Actes de CORIA-TALN 2026
Nantes, France, 29 juin au 3 juillet 2026

Stéganographie textuelle par paraphrase : une approche par *LLM*

Geoffrey Anquetil Jonathan Chevelu
Univ Rennes, IRISA, CNRS, 22300 Lannion, France
geoffrey.anquetil@irisa.fr, jonathan.chevelu@irisa.fr

RÉSUMÉ

La stéganographie textuelle par paraphrase permet de dissimuler un message secret tout en garantissant l'ancrage contextuel et la pertinence du texte généré vis-à-vis d'une source. Cet article formalise ce problème d'optimisation et établit un protocole d'évaluation robuste s'appuyant sur des métriques récentes. Sur le plan technique, nous proposons une méthode exploratoire exploitant un grand modèle de langue (*LLM*) guidé par une recherche en faisceau diversifiée (*DBS*) sous contrainte de parité de bloc. Afin d'isoler l'impact de l'insertion stéganographique, les textes produits sont systématiquement comparés à des paraphrases générées par le même modèle sans contrainte. Si l'approche confirme la faisabilité de l'insertion et atteint une capacité compétitive, des écarts significatifs sont observés concernant la perplexité et la fidélité sémantique fine. Ces différences illustrent les altérations induites par la génération sous contrainte stricte et ouvrent de nouvelles perspectives de recherche pour cette tâche.

ABSTRACT

Textual Steganography using Paraphrase : An LLM approach.

Textual steganography via paraphrasing allows a secret message to be hidden while ensuring the contextual anchoring and relevance of the generated text relative to a source. This paper formalizes this optimization problem and establishes a robust evaluation protocol relying on recent metrics. Technically, we propose an exploratory method leveraging a Large Language Model (LLM) guided by a Constrained Diverse Beam Search (DBS) using block parity. To isolate the impact of the steganographic insertion, the produced texts are systematically compared to unconstrained paraphrases generated by the same model. While the approach confirms the feasibility of the insertion and achieves a competitive capacity, significant differences are observed regarding perplexity and fine-grained semantic fidelity. These variations illustrate the alterations induced by strictly constrained generation and open new research perspectives on this task.

MOTS-CLÉS : stéganographie textuelle, paraphrase, grand modèle de langue.

KEYWORDS: textual steganography, paraphrase, Large Language Model.

1 Introduction

La protection et la confidentialité des échanges numériques s'appuient historiquement sur trois domaines distincts : la cryptographie, le tatouage numérique et la stéganographie (Al-Yousuf & Din, 2020). Si la cryptographie vise à rendre un message inintelligible, et que le tatouage numérique cherche à insérer une marque robuste pour protéger la propriété intellectuelle (souvent contre l'altération), la stéganographie a pour objectif de dissimuler l'existence même d'une communication. Le message

Actes des 33ème Conférence sur le Traitement Automatique des Langues Naturelles.
Volume 1 : articles scientifiques originaux, pages 564–575.

secret est ainsi enfoui au sein d'un document d'apparence anodine, appelé couverture, de manière à n'éveiller aucun soupçon lors de son interception par un tiers.

Parmi les différents vecteurs possibles, le texte constitue un support de choix en raison de son omniprésence dans les communications quotidiennes (messageries, courriels). Toutefois, sa nature discrète rend l'insertion stéganographique particulièrement complexe comparée aux médias continus (images, audio) qui tolèrent d'infimes modifications imperceptibles (Almayyahi *et al.*, 2021). Pour contourner cette difficulté, la recherche s'est récemment orientée vers la génération textuelle *ex nihilo* (Ziegler *et al.*, 2019; Yang *et al.*, 2019; Lin *et al.*, 2024). Bien que performante en termes de capacité d'insertion, cette approche souffre d'un manque d'ancrage contextuel : les textes générés artificiellement s'intègrent difficilement dans un flux de discussion préexistant. C'est pourquoi la stéganographie par paraphrase s'impose comme une alternative pertinente. En reformulant une phrase source sans en altérer le sens profond, elle garantit la pertinence et l'innocence contextuelle du média échangé.

Pourtant, malgré la domination actuelle des grands modèles de langue (*LLM*) dans les tâches de génération textuelle, leur application spécifique à la génération de paraphrases stéganographiques reste largement inexplorée. Les méthodes existantes reposent encore souvent sur des manipulations syntaxiques locales ou des traductions pivots (Yang *et al.*, 2022), sans exploiter pleinement la flexibilité combinatoire offerte par les architectures des *LLM*.

Face à ce constat, la contribution majeure de cet article est double. Elle réside en premier lieu dans la formalisation du problème de la stéganographie textuelle générative par paraphrase, couplée à l'établissement d'un protocole d'évaluation robuste. Face aux limites des métriques traditionnelles pour juger du maintien du sens, ce protocole intègre des outils d'évaluation sémantiques comme ParaPLUIE (Lemesle *et al.*, 2025) et syntaxiques récents. Dans un second temps, une méthode exploratoire s'appuyant sur une recherche en faisceau diversifiée (DBS) (Vijayakumar *et al.*, 2016) sous contraintes est proposée. Si cette première approche démontre empiriquement la viabilité d'une insertion stéganographique par *LLM*, elle met également en évidence une certaine irrégularité dans la qualité de la génération. Cette dernière offre ainsi des perspectives d'amélioration.

