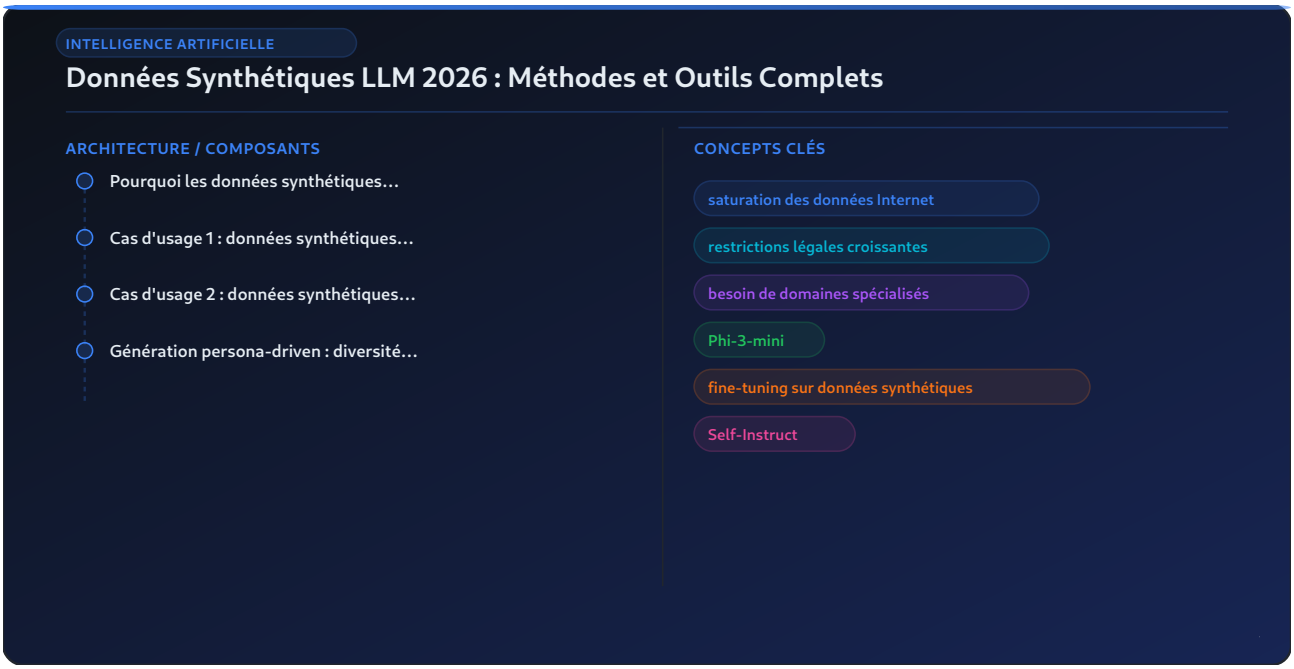
Données Synthétiques pour l'Entraînement LLM 2026 : Méthodes et Outils

> Guide données synthétiques pour l'entraînement LLM en 2026 — génération, validation, déduplication, privacy-preserving et outils Argilla, DataDreamer.

La génération de données synthétiques pour l'entraînement et l'évaluation des LLM constitue en 2026 l'une des directions techniques les plus actives de la recherche en intelligence artificielle, avec des implications profondes sur la qualité des modèles déployés, la conformité réglementaire, et l'économie du développement IA. Le constat qui a catalysé cet essor est paradoxal : les LLM les plus performants de 2025-2026 ont été entraînés en partie sur des données générées par des LLM précédents. Phi-3 de Microsoft, entraîné presque exclusivement sur des données synthétiques générées par GPT-4, démontre qu'un modèle compact peut atteindre des performances remarquables sans corpus de données réelles massif. Gemma de Google utilise des données synthétiques pour combler les lacunes de couverture de son corpus d'entraînement. Cette tendance reflète une réalité économique et réglementaire : les données réelles de haute qualité sont rares, coûteuses à annoter, souvent soumises à des restrictions légales (droits d'auteur, RGPD, secret médical), et difficiles à équilibrer selon les dimensions souhaitées. Les données synthétiques offrent une alternative qui peut être générée à la demande, dans n'importe quelle langue, sur n'importe quel domaine, avec n'importe quelle distribution, et sans les contraintes légales des données réelles — sous réserve que la génération elle-même respecte certaines conditions. Ce guide exhaustif couvre les cas d'usage des données synthétiques (fine-tuning, RAG, évaluation), les méthodes de génération (self-instruct, persona-driven, constitutional), les pipelines de validation et déduplication, la protection de la vie privée (DP-SGD, synthétisation de données sensibles), les outils majeurs (Argilla, DataDreamer, Distilabel), et le cadre légal RGPD applicable aux organisations françaises.



Pourquoi les données synthétiques sont devenues incontournables en 2026

Trois facteurs convergents expliquent l'explosion de l'intérêt pour les données synthétiques dans le développement LLM. Premièrement, la **saturation des données Internet** : les estimations suggèrent que les LLM de dernière génération ont consommé une part substantielle du texte disponible publiquement sur Internet. Les nouvelles données de haute qualité se raréfient tandis que la demande pour entraîner et affiner des modèles toujours plus performants continue de croître. Deuxièmement, les **restrictions légales croissantes** : la directive sur le droit d'auteur de l'UE et des décisions judiciaires récentes ont créé une incertitude légale sur l'utilisation de données scraped pour l'entraînement IA, poussant les organisations à chercher des alternatives légalement plus solides. Troisièmement, le **besoin de domaines spécialisés** : pour fine-tuner un LLM sur la cybersécurité, le droit fiscal français, ou la médecine urgentiste pédiatrique, les données réelles de qualité sont extrêmement rares et souvent confidentielles. Les données synthétiques générées par des LLM généralistes guidés par des experts constituent une alternative pragmatique.

