

Autonomisation des femmes par l'acquisition de connaissances et de compétences en programmation pour des opportunités d'emploi dans le secteur des technologies de l'information



ENCODE-IT

Projet 2024-2-PT01-KA210-ADU-000265571



Cofinancé par
l'Union européenne

Financé par l'Union européenne. Les points de vue et opinions exprimés n'engagent toutefois que leur(s) auteur(s) et ne reflètent pas nécessairement ceux de l'Union européenne ou de l'Agence exécutive européenne pour l'éducation et la culture (EACEA). Ni l'Union européenne ni l'EACEA ne peuvent en être tenues responsables.



Logique algorithmique et Prompt Engineering



Cofinancé par
l'Union européenne



4.1. Qu'est-ce qu'un algorithme et pourquoi est-il important ?



Cofinancé par
l'Union européenne

4.1. Qu'est-ce qu'un algorithme et pourquoi est-il important ?

Un algorithme est une suite claire et ordonnée d'étapes suivies pour résoudre un problème ou accomplir une tâche. En réalité, nous créons constamment des algorithmes dans notre vie quotidienne : préparer du thé, se préparer avant de sortir ou choisir un trajet en transport public. Chacune de ces actions suit une logique du type « d'abord faire ceci, puis cela ».

Les ordinateurs fonctionnent exactement de cette manière : ils ont besoin d'étapes précises indiquant ce qui vient en premier, ce qui suit et quel chemin prendre selon la situation.

Un algorithme comporte trois éléments principaux : l'entrée, les étapes de traitement et la sortie.

- L'entrée correspond aux données du problème (par exemple : « deux nombres »).
- Les étapes de traitement expliquent quoi faire avec ces données (additionner, comparer, trier, etc.).
- La sortie est le résultat attendu (la somme, le plus grand nombre, la liste triée).

Comme les ordinateurs fonctionnent selon des règles et non des émotions, les instructions doivent être claires et précises. Des expressions comme « attendre un peu » ou « si nécessaire » sont ambiguës pour un ordinateur. Il faut plutôt utiliser des formulations précises comme « attendre 3 secondes » ou « si la valeur est supérieure à 10 ».



4.1. Qu'est-ce qu'un algorithme et pourquoi est-il important ?

La pensée algorithmique permet de diviser une tâche en petites étapes. Au lieu de résoudre un grand problème d'un seul coup, celui-ci est séparé en différentes actions comme : « d'abord récupérer les données, ensuite les vérifier, signaler les erreurs éventuelles puis continuer s'il n'y en a pas ».

Cette approche a deux avantages : elle réduit le risque d'erreurs et facilite la réutilisation. Avec quelques modifications, le même processus peut être utilisé dans d'autres projets.

Prenons un exemple du quotidien : un formulaire de candidature en ligne.

- Entrée : informations comme le nom, l'e-mail et le numéro de téléphone.
- Étapes : vérifier si des champs sont vides, si l'adresse e-mail est correcte, si le numéro contient uniquement des chiffres et si la case à cocher est sélectionnée.
- En cas d'erreur : informer l'utilisateur (« e-mail invalide »).
- Sinon : enregistrer les données et afficher un message de remerciement.
- Sortie : une candidature enregistrée dans la base de données.

La description de ce processus pour un ordinateur constitue un algorithme. Il peut être représenté sous forme de schéma ou simplement écrit en texte clair.



4.1. Qu'est-ce qu'un algorithme et pourquoi est-il important ?

Les algorithmes ne suivent pas toujours un chemin unique. Parfois, des conditions et des embranchements sont nécessaires. Par exemple :

« Si le mot de passe contient moins de 8 caractères, afficher un avertissement ; sinon, autoriser l'accès. »

Parfois, une boucle est nécessaire :

« Vérifier chaque élément de la liste un par un ; si un élément incorrect est trouvé, arrêter le processus. »

Les conditions et les boucles ressemblent beaucoup aux décisions prises dans la vie quotidienne. Par exemple, lorsque vous vérifiez votre panier avant de passer à la caisse au supermarché : « s'il y a des articles en trop, les retirer ; sinon, continuer vers la caisse ».

L'aspect le plus humain dans l'écriture d'algorithmes concerne les cas particuliers, appelés edge cases. La plupart des situations fonctionnent normalement ; les problèmes apparaissent dans des cas rares mais importants.

Par exemple, un utilisateur peut avoir ajouté un espace dans son numéro de téléphone, écrit son prénom et son nom en un seul mot, ou téléchargé un fichier dépassant la taille autorisée.