Après un rappel des évolutions de la stéganographie textuelle en section 2, le cadre formel et les exigences spécifiques de la tâche par paraphrase sont définis section 3. Ce socle théorique permet d'introduire l'approche d'encodage par *LLM* et recherche en faisceau contrainte en section 4. Enfin, la viabilité et les limites de cette méthode sont évaluées via un protocole expérimental dédié en section 5, dont les résultats quantifient précisément le compromis entre capacité d'insertion, indiscernabilité statistique et maintien du sens en section 6.

2 Approches de la stéganographie textuelle

La stéganographie linguistique a connu une évolution majeure, passant de simples modifications locales (Singh *et al.*, 2009) à des approches neuronales génératives (Wu *et al.*, 2024).

Historiquement, les premières méthodes inséraient des informations secrètes en altérant légèrement un texte de couverture préalablement fourni. Ces approches opèrent généralement au niveau du mot, en s'appuyant sur des dictionnaires pour substituer des synonymes (Winstein, 1998; Huo & Xiao, 2016) ou introduire volontairement des erreurs typographiques (Topkara *et al.*, 2007). Cependant, ces remplacements partiels engendrent très souvent des ambiguïtés sémantiques et souffrent d'une

capacité d'insertion extrêmement faible.

Avec l'avènement des réseaux de neurones récurrents puis des architectures *transformers* (Vaswani *et al.*, 2017), le paradigme a basculé vers la génération *ex nihilo*. Au lieu de modifier un texte existant, ces méthodes altèrent la distribution de probabilités du modèle de langue lors de l'inférence pour échantillonner des *tokens* correspondant au secret, via des techniques comme le *Bin-Coding* ou le codage arithmétique (Yang *et al.*, 2019; Ziegler *et al.*, 2019; Zhang *et al.*, 2021). Récemment, l'utilisation de méthodes génératives conditionnées a permis de guider la création du texte en fournissant en entrée un thème précis ou une émotion (Wu *et al.*, 2024; Lin *et al.*, 2024). Si ces approches atteignent des capacités d'insertion très élevées, elles manquent d'ancrage contextuel. L'incapacité de ces textes à s'intégrer naturellement dans un flux de discussion préexistant pose un problème majeur quant à l'innocence du média échangé.

Pour résoudre cette difficulté d'intégration contextuelle, la stéganographie par paraphrase s'efforce de dissimuler le secret tout en préservant au mieux le sens d'une phrase source (Chang & Clark, 2010). Des travaux récents ont relancé cette voie en combinant par exemple des manipulations syntaxiques et lexicales conjointes à de multiples granularités (Ou *et al.*, 2025), ou en exploitant la traduction pivot (langue source vers langue pivot puis retour) couplée à un codage par intervalles sensible à la sémantique (Yang *et al.*, 2022).

Toutefois, bien que les *LLMs* basés sur des architectures *decoder-only* dominent aujourd'hui la génération de texte, leur utilisation reste massivement cantonnée à la génération *ex nihilo*. L'exploitation spécifique de ces modèles pour la génération de paraphrases stéganographiques n'a pas encore été explorée, constituant ainsi l'objet central de cette étude.

3 Formulation du problème

La stéganographie générative textuelle par paraphrase est un problème d'optimisation. À partir d'un texte d'origine appelé couverture C et d'un message secret à dissimuler M, le défi de l'encodeur consiste à générer un nouveau texte, le stégo-texte S, qui optimise quatre propriétés fondamentales. Ces propriétés sont illustrées sur la figure 1 :

1. **Capacité :** le rapport entre la taille en bits de M et la longueur de S en mots (bpw) ou en *tokens* (bpt). C'est la charge utile secrète du message stéganographié.
2. **Qualité de la paraphrase :** S doit être correct syntaxiquement et la distance sémantique entre S et C doit être minimale.
3. **Réversibilité :** le message secret doit pouvoir être extrait intégralement et sans erreur à partir du stégo-texte. La fonction de décodage doit donc garantir une extraction exacte : $D(S) = M$.
4. **Indiscernabilité :** parmi les candidats satisfaisant les conditions précédentes, S ne doit pas être identifiable statistiquement comme étant porteur d'un message caché. Étant donné la banalisation de l'usage des *LLM*, S doit tromper un classifieur stéganalytique entraîné à différencier des textes générés par un modèle de langue sans contrainte d'encodage des stégo-textes. L'objectif est que le stégo-texte se confonde avec une génération *LLM* classique pour les intermédiaires de surveillance, maximisant ainsi l'erreur de classification de l'intermédiaire.