La démonstration la plus frappante de la viabilité des données synthétiques est le travail de Microsoft sur **Phi-3-mini** : un modèle de 3.8 milliards de paramètres entraîné principalement sur des données synthétiques (des "livres synthétiques" générés par GPT-4 couvrant des domaines

variés à un niveau de difficulté adapté) atteignant des performances proches de Llama 3 8B sur de nombreux benchmarks malgré sa taille significativement plus petite. Ce résultat démontrant qu'une données synthétiques de haute qualité peut surpasser quantitativement des données réelles de qualité moindre a changé la perception de la communauté.

Cas d'usage 1 : données synthétiques pour le fine-tuning

Le **fine-tuning sur données synthétiques** vise à adapter un LLM généraliste à un domaine ou une tâche spécifique sans disposer de données d'entraînement réelles suffisantes. Le processus typique : identifier les types de tâches que le modèle fine-tuné devra accomplir (classification d'incidents de sécurité, génération de rapports de vulnérabilités, réponses à des questions de conformité RGPD), générer synthétiquement des milliers à centaines de milliers d'exemples (instruction + réponse attendue) couvrant ces tâches, valider la qualité de ces exemples, et effectuer le fine-tuning via des techniques efficaces comme LoRA ou QLoRA.

> ⚡ **Retour terrain**
>
> Pour une banque régionale qui voulait automatiser la rédaction de ses synthèses de risque, j'ai benchmarké GPT-4o, Claude 3.5 Sonnet et Mistral Large sur un corpus de 200 notes anonymisées. La métrique critique n'était pas la précision brute mais le taux de fabrication de chiffres — seul Claude atteignait 0 % sur ce critère sur ce corpus précis. La conclusion : choisir un modèle pour une tâche critique exige des benchmarks sur vos propres données, pas sur les leaderboards publics.

La technique **Self-Instruct** (Wang et al., 2022) popularise l'approche bootstrapping : en partant d'un ensemble réduit d'instructions "seed" rédigées manuellement par des experts du domaine, un LLM génère itérativement de nouvelles instructions similaires mais distinctes. Ces nouvelles

instructions sont filtrées pour éliminer les duplicates et les exemples de mauvaise qualité, puis utilisées pour générer des paires instruction/réponse qui alimentent le fine-tuning. Des variantes comme **Alpaca** (Stanford) et **Vicuna** ont démontré qu'un modèle Llama fine-tuné sur 50 000 à 200 000 exemples Self-Instruct peut approcher les performances de ChatGPT sur des tâches de conversation générale — une démonstration spectaculaire du potentiel de cette approche à faible coût.

Cas d'usage 2 : données synthétiques pour le RAG et les datasets d'évaluation

La **génération de données synthétiques pour les systèmes RAG** couvre deux besoins distincts : l'entraînement de composants du pipeline RAG (notamment les embeddings et les rerankers) et la construction de datasets d'évaluation pour mesurer les performances du système sur des questions représentatives du cas d'usage. Pour l'évaluation, générer synthétiquement un dataset de paires (question, document pertinent, réponse correcte) à partir d'un corpus de documents réels permet de construire rapidement un dataset golden RAG sans annotation humaine coûteuse.

La technique **RAG dataset generation** utilise un LLM pour lire chaque document du corpus et générer des questions auxquelles ce document répond, ainsi que les réponses correspondantes. Des variantes plus sophistiquées génèrent des questions multi-hop (nécessitant de combiner des informations de plusieurs documents), des questions de raisonnement, et des questions piège où le document contient une information trompeuse. Le framework **RAGAS** propose des fonctionnalités intégrées pour la génération de datasets d'évaluation synthétiques pour les systèmes RAG, couvrant les dimensions de faithfulness, relevancy, et context precision.

À RETENIR

À retenir — Données Synthétiques pour les LLM

Self-Instruct : bootstrap d'un dataset depuis 100 seeds → 100K exemples de qualité

Phi-3 : preuve que données synthétiques seules peuvent produire un modèle compétitif

Distilabel : pipeline Python modulaire de génération + validation + déduplication

Argilla : annotation et curation humaine du dataset synthétique, indispensable

DP-SGD : entraînement différentiellement privé pour données synthétiques de données sensibles

Qualité > Quantité : 10K exemples excellents battent 100K exemples médiocres

Génération persona-driven : diversité et représentativité

L'approche **persona-driven generation**, popularisée notamment par les travaux de Tulu 3 (Allen AI, 2024) et Magpie (diverses variantes), vise à améliorer la diversité du dataset synthétique en définissant des personas variées (profils d'utilisateurs avec des caractéristiques différentes : niveau d'expertise, domaine professionnel, style de communication, contexte culturel) et en générant des instructions dans le style de chaque persona. Cette diversification réduit les biais de distribution inhérents à la génération avec un seul "style LLM" et produit des modèles fine-tunés plus robustes à la variété des utilisateurs réels.

L'outil **Magpie** exploite une propriété spécifique des LLMs instruction-tuned pour générer des paires (instruction, réponse) sans avoir à fournir des instructions seed : en soumettant uniquement le template de début de conversation utilisateur au modèle, il génère spontanément une instruction plausible. Ce processus, répété des milliers de fois, produit un dataset diversifié représentant naturellement les types de requêtes que les utilisateurs formulent réellement au modèle — une source de données beaucoup plus représentative de la distribution réelle d'usage que des instructions manuellement conçues.

