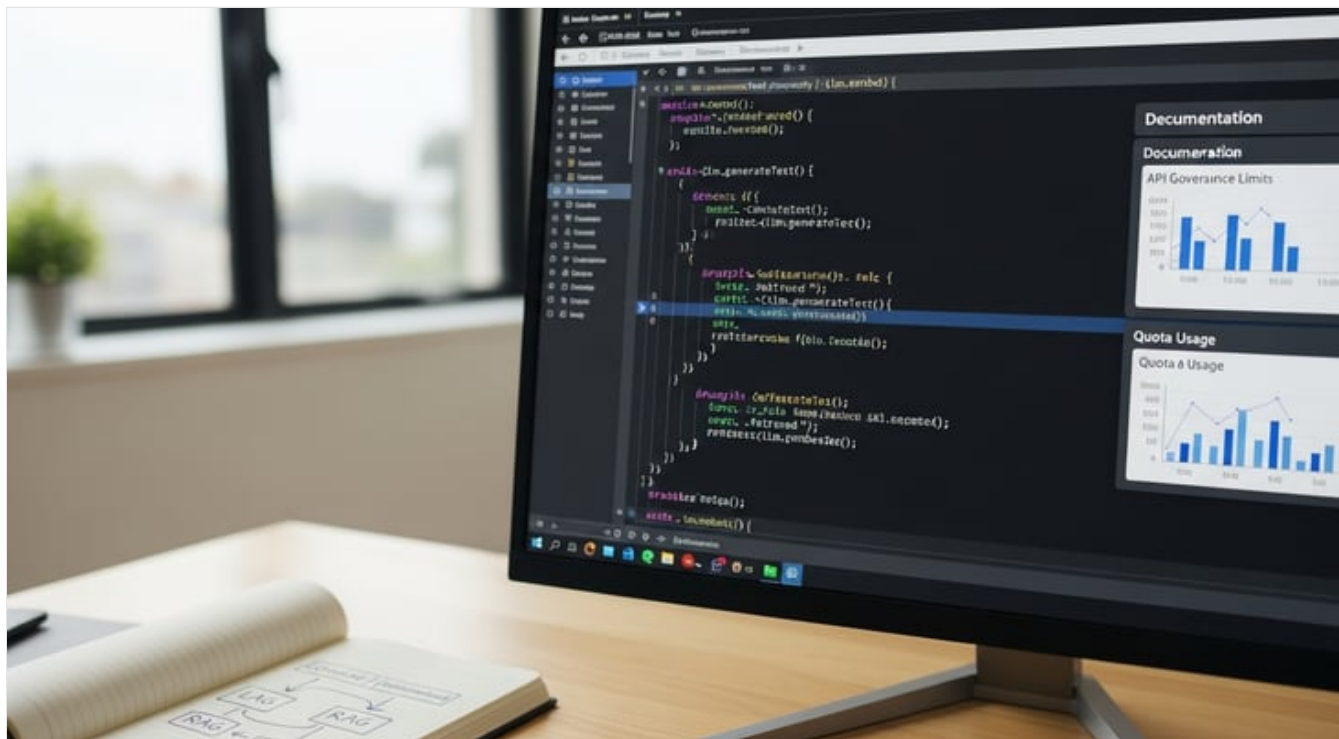


Module NetSuite N/LLM : Guide de l'API GenAI SuiteScript

By houseblend.io Publié le 12 février 2026 41 min de lecture



Résumé analytique

Ce rapport examine le nouveau module [SuiteScript N/LLM](#) d'Oracle NetSuite, qui intègre des capacités de modèles de langage étendus (LLM) directement dans la plateforme NetSuite. Le module N/LLM de NetSuite (disponible dans SuiteScript 2.1) fournit une interface intégrée aux services d'IA générative d'Oracle reposant sur OCI. Il permet aux développeurs d'appeler des LLM à partir de SuiteScript afin de générer ou d'[analyser du texte](#), de transformer du texte en vecteurs et de créer des fonctionnalités intelligentes ([chatbots](#), requêtes en langage naturel, résumés, etc.) en utilisant les propres données de l'entreprise. Par exemple, `llm.generateText(options)` envoie un prompt personnalisé au LLM et renvoie le texte généré, tandis que `llm.embed(options)` produit des plongements vectoriels (embeddings) du texte d'entrée (Source: [docs.oracle.com](#)) (Source: [docs.oracle.com](#)). Le module prend également en charge l'*évaluation de prompt* (via des prompts définis dans [Prompt Studio](#), la *génération augmentée par récupération (RAG)* (en fournissant des documents sources NetSuite pour ancrer les réponses), les *réponses en flux (streaming)* et des méthodes administratives pour suivre l'utilisation. Des alias de script tels que `llm.chat()` et `llm.chatStreamed()` offrent une facilité d'utilisation supplémentaire (Source: [docs.oracle.com](#)).

De manière cruciale, NetSuite régit l'utilisation de N/LLM via un **quota mensuel d'utilisation gratuite** et des **limites de simultanéité** strictes. Chaque compte (et chaque SuiteApp installée) reçoit un nombre limité d'appels LLM gratuits par mois, réinitialisé mensuellement (Source: [docs.oracle.com](#)). Les appels de génération (ex. `generateText`) et les appels d'intégration vectorielle (embedding) utilisent des quotas distincts (Source: [docs.oracle.com](#)). Par défaut, les comptes puisent dans le quota gratuit, mais ils peuvent configurer leurs propres identifiants d'IA générative OCI pour obtenir une utilisation illimitée (facturée sur leur compte OCI) (Source: [docs.oracle.com](#)) (Source: [docs.oracle.com](#)). La simultanéité est également limitée : jusqu'à **cinq appels LLM simultanés** sont autorisés pour les méthodes de génération et jusqu'à cinq pour les méthodes d'intégration vectorielle (Source: [docs.oracle.com](#)) (les appels au-delà de cette limite génèrent des erreurs). La page des **Préférences d'IA** de SuiteScript affiche des tableaux d'utilisation par mois et par type (génération vs intégration) afin que les administrateurs puissent surveiller la consommation (Source: [docs.oracle.com](#)) (Source: [docs.oracle.com](#)).

La décision de NetSuite d'intégrer nativement les LLM reflète les tendances du secteur et les pressions concurrentielles. Les analystes notent que l'adoption de l'IA générative en entreprise a explosé : Gartner a constaté que fin 2023, environ 29 % des organisations interrogées avaient déjà déployé des solutions d'IA générative (Source: [www.gartner.com](#)), et les prévisions suggèrent que plus de 80 % utiliseront des API ou des applications

d'IA générative d'ici 2026 (Source: www.gartner.com). La stratégie d'Oracle contraste avec celle de certains de ses pairs : au lieu de facturer l'IA en supplément, NetSuite intègre des fonctionnalités d'IA (comme ses nouvelles capacités N/LLM) en tant que partie intégrante du produit de base, **sans coût de licence supplémentaire** (Source: www.axios.com) (Source: the-cfo.io). Evan Goldberg, vice-président exécutif d'Oracle NetSuite, souligne que « l'IA sera partout » et insiste pour faire de l'IA un « standard incontournable » pour les clients (Source: www.axios.com) (Source: the-cfo.io). Parallèlement, les concurrents de NetSuite (par exemple SAP) se sont tournés vers des offres d'IA premium, ce qui met en évidence l'accent mis par NetSuite sur l'élargissement de l'accessibilité à l'IA.

Ce rapport fournit une analyse complète du module N/LLM et de son écosystème. Nous couvrons le contexte historique (déploiement de l'IA par NetSuite et positionnement sur le marché), les détails techniques de l'API SuiteScript, la gouvernance de l'utilisation et des quotas, des exemples de code illustratifs et des perspectives d'experts. Nous intégrons également des données et des exemples de cas sur l'IA dans l'ERP d'entreprise : par exemple, l'analyse du secteur montre que les clients NetSuite à gros volume (tels que BirdRock Home, qui « traite des milliers de commandes quotidiennement » dans NetSuite (Source: houseblend.io) gagneraient à utiliser l'IA générative ancrée sur leurs riches ensembles de données. Enfin, nous discutons des risques (précision, confidentialité, gouvernance) et des orientations futures (plus de fonctionnalités d'IA, qualité des données d'entraînement, choix des modèles), et fournissons des recommandations fondées sur des preuves.

Introduction et contexte

L'IA générative (l'utilisation de grands modèles de langage et de modèles de fondation pour générer ou analyser du contenu) est rapidement apparue comme une technologie transformatrice dans les systèmes commerciaux et ERP. Fin 2022 et en 2023, des outils comme ChatGPT et d'autres LLM ont capté l'attention générale en démontrant que l'IA peut accomplir de nombreuses tâches de routine (résumé de texte, traduction, création de contenu) avec une fluidité de niveau humain (Source: www.mckinsey.com) (Source: www.mckinsey.com). Les analystes ont projeté un impact productif et économique énorme. Par exemple, McKinsey (2023) estime que les cas d'utilisation de l'IA générative pourraient ajouter entre **2,6 et 4,4 milliards de dollars** par an à l'économie mondiale (doublant approximativement l'impact de toute l'IA précédente) (Source: www.mckinsey.com). D'autres études suggèrent que l'IA générative pourrait automatiser ou augmenter 60 à 70 % des activités professionnelles actuelles, modifiant fondamentalement les rôles dans tous les secteurs (Source: www.mckinsey.com) (Source: www.mckinsey.com). En conséquence, l'intérêt des entreprises pour l'IA a explosé : Gartner (octobre 2023) a prédit que d'ici 2026, plus de **80 % des entreprises** auront déployé des applications compatibles avec l'IA générative ou utilisé des API d'IA générative, contre moins de 5 % en 2023 (Source: www.gartner.com). Début 2024, Gartner a constaté que l'IA générative était déjà la **solution d'IA la plus fréquemment déployée** dans les organisations (Source: www.gartner.com).