Un bon algorithme anticipe et gère ces situations dès le départ. Cette prévoyance rend le logiciel plus fiable et améliore l'expérience utilisateur.



4.1. Qu'est-ce qu'un algorithme et pourquoi est-il important ?

Un autre point essentiel est la simplicité. Souvent, une solution « plus courte et plus claire » est préférable à une solution « plus complexe mais plus ingénieuse ».

Un algorithme simple est plus facile à tester et plus rapide à mettre à jour lorsqu'une maintenance est nécessaire. Dans une équipe ayant différents niveaux d'expérience, des processus clairs renforcent également la collaboration. La simplicité est donc synonyme de qualité et de durabilité.

Un algorithme n'a pas besoin d'être le « plan parfait ». Dans certains cas, on recherche non pas la meilleure solution possible, mais une solution suffisamment bonne et suffisamment rapide ; on parle alors d'approche heuristique. Par exemple, calculer toutes les possibilités pour déterminer un itinéraire de livraison pourrait prendre plusieurs jours. Une méthode rapide et efficace apporte souvent davantage de valeur à l'utilisateur. Cet équilibre entre précision, rapidité et coût est particulièrement important dans les projets disposant de ressources limitées.



4.1. Qu'est-ce qu'un algorithme et pourquoi est-il important ?

Enfin, un bon algorithme repose sur une habitude essentielle : le retour d'information et la correction. Une fois le processus établi, il doit être testé avec de petits exemples afin d'identifier les erreurs et clarifier les étapes. Ce cycle d'essai, d'observation, d'amélioration et de développement est au cœur de tout apprentissage efficace. Il ne faut pas en avoir peur ; au contraire, c'est la clé d'un progrès sûr et durable.

En résumé, un algorithme est l'art d'expliquer à un ordinateur non seulement « quoi faire », mais aussi « quoi faire, dans quel ordre et sous quelles conditions ».

Qu'il s'agisse de la validation d'un formulaire ou du fonctionnement d'une plateforme éducative, un algorithme bien conçu simplifie le travail, réduit les erreurs et offre à l'utilisateur une expérience fluide et claire.

Une fois cette logique comprise, commencer à coder devient beaucoup moins intimidant, car vous comprenez clairement ce que vous faites, pourquoi vous le faites et comment cela fonctionne.





4.2. Pensée algorithmique et étapes de résolution de problèmes



Cofinancé par
l'Union européenne

4.2. Pensée algorithmique et étapes de résolution de problèmes

Résoudre un problème consiste souvent moins à chercher ce qui a été mal fait qu'à savoir par où commencer. La pensée algorithmique nous enseigne précisément cela : décomposer un problème complexe en petites parties gérables et organiser chaque étape dans un ordre logique.

Tout commence par la définition du problème. Que faut-il faire ? Quelles informations sont nécessaires ? Quelles conditions fixent les limites ?

Toute action entreprise sans répondre à ces questions ressemble à une flèche tirée dans la mauvaise direction. Un problème bien défini représente déjà la moitié de la solution.

L'étape suivante consiste à diviser le problème en plusieurs parties. Essayer de résoudre directement un problème important paraît souvent compliqué. Mais lorsqu'il est découpé en petites étapes, tout devient plus simple.

Par exemple, si vous souhaitez créer un formulaire web, vous réfléchirez séparément à l'interface, à la validation des données puis au processus d'enregistrement.

Cette méthode facilite la réflexion, comme lorsqu'on commence un puzzle par les coins.



4.2. Pensée algorithmique et étapes de résolution de problèmes

La reconnaissance des modèles constitue la deuxième étape de ce processus.

Comment avez-vous résolu un problème similaire auparavant ? Ou quelqu'un d'autre l'a-t-il déjà résolu ?

L'informatique repose largement sur la reconnaissance de modèles. Lorsqu'un système rencontre une situation similaire à une expérience passée, il peut réutiliser cette expérience dans un nouveau contexte.

Le cerveau humain fonctionne de manière comparable : nous n'inventons pas une nouvelle méthode chaque fois que nous ouvrons une porte ; une fois appris, le geste est répété automatiquement.

L'étape d'abstraction consiste à mettre de côté les détails complexes afin de se concentrer sur l'essentiel.

Par exemple, lors de la création d'un programme de calcul, il est plus important de réfléchir d'abord à la manière dont l'utilisateur saisira les données et à la façon dont le système les traitera, plutôt qu'au design de l'interface.

L'abstraction évite que des détails inutiles ne détournent l'attention et permet de concentrer l'énergie sur les points les plus importants.