Pipelines de validation et contrôle qualité

La qualité d'un dataset de données synthétiques dépend directement de la rigueur du pipeline de validation. Des données synthétiques non validées peuvent introduire des biais systématiques, des hallucinations persistantes (le LLM génère des "faits" incorrects qui s'accumulent dans le dataset), ou des formats incorrects qui dégradent le fine-tuning. Un pipeline de validation complet inclut plusieurs étapes distinctes.

La **déduplication** est la première étape indispensable. Des embeddings des instructions générées sont calculés, et des paires avec une similarité cosinus supérieure à un seuil (0.85 typiquement) sont identifiées comme quasi-duplicates et dédupliquées. Des outils comme **datasketch** (MinHash LSH) ou **semdedup** (déduplication par similarité sémantique) automatisent ce processus à grande échelle. La **validation de format** vérifie que les réponses générées respectent les contraintes de format attendues (longueur, structure, présence des éléments requis). La **validation qualitative** utilise un LLM-as-judge pour évaluer la pertinence, l'exactitude, et la complétude de chaque paire instruction/réponse, filtrant les exemples sous un seuil de qualité minimum.

La **détection de contamination** est une étape souvent négligée mais critique : s'assurer que les données synthétiques ne "fuient" pas des exemples des benchmarks d'évaluation standard (MMLU, HumanEval, etc.) qui seraient alors mémorisés et produiraient des performances artificiellement gonflées. Des outils comme **n-gram overlap detection** ou des modèles de détection de contamination spécifiques permettent d'identifier et d'éliminer ces fuites potentielles.

Distilabel : le pipeline open-source de référence

Distilabel, développé par Argilla, est devenu en 2025-2026 le framework open-source de référence pour la génération et la validation de datasets LLM. Son architecture modulaire basée sur des "steps" Python composables permet de construire des pipelines de génération complexes facilement : un step de génération d'instructions (via un LLM), un step de génération de réponses

(via un LLM différent ou le même), un step de scoring qualité (LLM-as-judge), un step de déduplication, et un step d'export dans les formats HuggingFace standard.

Distilabel supporte nativement les principaux providers LLM (OpenAI, Anthropic, Google Gemini, Hugging Face Inference) et des modèles open-source via vLLM ou Ollama. Son intégration avec **Argilla** permet un workflow de génération + annotation humaine fluide : les données générées synthétiquement peuvent être exportées vers Argilla pour révision et correction par des experts du domaine, puis réintégrées dans le pipeline. La version 1.x de Distilabel (2024-2025) a introduit une API streaming permettant de traiter des datasets de taille arbitraire sans surcharger la mémoire — une amélioration critique pour les pipelines produisant des millions d'exemples.

Argilla : annotation humaine et curation du dataset synthétique

Argilla (open-source, auto-hébergeable) est la plateforme d'annotation de données pour LLM la plus utilisée en France et en Europe en 2026. Elle permet aux experts du domaine de revoir, corriger, et approuver les données générées synthétiquement via une interface web intuitive, produisant le "human-in-the-loop" indispensable pour garantir la qualité des données sensibles (médecine, droit, finance). L'intégration d'Argilla dans le pipeline Distilabel est native et documentée, permettant un workflow continu de génération → annotation → fine-tuning.

L'annotation humaine dans Argilla ne nécessite pas nécessairement de réviser chaque exemple synthétique. Pour des datasets de grande taille (100K+ exemples), une stratégie d'échantillonnage stratifié identifie les exemples les plus représentatifs et les plus incertains (score de qualité LLM-as-judge intermédiaire) pour la révision humaine, en laissant les exemples à haute confiance et basse confiance être filtrés automatiquement. Cette approche "human-in-the-loop efficace" optimise l'investissement en annotation humaine tout en maintenant la qualité du dataset final.

DataDreamer : workflows de données synthétiques pour la recherche

DataDreamer (Patel & Rafiei, 2024) est un framework Python orienté recherche pour la création de pipelines reproductibles de génération de données LLM et de fine-tuning. Conçu pour faciliter la reproductibilité des expériences de recherche, DataDreamer met l'accent sur la traçabilité des données générées (chaque exemple est associé aux métadonnées de génération : modèle utilisé, version, paramètres, timestamp) et sur la mise en cache intelligente pour éviter de re-générer des données identiques entre expériences.

DataDreamer propose des abstractions haut niveau pour les tâches courantes : génération d'instructions via self-instruct ou persona-driven, constitution de datasets de préférence (paires de réponses ranked pour le RLHF/DPO), et évaluation via LLM-as-judge. Son architecture de "Tasks" et "Trainers" permet de chaîner les étapes de génération et de fine-tuning en quelques dizaines de lignes de Python, réduisant considérablement le boilerplate de code nécessaire pour les expériences de recherche reproduisant ou étendant des travaux existants.

Privacy-preserving : DP-SGD et données médicales

La génération de données synthétiques à partir de données réelles sensibles — dossiers médicaux, transactions financières, données RH — pose des défis de protection de la vie privée spécifiques. Une donnée synthétique n'est pas intrinsèquement anonyme : si le processus de synthèse "mémorise" des cas individuels rares ou des configurations de features exceptionnelles, un adversaire disposant d'informations partielles peut remonter aux données originales via des attaques de réidentification.

La **Differential Privacy (DP)** offre des garanties formelles contre ce risque. L'algorithme **DP-SGD** (Differentially Private Stochastic Gradient Descent), implémenté dans TensorFlow Privacy et Opacus (PyTorch), ajoute du bruit calibré aux gradients pendant l'entraînement, garantissant que la présence ou l'absence d'un individu dans le dataset d'entraînement ne peut pas être détectée avec une probabilité supérieure à un paramètre epsilon contrôlable. Les données synthétiques générées par ce modèle DP héritent de ses garanties formelles. Le coût : une

dégradation des performances du modèle synthétique qui dépend du niveau de privacy souhaité (epsilon faible = forte privacy = plus faible utilité).