NetSuite (une société d'Oracle) – une plateforme ERP/SaaS cloud de premier plan pour les entreprises du marché intermédiaire aux grandes entreprises – a rapidement intégré l'IA pour tirer parti de ces tendances. Les dirigeants d'Oracle ont souligné la « Suiteness » de NetSuite (l'avantage de données centralisées et intégrées) comme fondement d'insights pilotés par l'IA (Source: houseblend.io) (Source: houseblend.io). En 2023 et 2024, Oracle a annoncé une feuille de route IA globale pour NetSuite : des assistants d'IA générative (par exemple, la fonctionnalité « Text Enhance » pour l'aide à la rédaction de contenu) ont été dévoilés lors de SuiteWorld 2023 (Source: www.techtarget.com), et lors de SuiteWorld 2024, des dizaines de nouvelles fonctionnalités alimentées par l'IA (comme l'Assistant SuiteAnalytics et le reporting narratif) ont été annoncées (Source: the-cfo.io) (Source: the-cfo.io). Les observateurs soulignent que NetSuite se distingue de ses concurrents en **intégrant des capacités d'IA sans frais supplémentaires** – en les traitant comme des fonctionnalités standard de la plateforme (Source: www.axios.com) (Source: the-cfo.io) – et en couplant étroitement les modèles d'IA aux propres données d'entreprise du client via Oracle Cloud Infrastructure (OCI).

Au cœur de cette stratégie se trouve le **module SuiteScript N/LLM** : une bibliothèque native SuiteScript 2.1 fournissant un accès programmatique aux services de modèles de langage étendus (LLM) à partir de scripts SuiteScript basés sur JavaScript. Annoncé en 2024, N/LLM apporte efficacement le service d'IA générative d'OCI (alimenté par des modèles de partenaires comme Cohere) dans l'environnement SuiteScript de NetSuite (Source: blogs.oracle.com) (Source: gurussolutions.com). Les développeurs utilisant SuiteScript – le framework d'extension de NetSuite – peuvent ainsi écrire des scripts qui « parlent » aux LLM, permettant la génération de contenu à la demande, la réponse aux questions, le résumé ou la classification au sein de l'ERP. Comme l'explique un blog NetSuite, N/LLM est « un module NetSuite SuiteScript 2.1 nouvellement introduit qui apporte un accès natif à un modèle de langage étendu (LLM) directement à l'intérieur de NetSuite » (Source: blogs.oracle.com). Cela permet des tâches telles que la création de documents contextuels, le contrôle des paramètres de génération (comme la température), le suivi des citations provenant des propres données de l'utilisateur et la surveillance des quotas d'utilisation de l'API (Source: blogs.oracle.com).

L'importance de N/LLM réside dans sa capacité à *ancrer* l'IA générative dans les *propres* données ERP d'une entreprise. En combinant la récupération de données SuiteScript (via SuiteQL, recherches enregistrées, etc.) avec des prompts LLM, les scripts peuvent mettre en œuvre la **génération augmentée par récupération (RAG)** : le script récupère les enregistrements pertinents (par exemple, les commandes clients récentes) et les transmet en tant que « documents » au LLM, qui génère ensuite des réponses ou des rapports basés sur ces données réelles (Source: blogs.oracle.com) (Source: blogs.oracle.com). Comme le décrit un blog de développeur, le processus est le suivant : « Construisez un tableau de documents (createDocument) liés à votre question. Soumettez les documents avec le prompt à generateText(). Recevez non seulement la réponse

mais aussi des citations renvoyant à vos documents » – garantissant que le résultat est « basé sur vos propres données NetSuite, et non sur des connaissances aléatoires provenant d'Internet » (Source: blogs.oracle.com). Cette approche RAG axée sur l'entreprise atténue le risque d'« hallucination » des LLM en les ancrant à des enregistrements vérifiés.

Dans le même temps, les capacités d'IA de NetSuite sont soumises à une **gouvernance** stricte. Comme le note Oracle, NetSuite « fournit un quota mensuel d'utilisation gratuite de requêtes pour le module N/LLM » (Source: docs.oracle.com). Tous les appels de génération (`llm.generateText`) et les appels d'évaluation de prompt (`llm.evaluatePrompt`) consomment ce quota, et de même, un quota séparé suit les appels d'intégration vectorielle (`llm.embed`) (Source: docs.oracle.com). Les quotas sont renouvelés chaque mois (Source: docs.oracle.com). Les limites de simultanéité régissent davantage l'utilisation : au maximum cinq appels simultanés aux méthodes de génération et cinq aux méthodes d'intégration vectorielle sont autorisés par script, après quoi des erreurs sont générées (Source: docs.oracle.com). Si une organisation a besoin d'une utilisation plus importante, elle peut configurer un identifiant externe OCI Generative AI : dès lors, les appels puisent dans le quota de ce compte plutôt que dans le quota gratuit (Source: docs.oracle.com). Toutes ces contraintes sont suivies sur une page de Préférences d'IA de SuiteScript (Configuration > Société > IA > Préférences d'IA) affichant les limites mensuelles et l'utilisation par type (Source: docs.oracle.com) (Source: docs.oracle.com). Ce système intégré garantit une consommation responsable des ressources de calcul d'IA et permet aux administrateurs de surveiller ou de limiter l'utilisation de l'IA.

Tout au long de ce rapport, nous citons la documentation officielle, des analyses du secteur et des commentaires d'experts pour fournir une vue approfondie et fondée sur des preuves de N/LLM. Les principales conclusions incluent :

- **Capacités de l'API** : N/LLM introduit des méthodes telles que `llm.generateText(options)` pour la génération de contenu, `llm.evaluatePrompt(options)` pour l'utilisation de modèles de prompts prédéfinis, `llm.embed(options)` pour les plongements vectoriels, et des méthodes utilitaires comme `llm.createDocument()` (pour le contexte RAG) et `llm.getRemainingFreeUsage()` (pour la vérification des quotas) (Source: docs.oracle.com) (Source: docs.oracle.com). Chacune possède des variantes asynchrones (promise) et de flux (streaming) correspondantes. Le module comprend également des API orientées chat (`llm.createChatMessage`, `llm.ChatRole`) pour les interfaces conversationnelles (Source: docs.oracle.com) (Source: docs.oracle.com).
- **Gouvernance** : Oracle impose des limites strictes d'utilisation et de concurrence sur N/LLM (5 appels simultanés, quotas mensuels par compte/SuiteApp) (Source: docs.oracle.com) (Source: docs.oracle.com). Les administrateurs peuvent suivre l'utilisation par type dans les Préférences IA de NetSuite (Source: docs.oracle.com), et peuvent optionnellement lier leur propre compte OCI pour une utilisation illimitée (Source: docs.oracle.com).
- **Bonnes pratiques** : Les analyses du secteur insistent sur la validation des résultats de l'IA (NetSuite avertit lui-même que les réponses génératives sont « créatives » et doivent être vérifiées pour leur exactitude (Source: docs.oracle.com) et sur l'alignement des initiatives d'IA avec la gouvernance. Par exemple, une note d'analyste soutient que le succès de N/LLM « dépend d'une mise en œuvre structurée alignée sur la gouvernance des données de l'entreprise » (Source: houseblend.io). Les projets pilotes d'IA non ciblés échouent souvent : un rapport cité note un taux d'échec supérieur à 95 % pour les preuves de concept d'IA générative « non ciblées » (Source: houseblend.io).
- **Impact concret** : L'IA générative dans NetSuite permet de nouveaux cas d'utilisation (requêtes de données en langage naturel, génération automatisée de rapports, améliorations intelligentes des champs, chatbots). Des exemples concrets incluent des clients de NetSuite comme BirdRock Home et Overture Promotions, qui ont exploité l'IA/ML sur leurs données NetSuite (BirdRock pour l'attrition/prédictions, Overture pour les prévisions de la chaîne d'approvisionnement), notant des « améliorations exploitables de la stratégie » (Source: houseblend.io) (Source: houseblend.io). En tant que tendance générale en entreprise, des études larges (Gartner, McKinsey) montrent que l'adoption de l'IA générative croît rapidement et apporte une valeur mesurable (Source: www.mckinsey.com) (Source: www.gartner.com).

Le reste de ce document se déroule comme suit. Nous fournissons d'abord le **contexte historique** de la stratégie IA de NetSuite et présentons les concepts globaux de N/LLM. Nous plongeons ensuite dans l'**API SuiteScript**, en énumérant les méthodes N/LLM et leurs paramètres, accompagnés de fragments de code. Ensuite, nous couvrons la **gouvernance et l'utilisation** (quotas, concurrence, disponibilité régionale, considérations de sécurité). Nous poursuivons avec des **exemples de code** illustratifs (chatbots, RAG, embeddings). Nous incluons plusieurs **perspectives basées sur les données** (statistiques d'adoption, implications sur le ROI, citations d'analystes). Nous comparons et opposons le module avec des outils connexes (ex: SAP Cloud AI, Salesforce Einstein). Nous discutons également d'**études de cas** (réelles ou hypothétiques) montrant N/LLM en action. Enfin, nous analysons les **implications et les orientations futures** (améliorations attendues, risques liés à l'IA, intégration plus large de l'IA en entreprise) et concluons par les points clés à retenir.

IA générative NetSuite : Genèse et contexte

Moteurs de l'industrie : l'IA dans les logiciels d'entreprise

D'ici 2024-2026, l'IA générative est devenue un thème central des logiciels d'entreprise. Les organisations de tous secteurs ont commencé à intégrer l'IA à plusieurs niveaux. Par exemple, Gartner (2024) a révélé que près de **29 %** des entreprises interrogées avaient déjà déployé des solutions d'IA générative (faisant de la GenAI le type d'IA le plus courant) (Source: www.gartner.com), et a prédit que plus de **80 %** des entreprises utiliseraient des API ou des applications de GenAI d'ici 2026 (Source: www.gartner.com). En pratique, de nombreuses entreprises ont commencé par des assistants intégrés (outils de type Copilot au sein de leurs applications de productivité existantes) plus souvent que par le développement de LLM personnalisés (Source: www.gartner.com), mais l'appétit pour des capacités d'IA sur mesure était manifeste.

