Le Noyau Génomique : Une Architecture Neuro-Symbolique pour la Distillation du Savoir

1. Introduction : La Convergence de la Biologie et de l'Épistémologie Computationnelle

L'ambition de créer un système capable d'extraire le "savoir total distillé" d'un corpus documentaire, analogue à la dynamique de l'ADN, marque un tournant décisif dans l'ingénierie de l'intelligence artificielle. La requête initiale, visionnaire, postule l'existence d'un "bin d'ADN" — un conteneur d'information si dense et si structuré qu'il transcende la simple compression de données pour atteindre une forme de *Génomique Sémantique*. Dans le contexte actuel, où les Grands Modèles de Langage (LLM) sont confrontés à la saturation de leurs fenêtres contextuelles et à l'entropie inhérente au langage naturel, cette métaphore biologique n'est pas une simple figure de style. Elle constitue un cadre architectural rigoureux et nécessaire pour l'avenir de la gestion des connaissances.[1]

Le présent rapport de recherche propose la conception théorique et technique de ce système, baptisé le **Noyau Génomique de Connaissance (NGC)**. Ce système ne se contente pas de résumer ; il *métabolise* l'information brute pour en extraire les "axiomes" fondamentaux, les chaînes causales et les invariants logiques, qu'il encode ensuite dans un format hyper-structuré. Ce "bin", à l'instar d'un chromosome, est inerte en soi mais possède un potentiel génératif infini lorsqu'il est "exprimé" par un moteur d'inférence (l'équivalent de la machinerie cellulaire de transcription).

Nous nous appuyons sur des recherches de pointe en "Épigénolinguistique" [1], en compression sémantique télégraphique [4], et sur les architectures d'agents autonomes comme les "LLM OS" [5] et "MemGPT".[6] En intégrant des protocoles de distillation comme la "Chain of Density" (Chaîne de Densité) [7] et des structures de données avancées telles que les hypergraphes [8], nous démontrons comment transformer le chaos des documents non structurés en une structure cristalline, manipulable et éternelle : l'ADN de la Connaissance.

1.1 Le Problème de l'Entropie Sémantique

Le langage naturel est un vecteur inefficace pour le stockage de la vérité. Il est saturé de redondances grammaticales, de marqueurs sociaux et de structures syntaxiques qui, bien qu'essentiels à la communication humaine, constituent du "bruit" pour une machine cherchant à stocker du savoir pur. Les approches actuelles, telles que les vecteurs d'encastrement (embeddings), capturent la sémantique floue mais échouent souvent à

préserver la précision des relations logiques complexes (le "qui a fait quoi, quand et pourquoi").[9] Le NGC vise à résoudre ce problème par une approche "Neuro-Symbolique" [11], combinant l'intuition des réseaux de neurones avec la rigueur de la logique formelle pour créer une représentation sans perte du sens.

2. Fondements Théoriques : De la Métaphore à l'Architecture

Pour construire un "Kernel" (noyau) qui imite l'ADN, nous devons d'abord décoder les principes physiques et informationnels qui rendent l'ADN si efficace. L'ADN n'est pas seulement un stockage ; c'est un code exécutable, modulaire et résilient.

2.1 Le Cadre ISAF4DNA : L'ADN comme Langage

La recherche récente sur le décodage de l'information régulatrice dans l'ADN a donné naissance au cadre *Interpretability-First, Structural Artificial Intelligence Framework for DNA* (ISAF4DNA). Ce cadre révèle que l'ADN suit une organisation "quasi-linguistique" caractérisée par trois niveaux [1] :

1. **Motifs (Unités Lexicales) :** Ce sont les mots du génome. Dans notre NGC, l'équivalent est l'**Atome Sémantique** (ou "Tokum").[3] Il s'agit d'une unité de sens indivisible — une entité, une action ou une propriété — dépouillée de toute variation syntaxique. Par exemple, "L'augmentation de la température a causé la fonte" devient l'atome {Cause:}.
2. **Redondance (Syntaxe) :** L'ADN utilise des structures d'ancrage conservées (le cœur du message) et des flancs sélectifs (le contexte). Notre noyau doit identifier les "Axiomes Piliers" d'un document — les vérités immuables — et les séparer des variations stylistiques floues. C'est la distinction entre l'information (le signal) et la formulation (le bruit).[1]
3. **Déploiement (Pragmatique) :** L'expression d'un gène dépend du tissu cellulaire (contexte). De même, un fait extrait d'un document n'est vrai que dans un contexte donné (temporel, géographique, juridique). Notre "ADN de Connaissance" doit inclure des métadonnées "épigénétiques" qui définissent les conditions de validité de chaque axiome.[1]