En France, le cadre légal applicable aux données synthétiques médicales est en cours de clarification. La CNIL a publié en 2024 une note d'orientation précisant que les données synthétiques peuvent, sous certaines conditions, ne pas être considérées comme des données personnelles au sens du RGPD — ouvrant la voie à leur utilisation sans les contraintes des données réelles pour l'entraînement de modèles IA médicaux. Ces conditions incluent notamment la démonstration que le risque de réidentification est négligeable, via des méthodes d'audit formel comme la DP ou des tests empiriques de membership inference attack.

Outil	Type	Points forts	Licence
Distilabel	Pipeline génération	Modulaire, HF natif, streaming	Apache 2.0
Argilla	Annotation humaine	Interface web, self-hosted	Apache 2.0
DataDreamer	Framework recherche	Reproductibilité, traçabilité	MIT
Magpie	Génération auto instructions	Sans seed, représentatif usage réel	MIT
Opacus	DP-SGD PyTorch	Garanties DP formelles, intégration PyTorch	Apache 2.0

Légalité RGPD des données synthétiques

La question de la légalité des données synthétiques au regard du RGPD est nuancée et dépend fortement du contexte de génération. Si les données synthétiques sont générées à partir de données personnelles réelles (anonymisation synthétique), la CNIL considère qu'elles restent potentiellement dans le champ du RGPD si le risque de réidentification n'est pas négligeable. En revanche, des données synthétiques générées par un LLM sans référence directe à des individus réels ne constituent généralement pas des données personnelles et échappent au RGPD.

Le point critique est la **source** des données d'entraînement du LLM générateur : si ce LLM a été entraîné sur des données personnelles sans base légale adéquate, les données synthétiques qu'il

génère peuvent "hériter" de cette problématique. En pratique, l'utilisation de LLM commerciaux (GPT-4o, Claude) ou de modèles open-source entraînés sur du texte public pour générer des données synthétiques de domaines non personnels (cybersécurité, documentation technique, code) est généralement non problématique sous le RGPD. Pour les données synthétiques dans des domaines sensibles (santé, finances personnelles), une analyse juridique spécifique et potentiellement une consultation CNIL est recommandée.

La France dispose d'une position spécifique sur ce sujet via les travaux du **Health Data Hub** qui a développé des lignes directrices sur la génération de données médicales synthétiques pour la recherche, applicable par extension à d'autres domaines de données sensibles. Notre guide sur la gouvernance IA et l'AI Act complète l'analyse réglementaire avec les obligations applicables aux systèmes IA entraînés sur des données synthétiques.

Constitutional data generation : aligner les données sur des valeurs

La **constitutional data generation**, inspirée du Constitutional AI d'Anthropic, consiste à générer des données synthétiques conformes à un ensemble de principes définis explicitement. Pour générer un dataset de fine-tuning pour un assistant IA d'entreprise, une "constitution" pourrait inclure des principes comme "les réponses doivent toujours vérifier les informations avant d'affirmer des faits", "les réponses doivent recommander de consulter un expert pour les décisions à fort impact", et "les réponses doivent respecter les politiques de confidentialité de l'entreprise". Le LLM générateur applique ces principes lors de la génération de chaque réponse, et un modèle critique vérifie leur respect avant d'inclure l'exemple dans le dataset.

Cette approche garantit que le modèle fine-tuné sur le dataset constitutionnel intégrera ces valeurs dans son comportement — approche bien plus robuste que d'essayer d'imposer ces valeurs via un système prompt après fine-tuning. La difficulté réside dans la définition d'une constitution suffisamment précise pour guider le comportement sans être trop contraignante pour permettre la diversité nécessaire à un bon fine-tuning. Des itérations de validation avec des experts du domaine sont indispensables pour affiner la constitution jusqu'à produire un dataset de qualité satisfaisante.

DPO et préférence data : générer des datasets de comparaison

L'entraînement par **Direct Preference Optimization (DPO)** nécessite des datasets de préférence : des triplets (instruction, réponse préférée, réponse rejetée) permettant au modèle d'apprendre à favoriser certains types de réponses. La génération synthétique de ces datasets de préférence constitue un cas d'usage majeur des données synthétiques en 2026.

Le workflow typique : générer deux à quatre réponses différentes pour chaque instruction (en variant le modèle, la température, ou le prompt système), les faire évaluer par un LLM-as-judge ou par des annotateurs humains pour identifier la préférée et la rejetée, et construire le dataset de triplets. Des frameworks comme **UltraFeedback** (générer des données de préférence à grande échelle via GPT-4 comme juge) ou **Orca DPO Pairs** ont démontré que des datasets de préférence synthétiques de haute qualité pouvaient améliorer significativement les performances des modèles fine-tunés via DPO. La clé est la qualité du modèle-juge : utiliser GPT-4o ou Claude Opus comme juge produit des annotations de préférence de meilleure qualité que des modèles plus légers, mais à un coût plus élevé.

Questions fréquentes sur les données synthétiques pour LLM

Les données synthétiques peuvent-elles remplacer complètement les données réelles ?

Non dans le cas général, mais pour des tâches bien définies et avec des modèles générateurs de haute qualité, les données synthétiques peuvent approcher ou égaler la performance des données réelles. L'exemple Phi-3 (Microsoft) démontre que pour des tâches de raisonnement général, des données synthétiques peuvent suffire. Pour des tâches nécessitant des connaissances très spécialisées ou des nuances culturelles fines, un mélange de données réelles et synthétiques reste préférable.