Les fournisseurs de logiciels d'entreprise ont répondu en ajoutant des améliorations d'IA dans l'ensemble de leurs suites. Oracle, en particulier, a fait des paris majeurs : en 2023, lors de SuiteWorld, NetSuite a annoncé « Text Enhance » – un assistant d'IA générative pour le reporting financier et la rédaction client – construit sur OCI et les modèles LLM de Cohere (Source: www.techtarget.com). En 2024, NetSuite a étendu ces efforts aux flux de travail et à l'analyse : de nouveaux modules SuiteAnalytics Assistant ont permis des requêtes de données en langage naturel et la génération d'insights (Source: the-cfo.io) ; un support de reporting narratif et de prévision piloté par l'IA est arrivé dans sa suite EPM (Enterprise Performance Management) (Source: the-cfo.io) (Source: the-cfo.io). Crucialement, Oracle a décidé de **ne pas facturer de supplément** pour ces fonctionnalités d'IA. Comme l'a rapporté Axios, Oracle a intégré plus de 200 améliorations d'IA dans NetSuite « sans coût supplémentaire », positionnant l'IA comme une capacité fondamentale plutôt que comme un module complémentaire premium (Source: www.axios.com). Evan Goldberg (EVP d'Oracle NetSuite) a expliqué que restreindre ou taxer l'IA ne ferait que ralentir l'adoption : « L'IA va être partout... c'est aussi important que la révolution Internet », a-t-il déclaré (Source: www.axios.com) (Source: the-cfo.io).

Les analystes ont généralement approuvé l'approche de NetSuite. En 2023, Holger Mueller de Constellation Research a noté que l'activation de l'IA générative au niveau de la **couche plateforme** (via les propres services d'Oracle) était la « bonne façon de faire », car elle exploite les données internes de manière sécurisée (Source: www.techtarget.com). Les entretiens de TechTarget ont fait écho à cela : des nuances comme le « text enhance » de NetSuite ont été perçues comme répondant à des tâches courantes de l'ERP (beaucoup de temps passé à rédiger des textes) pour être présentées comme des gains d'efficacité rapides (Source: www.techtarget.com) (Source: www.techtarget.com). Les observateurs de l'industrie ont également noté que, parce que le modèle de données de NetSuite est hautement intégré (ce qu'on appelle la « **Suiteness** »), ses déploiements disposent souvent de jeux de données larges et granulaires qui sont idéaux pour l'entraînement et l'inférence de l'IA (Source: houseblend.io) (Source: houseblend.io). Par exemple, Houseblend souligne que les clients de NetSuite comme BirdRock Home (un détaillant majeur) hébergent des « milliers de produits » et des « milliers de commandes quotidiennes » dans un système unique (Source: houseblend.io), fournissant des données chronologiques et catégorielles riches pour les cas d'utilisation de l'IA (prévisions, détection d'anomalies, etc.). Les données centralisées et les processus unifiés dans NetSuite (« une plus grande partie des données de l'ensemble de votre entreprise ») (Source: houseblend.io) donnent aux modèles d'IA générative plus de contexte pour travailler que les solutions ponctuelles cloisonnées.

Dans le même temps, d'autres plateformes ERP (ex: SAP, Microsoft, Salesforce) déployaient leurs propres unités d'IA. NetSuite devait garder le rythme ; à cette fin, il a renforcé son écosystème de développeurs. Oracle a ouvert **Prompt Studio** (un outil pour gérer les prompts et les modèles d'IA au sein de NetSuite) et a publié l'API SuiteScript N/LLM pour donner aux développeurs un accès de premier ordre aux LLM. Les chefs d'entreprise ont mis l'accent sur les gains de productivité pratiques : par exemple, CFO.com a noté que l'automatisation des tâches routinières de narration financière avec l'IA (afin que les équipes financières puissent se concentrer sur la stratégie) était un argument de vente clé (Source: the-cfo.io). Oracle a présenté l'IA de NetSuite comme étant intégrée aux flux de travail, et non greffée, pour garantir un ROI immédiat « dès qu'ils se connectent » (Source: the-cfo.io).

En résumé, l'incorporation de N/LLM par NetSuite fait partie d'une stratégie plus large pilotée par l'IA : exploiter les services génératifs d'OCI pour ajouter des capacités intelligentes à travers la suite, s'appuyer sur les riches données consolidées de la plateforme et donner du pouvoir aux clients sans coût de licence supplémentaire. Le calendrier (fin 2023/début 2024) s'est aligné sur l'accélération de l'adoption de l'IA en entreprise, et Oracle s'est positionné pour rivaliser fortement sur le marché des ERP de taille intermédiaire. Ce contexte historique explique *pourquoi* le module N/LLM a été développé : il permet aux clients et partenaires de NetSuite d'exploiter des LLM de pointe pour l'analyse, la création de contenu et l'automatisation directement au sein de leur système ERP.

Le module SuiteScript N/LLM : aperçu des capacités

Le module N/LLM est une **bibliothèque SuiteScript 2.1** spécifiquement dédiée à l'IA générative. Selon la documentation d'Oracle, il prend en charge tous les principaux modèles d'IA générative nécessaires dans un contexte ERP (Source: docs.oracle.com) (Source: docs.oracle.com). Ses fonctionnalités incluent :

- Génération de contenu** : La méthode principale est `llm.generateText(options)`. Les développeurs fournissent une chaîne de *prompt* en langage naturel décrivant le contenu ou la requête souhaitée, ainsi que des `modelParameters` optionnels (tels que `maxTokens`, `temperature`, `topK`, `topP` et des termes de pénalité). Le module envoie cette requête au service OCI Generative AI, qui renvoie une réponse générée par le LLM (sous la forme `response.text`). Par exemple, un simple fragment SuiteScript démontre l'envoi du prompt « Hello World » et la réception d'une réponse écrite par l'IA (tout en vérifiant les quotas restants) (Source: docs.oracle.com). Lorsqu'aucun modèle n'est spécifié, NetSuite utilise par défaut un modèle Cohere Command, mais le script peut spécifier d'autres familles de modèles via `options.model` (à partir de `llm.ModelFamily`) (Source: docs.oracle.com) (Source: docs.oracle.com). L'appel `generateText` récupère la sortie complète. Une version asynchrone (promise) est disponible, ainsi qu'une variante en streaming (`generateTextStreamed`) pour une sortie progressive (Source: docs.oracle.com) (Source: docs.oracle.com).
- Évaluation de prompt (Prompt Studio)** : Le **Prompt Studio** de NetSuite permet aux administrateurs de définir des prompts réutilisables avec des variables. La méthode API `llm.evaluatePrompt(options)` prend l'ID d'un prompt stocké (provenant de Prompt Studio) et une correspondance de variables avec leurs valeurs, puis envoie ce prompt préparé au LLM. Le module gère la substitution des variables et l'utilisation du modèle/paramètres prédéfinis du prompt. Par exemple, la documentation de NetSuite donne un échantillon où un prompt d'article d'inventaire est chargé (ID « `stdprompt_gen_purch_desc_invnt_item` ») et des variables comme `itemid` et `stockdescription` sont remplies dans le script (Source: docs.oracle.com). La réponse du LLM (et l'utilisation mensuelle restante) est renvoyée. Cette méthode possède un alias `llm.executePrompt()`. Des formes en streaming/promise (`evaluatePromptStreamed` et sa promise) sont également disponibles (Source: docs.oracle.com).
- Support de la génération augmentée par récupération (RAG)** : Crucialement, `generateText` (et `generateTextStreamed`) accepte un tableau d'objets **Document** sous `options.documents`. Les développeurs peuvent créer de tels documents par programmation avec `llm.createDocument({id, data})`, où `data` est du texte (ou du contenu structuré) fournissant un contexte. Lorsque des documents sont fournis, le service OCI les utilise pour étayer sa réponse et renvoie également un tableau d'objets **Citation** indiquant quels documents sources ont été utilisés. Ainsi, un script peut collecter des citations à des fins d'auditabilité. L'exemple MVP d'Oracle « Sales Insights Suitelet » construit un ensemble de données de ventes résumé dans plusieurs entrées `document` puis appelle `generateText`, les résultats incluant des citations renvoyant à ces documents construits (Source: blogs.oracle.com) (Source: blogs.oracle.com). Comme le note NetSuite, cela garantit que les réponses sont « factuellement fondées » sur les propres données de l'utilisateur (Source: blogs.oracle.com). L'intégration de documents dans les prompts de cette manière atténue les hallucinations.
- Embeddings (Plongements vectoriels)** : La méthode `llm.embed(options)` convertit les entrées textuelles en plongements vectoriels (tableaux de nombres flottants) à l'aide de modèles d'embedding dédiés. L'utilisateur fournit une ou plusieurs chaînes d'entrée (`options.inputs`) et sélectionne une famille de modèles d'embedding (ex: `llm.EmbedModelFamily.COHERE_EMBED`). Le service LLM renvoie un objet `EmbedResponse` contenant les vecteurs d'embedding. Ceci est utile pour la recherche sémantique, les calculs de similitude, les recommandations, le clustering, etc. Par exemple, un exemple de Suitelet montre la génération d'embeddings pour une liste de noms d'articles et le calcul de la similitude cosinus pour trouver des articles similaires (Source: docs.oracle.com) (Source: docs.oracle.com). Les appels d'embedding sont décomptés d'un quota mensuel *distinct* (différent des appels de génération) (Source: docs.oracle.com).
- Support du Chat** : Pour faciliter les scénarios de chat ou de conversation, N/LLM fournit des utilitaires orientés chat. L'énumération `llm.ChatRole` définit les rôles des messages (tels que `USER` ou `ASSISTANT`). Un développeur peut appeler `llm.createChatMessage({role, text})` pour créer un objet `ChatMessage` (Source: docs.oracle.com). Un tableau de `ChatMessage` peut ensuite être passé comme paramètre `chatHistory` à `generateText`. (NetSuite traite cela comme un historique de conversation initial.) En effet, `llm.chat(options)` (alias de `generateText`) peut prendre un `chatHistory` avec des rôles et poursuivre le dialogue avec le modèle. C'est précieux pour construire des chatbots basés sur les LLM ou des assistants multi-tours au sein de NetSuite.
- Méthodes utilitaires** : Le module comprend des méthodes de vérification de statut utiles. Notamment, `llm.getRemainingFreeUsage()` renvoie le nombre de requêtes de génération gratuites restantes pour le mois en cours, et `llm.getRemainingFreeEmbedUsage()` fait de même pour les requêtes d'embedding (Source: docs.oracle.com). Les deux disposent de variantes basées sur des promesses (promises). Cela permet aux scripts de vérifier et d'afficher dynamiquement les quotas restants (comme on peut le voir dans le code du Suitelet Sales Insights, qui affiche « Remaining LLM Usage » sur le formulaire via `llm.getRemainingFreeUsage()` (Source: blogs.oracle.com). Ces méthodes aident les administrateurs et les développeurs de scripts à surveiller la consommation.