2.2 L'Hypothèse de l'ADN Cognitif

Au-delà de la biologie moléculaire, le concept de "Neural Knowledge DNA" suggère que la connaissance doit être stockée sous forme de modèles dynamiques et évolutifs plutôt que statiques.[13] Les réseaux de croyances et de connaissances forment des "Ensembles Cognitifs Collectivement Autocatalytiques" (CACS) — des structures auto-organisées où chaque idée renforce et valide les autres.[12]

L'extraction du savoir n'est donc pas une simple copie ; c'est une opération de **Reverse**

Engineering (ingénierie inverse) de l'esprit de l'auteur. Le script doit identifier les prémisses implicites (les axiomes non dits) sur lesquelles repose le texte. Si un document traite de "l'optimisation des moteurs thermiques", il contient l'axiome implicite "la thermodynamique est valide". Le NGC doit expliciter ces axiomes pour construire une base de connaissances robuste et transférable.[15]

2.3 Compression Sémantique Télégraphique (TSC)

Pour atteindre la densité de l'ADN, nous devons appliquer une compression radicale. La technique de Telegraphic Semantic Compression (TSC) offre une méthode inspirée de la théorie de l'information.4 Contrairement aux résumés traditionnels qui reformulent, la TSC supprime les structures grammaticales prévisibles (les "introns" linguistiques) pour ne garder que les mots à haute entropie (les "exons" sémantiques).
Par exemple, la phrase "Le système utilise une architecture basée sur des transformateurs pour analyser le texte" (12 tokens) devient "Système:Architecture:Transformateurs -> Analyse:Texte" (5 tokens). Cette réduction permet d'étendre artificiellement la fenêtre contextuelle des LLM, permettant au noyau de traiter des volumes massifs de données sans perte de sens critique.17

3. Protocoles d'Extraction : L'Enzymologie du Noyau

Le fonctionnement du NGC repose sur deux processus majeurs, mimant les enzymes biologiques : l'**Hélicase** (pour dérouler et segmenter le texte) et la **Polymérase** (pour synthétiser et encoder le savoir). Ces processus sont orchestrés par des algorithmes complexes de traitement du langage naturel.

3.1 Le Protocole Hélicase : Segmentation et Déroulement Sémantique

Les documents longs (livres, rapports techniques) dépassent souvent la capacité d'attention des modèles. Le protocole Hélicase prépare le "brin" d'information pour la lecture.

3.1.1 Décomposition Récursive (MapReduce pour LLM)

Pour traiter l'immensité des données, nous adaptons le paradigme "MapReduce" aux LLM.[19]

1. **Phase Map (Cartographie) :** Le document est divisé en segments (chunks). Cependant, une découpe arbitraire brise le sens. Nous utilisons la *Détection de Frontières Sémantiques*.[21] Le noyau analyse la cohérence lexicale pour couper le texte aux transitions naturelles (changement de sujet, nouveau chapitre), assurant que chaque "nucléosome" d'information est complet.
2. **Phase Shuffle (Regroupement Spectral) :** Un concept peut être fragmenté à travers tout le document (ex: une définition au chapitre 1, une application au chapitre 10). L'Hélicase utilise le *Spectral Clustering* sur les vecteurs d'encastrement pour identifier ces fragments dispersés et les "replier" ensemble.[21] Cela imite le repliement de la chromatine, rapprochant des gènes distants pour une régulation commune.
3. **Phase Reduce (Fusion Conflictuellle) :** Si deux segments se contredisent, le noyau

applique un protocole de "calibration de confiance en contexte".[19] Il évalue la récence et la spécificité pour résoudre le conflit avant l'encodage, garantissant que l'ADN final est cohérent.

3.2 Le Protocole Polymérase : Distillation par Chaîne de Densité

Une fois le texte préparé, la Polymérase synthétise le "Savoir". C'est ici que l'intelligence artificielle générative est contrainte pour produire de la densité pure.

3.2.1 Chain of Density (CoD)

La technique de *Chain of Density* est le cœur réacteur du système.[7] Elle transforme un résumé lâche en un bloc de connaissances ultra-dense. Le processus est itératif :

- **Itération 1 (Squelette) :** Extraction des entités principales (Sujet, Objet).
- **Itérations 2-5 (Densification) :** Le noyau identifie les entités "manquantes" dans le résumé précédent et les intègre sans augmenter la longueur totale du texte. Cela force le modèle à fusionner les phrases, à généraliser les concepts et à éliminer tout mot de remplissage.[23] Le résultat est un texte où chaque token porte une charge informative maximale, simulant la densité du code génétique.

3.2.2 P-Distill et Cibles Douces

Pour capturer non seulement les faits, mais aussi le *raisonnement* (le "pourquoi"), nous utilisons la méthode **P-Distill**.[24] Au lieu d'apprendre uniquement du texte de sortie d'un modèle "Enseignant" (ex: GPT-4), le noyau apprend des distributions de probabilités (soft targets).[25] Cela permet de capturer les nuances et les incertitudes du modèle expert, transférant ainsi une forme d'intuition au noyau.