4.2. Pensée algorithmique et étapes de résolution de problèmes

Après ces étapes, il est temps de développer une solution pas à pas.

On définit l'objectif de chaque étape, le moment où elle s'exécute et les conditions qui peuvent la modifier. C'est le langage des ordinateurs : clair, séquentiel et conditionnel.

Par exemple : « si l'utilisateur est connecté, le rediriger vers la page d'accueil ; sinon, afficher un avertissement » constitue une petite chaîne logique, mais très claire.

Ensuite, il reste à tester et corriger les erreurs. Aucun processus n'est parfait du premier coup. Trouver des erreurs est souvent l'une des meilleures façons d'apprendre, car cela nous pousse à demander : « Pourquoi cela n'a-t-il pas fonctionné ? »

Cette façon de penser dépasse le monde du logiciel. Elle s'applique aussi à la préparation d'une présentation, à l'organisation d'un événement ou à la création d'un workflow : définir le problème, le diviser en parties, reconnaître les modèles, simplifier et avancer étape par étape.

Cette méthode rend les problèmes complexes plus compréhensibles et augmente la productivité. La pensée algorithmique n'est pas seulement utile pour guider les ordinateurs ; elle aide aussi à organiser son propre esprit. Structurer la pensée est la meilleure façon de maîtriser le chaos.





4.3. La relation entre l'intelligence artificielle et les algorithmes



Cofinancé par
l'Union européenne

4.3. La relation entre l'intelligence artificielle et les algorithmes

Ce qui rend une machine « intelligente » repose en réalité sur les algorithmes.

Les systèmes d'intelligence artificielle, bien qu'ils paraissent complexes, restent fondamentalement des structures fonctionnant étape par étape, capables d'analyser des données et d'aboutir à des conclusions logiques.

La différence est qu'aujourd'hui, ces systèmes utilisent non seulement des règles fixes, mais aussi la capacité d'apprendre à partir de l'expérience.

Les algorithmes traditionnels produisent toujours le même résultat pour une même entrée. Une calculatrice effectuera toujours une addition de la même manière, quels que soient les nombres saisis.

L'intelligence artificielle ajoute une dimension supplémentaire à cette logique : elle mémorise des exemples passés, compare des situations similaires et génère de nouveaux résultats.

Ce processus d'apprentissage ressemble beaucoup au fonctionnement du cerveau humain.



4.3. La relation entre l'intelligence artificielle et les algorithmes

Prenons un exemple : le filtre de « courrier indésirable » dans une boîte e-mail.

Au départ, le système prend des décisions à partir de certains mots ou adresses spécifiques. Mais chaque fois qu'un utilisateur marque un message comme « spam », le système apprend quelque chose de nouveau.

Avec le temps, il met à jour ses propres règles et filtre automatiquement des e-mails similaires à l'avenir. Ce processus correspond à un algorithme qui s'entraîne à partir des données.

L'intelligence artificielle est en réalité une extension avancée des algorithmes. D'un côté, elle repose toujours sur une logique du type « si-alors » ; de l'autre, elle adapte cette logique grâce à l'expérience acquise.

Cette flexibilité transforme un logiciel statique en un système dynamique capable d'apprendre.



4.3. La relation entre l'intelligence artificielle et les algorithmes

À ce stade, la question ne devient plus seulement « comment cela fonctionne ? », mais plutôt « pourquoi le système a-t-il pris cette décision ? »

En effet, chaque décision prise par un système d'intelligence artificielle repose sur des milliers d'exemples déjà observés ainsi que sur les relations de probabilité qu'il a établies.

Comprendre l'intelligence artificielle revient donc à comprendre la manière dont l'algorithme réfléchit.