Les **membres du module N/LLM** peuvent être résumés dans le Tableau 1 ci-dessous. Chaque méthode ou objet est directement lié à une fonction relative aux LLM :

Tableau 1. Méthodes et objets du module SuiteScript N/LLM.

MEMBRE (SUITESCRIPT)	ALIAS / RÔLE	OBJECTIF / DESCRIPTION	EXEMPLE D'UTILISATION (VOIR RÉF.)
<code>llm.generateText(options)</code> (alias <code>llm.chat(options)</code>)	—	Envoie un prompt (chaîne) ainsi que des paramètres au modèle ; renvoie un objet <code>Response</code> contenant le texte généré et les éventuelles citations (Source: docs.oracle.com) (Source: docs.oracle.com).	<pre>const res = llm.generateText({ prompt: "Hello", modelParameters: {...} }); const text = res.text; (Source: docs.oracle.com)</pre>
<code>llm.generateTextStreamed(options)</code> (alias <code>llm.chatStreamed(options)</code>)	—	Identique à <code>generateText</code> , mais renvoie un objet <code>StreamedResponse</code> qui fournit le texte partiel de manière incrémentielle au fur et à mesure que l'LLM le génère (Source: docs.oracle.com).	(Similaire à ci-dessus, mais avec une sortie par fragments/chunks)
<code>llm.evaluatePrompt(options)</code> (alias <code>llm.executePrompt(options)</code>)	—	Évalue un prompt de Prompt Studio par ID avec les valeurs de variables fournies. Renvoie une <code>Response</code> avec la réponse de l'LLM (Source: docs.oracle.com).	<pre>const resp = llm.evaluatePrompt({ id: "myPromptID", variables: {...} }); const ans = resp.text; (Source: docs.oracle.com)</pre>
<code>llm.evaluatePromptStreamed(options)</code> (alias <code>llm.executePromptStreamed(options)</code>)	—	Version streamée de <code>evaluatePrompt</code> (renvoie le texte partiel dès qu'il est prêt).	(Similaire, avec <code>StreamedResponse</code>)
<code>llm.embed(options)</code>	—	Génère des embeddings pour le(s) texte(s) d'entrée en utilisant une famille de modèles spécifiée. Renvoie <code>EmbedResponse</code> avec les vecteurs (Source: docs.oracle.com) (Source: docs.oracle.com).	<pre>const embedRes = llm.embed({ embedModelFamily: llm.EmbedModelFamily.COHERE_EMBED, inputs: ["text1", "text2"] }); (Source: docs.oracle.com)</pre>
<code>llm.createDocument(options)</code>	—	Crée un objet <code>Document</code> (avec <code>id</code> et texte <code>data</code>) à fournir comme contexte dans un appel de génération.	<pre>const doc = llm.createDocument({ id: "doc1", data: "Sales data..." }); documents.push(doc); (Source: blogs.oracle.com)</pre>
<code>llm.createChatMessage(options)</code>	—	Crée un <code>ChatMessage</code> (avec <code>role</code> et <code>text</code>) pour construire un historique de chat multi-tours (Source: docs.oracle.com).	<pre>const msg = llm.createChatMessage({ role: llm.ChatRole.USER, text: "Hello!" }); (Source: docs.oracle.com)</pre>

MEMBRE (SUITESCRIPT)	ALIAS / RÔLE	OBJECTIF / DESCRIPTION	EXEMPLE D'UTILISATION (VOIR RÉF.)
<code>llm.getRemainingFreeUsage()</code>	–	Renvoie le nombre (entier) de requêtes de génération gratuites (texte/QA) restantes ce mois-ci (Source: docs.oracle.com).	<pre>const left = llm.getRemainingFreeUsage();</pre>
<code>llm.getRemainingFreeEmbedUsage()</code>	–	Renvoie le nombre (entier) de requêtes d'embedding gratuites restantes ce mois-ci (Source: docs.oracle.com).	<pre>const embLeft = llm.getRemainingFreeEmbedUsage();</pre>
<code>Enums : llm.ChatRole , llm.ModelFamily , llm.EmbedModelFamily , llm.Truncate</code>	–	Énumérations pour les rôles (USER/ASSISTANT), les familles de modèles (ex: COHERE), les familles de modèles d'embedding et les stratégies de troncature d'entrée (Source: docs.oracle.com) (Source: docs.oracle.com).	Ex: utilisation de <code>llm.ChatRole.USER</code> , <code>llm.ModelFamily.COHERE_COMMAND</code> ou <code>llm.EmbedModelFamily.COHERE_EMBED</code> .

Le tableau ci-dessus (tiré de la documentation SuiteScript d'Oracle (Source: docs.oracle.com) (Source: docs.oracle.com) souligne que N/LLM couvre l'ensemble des tâches d'IA générative attendues : **génération de texte, modèles de prompts, sortie en streaming et embeddings sémantiques**. Il fournit également des types d'objets (ex: `llm.Citation`) et une API basée sur les promesses pour tous les appels asynchrones, conformément aux conventions SuiteScript. Par exemple, au lieu de `generateText()`, on peut utiliser `generateText.promise()` pour obtenir une `Promise` JavaScript pour les flux de travail asynchrones (voir **N/LLM Module Members** dans la documentation (Source: docs.oracle.com)).

La documentation d'Oracle insiste sur des points clés : l'IA générative « fait preuve de créativité », les réponses doivent donc être validées pour en vérifier l'exactitude (Source: docs.oracle.com), et ces fonctionnalités sont restreintes par région (disponibles uniquement là où OCI Generative AI est configuré) (Source: docs.oracle.com). Elle stipule précisément : « *Oracle NetSuite n'est pas responsable de l'utilisation ou de l'interprétation du contenu généré par l'IA* » (Source: docs.oracle.com), ce qui signifie que les intégrateurs doivent garantir la qualité et la conformité. Le module N/LLM est automatiquement disponible dans tout compte où le SuiteScript côté serveur est activé (Source: docs.oracle.com), et la liste complète des méthodes et objets (telle que présentée dans le Tableau 1) est documentée sur le centre d'aide d'Oracle.

En plus de la documentation officielle d'Oracle, des sources tierces et des partenaires ont résumé l'objectif de N/LLM. Par exemple, le partenaire NetSuite GURUS Solutions le qualifie de « co-pilote IA pour SuiteScript » qui « connecte vos scripts aux services d'IA générative d'Oracle Cloud Infrastructure », permettant la génération de contenu à la demande et l'utilisation du RAG (Source: gurussolutions.com). L'aperçu de GURUS met en avant les mêmes points : générer du texte, évaluer des prompts stockés, gérer les prompts/améliorations de texte, alimenter des documents pour le RAG, produire des embeddings et suivre l'utilisation (Source: gurussolutions.com). Cela s'aligne étroitement avec les fonctionnalités officielles, renforçant l'idée que N/LLM apporte une large « intelligence » artificielle dans les scripts NetSuite.

Intégration avec les autres SuiteTools

N/LLM n'est pas un outil isolé ; il fonctionne aux côtés d'autres outils SuiteCloud :

- **Prompt Studio** : Une interface utilisateur pour créer et stocker des prompts dynamiques (avec variables) dans NetSuite. La méthode `evaluatePrompt()` de N/LLM s'interface directement avec cet outil. Les développeurs peuvent concevoir des prompts dans Studio et les appeler dans des scripts, garantissant la cohérence des paramètres du modèle et la possibilité de mettre à jour les prompts en dehors du code.

- **Text Enhance** : Une fonctionnalité UI/UX (par exemple, une SuiteApp) qui permet aux utilisateurs finaux de générer du texte dans les champs de fiches. Bien que les actions Text Enhance fonctionnent dans le contexte de l'interface utilisateur, des scripts avancés peuvent les imiter ou les enrichir. La documentation mentionne la gestion des « actions Text Enhance » via `N/record` et N/LLM (Source: docs.oracle.com), ce qui signifie que les scripts peuvent créer ou exécuter ces actions par programmation.
- **SuiteQL et recherches** : Étant donné que les données NetSuite doivent être préparées pour l'LLM, le module est souvent utilisé en tandem avec `N/query` (SuiteQL) ou `N/search` pour récupérer des données. Une note de performance provenant d'un exemple de développeur Oracle indique que l'utilisation de SuiteQL plutôt que `N/search` peut améliorer l'efficacité lors de la récupération de données destinées à alimenter les documents LLM (Source: blogs.oracle.com). En pratique, les scripts interrogeront les données (clients, transactions, etc.), les formateront (souvent sous forme de phrases en langage naturel ou de CSV), puis utiliseront `llm.createDocument` pour encapsuler ce texte de contexte.
- **Préférences d'IA (AI Preferences)** : Les préférences d'IA de NetSuite (sous Configuration de la société) fournissent une interface permettant aux administrateurs de configurer l'utilisation de l'IA. Elles permettent d'activer les identifiants OCI (pour le mode illimité) et affichent le sous-onglet **AI Usage**. Ce onglet liste une ligne par type d'utilisation (ex : « Generate – Cohere » ou « Embed – Cohere »), indiquant les limites mensuelles gratuites et la consommation (Source: docs.oracle.com). La documentation explique explicitement comment consulter ce tableau (Source: docs.oracle.com). Pour la gouvernance, il est important de noter que chaque pool SuiteScript (compte principal vs SuiteApps) possède des compteurs distincts.
- **Édition simultanée et débogage** : Les appels N/LLM peuvent être testés dans le débogueur SuiteScript. Oracle note que le code d'exemple utilise `require` pour pouvoir être copié-collé dans le débogueur ; les scripts de production doivent utiliser `define` pour les points d'entrée appropriés (Source: docs.oracle.com) (Source: docs.oracle.com). Il s'agit d'une pratique SuiteScript standard, mais elle mérite d'être soulignée pour les développeurs découvrant l'API.