Sous l'hypothèse d'un secret chiffré, il n'est pas question de mesurer la capacité d'un attaquant à reconstruire le secret original. Seule la capacité à détecter la présence de stéganographie est

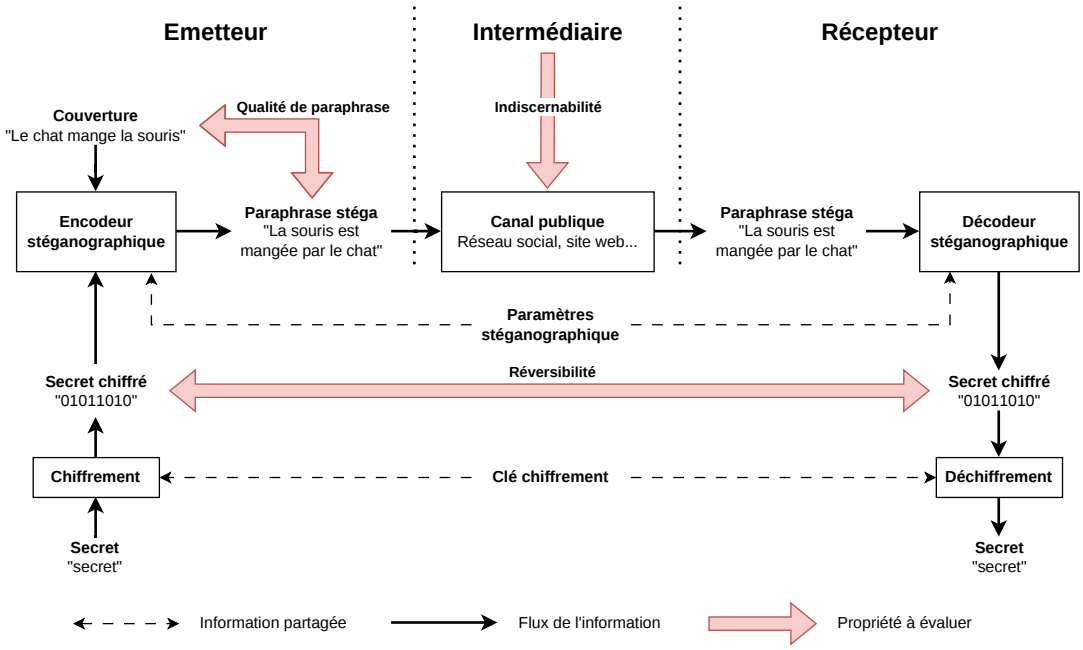

FIGURE 1 – Modèle conceptuel d'un système de stéganographie textuelle par paraphrase et propriétés à évaluer.

importante ici. Aussi, la robustesse à la modification n'est pas un objectif ici comme elle pourrait l'être en tatouage.

La satisfaction simultanée de ces quatre propriétés exige un compromis entre la capacité d'insertion, l'innocence du texte et la qualité de la paraphrase. En effet, augmenter la quantité d'informations dissimulées restreint naturellement l'espace de recherche des paraphrases valides. Cette restriction risque alors de dégrader la fidélité sémantique vis-à-vis de la couverture ou d'introduire des biais statistiques rendant le stégo-texte suspect face à l'intermédiaire.

4 Une approche par *LLM*

L'architecture générale propose d'encoder le secret à travers les identifiants des *tokens* générés par un *LLM*. La génération est initiée par un patron prenant en entrée la phrase de couverture et invitant le modèle à en générer une paraphrase. Un filtre est appliqué à chaque pas de génération du modèle pour garantir le choix de *tokens* encodant le secret.

L'encodage du secret se fait via une opération modulo n sur l'identifiant du *token* (pour $n = 2$, l'encodage est une condition de parité binaire).

Toutefois, la tâche de paraphrase exige de restituer fidèlement des séquences du texte de couverture comme des entités nommées ou des dates. Un encodage de parité strict *token* par *token* serait trop rigide : il empêcherait la génération de ces séquences forcées. Il est donc indispensable d'introduire de la flexibilité dans le processus de décodage du modèle pour concilier insertion du secret et préservation syntaxique et sémantique.

4.1 Recherche en faisceau diversifiée sous contraintes

Afin de surmonter le problème posé par les séquences de mots incompressibles, la génération s'appuie sur un algorithme de recherche en faisceau diversifiée (*DBS*) (Vijayakumar *et al.*, 2016) sous contraintes. Alors qu'un décodage glouton risquerait d'être bloqué par la nécessité de restituer fidèlement ces entités spécifiques, le *DBS* maintient en parallèle plusieurs hypothèses textuelles distinctes. À chaque pas de génération, un filtre d'élagage est appliqué :

- seuls les candidats encodant effectivement le secret sont conservés (voir section 4.2) ;
- le *token* doit respecter la contrainte de stabilité tokenisation-détokenisation (voir section 4.3) ;
- si aucun candidat ne satisfait la contrainte, le faisceau est élagué, forçant le modèle à explorer des chemins alternatifs.

L'utilisation d'une pénalité de diversité force le modèle à explorer des structures syntaxiques variées. Cet aspect est fondamental : il permet de déplacer les séquences de *tokens* incompressibles au sein de la phrase générée. À titre d'illustration, une information rigide telle que « 1789 » pourra être décalée dans plusieurs phrases générées : « La Révolution française a débuté en 1789 » et « En 1789 éclata la Révolution française ». Ce glissement syntaxique décale la position de l'entité contrainte lors de la génération, offrant ainsi la flexibilité nécessaire pour contourner un blocage d'encodage stéganographique tout en préservant l'intégrité sémantique et syntaxique.