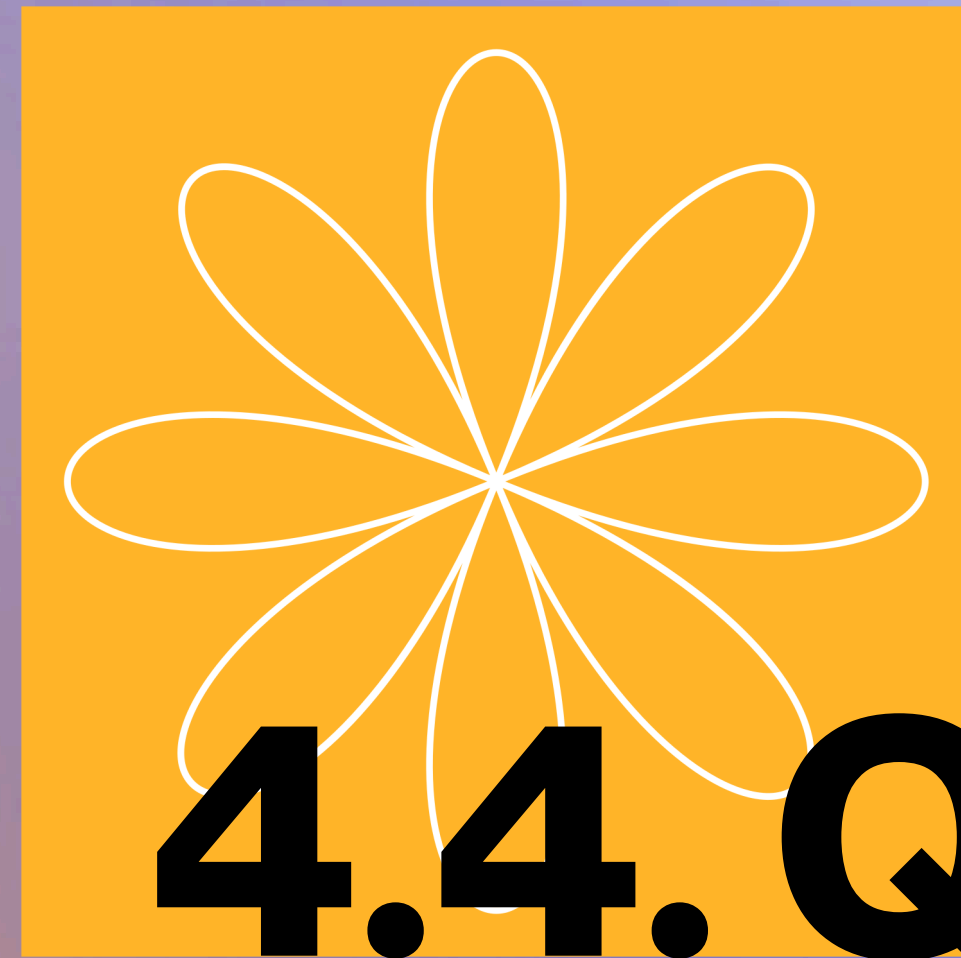
C'est ici qu'apparaît le point commun entre les humains et les machines. Tous deux comprennent le monde à travers des modèles, analysent les différences et tirent des conclusions.

La différence est que l'humain agit avec des émotions et de l'intuition, tandis que la machine fonctionne à partir de données et de probabilités. Pourtant, les deux poursuivent le même objectif : prendre de meilleures décisions.

Aujourd'hui, derrière chaque modèle d'intelligence artificielle se cachent des algorithmes composés de centaines de couches. De la reconnaissance d'images aux systèmes de traitement du langage, tous ces processus reposent sur des mécanismes déterminant comment les données sont analysées.

Les algorithmes ne sont donc pas seulement la base de l'intelligence artificielle ; ils reflètent aussi sa manière de penser.





4.4. Qu'est-ce qu'un prompt ?



Cofinancé par
l'Union européenne

4.4. Qu'est-ce qu'un prompt ?

Chaque commande donnée à un système d'intelligence artificielle est en réalité un prompt. Dans sa forme la plus simple, c'est la manière dont nous communiquons avec la machine.

Des phrases comme « Écris-moi une histoire », « Analyse ces données » ou « Explique ce sujet simplement » sont toutes des exemples de prompts.

Le système interprète ces instructions à la fois sur le plan linguistique et logique, puis génère la réponse la plus adaptée à partir des informations qu'il a apprises.

Un bon prompt ne se contente pas d'indiquer ce que nous voulons ; il explique aussi comment nous le voulons.

C'est comparable au fait de confier une tâche à quelqu'un : au lieu de dire simplement « fais ceci », on précise « fais-le de cette manière, en respectant ces règles ».

Cette différence est également essentielle pour la machine. Des instructions claires, explicites et accompagnées d'un contexte précis facilitent la compréhension correcte du système.





4.4. Qu'est-ce qu'un prompt ?

L'importance de cette forme de communication vient du fait que l'intelligence artificielle ne « comprend » pas totalement le langage humain.

Elle évalue les mots à partir de calculs de probabilités. Autrement dit, le sens d'une phrase dépend de la probabilité qu'elle représente dans son modèle mathématique.

Ainsi, plus nos demandes sont vagues, plus les réponses du système risquent d'être dispersées ou imprécises.

C'est pourquoi un bon prompt est avant tout une expression claire de la pensée. L'objectif doit être clairement défini, le contexte précisé et, si nécessaire, des exemples doivent être ajoutés.