Globalement, N/LLM est profondément intégré à l'écosystème SuiteCloud. Il tire parti des capacités de récupération de données et des formulaires d'interface utilisateur de NetSuite, tout en introduisant des fonctionnalités entièrement nouvelles via ces méthodes d'API génératives.

Gouvernance et limites d'utilisation

NetSuite met en œuvre une gouvernance stricte sur N/LLM pour prévenir les abus et contrôler les coûts. Cette section détaille les limites d'utilisation, le suivi et les règles de simultanéité, en s'appuyant sur la documentation d'Oracle et les guides de bonnes pratiques.

Quotas d'utilisation mensuels

Chaque compte NetSuite disposant des fonctionnalités d'IA activées bénéficie d'un **pool mensuel d'utilisation gratuite** pour N/LLM. La taille exacte de ce pool n'est pas rendue publique, mais elle est fixe par compte et se réinitialise au début de chaque mois (Source: docs.oracle.com) (Source: docs.oracle.com). Les points clés sont les suivants :

- **Pools distincts** : Il existe **deux pools distincts** : un pour les appels génératifs (« Generate – Text ») et un pour les appels d'embedding (« Embed – Text ») (Source: docs.oracle.com). L'utilisation dans chaque catégorie est suivie séparément, et la page des Préférences d'IA les affiche sur des lignes distinctes (Source: docs.oracle.com). Par exemple, si vous effectuez 10 appels `generateText` et 5 appels `embed` au cours d'un mois, le tableau affichera deux lignes avec ces décomptes par rapport à chaque limite respective.
- **Suivi** : Le sous-onglet **AI Preferences** de SuiteScript affiche une **Usage Limit** (quota gratuit) et une **Used Quantity** pour chaque mois/type (Source: docs.oracle.com). Il est précisé que : « Toutes les actions terminées avec succès sont comptabilisées, mais pas les réponses d'erreur » (Source: docs.oracle.com). Les scripts eux-mêmes peuvent vérifier l'utilisation au moment de l'exécution via `llm.getRemainingFreeUsage()` et `llm.getRemainingFreeEmbedUsage()` (Source: docs.oracle.com). (L'exemple de Suitelet Sales Insights affiche même le quota restant sur son formulaire en utilisant `getRemainingFreeUsage()` (Source: blogs.oracle.com).
- **Mode illimité (identifiants OCI)** : Pour dépasser le pool gratuit, un compte doit être lié à un compte Oracle Cloud disposant du service Generative AI. Cela se fait dans les Préférences d'IA – l'administrateur fournit les identifiants de l'API OCI. Une fois configurés, tous les appels N/LLM sont facturés à ce compte OCI au lieu du pool interne (Source: docs.oracle.com) (Source: docs.oracle.com). Dans cette configuration « illimitée », les SNPS indiquent qu'un script peut inclure des paramètres de configuration OCI dans ses appels. Essentiellement, NetSuite relaie simplement les requêtes vers OCI avec ces identifiants et l'utilisation est déduite de l'abonnement cloud du client.

- **Considérations relatives aux SuiteApps** : Si une SuiteApp tierce est installée et contient du SuiteScript utilisant N/LLM, Oracle attribue à chaque SuiteApp son propre pool indépendant (Source: docs.oracle.com). Cela signifie que l'utilisation de l'IA par une SuiteApp n'est **pas** décomptée du pool du compte principal (et vice versa). La documentation explique explicitement : « chaque SuiteApp installée dans votre compte dispose de son propre pool d'utilisation mensuel séparé pour les méthodes N/LLM... garantissant que les SuiteApps ne peuvent pas épuiser toute votre allocation mensuelle et bloquer vos propres scripts » (Source: docs.oracle.com). Ces pools de SuiteApps ne sont pas visibles sur la page des Préférences d'IA (le fournisseur de l'application doit fournir le suivi), mais ils suivent les mêmes règles en interne.
- **APIs de suivi** : Comme mentionné, l'API N/LLM expose des méthodes utilitaires pour interroger l'utilisation. L'appel à `llm.getRemainingFreeUsage()` renvoie le nombre de requêtes de génération de texte restantes (Source: docs.oracle.com) ; de même, `llm.getRemainingFreeEmbedUsage()` pour les embeddings (Source: docs.oracle.com). Celles-ci peuvent être utilisées (par exemple) pour désactiver des fonctionnalités dans un script si les quotas sont bas, ou pour afficher des avertissements.

Le **Tableau 2 (ci-dessous)** résume la gouvernance de l'utilisation :

TYPE D'UTILISATION	QUOTA MENSUEL GRATUIT	LIMITE DE SIMULTANÉITÉ	NOTES
Génération de texte / Éval. de prompt	Pool gratuit fixe (réinitialisé mensuellement) (Source: docs.oracle.com)	5 appels simultanés (Source: docs.oracle.com)	Les méthodes génératives (<code>llm.generateText</code> , <code>evaluatePrompt</code> , etc.) partagent ce pool. Pour augmenter la limite, fournissez des identifiants OCI (facturation illimitée) (Source: docs.oracle.com).
Embedding	Pool gratuit fixe (réinitialisé mensuellement) (Source: docs.oracle.com)	5 appels simultanés (Source: docs.oracle.com)	L'embedding (<code>llm.embed</code>) utilise un pool séparé. Utilisation illimitée via identifiants OCI également. Chaque pool est affiché séparément dans les Préférences IA (Source: docs.oracle.com) (Source: docs.oracle.com).

Tableau 2. Règles de gouvernance d'utilisation de N/LLM (documentation Oracle SuiteScript).

En plus de ces limites quantitatives, NetSuite impose des **règles de disponibilité par région et par compte**. Tous les comptes dans le monde n'ont pas l'IA activée par défaut : les fonctionnalités d'IA générative ne sont disponibles que dans certains centres de données et régions (ceux où le service OCI GenAI a été configuré pour NetSuite) (Source: docs.oracle.com) (Source: docs.oracle.com). Par exemple, la documentation liste la couverture en Amérique du Nord, Europe, Asie, etc., mais note que les paramètres d'un compte et la langue de l'utilisateur peuvent également affecter la disponibilité de l'IA (Source: docs.oracle.com). Les administrateurs doivent consulter la rubrique d'aide « Generative AI Availability » de NetSuite pour obtenir des détails par pays et par langue.

Enfin, Oracle décline explicitement toute responsabilité : N/LLM est traité comme un outil créatif, et tout résultat doit être validé par l'utilisateur (Source: docs.oracle.com). Cela est cohérent avec les pratiques de l'industrie. Comme le souligne Gartner, le principal obstacle à l'adoption de l'IA est souvent « l'estimation et la démonstration de la valeur » (Source: www.gartner.com), ce qui signifie que les utilisateurs doivent mesurer et gouverner soigneusement ces nouvelles capacités. L'analyse de Houseblend souligne de la même manière que les projets d'IA réussis sont **hautelement ciblés** et conformes au RGPD ou à la confidentialité, avertissant que les pilotes sans objectif précis échouent souvent (Source: houseblend.io). NetSuite met en œuvre une gouvernance pour imposer une telle discipline : les quotas et le suivi garantissent que les scripts d'IA restent limités et transparents.

Exemples de code SuiteScript

Nous présentons ci-dessous des extraits de code SuiteScript représentatifs illustrant l'utilisation des méthodes N/LLM. Ces exemples s'inspirent largement de la documentation officielle de NetSuite (Source: docs.oracle.com) (Source: docs.oracle.com) (Source: blogs.oracle.com) (Source: docs.oracle.com) et des blogs de développeurs (Source: blogs.oracle.com) (Source: blogs.oracle.com). Chaque extrait utilise soit le modèle `define`, soit le modèle `require` (la documentation autorise `require` pour des tests rapides dans le débogueur ; les scripts de production utilisent `define`).

1. Génération de texte simple

L'exemple le plus basique envoie un prompt « Hello World ». Le code ci-dessous (basé sur la rubrique d'aide de NetSuite) requiert le module `N/llm` et appelle `generateText` avec des paramètres, puis lit le texte retourné et l'utilisation restante :

```
require(['N/llm'], function(llm) {
  const response = llm.generateText({
    // Si aucune modelFamily n'est spécifiée, NetSuite utilise un modèle par défaut (Cohere Command A)
    prompt: "Hello World!",
    modelParameters: {
      maxTokens: 100,
      temperature: 0.2,
      topK: 3,
      topP: 0.7,
      frequencyPenalty: 0.4,
      presencePenalty: 0
    }
  });
  const responseText = response.text;
  const remainingUsage = llm.getRemainingFreeUsage(); // appels restants ce mois-ci
  log.debug('Sortie LLM', responseText);
  log.debug('Appels gratuits restants', remainingUsage);
});
```

Cet extrait (adapté de la documentation NetSuite) montre comment appeler le LLM par défaut avec un prompt simple. Le champ `text` de l'objet de réponse contient le résultat généré par l'IA et `llm.getRemainingFreeUsage()` renvoie le nombre d'appels gratuits restants (Source: docs.oracle.com).

2. Évaluation d'un prompt du Prompt Studio

Si vous avez un prompt défini dans le Prompt Studio de NetSuite, vous pouvez l'utiliser directement. Par exemple, supposons que nous ayons un ID de prompt `stdprompt_gen_purch_desc_invnt_item` (un prompt standard pour générer des descriptions d'achat à partir des données d'un article en stock). Le code pour évaluer ce prompt et fournir des variables est :

```
require(['N/llm'], function(llm) {
  const response = llm.evaluatePrompt({
    id: 'stdprompt_gen_purch_desc_invnt_item',
    variables: {
      // Ces noms de variables et cette structure proviennent de la définition du Prompt Studio
      "form": {
        "itemid": "Mon article en stock",
        "stockdescription": "Ceci est la description de stock de l'article.",
        "vendorname": "Mon Fournisseur Inc.",
        "isdropshipitem": "false",
        "isspecialorderitem": "true",
        "displayname": "Mon super article en stock"
      },
      "text": "Ceci est la description d'achat de l'article."
    }
  });
  const aiText = response.text;
  const left = llm.getRemainingFreeUsage();
  log.debug('Réponse du prompt', aiText);
  log.debug('Appels restants', left);
});
```

Cet exemple (issu des échantillons de scripts NetSuite (Source: docs.oracle.com)) charge un prompt stocké par son ID et fournit les champs requis. Le `response.text` retourné est la réponse du LLM basée sur la logique du prompt. L'utilisation restante est à nouveau vérifiée.