4.2 Encodage par parité de bloc

Préalablement à l'insertion, le message secret M est représenté en base n et donc en une séquence de symboles $m_i \in \{0, \ldots, n-1\}$. Pour procéder à sa dissimulation lors de la génération, le texte produit est alors segmenté en blocs successifs de k *tokens*. Une valeur $\mathcal{P}(B_i)$ est calculée pour chaque bloc $B_i = \{t_1^{(i)}, \ldots, t_k^{(i)}\}$ selon l'équation :

$$\mathcal{P}(B_i) = \left(\sum_{j=1}^{k} \mathrm{id}(t_j^{(i)}) \right) \bmod n$$

Où $\mathrm{id}(t_j^{(i)})$ est le nombre entier correspondant à l'identifiant dans le vocabulaire du modèle du j-ième *token* du bloc B_i.

Le bloc B_i est valide si $\mathcal{P}(B_i) = m_i$, où $m_i \in \{0, \ldots, n-1\}$ correspond au i-ème symbole du message secret. Dans le cas d'un encodage binaire privilégié dans cette étude ($n = 2$), cette somme modulaire équivaut à l'application d'un OU exclusif (*XOR*) sur la parité des identifiants du bloc.

Cette approche par bloc (dès lors que $k > 1$) diversifie les chemins permettant d'encoder un symbole unique. En se plaçant dans le cas d'un encodage binaire, sur les 2^k combinaisons de parités possibles pour un bloc, exactement la moitié (2^{k-1}) satisfait la condition $\mathcal{P}(B_i) = m_i$. Le modèle n'est ainsi jamais restreint à une configuration unique. Si la paraphrase exige une entité nommée forcée dont la parité des *tokens* s'avère défavorable, les autres *tokens* du bloc s'ajustent pour modifier la somme globale et satisfaire la contrainte. Cette marge combinatoire accroît la probabilité de générer une séquence valide et sémantiquement cohérente.

Outre cette flexibilité combinatoire, l'utilisation de la somme modulaire joue un rôle crucial dans l'indiscernabilité statistique du texte généré. Toute restriction de l'espace de décodage risque d'altérer

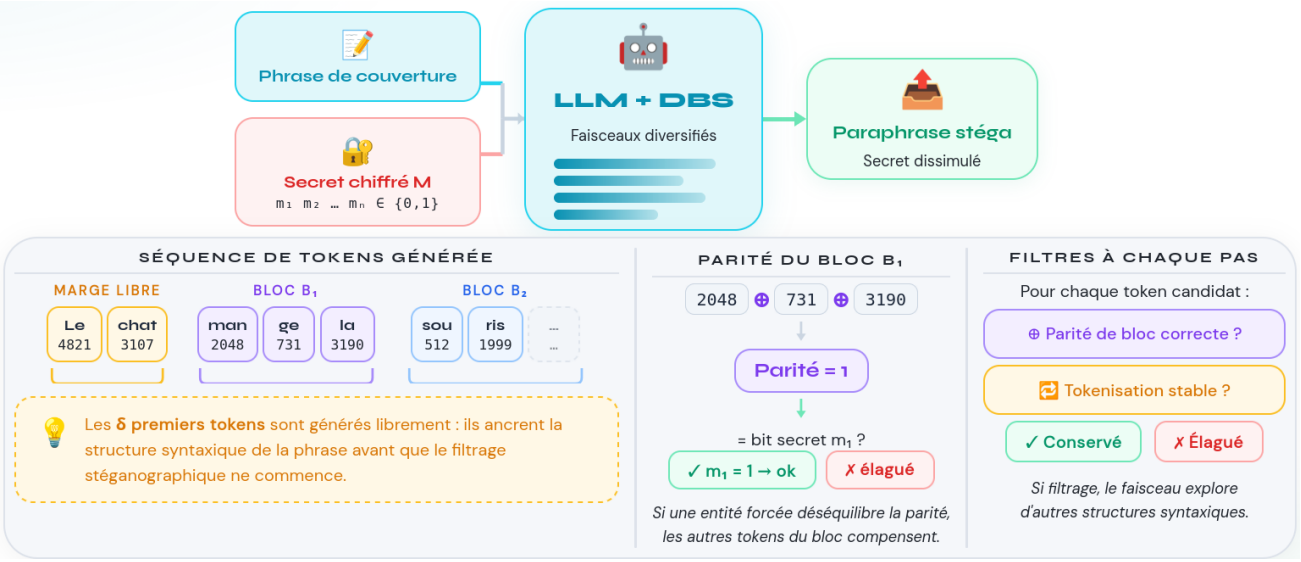

FIGURE 2 – Illustration des différentes étapes du processus d'encodage du secret.

la fréquence naturelle d'apparition des mots. Par exemple, si l'on utilisait une méthode d'agrégation naïve comme le vote majoritaire sur un bloc, encoder le bit 1 forcerait le modèle à générer artificiellement une majorité de *tokens* de parité impaire, biaisant mécaniquement la distribution globale du vocabulaire. À l'inverse, l'opération modulo préserve cette distribution : elle permet de satisfaire une condition de parité globale via de multiples motifs locaux équilibrés. De plus, sous l'hypothèse standard où le message M est préalablement chiffré (garantissant une entropie maximale), l'équirépartition des bits à dissimuler n'introduit aucun biais structurel : aucune combinaison de parité n'est artificiellement surreprésentée au cours de la génération. La méthode lisse ainsi son empreinte et évite toute altération suspecte de la distribution des unigrammes ou de la structure des n-grammes.