4.4. Qu'est-ce qu'un prompt ?

Par exemple, dire :

« Propose une idée de projet éducatif pour enseigner les compétences numériques à des femmes défavorisées »
est beaucoup plus précis que simplement demander :
« Propose une idée de projet éducatif ».

Cette approche améliore non seulement la qualité des réponses, mais développe également notre manière de réfléchir.

En effet, rédiger un bon prompt signifie savoir exactement ce que l'on veut. Définir une idée nous oblige à réfléchir au résultat que nous souhaitons obtenir.

C'est l'essence même de la communication avec l'intelligence artificielle : plus nos instructions sont claires, ciblées et cohérentes, meilleurs seront les résultats.

L'écriture de prompts est donc moins une compétence technique qu'une compétence d'organisation de la pensée.





4.5. La logique de création de prompts efficaces



Cofinancé par
l'Union européenne

4.5. La logique de création de prompts efficaces

Exprimer correctement une idée est la première étape pour obtenir un bon résultat. Il en va de même lorsque l'on travaille avec des systèmes d'intelligence artificielle.

Même le système le plus avancé ne pourra produire des réponses de qualité que si les instructions fournies sont elles-mêmes de qualité. Rédiger un prompt efficace revient donc à clarifier sa pensée.

La première règle consiste à définir l'objectif. Que voulons-nous obtenir ? Un texte, une analyse, une suggestion ?

Connaître précisément le but facilite le choix des mots. Un objectif flou peut conduire le système dans une mauvaise direction.



4.5. La logique de création de prompts efficaces

Par exemple, au lieu de dire :

« Donne-moi une idée de projet »,

il est plus efficace de préciser :

« Crée une idée de projet visant à développer les compétences numériques des femmes défavorisées ».

Cette formulation clarifie à la fois le sujet et l'orientation de la demande.

La deuxième règle est de fournir du contexte. L'intelligence artificielle ne peut pas deviner naturellement la situation comme le ferait un humain.

Il est donc important de préciser l'environnement, le public cible, la langue ou encore le format attendu.

Des indications comme :

« Rédige une publication pour les réseaux sociaux destinée aux jeunes »

ou

« Crée un brouillon d'e-mail formel »

aident le modèle à adapter correctement le ton et la forme de sa réponse.



4.5. La logique de création de prompts efficaces

La troisième étape consiste à guider avec des exemples. Même lorsqu'une demande semble claire, les exemples aident à rendre une idée abstraite plus concrète.

Des indications comme « adopte un ton similaire à cet article » ou « génère des suggestions dans ce style de titre » orientent le système dans la direction souhaitée.

Un autre point important est de définir des limites. L'intelligence artificielle peut générer une infinité d'options, ce qui peut parfois entraîner des détails inutiles.

Des consignes comme « écris sous forme de puces », « limite la réponse à 200 mots » ou « n'utilise pas de termes techniques » permettent de garder le focus.

Enfin, il est préférable d'avancer étape par étape. Plutôt que de donner une demande longue et complexe en une seule fois, guider le système progressivement facilite le travail pour l'utilisateur comme pour l'IA.

Obtenir d'abord une idée générale, puis demander des détails, donne souvent des résultats plus cohérents.

Un bon prompt commence par les bonnes informations, mais se termine par une bonne communication. Travailler avec l'intelligence artificielle ne consiste pas seulement à donner du code ou des commandes ; c'est une manière d'exprimer ses idées clairement, simplement et avec un objectif précis.



4.5. La logique de création de prompts efficaces

GPT-5 P.R.O.M.P.T. FRAMEWORK

You are a senior AI business strategist who helps SaaS founders design GTM strategies using AI agents and automation tools. Goal: produce a 90-day action plan to launch and scale a new SaaS product.

Allowed tools: internal CRM data + GPT-5 search; no web browsing.

Reasoning effort: high for deep market research.

Done = a complete, prioritized roadmap with timelines and KPIs.

You are acting as a hybrid business coach and AI workflow architect. You can design automation flows, recommend tools, and adjust strategies for different funding stages. Follow tool rules strictly; no speculative data.

Create a 3-step plan before doing anything (Research → Draft → Review).

Execute each step in sequence. Provide a short "Done" checklist confirming all deliverables. Keep going until the plan is fully complete.

Deliver output in Markdown with:

- Section headings for each quarter milestone.
- Bullet lists for tasks.
- Tables for KPIs, timelines, and resources.
- Restate formatting every 3–5 turns if the conversation is long.