Comment éviter le "model collapse" avec des données synthétiques ?

Le model collapse désigne la dégradation progressive des performances lorsqu'un modèle est entraîné itérativement sur ses propres outputs. Pour l'éviter : toujours maintenir un ancrage dans des données réelles de haute qualité (même en petite quantité), valider les données synthétiques avec des LLM-as-judge différents du modèle générateur, surveiller la diversité du dataset généré (entropie des embeddings), et limiter le nombre d'itérations bootstrapping. Des études théoriques (Shumailov et al., 2023) ont formalisé ce risque, et les bonnes pratiques actuelles préconisent un ratio minimum de 20 à 30% de données réelles dans les mixtures d'entraînement.

Quel modèle utiliser comme "teacher" pour générer les données synthétiques ?

Pour un fine-tuning de qualité, le modèle "teacher" (générateur des données) doit être significativement plus capable que le modèle "student" (fine-tuné). GPT-4o et Claude Opus sont les meilleurs teachers pour des datasets généralistes. Pour réduire les coûts, Claude Sonnet ou GPT-4o-mini offrent un bon compromis qualité/prix. Pour les équipes souhaitant éviter les dépendances cloud, Llama 3 70B ou Mixtral 8x22B auto-hébergés constituent des alternatives acceptables pour des domaines bien représentés dans leurs données d'entraînement.

Comment valider qu'un dataset synthétique améliore effectivement les performances ?

Diviser le dataset synthétique en un split d'entraînement et un split de validation (20%) ; effectuer le fine-tuning sur le split d'entraînement ; mesurer les performances sur le split de validation ET sur un dataset golden humain préexistant. Si le modèle fine-tuné surpasse le modèle de base sur les deux datasets, le dataset synthétique a apporté une valeur réelle. Si les gains ne se généralisent pas au dataset humain, les données synthétiques ont probablement introduit des biais spécifiques qui ne reflètent pas la distribution réelle des tâches.

Cas pratique : génération d'un dataset cybersécurité pour une MSSP française

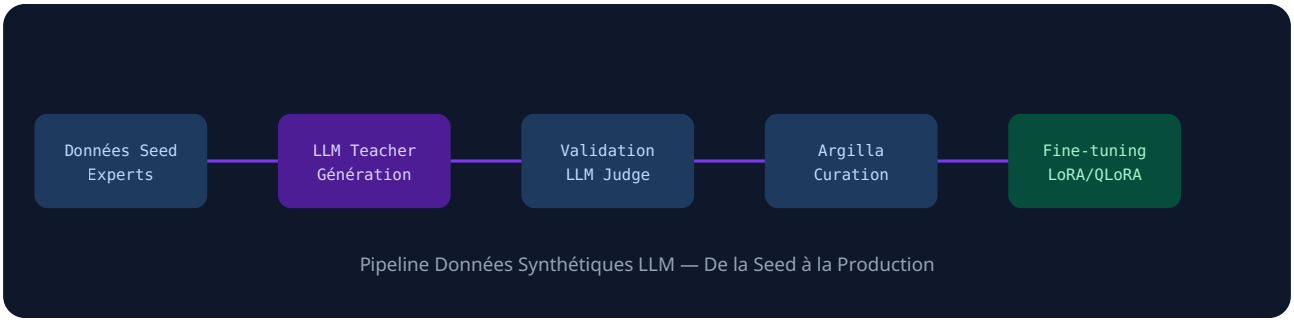
Un exemple concret (MSSP française, 2025, anonymisé) illustre la démarche complète. L'organisation souhaitait fine-tuner un LLM compact (Llama 3 8B) pour automatiser la classification et le résumé des alertes SOC selon sa taxonomie d'incidents interne. Les données réelles d'alertes historiques étaient disponibles mais insuffisamment labelisées (seulement 2 000 exemples annotés sur 50 000 alertes) et ne couvrant pas toutes les catégories d'incidents pertinentes.

La stratégie synthétique : utiliser les 2 000 exemples annotés réels comme données seed pour générer synthétiquement 50 000 exemples additionnels via Distilabel + Claude Sonnet. Chaque alerte synthétique était générée en demandant à Claude de créer une alerte réaliste correspondant à une catégorie et une sévérité spécifiques, avec les IOCs, les timestamps, et le contexte système. Les experts SOC ont validé via Argilla un échantillon de 5 000 exemples représentatifs. Le modèle fine-tuné sur ce dataset mixte (2 000 réels + 50 000 synthétiques) atteignait 89% de précision sur la classification, vs 72% pour le modèle de base et 85% pour un modèle entraîné uniquement sur les 2 000 exemples réels. Consultez notre guide LLMOps pour les aspects de monitoring de ce type de modèle fine-tuné en production. Pour un accompagnement dans la construction de datasets spécialisés cybersécurité, contactez notre équipe via la page consulting.

Perspectives : données synthétiques et réglementation IA en 2027

L'**AI Act européen** impose aux fournisseurs de modèles à usage général (GPAI) une documentation des données d'entraînement utilisées, incluant les données synthétiques. Cela signifie que les organisations développant des LLM sur des données synthétiques devront maintenir une traçabilité complète du pipeline de génération : quel modèle a servi de teacher, avec quels paramètres, sur quelle base légale, et avec quelles validations qualité. Cette exigence de traçabilité converge avec les bonnes pratiques de DataDreamer et renforce la nécessité de pipelines de données synthétiques documentés et auditables.