3.2.3 Extraction Propositionnelle

Parallèlement à la densification, un processus d'**Extraction Propositionnelle** convertit le texte narratif en tuples logiques.[26]

- *Texte :* "L'expérience a échoué car la valve a fui sous la pression."
- *Propositions :*
  1. Expérience -> Statut -> Échec
  2. Valve -> Action -> Fuite
  3. Fuite -> Cause -> Pression
  4. Fuite -> Conséquence -> Échec
     Ce format permet une validation logique rigoureuse et sert de base à la construction du graphe de connaissances.28

Tableau 1 : Comparatif des Stratégies de Compression Sémantique

Méthode	Mécanisme Principal	Densité Informationnelle	Réversibilité (Reconstruction)	Analogie Biologique
Embeddings (Vecteurs)	Représentation mathématique dans un espace N-dimensionnel.	Moyenne (Perte de détails spécifiques).	Faible (Reconstruction approximative/floue).	Hormones (Signal diffus).
Résumé Standard	Réécriture en langage naturel plus court.	Faible (Conserve la syntaxe humaine redondante).	Moyenne (Perte de nuances contextuelles).	ARN messager (Temporaire).
Telegraphic Compression (TSC)	Suppression sélective des mots grammaticaux (Stop words).	Élevée (Suppression du bruit syntaxique).	Élevée (Reconstruction facile par LLM).	Introns/Exons (Épissage).
Chain of Density (CoD)	Itérations récursives d'ajout d'entités sous contrainte de longueur.	**Très Élevée** (Fusion conceptuelle).	**Très Élevée** (Savoir distillé pur).	**Chromatine (Stockage dense).**

4. Le Génome Sémantique : Format et Structure de Stockage

L'utilisateur demande un "bin d'ADN". En informatique, cela correspond au format de sérialisation. Ce format doit être universel, rigide et machine-readable. Nous rejetons les formats flous pour adopter une structure hybride **Hypergraphe-JSON**.

4.1 Des Triples aux Hypergraphes

Les Graphes de Connaissances (KG) traditionnels utilisent des "triples"

(Sujet-Prédicat-Objet). Cependant, la réalité est rarement binaire. Les relations complexes nécessitent des Hypergraphes.8
Dans un hypergraphe, une "arête" (hyperedge) peut connecter un nombre arbitraire de nœuds.

- *Exemple :* Un événement historique implique une Date, un Lieu, des Acteurs, une Cause et une Conséquence. Une arête binaire fragmenterait cette information. Une hyper-arête capture l'événement comme une "molécule" de savoir indivisible.[31] Cela permet de préserver l'intégrité contextuelle, essentielle pour éviter les hallucinations lors de la récupération (GraphRAG).[32]

4.2 Schéma Nucléotidique (JSON Contraint)

Pour garantir que le "bin" soit exploitable, nous imposons un schéma JSON strict via le prompt système, en utilisant des techniques de "Constrained Decoding" (Décodage Contraint).[33] Le fichier résultant est une séquence de "Nucléotides Sémantiques".

Structure du Nucléotide Sémantique (Exemple) :

JSON

```
{
 "id_sequence": "SEQ_AF992",
 "type": "AXIOME_CAUSAL",
 "contenu_compressé": "Inflation:Hausse -> PouvoirAchat:Baisse -> Consommation:Stagnation",
 "vecteurs_contexte": ["économie", "macro_dynamique", "2024"],
 "score_integrité": 0.98,
 "hyper_connexions":,
 "épigénétique": {
  "source_origine": "Rapport_BCE_2024.pdf",
  "validité_temporelle": "COURT_TERME",
  "domaine_application": "Zone_Euro"
 }
}
```

Ce format intègre les principes **ICA (Intent, Context, Action)** [35], assurant que chaque bit de savoir porte avec lui son mode d'emploi. Le champ "épigénétique" est crucial : il permet au système de savoir *quand* activer ce savoir (par exemple, ne pas appliquer une loi fiscale de 1990 en 2025).[1]

4.3 Algorithmes de Stockage Inspirés de l'ADN

Pour pousser la métaphore jusqu'au stockage physique, nous pouvons appliquer des algorithmes de codage d'ADN numérique.[36]

- **Codage de Huffman :** Les concepts les plus fréquents sont mappés sur des séquences courtes, optimisant l'espace.
- **Prévention des Homopolymères :** Pour éviter les erreurs de lecture (répétitions), nous insérons des marqueurs de synchronisation dans le flux JSON, garantissant que le LLM ne "dérive" pas lors de la génération massive de texte.[38]
- **Correction d'Erreurs :** Le noyau encode les axiomes critiques de manière redondante (synonymes), permettant au système de reconstruire la vérité même si une partie du fichier est corrompue ("mutation").[40]