Tone: confident, encouraging, and precise.

Verbosity: medium for plans, high detail for KPI breakdowns.

Mix strategic vision with clear, actionable steps.

Cap tool calls at 3 per turn. If a data lookup fails, retry once, then proceed with available info. Always verify facts before including them.

Purpose + Role + Order of Action + Mould the Format

Personality + Tight Controls

1. Objectif : Que souhaitez-vous accomplir ?
2. Rôle : Dans quel rôle l'IA doit-elle agir ? Expert, mentor, analyste ?
3. Ordre des actions : Que faut-il faire en premier et que faut-il vérifier ensuite ?
4. Format de réponse : Sous quelle forme la réponse doit-elle être présentée ? Liste à puces, tableau, texte narratif ?
5. Personnalité : Quel ton doit être utilisé ? Formel, strict, créatif ?
6. Contraintes : Quelles sont les limites ? Que doit-elle éviter ? À quoi doit-elle faire attention ?





4.6. Comment l'IA interprète les prompts ?



Cofinancé par
l'Union européenne

4.6. Comment l'IA interprète les prompts ?

Chaque prompt écrit pour un système d'intelligence artificielle est en réalité transformé en une série de probabilités numériques dans l'univers du modèle.

Ce que nous percevons comme du « sens » correspond, pour l'IA, à un réseau de relations entre les mots. Le système calcule à quel point un mot est lié à un autre dans une phrase, quel sens il évoque et dans quel contexte il est utilisé. Même si ce processus paraît complexe, il reproduit mathématiquement quelque chose que l'esprit humain fait intuitivement.

Comme les modèles d'IA sont entraînés sur d'immenses quantités de textes et d'informations, ils apprennent à prédire les significations possibles d'un mot ou d'une expression.

Par exemple, lorsque le mot « chat » apparaît, le système ne pense pas seulement à l'animal lui-même, mais aussi à des milliers de concepts associés comme la maison, les soins, les miaulements ou la tendresse.

Ainsi, lorsqu'il génère une réponse, il ne se base pas uniquement sur le mot, mais aussi sur tout son champ sémantique.



4.6. Comment l'IA interprète les prompts ?

Lorsqu'un prompt est reçu, le système cherche d'abord à comprendre le contexte : s'agit-il d'une question, d'une tâche ou d'une demande d'explication ?

Il analyse ensuite la structure de la phrase, identifie les significations possibles et évalue les probabilités de réponse pour chacune d'elles.

La combinaison ayant la probabilité la plus élevée devient la « réponse » du système. C'est pourquoi une petite différence dans la formulation peut parfois entraîner un changement important dans le résultat obtenu.

Les indications concernant le ton et le style influencent également directement la manière dont le modèle répond.

Par exemple, lorsqu'on demande « explique avec un ton amical », le système privilégie des mots plus simples et plus conversationnels.

À l'inverse, lorsqu'on demande « utilise un style académique », il favorise des phrases plus longues et plus formelles.

Ainsi, un prompt ne transmet pas seulement une information ; il transmet aussi un signal de style.

Cependant, chaque modèle possède ses limites. L'intelligence artificielle ne ressent pas l'« intention » ou les « émotions » comme un être humain.

Elle reproduit simplement des schémas observés dans des textes similaires. Derrière chaque réponse se trouve donc un calcul de probabilités et un processus de prédiction.

C'est finalement l'utilisateur qui interprète ces prédictions.





4.7. Pourquoi est-il important de rédiger de bons prompts ?



Cofinancé par
l'Union européenne

4.7. Pourquoi est-il important de rédiger de bons prompts ?

La qualité des résultats obtenus avec les systèmes d'intelligence artificielle dépend en grande partie de la manière dont la question est formulée.

Rédiger un bon prompt est important non seulement pour obtenir la bonne réponse, mais aussi pour apprendre à penser de manière claire et structurée. Une formulation précise détermine à la fois ce que nous voulons et la façon dont le système va répondre.

La différence la plus fondamentale concerne l'efficacité. Un prompt ambigu gaspille la capacité prédictive du modèle, tandis qu'une instruction claire et ciblée accélère le processus.

Par exemple, il existe une grande différence entre :

« Donne-moi une idée de projet » et...



4.7. Pourquoi est-il important de rédiger de bons prompts ?

« Propose trois idées de projets visant à développer les compétences numériques des femmes défavorisées ».

La deuxième formulation définit à la fois l'objectif et les limites de la demande, ce qui aide le système à se concentrer sur le bon sujet.