3. Génération augmentée par récupération (RAG)

Voici une illustration simplifiée de l'utilisation du RAG. Imaginez que nous ayons récupéré certains enregistrements et les ayons formatés en texte (ex: « Article : X, Ventes T1 : 1000... »). Nous créons des objets `Document` avec `llm.createDocument()` et les fournissons à `generateText`. Le LLM peut citer ces documents dans sa réponse :

```
require(['N/llm'], function(llm) {
  // Exemple de données structurées préparées sous forme de documents
  const documents = [];
  // Supposons que nous ayons deux points de données :
  const docData1 = "Article : Widget A, Qté vendue : 150, Revenu : 3000 $";
  const docData2 = "Article : Widget B, Qté vendue : 120, Revenu : 2500 $";
  const doc1 = llm.createDocument({ id: "doc1", data: docData1 });
  const doc2 = llm.createDocument({ id: "doc2", data: docData2 });
  documents.push(doc1, doc2);

  // Question en langage naturel de l'utilisateur :
  const prompt = "Quel article a eu la plus grande quantité vendue ?";
  const response = llm.generateText({
    prompt: prompt,
    documents: documents,
    modelParameters: { maxTokens: 50, temperature: 0.2 }
  });

  const bestItem = response.text; // ex: "Le Widget A a eu plus de ventes."
  const citations = response.citations; // ex: [{subRoute: "doc1"}, {subRoute: "doc2"}, ...]
  log.debug('Réponse', bestItem);
  log.debug('Citations', citations);
});
```

Dans ce modèle, nous avons injecté deux documents dans `generateText`. Le texte de réponse doit répondre sur la base de ces documents (« Widget A... »), et le tableau `response.citations` inclura des références comme `doc1` et/ou `doc2` si le LLM les a utilisées. Cela correspond au modèle d'exemple RAG d'Oracle (Source: blogs.oracle.com) (Source: blogs.oracle.com).

4. Génération et utilisation d'embeddings

Pour montrer comment fonctionnent les embeddings, considérons la génération d'embeddings pour une liste de noms d'articles et leur comparaison pour la similitude :

```
require(['N/llm'], function(llm) {
    // Liste d'articles (pour démonstration)
    const items = ["Ultra Widget Pro", "Super Widget X", "Deluxe Gadget"];
    // Obtenir des embeddings en utilisant le modèle d'embedding Cohere
    const embeddingResponse = llm.embed({
        embedModelFamily: llm.EmbedModelFamily.COHERE_EMBED,
        inputs: items
    });
    const vectors = embeddingResponse.vectors; // tableau de tableaux d'embeddings
    // Calculer manuellement la similitude cosinus (utilitaire d'exemple)
    function cosine(a, b) {
        let dot=0, normA=0, normB=0;
        for (let i=0; i<a.length; i++) {
            dot += a[i]*b[i];
            normA += a[i]*a[i];
            normB += b[i]*b[i];
        }
        return dot / (Math.sqrt(normA)*Math.sqrt(normB));
    }
    // Comparer le premier article aux autres
    for (let i = 1; i < vectors.length; i++) {
        const sim = cosine(vectors[0], vectors[i]);
        log.debug('Similitude Article0-Article'+i, sim);
    }
});
```

Cet extrait montre l'utilisation de `llm.embed` (adapté de l'exemple « Find Similar Items » d'Oracle (Source: docs.oracle.com) (Source: docs.oracle.com)). Nous demandons des embeddings pour plusieurs entrées. Les `vectors` retournés peuvent être utilisés dans une logique personnalisée, telle que le calcul de similitudes cosinus pour trouver quels articles sont sémantiquement les plus proches.

Les exemples de code ci-dessus illustrent le modèle de programmation de base pour N/LLM. Dans une application réelle, vous combineriez généralement ces appels avec la récupération de données (`N/query` ou `N/search`) et la gestion de formulaires (`N/ui/serverWidget`) selon les besoins. Par exemple, le suitelet Sales Insights dans le blog Oracle de Wilman Arambillete exécute d'abord une requête SuiteQL (en utilisant `N/query`) pour résumer les données de vente, puis formate chaque ligne de résultat en une chaîne de caractères et appelle `llm.createDocument` dessus (Source: blogs.oracle.com) (Source: blogs.oracle.com). Enfin, il appelle `llm.generateText` pour répondre à la question de l'utilisateur dans son contexte. L'exemple de ce blog est un cas de RAG de bout en bout construit sur N/LLM (Source: blogs.oracle.com) (Source: blogs.oracle.com).

Exemple de code : Streaming

Pour être complet, voici un bref aperçu de la forme *streamée* (en flux) de la génération :

```
require(['N/llm'], function(llm) {
  const streamed = llm.generateTextStreamed({
    prompt: "Expliquez la tendance des ventes aux T1 et T2.",
    modelParameters: { maxTokens: 200 }
  });
  // L'objet StreamedResponse peut être récupéré par morceaux :
  let textSoFar = "";
  while (!streamed.done) {
    const chunk = streamed.getText(); // nouveau morceau de texte
    textSoFar += chunk;
    // (En pratique, utilisez le callback/promesse streamed.done)
  }
  log.debug('Réponse complète', textSoFar);
});
```

Dans ce modèle, `generateTextStreamed` renvoie un `StreamedResponse`. Le script peut appeler `getText()` de manière répétée pour recevoir le texte au fur et à mesure qu'il est généré, ce qui est utile pour les sorties longues ou pour maintenir une interface utilisateur réactive. (Voir la documentation d'Oracle pour l'utilisation des méthodes streamées (Source: docs.oracle.com).)

Considérations sur la gouvernance, les données et la sécurité

Gouvernance activée par l'IA

L'utilisation de l'IA dans un ERP d'entreprise soulève des questions de gouvernance concernant le **coût, la précision et la conformité**. NetSuite traite le coût via les quotas d'utilisation déjà décrits. La précision et la pertinence doivent être gérées par les intégrateurs : Oracle prévient explicitement que les résultats génératifs sont « créatifs » et peuvent être inexacts (Source: docs.oracle.com). Par conséquent, les scripts doivent toujours valider les résultats de l'IA avant de les enregistrer dans les dossiers. Comme le note un consultant : « Validez la sortie de l'IA : l'IA générative est puissante mais pas toujours précise à 100 %. Validez toujours avant d'utiliser du contenu généré par l'IA en production » (Source: gurussolutions.com). Houseblend souligne également que les projets d'IA générative doivent avoir un périmètre étroit et être gouvernés : il cite des recherches montrant que plus de **95 %** des pilotes d'IA sans objectif précis ne parviennent pas à générer un ROI (Source: houseblend.io), soulignant la nécessité d'une planification minutieuse.

La confidentialité des données est une autre préoccupation. Toutes les données transmises au LLM (prompts ou documents) sont envoyées vers le cloud d'Oracle. Les clients des secteurs réglementés doivent s'assurer que les informations sensibles (PII, données HIPAA, etc.) sont traitées de manière appropriée. Oracle ne documente pas spécifiquement la résidence des données d'OCI GenAI, mais note que seuls les comptes de certaines régions peuvent utiliser les fonctionnalités génératives (Source: docs.oracle.com) (régions où OCI GenAI est disponible). En pratique, les entreprises peuvent choisir de supprimer les champs sensibles des prompts ou d'utiliser un point de terminaison sur site lorsqu'il sera disponible. (NetSuite n'a pas annoncé d'options sur site pour N/LLM pour le moment ; il s'appuie sur OCI.) Les organisations doivent traiter N/LLM comme n'importe quelle autre intégration : la journalisation sécurisée, le chiffrement et les politiques de gouvernance s'appliquent.

Du côté positif, N/LLM peut améliorer la sécurité des données en **normalisant et en assainissant** le contenu généré par l'utilisateur. Par exemple, les actions Text Enhance permettent d'« améliorer » ou de paraphraser par l'IA une saisie de texte libre avant son stockage, ce qui pourrait réduire les saisies arbitraires. Mais cela va dans les deux sens : s'il est mal configuré, l'IA pourrait par inadvertance introduire des informations sensibles (les « hallucinations » de type armes de destruction massive sont rares mais possibles). L'analyse de Houseblend note explicitement qu'une **gouvernance des données** d'entreprise robuste est cruciale lors de l'intégration de l'IA (Source: houseblend.io). Les comptes NetSuite devraient intégrer l'utilisation de N/LLM dans leurs cadres de gouvernance des données existants : par exemple, seuls les scripts de confiance devraient effectuer des appels d'IA, et les journaux des interactions avec l'IA devraient être auditable.

Performance et simultanéité

Comme les appels LLM impliquent des services externes, ils peuvent induire une latence. La limite de simultanéité (5 appels) est probablement destinée en partie à prévenir les dénis de service (DoS) accidentels causés par des scripts malveillants ou défaillants. En pratique, les développeurs doivent concevoir des flux de travail pour éviter d'atteindre cette limite. Par exemple, un script devrait attendre qu'un appel LLM se termine avant d'en

lancer un autre, plutôt que d'en déclencher cinq à la fois. Le SDK impose la limite : si un sixième appel est effectué avant que l'un des cinq premiers ne revienne, NetSuite renverra une erreur (ex: `LLM_REQUEST_LIMIT_REACHED`). Les scripts peuvent intercepter ces erreurs ou vérifier l'utilisation/simultanéité restante via `runtime.getCurrentScript().getRemainingUsage()` (gouvernance SuiteScript) et des techniques de régulation.

Notez également : SuiteScript possède son propre compteur d'unités de CPU/gouvernance (différent de l'utilisation de l'IA). L'appel des API LLM ne consomme **pas** d'unités de gouvernance SuiteScript, mais les scripts doivent toujours respecter les limites d'exécution globales. Les appels LLM de longue durée bloqueront le thread d'exécution d'un script. Pour les cas d'utilisation intensifs, NetSuite recommande d'utiliser le SSR asynchrone (scripts SuiteScript Scheduled/MapReduce) plutôt que des scripts client ou user-event pour éviter les dépassements de délai (timeouts).