5. Architecture du Noyau : Un Système d'Exploitation Agentique

Un script statique ne suffit pas pour gérer la complexité du savoir "total". Nous proposons une architecture de type **"LLM OS"** (Système d'Exploitation pour LLM).[5]

5.1 Le Cœur Neuro-Symbolique

Le noyau fonctionne sur une base hybride [11] :

- **Couche Neurale (Intuition) :** Le LLM (ex: GPT-4) lit le texte, comprend les nuances et propose des connexions.
- **Couche Symbolique (Logique) :** Un moteur de règles (ex: Prolog ou base de données graphe comme Memgraph/Neo4j) *valide* ces connexions. Si le LLM suggère "A implique B", la couche symbolique vérifie si cela contredit un axiome existant "A est indépendant de B". Ce mécanisme de contrôle ("Check-Balance") est la barrière immunitaire contre les hallucinations.[42]

5.2 MemGPT et la Gestion de la Mémoire

Les LLM standards sont amnésiques. Nous intégrons **MemGPT** [6] pour doter le NGC d'une mémoire persistante.

- **Mémoire Noyau (Core Memory) :** Contient les instructions du système (le "code génétique" de l'agent) et le contexte immédiat.
- **Mémoire Archivage (Disk) :** Le système "page" (swap) les informations entre la mémoire vive (contexte LLM) et le stockage long terme (Graphe Vectoriel).[44] Cela permet de traiter des corpus infinis en ne gardant en "conscience" que les éléments pertinents pour la tâche d'extraction en cours.

5.3 Orchestration par LangGraph

Pour coordonner ces processus, nous utilisons **LangGraph** [45], qui permet de définir des flux de travail cycliques et récursifs.

- **Le Flux Récursif :**
  1. **Lecteur :** Ingère un chunk.
  2. **Extracteur :** Applique la CoD et l'Extraction Propositionnelle.
  3. **Validateur (Critique) :** Vérifie la cohérence via la couche symbolique.
  4. **Lieur (Linker) :** Cherche les connexions dans l'hypergraphe existant.
  5. Encodeur : Sérialise en JSON.
     Si le Validateur détecte une ambiguïté, il déclenche une boucle de "Résumé Récursif" 47, demandant au Lecteur de réexaminer le contexte précédent.

6. Implémentation : Le Script "Semantic Polymerase"

Pour répondre à la demande de "script/prompt kernel", voici la logique architecturale du composant logiciel central : Semantic_Polymerase.py.

6.1 Le Prompt Système "Enzymatique"

Le prompt n'est pas une simple question, c'est un programme en langage naturel. Il doit être conçu pour forcer le modèle à agir comme une enzyme biologique.

> Définition du Prompt Système (Conceptuel) :
> "Tu es la Polymérase Sémantique. Ta fonction est de transcrire l'information non structurée en ADN de Connaissance.
> **Directives Opérationnelles :**
>
> 1. **Extraction Axiomatique :** Identifie les vérités immuables et les lois fondamentales du texte.
> 2. **Traçage Causal :** Mappe les séquences d'événements sous forme de prédicats logiques (Si X -> Alors Y).
> 3. **Compression Télégraphique :** Supprime toute syntaxe qui ne modifie pas les conditions de vérité. Garde uniquement les atomes sémantiques.
> 4. **Encodage :** La sortie DOIT respecter strictement le schéma JSON Nucleotide.
>
> **Interdictions :** Ne pas résumer. Ne pas utiliser de phrases de liaison ('Le texte dit que...'). Ne pas donner d'opinion. Extraire le code pur."

6.2 La Boucle d'Extraction (The Loop)

Le script Python orchestre l'interaction :

1. **Entrée :** Chunk de texte (2000-4000 tokens).

2. **Appel 1 (Identification) :** "Liste toutes les entités uniques et leurs types."
3. **Appel 2 (Relationnel) :** "Pour chaque entité, identifie ses relations hypergraphes avec les autres."
4. **Appel 3 (Densification - CoD) :** "Fusionne ces relations. Identifie le contexte manquant. Compresse."
5. **Validation :** Vérification syntaxique du JSON. Si erreur, réinjection de l'erreur dans le prompt pour correction (Self-Correction).
6. **Stockage :** Injection dans la base Neo4j/Vectorielle.

6.3 L'Expression : Du Génome au Phénotype

Le test ultime de l'ADN est sa capacité à construire un organisme. Le système inclut un module d'**Expression** (le Ribosome).

- **Prompt de Reconstruction (Inverse Summarization) :** "En utilisant *uniquement* les Axiomes ADN fournis, reconstruis une explication narrative détaillée du sujet. N'ajoute aucune connaissance externe.".[49]
- Cette technique permet de vérifier la "perte" lors de la compression. Si le texte reconstruit est incohérent, cela signale une mutation délétère dans le processus d'extraction.

