'analyse de données par IA
2006 : Google Translate = 1er système de traduction
2010 : DeepMind = 1er logiciel d'apprentissage profond
2011 : Siri = 1er assistant basé sur reconnaissance vocale
2016 : AlphaGo = 1ère victoire vs champion de Go
2018 : GPT-1 = 1er générateur non-déterministe de textes
2021 : Dall-E = 1er générateur non-déterministe d'images
2022 : ChatGPT = 1e agent conversation non-déterministe
2023 : Auto-GPT = 1er système IAG agentique autonome
2024 : Gemini & GPT-4o = 1ers systèmes IAG multimodaux
2025 : DeepSeek = 1er système IAG en open-source
2026 : Inclusion de l'IAG agentique dans tous les moteurs

VOCABULAIRE ET ACRONYMES

- **NN :** Neural Network (réseau de neurones artificiels)
- **DNN :** Deep Neural Network (= nombreuses couches)
- **[U]SML :** (Un)Supervised Machine Learning (apprentissage supervisé ou non-supervisé)
- **RLHF :** Reinforcement Learning by Human Feedback (apprentissage par renforcement supervisé)
- **NLP :** Natural Language Processing (traitement automatique des langues humaines)
- **LLM :** Large Language Model (système permettant d'analyser ou de générer du texte en langage naturel)

VOCABULAIRE ET ACRONYMES

- **Token :** unité atomique pour l'encodage par les LLM (selon les cas : mot, syllabe ou signe de ponctuation)
- **Transformer :** Système de LLM basé sur la prédiction des tokens à venir en fonction du contexte
- **PT :** Pre-Trained Transformer (inclusion d'un processus d'apprentissage pour améliorer la prédiction)
- **Corpus :** soupe de données textuelles utilisée pour l'apprentissage non-supervisée du Transformer
- **GPT :** Generative Pre-Trained Transformer (utilisation du modèle pour effectuer de la génération de texte)

HISTORIQUE IA

1950-1960 : Algorithme alpha-beta (jeux à 2 joueurs)

1960-1970 : Systèmes experts et NN simple couche

1970-1980 : Logique floue et modèles de Markov cachés

1980-1990 : NN multi-couches et rétro-propagation

1990-2000 : NN récursifs et apprentissage automatique

2000-2010 : NN profonds et analyse auto de données

2010-2020 : App par renforcement et analyse multimédia

2020-2030 : IAG, multimodalité, agents autonomes, …

HISTORIQUE IA GÉNÉRATIVE

2006 : Réseau de neurones profond (DNN)

2013 : Auto-encodeur variationnel (VAE)

*2 DNN en collaboration : **transcodeur** et **encodeur***

2015 : Réseau antagoniste génératif (GAN)

*2 DNN en compétition : **générateur** et **discriminateur***

2017 : Transformateur (Transformer ou T)

Analyse non-séquentielle → apprentissage parallèle

2018 : Transformateur Génératif Pré-entrainé (GPT)

Prédiction d'éléments à partir de données contextuelles

PRINCIPAUX OUTILS IAG

- **OpenAI :** ChatGPT *(Free, Plus, Pro, Enterprise, Edu)*
- **Microsoft :** Copilot *(Free, Studio, Enterprise)*
- **Google :** Gemini *(Free, Plus, Pro, Ultra)*
- **Anthropic :** Claude *(Free, Pro, Team, Enterprise)*
- **Mistral :** Le Chat *(Free, Pro, Team, Enterprise)*
- **Meta, Perplexity, … :** *outils IA spécialisés*
- **Poe, ChatHub, Monica, Magai, … :** *intégrateurs*

PRINCIPAUX OUTILS IAG

Trois types d'interfaces pour les outils IAG :

1 - Chatbot autonome : site web ou application

2 - Chatbot embarqué : IDE

3 - Package logiciel : API (multi-langages)

Les outils gratuits sont presque tous de Type 1

Sauf **Copilot + VSCode** *(étudiants uniquement)* et **API Gemini 3.1** *(avec limites assez souples)*

COMPOSANTES D'UNE IAG

- **DNN** avec rôles différents par zone de couches
- **Corpus** (10G à 10T mots) pour apprentissage *Common Crawl, Wikipedia, GitHub, MassiveText*
- **Matrices de paramètres** (en G ou T) pour les couches de neurones, le contexte et la mémoire
- **Modèle de prédiction** des tokens, avec valeur de température pour variabilité des réponses
- **Classification** des possibilités de génération
- **Requêtes et données**, fournies par utilisateur et limitées par une taille de contexte en tokens

PARAMÈTRES ET TOKENS

Evolution des modèles développés par OpenAI

- **G1** *(06/2018)* = 117M paramètres, 0.5k tokens
- **G2** *(02/2019)* = 1.5G paramètres, 1k tokens
- **G3** *(06/2020)* = 175G paramètres, 4k tokens
- **G4** *(09/2022)* = 1T paramètres, 8k tokens
- **G4o** *(04/2023)* = 4T paramètres, 64k tokens
- **O1** *(09/2024)* = ??? paramètres, 200k tokens
- **G5** *(08/2025)* = ??? paramètres, 400k tokens
- **G5.5** *(04/2026)* = ??? paramètres, 1M tokens