Une tendance émergente est la **certification des datasets synthétiques** : des tiers de confiance (organismes de certification comme le BSI en Allemagne, ANSSI en France dans une approche adaptée) pourraient à terme attester de la qualité et de la conformité des datasets synthétiques utilisés pour entraîner des systèmes IA à haut risque. Cette certification faciliterait la démonstration de conformité AI Act et créerait un marché pour les datasets synthétiques "premium" validés par des experts. Les organisations françaises proactives qui documentent rigoureusement leurs pipelines de données synthétiques aujourd'hui seront mieux positionnées pour cette évolution réglementaire.



Reward modeling et données de préférence pour RLHF

Le **Reinforcement Learning from Human Feedback (RLHF)** et ses variantes modernes (RLAIF — de l'IA plutôt que d'humains, DPO — Direct Preference Optimization, KTO — Kahneman-Tversky Optimization) nécessitent tous des données de préférence : des paires de réponses comparant une réponse "meilleure" à une réponse "moins bonne" selon des critères définis. La génération synthétique de ces paires de préférence à grande échelle est l'un des usages les plus transformateurs des données synthétiques dans le développement LLM en 2026.

Le workflow de génération de **reward data synthétique** fonctionne ainsi : pour chaque instruction du dataset, générer plusieurs réponses avec différentes configurations (modèles différents, températures différentes, prompts système variés), les soumettre à un LLM-as-judge fort (GPT-4o, Claude Opus) qui les rank selon des critères explicites (exactitude, utilité, harmlessness, longueur appropriée), et construire des paires (réponse préférée, réponse rejetée) à partir de ces rankings. La qualité du modèle de reward appris depuis ces données détermine

largement la qualité du modèle RLHF final — ce qui explique l'attention particulière apportée à la qualité du LLM-as-judge et à la cohérence des critères de préférence.

Des datasets de préférence synthétiques ouverts sont devenus des ressources importantes pour la communauté : **UltraFeedback** (64K comparaisons générées par GPT-4 sur des instructions Alpaca et Evol-Instruct), **Orca DPO Pairs** (paires de préférence sur des raisonnements multi-étapes), et **HelpSteer2** (Nvidia, avec annotations de plusieurs dimensions orthogonales de préférence). Ces datasets open-source permettent à des équipes sans ressources pour générer leurs propres données de préférence de tirer parti du travail de la communauté.

Code synthétique et datasets de programmation

La génération de **code synthétique pour le fine-tuning** de modèles de codage représente un cas d'usage mature et bien documenté des données synthétiques. Des modèles comme **DeepSeek Coder**, **Starcoder2**, et les variantes Code Llama ont tous utilisé des données synthétiques pour améliorer leurs performances sur des tâches de codage spécifiques. La génération synthétique permet de créer des exercices de codage dans des langages moins représentés dans les données d'entraînement classiques (Rust, Go, Julia), des problèmes avec des contraintes spécifiques (optimisation de performance, sécurité, accessibilité), et des exemples de débogage ou de refactoring difficiles à trouver dans du code open-source réel.

La validation du code synthétique est particulièrement efficace car elle peut être en partie **automatisée** : le code généré peut être exécuté et testé automatiquement, et les exemples dont le code ne passe pas les tests sont éliminés du dataset. Cette propriété de validabilité automatique du code rend la génération de datasets de code synthétique plus fiable que la génération de données textuelles qui nécessitent un LLM-as-judge subjectif. Des outils comme **EvalPlus** et **HumanEval** fournissent des harnesses de test standard pour évaluer la correction du code généré synthétiquement avant inclusion dans le dataset d'entraînement.

Gestion de la diversité linguistique dans les datasets synthétiques

La génération de données synthétiques en **français** et dans d'autres langues non-anglaises est un enjeu stratégique pour les organisations françaises souhaitant fine-tuner des LLM performants dans leur langue. Les LLM généralistes tendent à produire des données synthétiques de meilleure qualité en anglais qu'en français — leurs données d'entraînement étant dominées par du texte anglais. Cette inégalité se traduit par une qualité généralement légèrement inférieure des données synthétiques en français comparées à leurs équivalents anglais.

Des approches pour améliorer la qualité des données synthétiques en français : utiliser des modèles ayant un meilleur équilibre multilingue comme **Mistral Large** ou **Llama 3 70B** (entraîné sur plus de données multilingues que les versions précédentes), traduire des datasets synthétiques de haute qualité en anglais vers le français via des LLM de traduction, et combiner génération directe et traduction dans une proportion à calibrer selon la tâche. Le projet **CroissantLLM** (Université Paris-Saclay) développe des LLM bilingues franco-anglais de haute qualité avec des datasets d'entraînement soigneusement équilibrés entre les deux langues — un modèle potentiellement excellent teacher pour la génération de données synthétiques en français.

Détection de données synthétiques et intégrité des benchmarks

La prolifération des données synthétiques soulève un risque d'intégrité important pour les benchmarks d'évaluation des LLM : si les données d'entraînement d'un modèle contiennent des exemples des benchmarks publics (contamination), les performances mesurées sur ces benchmarks seront artificiellement gonflées et non représentatives des capacités réelles du modèle. La détection de cette **contamination de benchmark** est devenue un sujet de préoccupation majeur dans la communauté IA.

Des techniques de détection de contamination incluent : le **membership inference attack** (tester si le modèle "reconnaît" des exemples spécifiques de benchmarks en analysant sa distribution de probabilités sur ces exemples), la **canary insertion** (insérer des exemples synthétiques uniques dans les benchmarks et vérifier si les modèles les reproduisent), et l'**overlap n-gram detection**

(comparer les n-grammes des données d'entraînement avec ceux des benchmarks). Ces techniques sont imparfaites mais fournissent des signaux utiles pour évaluer la confiance à accorder aux performances déclarées d'un modèle. La communauté open-source de benchmarking (EleutherAI, Hugging Face) investit significativement dans le développement de benchmarks "dynamic" qui se renouvellent régulièrement pour réduire le risque de contamination.