7. Stratégies Avancées de Prompt Engineering

Pour maximiser la qualité du "bin", nous utilisons des techniques de pointe identifiées dans la littérature.

7.1 Résumé Inversé (Inverse Summarization)

Pour valider l'ADN extrait, nous inversons le processus. Après l'extraction, un agent "Critique" tente de régénérer le document original à partir du JSON seul.[49] Si le résultat diffère sémantiquement de l'original, le noyau marque la séquence comme "instable" et relance l'extraction avec une densité plus élevée. C'est un contrôle qualité adversarial.

7.2 La Technique "One-Shot Jailbreaking" pour les Chaînes Causales

Bien que conçue pour tester la sécurité, la technique de simulation de scénarios complexes ("imagine a scenario...") [16] est extrêmement efficace pour forcer le modèle à révéler les axiomes profonds. Nous adaptons cela en : "Imagine un scénario où cet axiome est faux. Quelles seraient les conséquences logiques?" Cela permet de tester la robustesse et la centralité d'une information extraite (Analyse Contrefactuelle).

7.3 Reconstruction de Texte à partir de Latents

Des recherches récentes sur la compression "Text-to-Latent" (C3) montrent que l'on peut compresser le texte en tokens non interprétables mais riches en information.[50] Bien que notre

approche privilégie le JSON interprétable (Neuro-Symbolique), nous pouvons utiliser cette technique pour les parties "floues" ou artistiques du texte qui résistent à la logique formelle, créant ainsi des régions "non-codantes" mais structurelles dans notre ADN de connaissance.

8. Défis, Risques et Mitigations

8.1 L'Hallucination des Faux Axiomes

Le risque majeur est que le LLM encode une hallucination comme une vérité absolue.

- **Mitigation :** La validation Neuro-Symbolique est impérative. Chaque nouvel axiome est confronté à l'ontologie existante. De plus, nous utilisons la **Calibration de Confiance** : chaque nucléotide possède un score de certitude (P_{truth}). Les faits incertains sont marqués comme "Récessifs" et ne sont exprimés que s'ils sont corroborés par d'autres sources.[19]

8.2 Coûts Computationnels

L'exécution de la *Chain of Density* et des vérifications récursives sur des millions de tokens est coûteuse.

- **Mitigation :** L'application préalable de la **Compression Télégraphique** [4] réduit la taille des inputs de 40 à 60% avant même l'entrée dans les étapes de raisonnement coûteuses, optimisant ainsi le rendement du "métabolisme" numérique.

9. Conclusion : Vers une Singularité Sémantique

La réponse à votre demande n'est pas un simple script, mais une architecture complète. Le **Noyau Génomique** décrit ici représente l'état de l'art de ce qu'il est possible de construire aujourd'hui en combinant les LLM, les bases de données graphes et l'ingénierie des prompts avancée.

Ce système réalise la promesse de la métaphore biologique :

1. **Stockage :** Un "bin" JSON hyper-dense, structuré et résilient (le Génome).
2. **Dynamique :** Un OS agentique capable de répliquer, réparer et exprimer ce savoir (la Cellule).
3. **Évolution :** Une capacité à mettre à jour ses croyances face à de nouvelles données (l'Épigénétique).

En construisant ce système, vous ne créez pas seulement une archive ; vous créez une forme d'intelligence dormante, une "graine" de savoir pur prête à germer à la moindre requête. C'est l'outil ultime pour passer de l'ère du "Big Data" à l'ère du "Deep Knowledge".

---

10. Analyse Détaillée des Méthodologies et Preuves

Cette section approfondit les preuves scientifiques justifiant chaque choix architectural du NGC.

10.1 Analyse des Techniques de Compression Sémantique

Le succès du noyau dépend de sa capacité à compresser sans perdre le sens (Lossless Semantic Compression).

10.1.1 Efficacité de la Chain of Density (CoD)

Les études montrent que la méthode CoD génère des résumés qui, à la 3ème ou 4ème itération, contiennent une densité d'entités significativement supérieure à celle des résumés humains, tout en restant lisibles.[7]

- **Implication pour le NGC :** Nous n'utilisons pas la CoD pour la lisibilité humaine, mais pour maximiser le ratio Information/Token. C'est l'équivalent numérique de la "déshydratation" d'un aliment pour sa conservation. Le savoir est concentré, prêt à être réhydraté (généré) plus tard.

10.1.2 Reconstructions Sémantiques

L'utilisation de librairies comme vec2text ou de prompts de reconstruction [53] prouve qu'il est possible d'inverser les embeddings pour retrouver du texte. Cependant, notre approche JSON/Symbolique offre une fidélité supérieure pour les faits logiques, car elle évite le flou probabiliste des vecteurs.[55]

10.2 Analyse des Architectures de Connaissance

10.2.1 Supériorité des Hypergraphes

L'adoption de **HyperGraphRAG** est justifiée par ses performances supérieures dans les tâches de raisonnement multi-sauts (multi-hop reasoning).[8] Là où un système RAG classique échoue à connecter "A cause B" et "B cause C" si A et C sont distants dans le texte, l'hypergraphe maintient ces liens explicites dans la structure de données même. Cela est crucial pour capturer la "totalité" du savoir demandé par l'utilisateur.