4.3 Stabilité du *tokenizer*

Un défi majeur pour l'encodage par parité de token est l'instabilité de la tokenisation : une séquence de tokens, une fois dé-tokenisée en texte puis re-tokenisée, peut produire une séquence d'identifiants différente (ex : fusion de sous-mots {télé,vision} devient {télévision}, gestion des espaces {_Le} devient {Le}). Pour garantir un taux de recouvrement de 100%, nous implémentons une vérification de stabilité à la volée. Pour toute séquence candidate partielle $S_{partiel}$, nous vérifions :

$$\text{Tokeniser}(\text{Detokeniser}(S_{partiel})) \stackrel{?}{=} S_{partiel} \tag{1}$$

Tout candidat ne respectant pas l'équation (1) est immédiatement rejeté, garantissant que le récepteur, en re-tokenisant le texte reçu, retrouvera exactement les mêmes IDs que ceux utilisés par l'émetteur.

Le mécanisme d'encodage du secret est illustré à la figure 2.

4.4 Marge initiale

Les premiers *tokens* générés jouent un rôle critique dans la paraphrase à cause de la nature auto-régressive des *LLM* : ils conditionnent la structure syntaxique de toute la séquence ultérieure. Imposer une contrainte de parité dès les premiers pas de temps reviendrait à élaguer arbitrairement des amorces possibles, risquant d'éliminer prématurément des structures de phrases pertinentes ou originales même avec le *DBS*.

Pour éviter cet effondrement précoce de la diversité, une marge initiale δ est introduite. Les δ premiers tokens sont générés librement par le modèle, sans contrainte stéganographique. Cette fenêtre initiale permet au *DBS* de déployer un éventail de contextes variés avant que le filtrage stéganographique ne commence, garantissant ainsi que l'élagage ultérieur s'opère sur un faisceau de candidats déjà syntaxiquement riches et distincts.

4.5 Extraction du secret

Le processus de décodage est déterministe. Contrairement à l'étape d'encodage, il ne nécessite aucune inférence du modèle de langage et a donc un **coût GPU nul**. Il repose strictement sur la possession du *tokenizer* partagé :

Le bloc « Décodeur stéganographie » (Fig 1) fonctionne comme suit :

1. **Tokenisation :** le texte reçu S est converti en une suite d'identifiants numériques via le *tokenizer* spécifique du modèle utilisé.
2. **Alignement :** les δ premiers tokens sont ignorés.
3. **Segmentation :** la séquence restante est découpée en blocs B_i de taille k.
4. **Reconstruction :** la parité $\mathcal{P}(B_i)$ pour chaque bloc est calculée afin d'extraire les bits du message M.

5 Protocole expérimental

Notre question de recherche se concentrant spécifiquement sur l'impact de l'ajout de contraintes lors de la génération de paraphrases, l'évaluation se place à l'échelle de la phrase. Par conséquent, nous restreignons le cadre expérimental à l'insertion d'un secret de longueur fixe, notée L, pour chaque phrase de couverture. Le modèle d'échange suppose ainsi uniquement un partage préalable des paramètres statiques : le modèle de *tokenizer*, la taille de bloc k, le décalage initial δ et la longueur du secret L.

L'objectif de ce protocole expérimental est d'isoler le coût de l'insertion stéganographique par rapport à une génération de paraphrase standard. Dans un contexte de banalisation des *LLMs*, le simple fait de détecter qu'un texte a été généré artificiellement ne suffit plus à le qualifier de suspect. Par conséquent, l'évaluation repose sur la comparaison systématique des stégo-textes avec une méthode de référence : des paraphrases générées par le même modèle, utilisant le même algorithme *DBS*, mais exemptes de contraintes stéganographiques.

5.1 Corpus et critères de sélection

Le jeu de données *Wikipedia-en-sentences* [1] a été retenu pour l'évaluation. C'est une extraction de phrases issues d'articles Wikipedia anglais. Ce choix est motivé par la nature factuelle et dense de son contenu (entités nommées, dates, faits historiques), qui constitue un environnement particulièrement hostile et exigeant pour la paraphrase stéganographique. Aucun filtre lexical n'a été appliqué : les caractères spéciaux, le vocabulaire rare et les structures syntaxiques complexes ont été délibérément conservés pour éprouver la robustesse du modèle.