Retour terrain : ce que les praticiens français découvrent

L'expérience terrain des praticiens français qui implémentent des pipelines de données synthétiques en production révèle plusieurs enseignements non évidents. Premier constat : la **diversité des prompts de génération** est aussi importante que la qualité du modèle teacher. Un pipeline qui génère tous ses exemples avec la même formulation de prompt produit des datasets homogènes avec des biais systématiques — même si chaque exemple individuel est de haute qualité. Varier les formulations, les longueurs de réponse attendues, les niveaux de détail, et les styles rédactionnels est indispensable pour obtenir la diversité qui améliore la généralisation.

Deuxième constat : la **supervision humaine reste indispensable pour les domaines critiques**. Un MSSP qui avait généré 100 000 exemples d'analyse d'alertes SOC sans révision humaine a découvert en production que son modèle fine-tuné propageait systématiquement des erreurs d'attribution de tactiques MITRE ATT&CK présentes dans les données synthétiques — erreurs que le LLM teacher avait générées de manière cohérente mais incorrecte. Une révision d'Argilla sur un échantillon de 5% du dataset aurait identifié ces erreurs systématiques avant le fine-tuning. Troisième constat : le suivi de la **distribution des labels et sujets** dans le dataset généré est crucial pour éviter les surreprésentations accidentelles. Pour une consultation dédiée sur la construction de pipelines de données synthétiques pour des usages en cybersécurité, notre équipe est disponible via la page consulting. Voir aussi notre article sur les meilleures pratiques LLMOps pour le monitoring continu des modèles fine-tunés en production.

Ressources et communautés pour la génération de données synthétiques

L'écosystème des données synthétiques pour LLM bénéficie d'une communauté open-source très active. Sur HuggingFace, le dataset UltraFeedback binarized est l'un des datasets de préférence synthétiques les plus utilisés pour le fine-tuning DPO. Le repository Distilabel sur GitHub maintient une documentation détaillée avec des exemples couvrant de nombreux cas d'usage. Pour les aspects théoriques de la differential privacy, le cours "Protecting Privacy in the Age of AI" de l'université de Boston constitue une référence accessible.

En France, le **Health Data Hub** publie régulièrement des ressources sur la synthétisation des données de santé, et la CNIL propose des guides pratiques sur l'anonymisation et la pseudonymisation des données personnelles — ressources directement pertinentes pour les organisations souhaitant générer des données synthétiques à partir de datasets contenant des informations personnelles. Le projet **FrenchBench** développe des benchmarks d'évaluation spécifiquement en français pour comparer les performances des LLM sur des tâches francophones, utile pour valider les modèles fine-tunés sur des données synthétiques françaises. Notre article sur les architectures GraphRAG complète ce guide avec les aspects de structure de connaissance applicable aux datasets complexes.

Évaluation de la qualité des données synthétiques générées

La mesure objective de la qualité d'un dataset synthétique avant de l'utiliser pour le fine-tuning est indispensable mais non triviale. Plusieurs dimensions doivent être évaluées simultanément : la **fidélité** (les exemples synthétiques ressemblent-ils à de vraies données du domaine ?), la **diversité** (le dataset couvre-t-il bien l'espace des tâches et des formulations possibles ?), la **qualité intrinsèque** (chaque exemple est-il factuelle-ment correct et bien formulé ?), et l'**utilité pour le fine-tuning** (le modèle fine-tuné sur ce dataset améliore-t-il effectivement ses performances sur les tâches cibles ?).

Des métriques automatiques pour évaluer ces dimensions : la **Vendi Score** (mesure de diversité basée sur les eigenvalues de la matrice de similarité des embeddings du dataset), la **FID (Fréchet**

Inception Distance) adaptée au texte via des embeddings, et des scores de qualité LLM-as-judge agrégés sur un échantillon représentatif. Mais la validation ultime reste la mesure des performances du modèle fine-tuné sur un held-out dataset humain — mesure directe de l'utilité effective du dataset synthétique pour l'objectif opérationnel final. Aucune métrique automatique ne remplace cette validation end-to-end.

Les bonnes pratiques recommandent une approche d'**évaluation multi-étapes** : filtrage automatique rapide (format, longueur, score LLM-as-judge minimum) pour éliminer les exemples manifestement mauvais, suivi d'une évaluation manuelle sur un échantillon stratifié de 1 à 5% du dataset total, et enfin une évaluation d'impact end-to-end via fine-tuning sur un subset et mesure des performances. Cette approche multi-étapes optimise le rapport temps/qualité de validation et garantit que seuls les datasets véritablement utiles atteignent le processus de fine-tuning, évitant les coûts de calcul GPU d'un fine-tuning sur un dataset sous-qualité.

Synthèse tabular : générer des données structurées pour le ML

Au-delà du texte, la **génération de données tabulaires synthétiques** pour l'entraînement de modèles ML classiques (tabular ML, fraud detection, scoring) est un domaine distinct mais important. Des approches comme les **GAN conditionnels (CTGAN, TVAE)** et les modèles de diffusion (TabDDPM) génèrent des données tabulaires synthétiques préservant les distributions marginales et les corrélations entre colonnes des données originales. Ces données sont utilisées pour augmenter des datasets limités, tester des pipelines ML sans exposer des données sensibles, et créer des données de test réalistes pour les environnements de développement.