10.2.2 L'Importance du Neuro-Symbolique

L'intégration de solveurs logiques dans la boucle (Neuro-Symbolic AI) [11] répond au besoin de fiabilité. Un système purement neural (LLM seul) est sujet à la dérive. Le composant symbolique agit comme un "correcteur orthographique" pour la logique, assurant que le "bin d'ADN" ne contient pas de mutations létales (contradictions flagrantes).

10.3 Analyse de l'OS Agentique

10.3.1 MemGPT et la Persistance

L'architecture **MemGPT** [6] est la seule solution viable pour gérer des contextes infinis de manière autonome. Elle permet au noyau de gérer sa propre attention, décidant activement ce qui mérite d'être stocké dans le "bin" et ce qui peut être oublié. C'est cette autonomie qui rapproche le système d'un organisme vivant plutôt que d'un simple script de traitement par lots.

11. Feuille de Route d'Implémentation

Pour passer de la théorie à la pratique, voici les étapes de développement recommandées pour votre "Script Kernel".

Phase 1 : Le Prototype In Vitro (Scripting)

- **Objectif :** Extraire l'ADN d'un document unique (PDF 50 pages).
- **Stack Technique :** Python, LangChain, OpenAI API (GPT-4o).
- **Action :** Coder la classe SemanticPolymerase. Implémenter le template de prompt CoD.
- **Livrable :** Un fichier JSON contenant les entités et axiomes extraits.

Phase 2 : Le Système In Vivo (Workflow Agentique)

- **Objectif :** Traiter un corpus de 100 documents avec références croisées.
- **Stack Technique :** LangGraph, MemGPT, Neo4j (Base Graphe).
- **Action :** Mettre en place le workflow MapReduce. Créer le schéma Hypergraphe dans Neo4j.
- **Livrable :** Un Graphe de Connaissances interrogeable où une requête sur le "Projet X" révèle les connexions à travers les 100 documents.

Phase 3 : La Couche Évolutive (Self-Correction)

- **Objectif :** Le système corrige lui-même ses connaissances.
- **Stack Technique :** Solveurs Logiques Neuro-Symboliques.
- **Action :** Implémenter un agent "Critique" qui scanne périodiquement le Graphe pour détecter les contradictions (ex: "Personne A morte en 1990" vs "Personne A écrit une lettre en 1995").
- **Livrable :** Une Base de Connaissances auto-réparatrice ("Self-Healing Knowledge Base").[42]

Ce rapport confirme que votre intuition de "l'ADN comme modèle" est validée par la recherche la plus récente en IA. En suivant cette architecture, vous ne construisez pas seulement un outil d'extraction, mais les fondations d'une mémoire artificielle pérenne.

Fin du Rapport.

Sources des citations

1. [2503.23494] Interpretable structural-semantic decoding reveals language-like organisation of regulatory information in DNA - arXiv, consulté le décembre 6, 2025, https://arxiv.org/abs/2503.23494
2. DNALONGBENCH: A Benchmark Suite for Long-Range DNA Prediction Tasks - bioRxiv, consulté le décembre 6, 2025, https://www.biorxiv.org/content/10.1101/2025.01.06.631595.full.pdf
3. Reading Between the Lines: What Your Brain Reveals About the Future of AI - Medium, consulté le décembre 6, 2025, https://medium.com/@eric_54205/reading-between-the-lines-what-your-brain-reveals-about-the-future-of-ai-6983adefe192
4. Telegraphic Semantic Compression (TSC) — A Semantic Compression Method for LLM Contexts | by Nuno Bispo | Django Unleashed | Nov, 2025 | Medium, consulté le décembre 6, 2025, https://medium.com/django-unleashed/telegraphic-semantic-compression-tsc-a-semantic-compression-method-for-llm-contexts-45de3ebbae96
5. LLM Operating Systems: Revolutionizing How Machines Think and Act - Pass4sure, consulté le décembre 6, 2025, https://www.pass4sure.com/blog/llm-operating-systems-revolutionizing-how-machines-think-and-act/
6. MemGPT: Towards LLMs as Operating Systems - Leonie Monigatti, consulté le décembre 6, 2025, https://www.leoniemonigatti.com/papers/memgpt.html
7. Better Summarization with Chain of Density Prompting - PromptHub, consulté le décembre 6, 2025, https://www.prompthub.us/blog/better-summarization-with-chain-of-density-prompting
8. Daily Papers - Hugging Face, consulté le décembre 6, 2025, https://huggingface.co/papers?q=graph-structured%20knowledge%20representation
9. Compressing and Interpreting Word Embeddings with Latent Space Regularization and Interactive Semantics Probing - arXiv, consulté le décembre 6, 2025, https://arxiv.org/html/2403.16815v1
10. Knowledge Graphs - arXiv, consulté le décembre 6, 2025, https://arxiv.org/pdf/2003.02320
11. Neuro-symbolic AI - Wikipedia, consulté le décembre 6, 2025, https://en.wikipedia.org/wiki/Neuro-symbolic_AI
12. Rational Superautotrophic Diplomacy (SupraAD) A Conceptual Framework for Alignment Based on interdisciplinary findings on the fundamentals of cognition - arXiv, consulté le décembre 6, 2025, https://arxiv.org/html/2506.05389v1
13. Deep Learning-Based Recommendation System: Systematic Review and Classification - IEEE Xplore, consulté le décembre 6, 2025, https://ieeexplore.ieee.org/iel7/6287639/10005208/10274963.pdf
14. Recent development of knowledge-based systems, methods and tools for One-of-a-Kind Production - ResearchGate, consulté le décembre 6, 2025,