Cependant, une contrainte de longueur minimale sur C est indispensable pour garantir que la phrase générée S offre une capacité d'insertion suffisante. Sous l'hypothèse de base qu'une paraphrase conserve approximativement la longueur de sa couverture ($len(S) \approx len(C)$), seules les couvertures C satisfaisant la contrainte suivante ont été sélectionnées :

$$len(C) \geq (L \times k) + \delta + \xi \tag{2}$$

où ξ représente une marge de sécurité en *tokens* permettant de pallier d'éventuels raccourcissements naturels lors de la reformulation. Dans le cadre de cette étude, les paramètres ont été fixés à $L = 16$, $k = 3$, $\delta = 3$ et $\xi = 13$, imposant ainsi une longueur minimale de couverture de 64 *tokens*. Sur cette base, un sous-ensemble aléatoire de 2500 couvertures C éligibles a été extrait. Pour chacune de ces couvertures, un secret M de $L = 16$ bits est généré aléatoirement, un unique stégo-texte S encodant ce secret est généré ainsi que, dans un objectif de comparaison, une paraphrase de référence sans contrainte stéganographique.

5.2 Métriques d'évaluation

L'évaluation du système s'articule autour de trois dimensions : la qualité linguistique (sémantique et syntaxique), l'efficacité d'insertion et l'indiscernabilité par perplexité d'un modèle de langue.

— **Qualité sémantique :** la similarité sémantique est mesurée via ParaPLUIE (Lemesle *et al.*, 2025). Ce score est généré en exploitant le modèle *Phi-4-mini-instruct* [2] en demi-précision couplé au patron d'évaluation *FS-DIRECT*. Évoluant sur une échelle de $-\infty$ à $+\infty$, un score strictement positif indique une paraphrase valide.
— **Qualité syntaxique :** une déclinaison spécifique de l'outil précédent, SyntaxPLUIE, est utilisée. Elle s'appuie sur le même évaluateur *Phi-4-mini-instruct* avec un patron dédié afin de juger exclusivement la grammaticalité et la fluidité de la phrase produite, indépendamment de sa fidélité au sens d'origine. De la même manière, un score strictement positif atteste d'une syntaxe correcte.
— **Capacité d'insertion :** la densité d'information est mesurée en bits par mot (bpw) et en bits par *token* (bpt) dans S. L'objectif théorique ciblé par l'hyperparamètre $k = 3$ correspond à une capacité d'environ $0,33$ bpt.
— **Perplexité :** la discrétion de la perturbation probabiliste introduite par l'encodeur est évaluée par l'analyse de la perplexité. Celle-ci est calculée à l'aide d'un modèle de langue externe et indépendant, ici *Phi-2* [3].

1. https://huggingface.co/datasets/sentence-transformers/wikipedia-en-sentences
2. https://huggingface.co/microsoft/Phi-4-mini-instruct
3. https://huggingface.co/microsoft/phi-2

5.3 Modèle et paramétrage de génération

Les expériences s'appuient sur le *LLM Mistral-7B-Instruct-v0.2* [4], instancié en précision demi-flottante (*float16*). L'ensemble des générations a été exécuté sur une unique carte graphique grand public NVIDIA RTX 4090, 24 Go de VRAM.

Cette configuration matérielle définit la borne supérieure pour la taille du faisceau qui a été fixée à 60. Ces faisceaux sont répartis en 4 groupes indépendants. Enfin, les hyperparamètres régissant la recherche s'alignent sur les recommandations de l'étude originale introduisant le *DBS* (Vijayakumar *et al.*, 2016) : la pénalité de diversité est établie à $0{,}7$ et la pénalité de répétition à $1{,}2$.

6 Résultats et discussion

L'évaluation des performances de l'approche stéganographique proposée s'articule autour de trois axes principaux : la capacité d'insertion couplée à la neutralité structurelle, la perplexité et le maintien de l'intégrité sémantique et syntaxique.

Les résultats quantitatifs globaux sont synthétisés dans la Table 1.

Méthode	**Perplexité**		**ParaPLUIE (↑)**		**SyntaxPLUIE (↑)**	
	Score	**Δ (↓)**	**Validité**	**Score**	**Validité**	**Score**
Paraphrase	$14{,}3 \pm 1{,}2$	$-1{,}8 \pm 1{,}2$	$(94{,}5 \pm 0{,}9)\,\%$	$5{,}7 \pm 0{,}1$	$(94{,}6 \pm 1{,}0)\,\%$	$6{,}2 \pm 0{,}1$
Stégo-texte	$33{,}9 \pm 1{,}1$	$17{,}8 \pm 1{,}0$	$(77{,}7 \pm 1{,}6)\,\%$	$3{,}7 \pm 0{,}1$	$(81{,}7 \pm 1{,}5)\,\%$	$4{,}1 \pm 0{,}1$

TABLE 1 – Comparaison des paraphrases produites par un modèle non contraint et des paraphrases avec une contrainte stéganographique. Les scores de similarité et de syntaxe sont calculés exclusivement sur les générations considérées comme valides (ParaPLUIE > 0 et SyntaxPLUIE > 0). La précision indique la proportion de ces générations valides. Les moyennes sont accompagnées de leur intervalle de confiance à $95\,\%$. Pour référence, la perplexité moyenne du texte de couverture s'établit à $16{,}04 \pm 0{,}54$.

La méthode atteint une capacité d'insertion moyenne de $0{,}21$ bpt et $0{,}39$ bpw. Ce taux de dissimulation s'avère compétitif vis-à-vis des approches fondées sur la paraphrase. A titre de comparaison, la méthode par langue pivot proposée par (Yang *et al.*, 2022) obtient ses meilleurs compromis linguistiques pour une capacité de $0{,}33$ bpw.