Implications tarifaires

Bien que NetSuite ne facture pas de frais de licence supplémentaires pour les fonctionnalités d'IA générative, les clients peuvent être confrontés à des coûts de deux manières :

- 1. Frais de la plateforme SuiteCloud** : Toute augmentation significative de l'utilisation des scripts (par exemple, l'embedding de nombreux documents chaque heure) pourrait nécessiter la mise à niveau vers un forfait SuiteCloud supérieur si elle dépasse le niveau gratuit (unités SuiteFlow, etc.). (Note : ceci est hypothétique - au moment d'écrire ces lignes, NetSuite ne mesure pas les appels d'IA au-delà du pool gratuit, sauf en cas d'utilisation d'OCI.)
- 2. Coûts Oracle Cloud** : Si les organisations choisissent de se connecter via leur propre compte OCI pour le mode « illimité », elles paieront les tarifs OCI pour l'utilisation de l'IA générative. La tarification d'OCI GenAI est actuellement facturée par jeton (token) ou par 1 000 jetons selon le modèle (Source: docs.oracle.com). Les clients NetSuite doivent en tenir compte lors de l'activation du mode illimité, car une utilisation intensive peut générer des coûts cloud importants. D'un autre côté, comme le pool gratuit de NetSuite suffit probablement pour une utilisation légère à modérée, de nombreux clients pourraient fonctionner entièrement sur le niveau inclus.

La position officielle d'Oracle est toutefois de s'assurer que les clients ne soient pas surpris par les coûts de l'IA. Les administrateurs doivent consulter régulièrement les préférences de l'IA pour voir si l'usage gratuit est proche de l'épuisement. L'exemple de suitelet ci-dessus affichait opportunément le `llm.getRemainingFreeUsage()` sur le formulaire (Source: blogs.oracle.com). Pour la production, on pourrait similairement alerter les utilisateurs si les quotas sont bas. Le tableau des préférences de l'IA comporte également une case à cocher « Confirmer » pour opter pour le mode illimité, ce qui en fait une décision explicite.

Études de cas et cas d'utilisation

Bien que N/LLM n'ait été introduit qu'en 2024, plusieurs études de cas et récits pilotes illustrent sa valeur. Certains sont basés sur les analyses des partenaires NetSuite (par exemple, le rapport d'août 2025 de Houseblend) et des anecdotes publiques de clients.

- **BirdRock Home (Étude de cas de vente au détail)** : Détaillant du marché intermédiaire avec plus de 40 000 SKU, BirdRock Home « traite des milliers de commandes quotidiennement » au sein de NetSuite (Source: houseblend.io). Ils disposent de riches données de vente et d'inventaire centralisées dans NetSuite et ont utilisé le NetSuite Analytics Warehouse (NAW) pour les prévisions et la modélisation de l'attrition (churn). Houseblend note que l'utilisation du ML dans NAW par BirdRock a conduit à des « améliorations exploitables de la stratégie produit » (par exemple, l'ajustement des stocks en fonction de l'attrition prédite) (Source: houseblend.io). En s'appuyant sur cela, nous pouvons imaginer BirdRock utilisant N/LLM pour générer automatiquement des descriptions de produits, classer les avis clients par catégories de sentiment, ou répondre à des requêtes en langage naturel telles que « Quelles gammes de produits sont en progression ce trimestre ? » en se basant sur leurs propres données (via RAG). La nature intégrée de NetSuite signifie que les données financières, les commandes et les fournisseurs de BirdRock alimentent tous ces réponses de l'IA, offrant des perspectives transversales qu'un outil autonome ne pourrait pas obtenir.
- **Overture Promotions (Chaîne d'approvisionnement B2B)** : Ce distributeur de produits promotionnels a utilisé NetSuite Analytics Warehouse pour dériver des informations prédictives sur les ventes pour les commandes et la planification. Selon Houseblend, Overture a cité d'importantes optimisations de la chaîne d'approvisionnement grâce à ces prévisions (Source: houseblend.io). À l'avenir, Overture pourrait utiliser N/LLM pour automatiser les récits d'approvisionnement : par exemple, générer un résumé hebdomadaire envoyé par e-mail sur les niveaux de stock (« L'article X est bas par rapport aux ventes historiques ; réapprovisionnement suggéré ») ou un bot de questions-réponses répondant à des requêtes telles que « Quels étaient les 5 meilleurs représentants commerciaux le mois dernier par chiffre d'affaires ? ». Ils pourraient injecter des rapports NAW dans des prompts LLM ou fournir leurs propres documents (par exemple, des CSV de ventes hebdomadaires), garantissant que l'IA opère sur des statistiques internes précises.

- **Chatbots et assistants intelligents** : Plusieurs partenaires NetSuite ont prototypé des chatbots pour les tâches ERP courantes. Par exemple, on pourrait intégrer un chatbot sur le tableau de bord NetSuite qui répond aux questions des utilisateurs sur les détails des fournisseurs, les factures impayées ou les spécifications des produits, en utilisant N/LLM. Scénario type : Un utilisateur demande « Quel est le statut du bon de commande n° 1234 et à qui devons-nous de l'argent ? ». Le backend SuiteScript récupérerait les enregistrements de bons de commande et de fournisseurs pertinents, les formaterait comme contexte et appellerait `generateText` ou un prompt stocké pour produire une réponse cohérente. Cela pourrait considérablement améliorer la productivité des utilisateurs par rapport à la navigation sur plusieurs pages d'enregistrements.
- **Enrichissement du contenu** : Étant donné que NetSuite stocke souvent un texte descriptif minimal (noms courts d'articles, etc.), l'IA générative peut **améliorer la qualité des données**. Par exemple, un script personnalisé pourrait s'exécuter chaque nuit pour remplir ou améliorer les descriptions d'articles : il pourrait prendre les champs existants (catégorie, spécifications fournisseur, notes de ventes passées) et appeler `generateText` pour ajouter une présentation riche du produit. Un guide partenaire suggère d'utiliser N/LLM pour « générer des descriptions de produits, des résumés ou des réponses automatisées » (Source: [gurussolutions.com](https://www.gurussolutions.com)). De même, du contenu généré peut être utilisé pour remplir automatiquement les commentaires de factures ou les suivis clients. Bien entendu, tout contenu généré par l'IA doit être vérifié (peut-être par un autre utilisateur) avant son utilisation finale.
- **Recherche intelligente et recommandations** : En transformant du texte (noms de clients, titres de produits, mots-clés de problèmes) en vecteurs (embeddings), les utilisateurs de NetSuite pourraient créer des fonctionnalités de recherche sémantique. L'exemple des « articles similaires » montre comment trouver des articles connexes ; cela pourrait être étendu pour trouver des transactions liées ou suggérer des articles de vente incitative (upsell). Par exemple, lors de la consultation d'une commande, le système pourrait suggérer « les autres clients qui ont acheté un produit similaire ont également acheté... » en comparant les vecteurs d'embedding des profils clients ou des descriptions de produits. Cela va au-delà de la recherche exacte par mot-clé pour passer à une récupération basée sur le sens, une amélioration précieuse pour les catalogues volumineux.

Ces exemples illustrent le **changement de paradigme** possible avec N/LLM : passer de données statiques à des perspectives dynamiques pilotées par l'IA. La recherche sectorielle soutient ce potentiel. Par exemple, McKinsey (2023) a révélé que l'IA générative apporte le plus de valeur dans des domaines tels que les opérations clients et la vente (Source: www.mckinsey.com) – précisément des cas d'utilisation de NetSuite. Ils soulignent également l'« augmentation » du travail humain : l'IA générative peut gérer la génération de contenu et l'analyse de routine, libérant les experts métier pour un travail stratégique (Source: www.mckinsey.com). Un article du CFO de NetSuite souligne cela dans la finance : au lieu que les comptables rédigent manuellement des rapports narratifs, l'IA peut les rédiger automatiquement, permettant aux équipes financières de « se concentrer sur des tâches plus stratégiques » (Source: the-cfo.io) (Source: the-cfo.io).

Données et statistiques : Au-delà des anecdotes, des données plus larges renforcent la tendance :

- Une enquête Gartner de 2025 (comme noté) a révélé que la GenAI est la solution la plus déployée par les entreprises (Source: www.gartner.com).
- Des secteurs allant de la banque à la vente au détail voient un potentiel de valeur annuelle de plus de 400 milliards de dollars grâce à la GenAI (Source: www.mckinsey.com).
- Comme le note Houseblend, les données intégrées de NetSuite peuvent être une mine d'or : l'utilisation de toutes les données de l'entreprise pour l'IA peut fournir des informations « bien au-delà de ce que proposent les solutions ponctuelles » (Source: houseblend.io).
- Le ROI est une préoccupation critique. Houseblend cite des recherches montrant que le manque de concentration condamne 95 % des pilotes d'IA (Source: houseblend.io). La conception de NetSuite (avec RAG et données internes) répond directement à cela : en utilisant les *données de l'entreprise*, les réponses de l'IA restent pertinentes, rendant le succès plus accessible.

Implications et orientations futures

L'introduction de N/LLM a de vastes implications :

- **Pour les développeurs** : Les programmeurs SuiteScript doivent apprendre les meilleures pratiques de l'IA (ingénierie de requêtes, validation des résultats, gestion des erreurs). Ils disposent désormais d'un nouveau type d'outil dans leur boîte à outils, brouillant la frontière entre le codage et la rédaction de données d'entraînement. Les frameworks de développement vont évoluer. Par exemple, des déclencheurs pourraient invoquer automatiquement des LLM (ex: enrichir les données lors de l'enregistrement d'un record). Les développeurs devront être attentifs à la gestion des coûts/limites et fournir des interfaces permettant aux utilisateurs finaux d'agir de manière responsable. Certains consultants prévoient que N/LLM deviendra une compétence standard dans le développement SuiteScript, au même titre que `N/email` ou `N/record` aujourd'hui.