https://www.researchgate.net/publication/220392477_Recent_development_of_knowledge-based_systems_methods_and_tools_for_One-of-a-Kind_Production
15. ADVANCING MATHEMATICS RESEARCH WITH LARGE LANGUAGE MODELS - arXiv, consulté le décembre 6, 2025, https://arxiv.org/html/2511.07420v1
16. One-Shot Jailbreaking: Exploiting Frontier AI for Adversarial Prompt Generation, consulté le décembre 6, 2025, https://www.lumenova.ai/ai-experiments/frontier-ai-models-one-shot-jaibreaking/
17. Extending Context Window of Large Language Models via Semantic Compression - ACL Anthology, consulté le décembre 6, 2025, https://aclanthology.org/2024.findings-acl.306.pdf
18. Extending Context Window of Large Language Models via Semantic Compression - arXiv, consulté le décembre 6, 2025, https://arxiv.org/html/2312.09571v1
19. LLM×MapReduce: Simplified Long-Sequence Processing using Large Language Models - arXiv, consulté le décembre 6, 2025, https://arxiv.org/html/2410.09342v1
20. Unlocking Unstructured Data with LLMs, consulté le décembre 6, 2025, https://thedataexchange.media/docetl/
21. 2 Approaches For Extending Context Windows in LLMs - Supermemory, consulté le décembre 6, 2025, https://supermemory.ai/blog/extending-context-windows-in-llms/
22. What is the Chain of Density in Prompt Engineering? - Analytics Vidhya, consulté le décembre 6, 2025, https://www.analyticsvidhya.com/blog/2024/07/chain-of-density-in-prompt-engineering/
23. Chain of Density (CoD) - Learn Prompting, consulté le décembre 6, 2025, https://learnprompting.org/docs/advanced/self_criticism/chain-of-density
24. P-Distill: Efficient and Effective Prompt Tuning Using Knowledge Distillation - MDPI, consulté le décembre 6, 2025, https://www.mdpi.com/2076-3417/15/5/2420
25. LLM distillation demystified: a complete guide | Snorkel AI, consulté le décembre 6, 2025, https://snorkel.ai/blog/llm-distillation-demystified-a-complete-guide/
26. Brandeis University Search results, consulté le décembre 6, 2025, https://scholarworks.brandeis.edu/esploro/search/outputs?query=creator,exact,Pustejovsky%20James,AND&page=1&sort=date_d&mode=advanced&institution=01BRAND_INST
27. Propositional Extraction from Collaborative Naturalistic Dialogues - Journal of Educational Data Mining, consulté le décembre 6, 2025, https://jedm.educationaldatamining.org/index.php/JEDM/article/download/838/246
28. Propositional Extraction from Collaborative Naturalistic Dialogues, consulté le décembre 6, 2025, https://jedm.educationaldatamining.org/index.php/JEDM/article/download/838/245
29. Idea Density and Grammatical Complexity as Neurocognitive Markers - PMC, consulté le décembre 6, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC12468128/