L'évaluation de la naturalité, réalisée à l'aide d'un modèle de langage tiers (*Phi-2*), permet d'observer l'impact statistique de l'insertion. Le texte d'origine présente une perplexité moyenne de $16{,}04 \pm 0{,}54$. La génération de paraphrases de référence (sans secret) tend à produire un texte légèrement plus prédictible pour le modèle évaluateur, marquant une baisse de la perplexité d'environ $11\,\%$ (Δ moyen de $-1{,}8 \pm 1{,}2$). Cette baisse s'explique mécaniquement par le processus de génération : l'algorithme de décodage tend à privilégier les séquences à haute probabilité, opérant un lissage statistique qui réduit l'entropie du texte par rapport au corpus d'origine. En revanche, l'application des contraintes stéganographiques entraîne une dégradation significative de la naturalité : la perplexité globale fait plus que doubler pour atteindre $33{,}85 \pm 1{,}09$, soit une hausse d'environ $111\,\%$ (écart moyen de

4. https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

$17{,}81 \pm 0{,}97$) par rapport à la couverture. Cette élévation reflète la mécanique de l'encodeur : pour satisfaire la condition stricte de parité, le modèle est ponctuellement forcé de dévier de ces chemins optimaux, ce qui altère de nouveau l'empreinte statistique du texte généré.

La nécessité d'altérer la distribution probabiliste pour dissimuler le secret entraîne logiquement un risque de dégradation sémantique et syntaxique. L'utilisation de métriques comparatives basées sur des *LLMs* (PLUIE) met en évidence l'irrégularité induite par l'insertion stéganographique. La proportion de paraphrases sémantiquement valides (précision ParaPLUIE > 0) passe de $(94{,}5 \pm 0{,}9)\,\%$ pour la méthode de référence à $(77{,}7 \pm 1{,}6)\,\%$ pour les stégo-textes. Sur ce sous-ensemble valide, le score d'adéquation sémantique moyen diminue également, passant de $5{,}70 \pm 0{,}07$ à $3{,}74 \pm 0{,}09$. Une tendance similaire est observée pour la grammaticalité (SyntaxPLUIE), avec une baisse de précision ($(81{,}7 \pm 1{,}5)\,\%$ contre $(94{,}6 \pm 0{,}9)\,\%$) et un score moyen réduit ($4{,}06 \pm 0{,}10$ contre $6{,}20 \pm 0{,}08$).

Cette baisse peut s'expliquer par les contraintes du décodage. Pour encoder le secret binaire tout en préservant les entités incompressibles, l'algorithme modifie l'ordre des mots, substitue des termes spécifiques ou complexifie les tournures syntaxiques (comme illustré dans la Table 2). Si le sens global demeure identifiable, ces variations linguistiques dégradent ponctuellement la fluidité et la fidélité stricte attendues d'une paraphrase naturelle.

Afin d'illustrer concrètement l'impact de l'encodeur stéganographique sur le texte généré, la Table 2 présente un exemple de paraphrase issue du corpus expérimental.

7 Conclusion et perspectives

La stéganographie textuelle générative par paraphrase offre une solution d'ancrage contextuel face aux limites des approches non guidées. Dans cet article, nous avons formalisé ce problème d'optimisation et introduit une méthode exploratoire reposant sur un grand modèle de langue couplé à une recherche en faisceau diversifiée, en ajoutant une contrainte de parité de bloc. Évaluée sur le corpus *Wikipedia-en-sentences* à l'aide de métriques récentes (ParaPLUIE, SyntaxPLUIE), cette approche confirme la faisabilité de l'insertion par paraphrase. La méthode atteint une capacité de $0{,}39$ bpw, légèrement supérieure à l'état de l'art, tout en garantissant une extraction exacte et déterministe du secret.

Toutefois, la comparaison systématique avec des paraphrases générées sans contrainte met en évidence le coût de cette dissimulation. L'application stricte des règles d'encodage se traduit factuellement par une élévation significative de la perplexité globale et une diminution de la proportion de paraphrases jugées syntaxiquement et sémantiquement valides.

Ces observations ouvrent plusieurs perspectives d'optimisation. À court terme, une étude systématique des hyperparamètres de l'encodeur (augmentation de la base modulaire $n > 2$, taille des blocs k, largeur du faisceau) permettra de cartographier plus finement les limites opérationnelles du système. Il conviendra également de confronter ces stégo-textes à des classifieurs stéganalytiques neuronaux pour en valider l'indiscernabilité pratique.

Enfin, sur le plan algorithmique, l'exploration d'une taille de bloc d'encodage dynamique constitue une piste de recherche prometteuse pour mieux s'adapter aux variations d'entropie de la langue.

Remerciements

Nous remercions Etienne Cauty pour sa contribution aux phases exploratoires de ce projet.

Annexes

Un exemple de production est présenté dans la Table 2.