FONCTIONNEMENT

- Une IAG génère un texte **token par token** en choisissant le token le plus probable en fonction des données utilisées pour l'entrainement
- **Aucun** raisonnement de nature cognitive
- **Aucune** compréhension des textes en entrée
- **Aucune** compréhension des textes en sortie
- **Aucune** vue globale sur le texte généré
- Seule la **requête** fournie par l'utilisateur ***(prompt)*** contient potentiellement une plus-value cognitive

EXEMPLES

- **Le ciel est … :** 3 tokens
  bleu 35%, gris 20%, nuageux 15%, sombre 10%, dégagé 10%, clair 5%, orageux 5%
- **Il fait beau, le ciel est ... :** 7 tokens
  dégagé 35%, bleu 30%, clair 15%, ensoleillé 10%, lumineux 5%, limpide 5%
- **L'orage se lève, le ciel est ... :** 8 tokens
  sombre 30%, noir 25%, menaçant 20%, gris 15%, nuageux 5%, turbulent 5%
- **Il fait beau, mais le ciel est ... :** 8 tokens
  nuageux 40%, couvert 30%, gris 15%, voilé 10%, menaçant 5%

EXEMPLES

Tout système IAG inclut un paramètre **température** qui permet de contrôler la variabilité des réponses

Ecrire 2 blagues sur le thème de l'argent, avec une température de 0 et 2, respectivement :

- **T = 0 :** Pourquoi les banquiers aiment-ils l'argent ? Parce qu'ils savent comment le faire fructifier !
- **T = 2 :** Pourquoi l'argent ne parle jamais ? Parce qu'il est trop occupé à changer de poche en dansant la samba avec les centimes !

Version 2023

EXEMPLES

Tout système IAG inclut un paramètre **température** qui permet de contrôler la variabilité des réponses

Ecrire 2 blagues sur le thème de l'argent, avec une température de 0 et 2, respectivement :

- **T = 0 :** Pourquoi les prix n'aiment pas les ascenseurs ? Parce qu'ils préfèrent monter tout seul !
- **T = 2 :** J'ai mis une pièce dans une fontaine pour faire un vœu. La fontaine m'a rendu un reçu fiscal et m'a proposé un crédit sur 36 mois !

Version 2026

ANTHROPOMORPHISME

Même si elle semble parler ou raisonner comme un humain, une IAG est un logiciel informatique. Il faut donc éviter les 4 formes d'anthropomorphisme :

- **Niveau 1 = Courtoisie / Respect :**

  Ne pas tutoyer ou vouvoyer l'IAG avec politesse

- **Niveau 2 = Encouragements / Insultes :**

  Ne pas remercier, féliciter ou insulter l'IAG

ANTHROPOMORPHISME

Même si elle semble parler ou raisonner comme un humain, une IAG est un logiciel informatique. Il faut donc éviter les 4 formes d'anthropomorphisme :

- **Niveau 3 = Personnification :**

  Ne pas assigner un rôle humain à l'IAG

- **Niveau 4 = Amitié / Confidence :**

  Ne pas établir une relation personnelle avec l'IAG

PROMPT ENGINEERING

- Une IAG génère un texte **token par token** en choisissant le token le plus probable en fonction des données utilisées pour l'entrainement
- Aucun raisonnement ni aucune compréhension
- Seule la **requête** fournie par l'utilisateur ***(prompt)*** contient potentiellement une plus-value cognitive
- Il existe un ensemble de **bonnes pratiques** pour la création de prompts qui permettent d'optimiser la réponse de l'IAG : ***prompt engineering***

PROMPT ENGINEERING

- **Task :** tâche à effectuer *(verbe + compléments)*
- **Process :** instructions sur le processus à suivre ou sur l'ordonnancement des tâches
- **Context :** ensemble des éléments de contexte destinés à focaliser la réponse
- **Examples :** exemples de réponses attendues *(one-shot learning ou few-shot learning)*
- **Target :** niveau d'expertise de la cible *(adjectifs ou liste de compétences)*
- **Format :** format ou style souhaité pour la réponse *(format de fichier standard ou éléments à inclure)*

PROMPT ENGINEERING

- **Prompt minimal :**

  Task + Format

- **Prompt classique pour code :**

  Task + Process + [Examples] + Format

- **Prompt classique pour documentation :**

  Task + Context + Target + Format

- **Prompt maximal :**

  Task + Process + Context + Examples + Target + Format

PROMPT ENGINEERING

- **Rédaction :** économiser le nombre de tokens *(en **anglais**, à l'**infinitif**, phrases **"sans gras"**)*
- **Focalisation :** chaînage de prompts focalisés, au lieu d'un prompt long avec nombreux éléments
- **Identificateurs :** définition par **{key} = value** puis réutilisation par **{key}** dans la suite du prompt
- **Délimiteurs :** utiliser syntaxe Python / Markdown

  "..." / """...""" / **(...)** / **[...]** / ***...*** / **– …**

TOKENS (ENG vs FRA)

OpenAI G3 (2020)

La Déclaration des Droits de l'Homme : 13 tokens

Declaration of Human Rights : 5 tokens

OpenAI G4o (2023)

La Déclaration des Droits de l'Homme : 10 tokens

Declaration of Human Rights : 4 tokens

OpenAI G5 (2025)

La Déclaration des Droits de l'Homme : 7 tokens

Declaration of Human Rights : 4 tokens