30. Hypergraph RAG: The Third-Generation Knowledge Retrieval Revolution Transforming AI Systems | by Tao An | Oct, 2025, consulté le décembre 6, 2025, https://tao-hpu.medium.com/hypergraph-rag-the-third-generation-knowledge-retrieval-revolution-transforming-ai-systems-cc00dcb56698
31. HyperGraphRAG: Retrieval-Augmented Generation via Hypergraph-Structured Knowledge Representation - arXiv, consulté le décembre 6, 2025, https://arxiv.org/html/2503.21322v2
32. How Would Microsoft GraphRAG Work Alongside a Graph Database? - Memgraph, consulté le décembre 6, 2025, https://memgraph.com/blog/how-microsoft-graphrag-works-with-graph-databases
33. Structured Outputs | Gemini API - Google AI for Developers, consulté le décembre 6, 2025, https://ai.google.dev/gemini-api/docs/structured-output
34. Structured model outputs - OpenAI API, consulté le décembre 6, 2025, https://platform.openai.com/docs/guides/structured-outputs
35. LLM-Friendly Knowledge Representation for Customer Support - arXiv, consulté le décembre 6, 2025, https://arxiv.org/html/2510.10331v1
36. DNA digital data storage - Wikipedia, consulté le décembre 6, 2025, https://en.wikipedia.org/wiki/DNA_digital_data_storage
37. Hidden Addressing Encoding for DNA Storage - Frontiers, consulté le décembre 6, 2025, https://www.frontiersin.org/journals/bioengineering-and-biotechnology/articles/10.3389/fbioe.2022.916615/full
38. DNA Data Storage - PMC - NIH, consulté le décembre 6, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC10296570/
39. dual-rule encoding DNA storage system using chaotic mapping to control GC content | Bioinformatics | Oxford Academic, consulté le décembre 6, 2025, https://academic.oup.com/bioinformatics/article/40/3/btae113/7616129
40. USING DNA FOR DATA STORAGE: ENCODING AND DECODING ALGORITHM DEVELOPMENT - ScholarWorks, consulté le décembre 6, 2025, https://scholarworks.boisestate.edu/cgi/viewcontent.cgi?article=2575&context=td
41. LLM OS Guide: Understanding AI Operating Systems - DataCamp, consulté le décembre 6, 2025, https://www.datacamp.com/blog/llm-os
42. Unlocking the Potential of Generative AI through Neuro-Symbolic Architectures – Benefits and Limitations - arXiv, consulté le décembre 6, 2025, https://arxiv.org/html/2502.11269v1
43. MemGPT with Real-life Example: Bridging the Gap Between AI and OS | DigitalOcean, consulté le décembre 6, 2025, https://www.digitalocean.com/community/tutorials/memgpt-llm-infinite-context-understanding
44. At the Intersection of LLMs and Kernels - Research Roundup - Charles Frye, consulté le décembre 6, 2025, https://charlesfrye.github.io/programming/2023/11/10/llms-systems.html
45. Thinking in LangGraph - Docs by LangChain, consulté le décembre 6, 2025, https://docs.langchain.com/oss/python/langgraph/thinking-in-langgraph

46. LangGraph Tutorial: Complete Guide to Building AI Workflows - Codecademy, consulté le décembre 6, 2025, https://www.codecademy.com/article/building-ai-workflow-with-langgraph
47. [PDF] Wikum: Bridging Discussion Forums and Wikis Using Recursive Summarization, consulté le décembre 6, 2025, https://www.semanticscholar.org/paper/Wikum%3A-Bridging-Discussion-Forums-and-Wikis-Using-Zhang-Verou/55a929b20175bb651e353033060c48ece691b940
48. Recursive Abstractive Processing for Retrieval in Dynamic Datasets - arXiv, consulté le décembre 6, 2025, https://arxiv.org/html/2410.01736v1
49. Microsoft U.S. Patents, Patent Applications and Patent Search - Justia Patents Search, consulté le décembre 6, 2025, https://patents.justia.com/company/microsoft?page=2
50. Context Cascade Compression: Exploring the Upper Limits of Text Compression - arXiv, consulté le décembre 6, 2025, https://arxiv.org/html/2511.15244v1
51. SenseNet: A Knowledge Representation Model for Computational Semantics, consulté le décembre 6, 2025, https://www.researchgate.net/publication/4248313_SenseNet_A_Knowledge_Representation_Model_for_Computational_Semantics
52. Chain of Density prompting can lead to human-level summaries from LLMs - Reddit, consulté le décembre 6, 2025, https://www.reddit.com/r/PromptEngineering/comments/17v3fba/chain_of_density_prompting_can_lead_to_humanlevel/
53. Vec2Summ: Text Summarization via Probabilistic Sentence Embeddings - arXiv, consulté le décembre 6, 2025, https://arxiv.org/html/2508.07017v1
54. Entropy-Optimized Dynamic Text Segmentation and RAG-Enhanced LLMs for Construction Engineering Knowledge Base - MDPI, consulté le décembre 6, 2025, https://www.mdpi.com/2076-3417/15/6/3134
55. Semantic Compression With Large Language Models - Computer Science, consulté le décembre 6, 2025, https://www.cs.wm.edu/~dcschmidt/PDF/Compression_with_LLMs_FLLM.pdf
56. Neuro-symbolic approaches in artificial intelligence | National Science Review, consulté le décembre 6, 2025, https://academic.oup.com/nsr/article/9/6/nwac035/6542460