Version	Texte généré et scores associés
Couverture (C)	*"The Dandenong railway line triplication project was an initiative of the state government of Victoria, Australia, to add sections of a third railway line from Caulfield to Dandenong to expand the capacity of and relieve congestion on the Pakenham and Cranbourne railway lines, part of the Melbourne suburban rail network."* **Scores :** Perplexité : 7,14
Paraphrase sans contraintes	*"To alleviate overcrowding on the Pakenham and Cranbourne railway lines, which are part of Melbourne's suburban rail network, the state government of Victoria, Australia, undertook a project to construct additional sections of a third railway line on the Dandenong railway line, extending from Caulfield to Dandenong."* **Scores :** ParaPLUIE = 7,26 | SyntaxPLUIE = 7,73 | Perplexité = 7,99
Paraphrase stega (S)	*"In Victoria, Australia, the Dandenong railway line tripling project was a state government undertaking aimed at alleviating overcrowding on the Pakenham and Cranbourne railway lines by introducing an additional railway line between Caulfield and Dandenong, expanding the rail network's capacity."* **Scores :** ParaPLUIE = 8,32 | SyntaxPLUIE = 8,08 | Perplexité = 8,90

TABLE 2 – Exemple de génération comparant un texte de couverture, une paraphrase de référence (sans contrainte) et une paraphrase stéganographique. Les scores individuels d'évaluation sémantique (ParaPLUIE), syntaxique (SyntaxPLUIE) et de naturalité (Perplexité) sont précisés pour chaque version générée. Le secret encodé dans la paraphrase stéganographiée est : 1001111000111001.

Références

AL-YOUSUF F. & DIN R. (2020). Review on secured data capabilities of cryptography, steganography, and watermarking domain. *Indonesian Journal of Electrical Engineering and Computer Science*, **17**, 1053. DOI : 10.11591/ijeecs.v17.i2.pp1053-1058.

ALMAYYAHI M., SULAIMAN R., SHUKUR Z. & HASAN M. K. (2021). A review on text steganography techniques. *Mathematics*, **9**, 2829. DOI : 10.3390/math9212829.

CHANG C.-Y. & CLARK S. (2010). Linguistic steganography using automatically generated paraphrases. In *North American Chapter of the Association for Computational Linguistics*.

HUO L. & XIAO Y. (2016). Synonym substitution-based steganographic algorithm with vector distance of two-gram dependency collocations. In *IEEE International Conference on Computer and Communications*, p. 2776–2780 : IEEE.

LEMESLE Q., CHEVELU J., MARTIN P., LOLIVE D., DELHAY A. & BARBOT N. (2025). Paraphrase generation evaluation powered by an LLM : A semantic metric, not a lexical one. In *Proceedings of the 31st International Conference on Computational Linguistics*, p. 8057–8087.

LIN K., LUO Y., ZHANG Z. & PING L. (2024). Zero-shot generative linguistic steganography. In K. DUH, H. GOMEZ & S. BETHARD, Éds., *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (Volume 1 : Long Papers)*, p. 5168–5182, Mexico City, Mexico : Association for Computational Linguistics. DOI : 10.18653/v1/2024.naacl-long.289.

OU C., XIANG L. & LIU Y. (2025). Promising multi-granularity linguistic steganography by jointing syntactic and lexical manipulations. *Proceedings of the AAAI Conference on Artificial Intelligence*, **39**(23), 24984–24992. DOI : 10.1609/aaai.v39i23.34682.

SINGH H., SINGH P. & SAROHA K. (2009). A survey on text based steganography. *Proceedings of the 3rd National Conference*.

TOPKARA M., TOPKARA U. & ATALLAH M. J. (2007). Information hiding through errors : a confusing approach. In *Proceedings of the SPIE*, volume 6505, p. 321–332.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. & POLOSUKHIN I. (2017). Attention is all you need. *CoRR*, **abs/1706.03762**.

VIJAYAKUMAR A. K., COGSWELL M., SELVARAJU R. R., SUN Q., LEE S., CRANDALL D. & BATRA D. (2016). Diverse beam search : Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv :1610.02424*.

WINSTEIN K. (1998). Lexical steganography through adaptive modulation of the word choice hash. Personal Communication, Online Available : http://web.mit.edu/keithw/tlex/.

WU J., WU Z., XUE Y., WEN J. & PENG W. (2024). Generative text steganography with large language model. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, p. 10345–10353 : ACM. DOI : 10.1145/3664647.3680562.

YANG T., WU H., YI B., FENG G. & ZHANG X. (2022). Semantic-preserving linguistic steganography by pivot translation and semantic-aware bins coding.

YANG Z., GUO X., CHEN Z.-M., HUANG Y. & ZHANG Y. (2019). Rnn-stega : Linguistic steganography based on recurrent neural networks. *IEEE Transactions on Information Forensics and Security*, **14**, 1280–1295.

ZHANG S., YANG Z., YANG J. & HUANG Y. (2021). Provably secure generative linguistic steganography. In C. ZONG, F. XIA, W. LI & R. NAVIGLI, Éds., *Findings of the Association for Computational Linguistics : ACL-IJCNLP 2021*, p. 3046–3055, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.findings-acl.268.

ZIEGLER Z., DENG Y. & RUSH A. (2019). Neural linguistic steganography. In K. INUI, J. JIANG, V. NG & X. WAN, Éds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 1210–1215, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1115.