Un autre avantage est la précision et la cohérence. Un prompt bien rédigé permet au modèle de produire des réponses plus stables et plus exploitables à chaque tentative.

Cet aspect est particulièrement important dans le travail en équipe. Lorsque différentes personnes obtiennent des résultats similaires autour d'un même objectif, cela garantit une meilleure cohérence et un standard de qualité commun.



4.7. Pourquoi est-il important de rédiger de bons prompts ?

De plus, un bon prompt accélère le processus d'apprentissage.

Les utilisateurs qui apprennent à formuler clairement leurs questions développent également leur manière de penser. Chaque instruction bien construite devient une occasion d'observer le comportement du système et de mieux le guider. Avec le temps, les utilisateurs commencent à communiquer plus efficacement non seulement avec l'intelligence artificielle, mais aussi avec leurs propres idées.

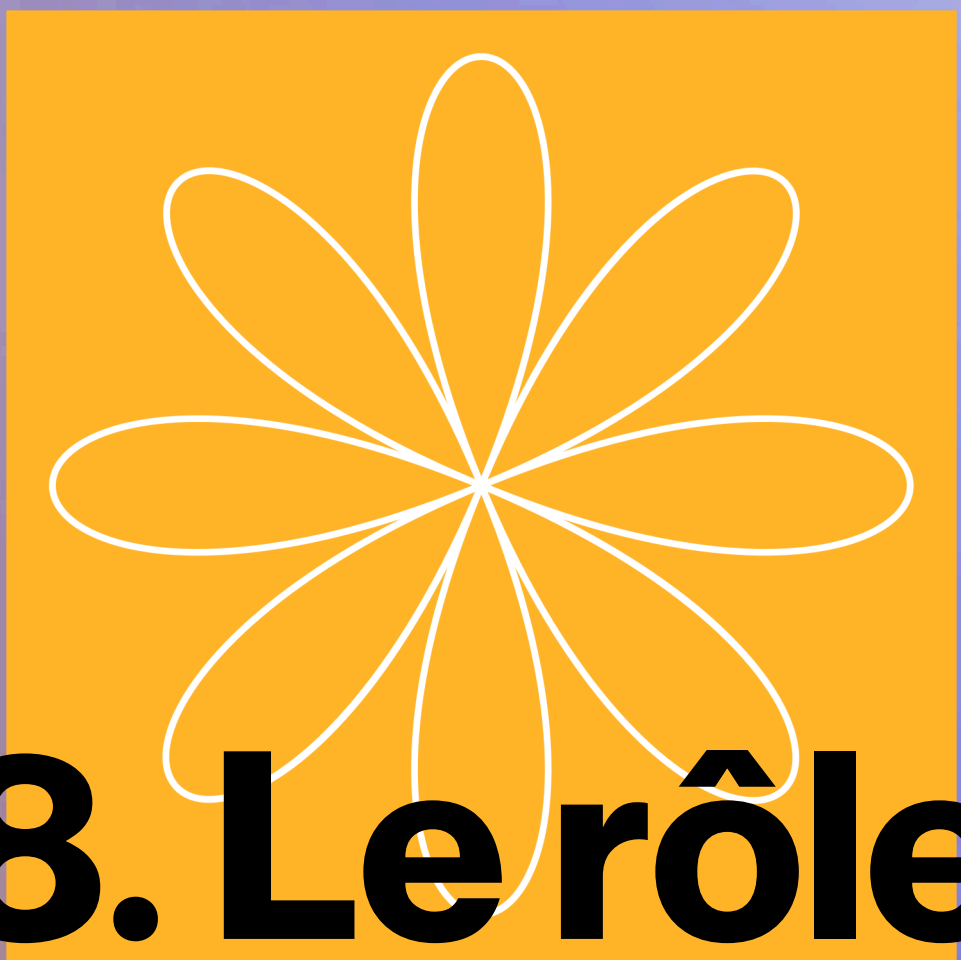
Un bon prompt permet également de gagner du temps et des ressources. Il réduit les explications inutiles, les erreurs répétitives et les résultats non pertinents.

Cette différence est particulièrement importante dans les domaines où le temps de production est limité, comme la rédaction de projets, la création de contenu ou la conception pédagogique.

En résumé, un bon prompt transforme non seulement la machine, mais aussi l'utilisateur. Il apprend à organiser ses pensées, à se concentrer sur l'objectif et à exprimer ses idées avec un langage clair et simple.

Cette compétence constitue une nouvelle forme de littératie à l'ère de l'intelligence artificielle : l'art de clarifier sa pensée.