En cybersécurité, la synthèse tabulaire trouve des applications directes dans la génération de logs réseau synthétiques pour l'entraînement de détecteurs d'intrusion, de journaux d'authentification synthétiques pour tester des systèmes de détection d'account takeover, et de données de transactions synthétiques pour les systèmes anti-fraude. Des bibliothèques Python comme **SDV (Synthetic Data Vault)** et **Synthpop** (R) proposent des interfaces accessibles pour la synthèse tabulaire sans expertise en deep learning. La validation de la qualité des données tabulaires synthétiques utilise des métriques spécifiques : column-wise statistical similarity, pair-wise

correlation similarity, et downstream ML utility (le modèle entraîné sur des données synthétiques perform-il aussi bien qu'un modèle entraîné sur des données réelles ?).

Infrastructure cloud pour la génération à grande échelle

La génération de datasets synthétiques à grande échelle (millions d'exemples) nécessite une infrastructure cloud adaptée pour être économiquement viable. L'approche naïve — appeler séquentiellement une API LLM pour chaque exemple — est trop lente pour des volumes significatifs. Des solutions d'accélération incluent le **batch processing API** d'Anthropic et d'OpenAI (réduction de 50% du coût, délai de 24h acceptable pour la génération offline), le déploiement de modèles open-source sur des instances GPU cloud (A100, H100) via vLLM pour un débit de 10 à 100x supérieur aux appels API séquentiels, et l'architecture multi-worker parallèle où des dizaines de workers génèrent simultanément des batches d'exemples indépendants.

Le coût estimé pour générer 100 000 exemples instruction/réponse de qualité (réponses de 200 à 500 tokens en moyenne) : avec GPT-4o-mini via l'API Batch, environ 50 à 150€ selon les longueurs ; avec Claude Haiku via Anthropic Batch API, environ 30 à 80€ ; avec Llama 3 70B auto-hébergé sur une instance GPU cloud A100, environ 100 à 300€ en coûts de compute (mais sans coût variable par token). Pour les organisations réalisant régulièrement de telles générations, l'auto-hébergement devient économiquement attractif à partir de quelques dizaines de milliers d'euros de génération mensuelle. Le calcul TCO doit inclure les coûts d'ingénierie pour maintenir l'infrastructure, qui peuvent dépasser les économies sur les APIs pour des équipes de petite taille.

Techniques de déduplication avancées pour grands datasets

La **déduplication** des datasets de données synthétiques à grande échelle (10M+ exemples) est un problème non trivial que des approches naïves (comparaison exacte de strings) ne peuvent pas

résoudre efficacement. Des techniques probabilistes comme le **MinHash LSH (Locality-Sensitive Hashing)** permettent d'identifier des quasi-duplicates textuels avec une complexité quasi-linéaire en taille du dataset. Des approches sémantiques basées sur des embeddings de phrases denses (MPNet, E5, BGE) suivies d'une recherche de voisins approximatifs (FAISS, ScaNN) capturent des duplicates sémantiques non détectables par les approches lexicales — deux exemples formulés différemment mais exprimant la même instruction de la même manière.

Le projet **SemDeDup** (Meta, 2023) a démontré que la déduplication sémantique d'un corpus d'entraînement pré-existant permettait d'entraîner des modèles plus performants avec moins de compute, en éliminant les exemples redondants qui apportent peu de signal additionnel. Cette observation s'applique directement aux datasets synthétiques : un dataset de 100K exemples bien dédupliqués est souvent plus efficace qu'un dataset de 500K exemples avec 70% de quasi-duplicates. L'implémentation pratique de la déduplication sémantique nécessite de calculer des embeddings pour tous les exemples (coûteux pour les très grands datasets) et de configurer soigneusement le seuil de similarité au-delà duquel deux exemples sont considérés comme redondants — seuil qui varie selon le cas d'usage et doit être calibré empiriquement.

Pour les équipes construisant des pipelines de données synthétiques pour la première fois, la recommandation pratique est de commencer petit : générer 1 000 exemples, les valider manuellement à 100%, identifier les problèmes systématiques, améliorer le pipeline, puis scaler progressivement. Cette approche d'itération rapide sur de petits volumes prévient les gaspillages de budget GPU sur de grands pipelines défectueux. Des organisations françaises pionnières dans ce domaine — notamment des équipes de recherche au CEA et des fintechs comme Qonto ou Alma — ont développé en 2024-2025 des pratiques internes documentées qui font l'objet d'une diffusion croissante dans la communauté via des talks aux conférences EGC, TALN, et JDoc. Ces retours d'expérience pratiques en contexte français sont précieux pour les équipes qui démarrent sur ce chemin.

L'opinion tranchée des experts les plus expérimentés dans la construction de datasets synthétiques est convergente sur un point que les vendeurs de solutions de génération automatique minimisent : **les données synthétiques ne sont jamais gratuites** — elles échangent le coût de l'annotation humaine contre le coût de la génération LLM (non négligeable à grande échelle), le coût de la validation qualité (indispensable), et le risque de propager des biais ou des erreurs systématiques du modèle teacher vers le modèle student via le dataset. Ce coût total est souvent inférieur à celui des données annotées humainement à qualité comparable, mais les

organisations qui sous-estiment la phase de validation et de curation finissent par déployer des modèles fine-tunés sous-performants ou biaisés. La discipline de la data engineering s'applique aussi rigoureusement aux données synthétiques qu'aux données réelles — peut-être plus, car les erreurs systématiques du modèle teacher peuvent contaminer silencieusement des milliers d'exemples sans qu'aucun signal d'erreur évident n'apparaisse dans les logs de génération.