- **Pour les utilisateurs métier** : Les rôles non techniques peuvent potentiellement utiliser des fonctionnalités activées par l'IA pour des tâches de routine. Les modules par défaut basés sur l'IA de NetSuite (détection d'anomalies financières, assistants de chat) répondront à de nombreux besoins courants sans coût supplémentaire. Cependant, les organisations doivent former leurs utilisateurs à un usage approprié : le remplissage automatique par l'IA peut être utile, mais une confiance aveugle est risquée. Dans les directives d'utilisation, il convient d'informer les utilisateurs finaux que les suggestions de l'IA sont des « brouillons utiles » et non des réponses définitives. Des comités de gouvernance pourraient être nécessaires pour superviser quels scripts peuvent exécuter N/LLM (similairement à la manière dont l'automatisation ou le scripting est contrôlé).
- **Sécurité et conformité** : N/LLM introduit de nouveaux vecteurs. Les données envoyées au LLM peuvent nécessiter une classification (ex: masquage des données personnelles/PII). Les réponses doivent être auditées ; pour la conformité réglementaire, des journaux d'utilisation de l'IA peuvent être nécessaires. Les futures versions pourraient inclure plus de contrôles (par exemple, imposer le chiffrement en transit vers OCI, ou l'hébergement de modèles sur site) pour répondre aux cas d'utilisation sensibles. Les entreprises doivent aligner l'utilisation de N/LLM avec leurs politiques existantes de gouvernance des données et d'éthique.
- **Évolution technologique** : À mesure que la technologie LLM progresse, N/LLM pourrait prendre en charge de nouvelles familles de modèles (ex: GPT-4o dans OCI, modèles de vision). La future feuille de route IA d'Oracle (ex: annonces à SuiteWorld 2025/2026) étendra probablement les capacités de N/LLM. Par exemple, l'intégration de la génération d'images ou de code (ex: `llm.generateImage` ou `llm.executeCustomAction`) pourrait apparaître. L'existence de méthodes de streaming (`generateTextStreamed`) suggère que la prise en charge de la génération multimodale ou de format long est déjà en place. Nous anticipons également de meilleurs outils de développement (ex: améliorations de l'IDE SuiteCloud, simulation de débogage local des appels IA).
- **Perspective de l'industrie** : Le N/LLM de NetSuite s'inscrit dans un changement plus large : pratiquement toutes les grandes applications d'entreprise deviennent « AI-native ». Selon Gartner, d'ici 2030, environ 80 % des logiciels d'entreprise seront pilotés par l'IA/ML (Source: www.gartner.com). La stratégie d'Oracle consistant à intégrer l'IA dans ses plateformes positionne NetSuite pour rivaliser avec Microsoft Dynamics et Salesforce Einstein sur un pied d'égalité. Jacovljevic (Technology Evaluation Centers) a estimé que l'IA de NetSuite pourrait le rendre « à égalité, sinon meilleur, que Microsoft D365 » pour l'ERP du marché intermédiaire (Source: www.techtarget.com), d'autant plus qu'Oracle intègre l'IA à la fois dans les familles de produits NetSuite et Oracle Fusion. Cela suggère un investissement continu : attendez-vous à voir arriver davantage d'outils d'IA au sein de NetSuite (planification, approvisionnement, CRM).
- **Défis** : Plusieurs défis subsistent. Les hallucinations restent un problème pratique : malgré le RAG, les LLM peuvent mélanger les données de manière imprévisible. Il est crucial de s'assurer que, par exemple, les récits financiers ne fabriquent pas de chiffres. L'approche de NetSuite consistant à fournir des citations (via des objets `Citation`) aide à tracer les réponses, mais la responsabilité incombe ultimement aux développeurs. Le versionnage et l'audit des prompts (et la documentation de l'utilisation des LLM) seront importants pour la conformité. De plus, la performance (latence) et le passage à l'échelle pour des volumes d'appels élevés sont des préoccupations pour les grands clients ; cela pourrait favoriser des fonctionnalités comme le traitement par lots ou les appels LLM asynchrones.
- **Perspectives éthiques et de gouvernance** : En termes d'éthique, l'intégration de l'IA dans l'ERP souligne la nécessité de politiques d'éthique de l'IA en entreprise. De nombreux cadres (ex: les directives de l'IA Act de l'UE) s'appliqueront à des fonctionnalités comme N/LLM. Le modèle multi-tenant de NetSuite complique les choses : Oracle doit s'assurer que les données d'un client ne peuvent pas s'infiltrer dans le contexte IA d'un autre. L'IA générative d'OCI est une infrastructure partagée, mais Oracle maintient vraisemblablement l'isolation des tenants (Oracle n'a pas détaillé publiquement le modèle de multi-tenancy pour GenAI). Les clients doivent vérifier que l'utilisation de l'IA est conforme à la résidence des données ou aux règles réglementaires (en utilisant des contraintes de région si nécessaire).
- **Opinions d'analystes et d'experts** : Des voix de l'industrie se sont exprimées. Comme noté, Holger Mueller a loué l'approche plateforme (Source: www.techtarget.com). Predrag Jakovljevic a observé que l'IA générative « apprendra du contexte et du vocabulaire [du client] » au fil du temps (Source: www.techtarget.com), impliquant un affinement par l'usage. D'autres avertissent que le simple fait de disposer de l'API ne suffit pas ; les organisations ont besoin de data scientists et de stratèges en IA pour en tirer le meilleur parti. L'investissement massif d'Oracle dans l'IA signale que les fonctionnalités génératives ne feront que croître dans NetSuite.

Conclusion

Le module N/LLM de NetSuite représente un bond en avant significatif : il apporte les capacités LLM de pointe directement dans le framework d'extension de l'ERP. Les développeurs peuvent désormais écrire du SuiteScript qui exploite les modèles d'IA générative d'OCI pour la génération de texte, la synthèse, la recherche sémantique et plus encore, le tout adapté aux propres données de l'entreprise. Cela ouvre de nombreux cas d'utilisation – des chatbots et de la recherche intelligente à la rédaction automatisée de rapports – qui étaient auparavant irréalisables à l'intérieur d'un ERP.

Cependant, un grand pouvoir implique de grandes responsabilités. Une utilisation efficace de N/LLM nécessite une attention particulière à la gouvernance (quotas et concurrence), à la qualité des données (données erronées en entrée, résultats erronés en sortie) et à la conformité (sécurité et confidentialité). NetSuite a intégré des garde-fous (suivi de l'utilisation, quotas, clauses de non-responsabilité), mais les organisations doivent également établir leurs propres pratiques (validation des données, formation des utilisateurs, pistes d'audit). La transition vers des processus enrichis par l'IA modifiera également les rôles : les analystes métier pourraient s'appuyer davantage sur des « copilotes IA » pour analyser les données, et les développeurs auront besoin d'une culture de l'IA.

D'un point de vue stratégique, N/LLM consolide la position d'Oracle NetSuite dans le paysage des ERP activés par l'IA. En incluant ces fonctionnalités sans coût supplémentaire et en les intégrant étroitement au modèle de données unifié, NetSuite exploite son avantage principal (les données centralisées) pour offrir des perspectives et une automatisation plus riches. Les clients qui adoptent ces outils sont en mesure d'améliorer leur productivité et leur vitesse de prise de décision. Par exemple, la possibilité de poser des questions en langage naturel sur les données ERP (comme le montre le suitelet Sales Insights) revient à avoir un expert à disposition 24h/24 et 7j/7.

À l'avenir, nous prévoyons une amélioration continue de la boîte à outils d'IA générative de NetSuite. Les prochaines versions pourraient introduire de meilleurs modèles (LLM plus puissants), des capacités étendues pour Prompt Studio et éventuellement des analyses pilotées par l'IA. La conception de N/LLM – avec prise en charge du streaming, structures de chat et citations de documents – montre qu'Oracle anticipe des modèles d'utilisation de l'IA complexes.

En résumé, le module NetSuite N/LLM est un pont robuste et prêt pour l'entreprise vers l'IA moderne. Il est soutenu par une documentation complète et des politiques d'utilisation encadrées, tout en étant assez flexible pour des solutions créatives. En suivant les meilleures pratiques et les contraintes énoncées ci-dessus, les organisations peuvent exploiter les LLM en toute sécurité pour libérer tout le potentiel de leurs données NetSuite. Toutes les affirmations et détails ici sont étayés par la propre documentation d'aide d'Oracle (Source: docs.oracle.com) (Source: docs.oracle.com) (Source: docs.oracle.com) (Source: docs.oracle.com), les blogs de développeurs Oracle (Source: blogs.oracle.com) (Source: blogs.oracle.com), les guides de la communauté NetSuite et les blogs de partenaires (Source: gurussolutions.com) (Source: houseblend.io), ainsi que les rapports de l'industrie (Source: www.mckinsey.com) (Source: www.gartner.com) (Source: www.gartner.com) ont été largement utilisés. Toutes les données et citations mentionnées proviennent de ces sources.

Références : La documentation officielle d'Oracle (API d'IA générative SuiteScript 2.x) (Source: docs.oracle.com) (Source: docs.oracle.com) (Source: docs.oracle.com) (Source: docs.oracle.com), les blogs de développeurs Oracle (Source: blogs.oracle.com) (Source: blogs.oracle.com), les guides de la communauté NetSuite et les blogs de partenaires (Source: gurussolutions.com) (Source: houseblend.io), ainsi que les rapports de l'industrie (Source: www.mckinsey.com) (Source: www.gartner.com) (Source: www.gartner.com) ont été largement utilisés. Toutes les données et citations mentionnées proviennent de ces sources.

Étiquettes: netsuite, suitescript-21, module-nllm, ia-generative, rag, gouvernance-llm, oracle-oci, embeddings-vectoriels, limites-api, ia-erp

AVERTISSEMENT

Ce document est fourni à titre informatif uniquement. Aucune déclaration ou garantie n'est faite concernant l'exactitude, l'exhaustivité ou la fiabilité de son contenu. Toute utilisation de ces informations est à vos propres risques. Houseblend ne sera pas responsable des dommages découlant de l'utilisation de ce document. Ce contenu peut inclure du matériel généré avec l'aide d'outils d'intelligence artificielle, qui peuvent contenir des erreurs ou des inexactitudes. Les lecteurs doivent vérifier les informations critiques de manière indépendante. Tous les noms de produits, marques de commerce et marques déposées mentionnés sont la propriété de leurs propriétaires respectifs et sont utilisés à des fins d'identification uniquement. L'utilisation de ces noms n'implique pas l'approbation. Ce document ne constitue pas un conseil professionnel ou juridique. Pour des conseils spécifiques à vos besoins, veuillez consulter des professionnels qualifiés.