4.8. Le rôle du Prompt Engineering dans le futur



Cofinancé par
l'Union européenne

4.8. Le rôle du Prompt Engineering dans le futur

À mesure que la technologie progresse, la relation entre les humains et les machines évolue également. Autrefois, les personnes contrôlaient les machines à l'aide de lignes de commande ; aujourd'hui, nous communiquons avec elles en langage naturel.

Au cœur de cette transformation se trouve le prompt engineering, c'est-à-dire l'art de concevoir des prompts efficaces. Dans un avenir proche, cette compétence deviendra aussi essentielle que savoir écrire ou préparer une présentation.

Dans le monde de production de demain, la question « comment écrire du code ? » sera progressivement remplacée par « comment donner le bon prompt ? ».

En effet, l'intelligence artificielle n'est plus seulement un outil, mais un véritable partenaire de production. Un prompt bien défini peut servir de base à des logiciels complexes ou à des contenus créatifs.

Exprimer une idée en quelques phrases peut réduire à quelques minutes des tâches qui prenaient auparavant des heures. Ce changement sera particulièrement visible dans l'éducation. Les étudiants apprendront moins à mémoriser qu'à exprimer clairement leurs idées, définir correctement les problèmes et guider consciemment les systèmes.

Pour un enseignant, un étudiant ou un entrepreneur, la compétence la plus précieuse sera désormais de « savoir poser les bonnes questions ».

Car les bonnes questions ont toujours plus de valeur que les bonnes réponses.



4.8. Le rôle du Prompt Engineering dans le futur

Dans le monde professionnel, le prompt engineering redéfinira la communication et la production. Les rapports, analyses, contenus et stratégies ne seront plus créés uniquement par des experts, mais par toute personne sachant formuler les bonnes instructions.

Cela augmentera la productivité tout en démocratisant l'accès à l'information.

Les futurs systèmes d'intelligence artificielle évolueront vers des structures capables de comprendre non seulement des commandes, mais aussi des intentions.

Les prompts multimodaux, combinant images, sons et mouvements, pourront transmettre non seulement du langage, mais aussi des émotions et du contexte.

La communication avec l'intelligence artificielle passera ainsi d'un simple échange de texte à une collaboration multidimensionnelle.

Toutes ces évolutions rendront le rôle de l'humain encore plus important, au lieu de le faire disparaître.

Car, quelle que soit l'intelligence des systèmes, le guidage restera entre les mains des humains. C'est à nous de décider quelles questions poser, quelles limites fixer et quelles valeurs préserver.

En résumé, le prompt engineering ne sera pas seulement une compétence technologique du futur ; il deviendra un nouveau langage de la pensée.

Le pouvoir des mots déterminera la direction des idées, et une idée exprimée correctement pourra être concrétisée de la bonne manière.





4.9. Résumé



Cofinancé par
l'Union européenne

4.9. Résumé

Les algorithmes et les prompts sont deux outils fondamentaux de réflexion à l'ère de l'intelligence artificielle.

L'un explique comment le système fonctionne, tandis que l'autre explique comment le guider. Lorsque ces deux compétences sont combinées, les individus ne deviennent plus seulement utilisateurs de la technologie, mais aussi acteurs capables de la façonner.

La pensée algorithmique nous apprend à résoudre les problèmes étape par étape. Elle permet de diviser des sujets complexes en éléments simples et compréhensibles, réduisant ainsi les erreurs et améliorant l'efficacité.

Cette approche ne s'applique pas uniquement au développement logiciel, mais aussi à de nombreuses situations du quotidien, comme la planification, la prise de décision ou l'élaboration de stratégies.

Le prompt engineering, quant à lui, transforme cette manière de penser en communication. Il permet d'exprimer ses idées de façon claire, cohérente et orientée vers un objectif précis.

Car l'essence du travail avec l'intelligence artificielle repose sur une communication claire et des instructions précises.

Un prompt bien conçu réduit la complexité, améliore la productivité et affine notre réflexion.

À l'avenir, la combinaison de ces deux domaines sera au cœur des compétences numériques. D'un côté se trouvent les algorithmes, c'est-à-dire la pensée logique et systématique ; de l'autre, les prompts, c'est-à-dire la capacité à utiliser efficacement le langage et l'intention.

Ces compétences rendent la technologie accessible à tous, démocratisent la production et permettent aux individus de concrétiser leurs idées dans le monde numérique.





EMPRESÁRIOS
PELA INCLUSÃO SOCIAL

ASSOCIAÇÃO PAREDES
PELA INCLUSÃO SOCIAL



SDSN

Sustainable
Development
Studies Network

Partenaires