La troisième voie des « Code LLM » ouverts et éthiques : une stratégie de différenciation durable face aux modèles extractivistes ?

Robert Viseur[1]

[1] UMONS, Faculté Warocqué d'Économie et de Gestion

robert.viseur@umons.ac.be

Résumé

L'article examine si la publication de « Code LLM » respectueux des licences libres ou open-sources peut constituer une stratégie viable de différenciation face aux modèles extractivistes. Il montre que la conformité juridique et la prise en compte des communautés (licences, attribution, pratiques de collecte) renforcent la légitimité et l'acceptabilité. Toutefois, leur performance encore inférieure limite aujourd'hui leur capacité à peser sur le marché des assistants à la programmation.

Mots-clés

Code LLM, open model, droit d'auteur, FLOSS, StarCoder.

Abstract

This paper examines whether releasing Code LLMs that respect free and open-source software licences can constitute a viable differentiation strategy vis-à-vis extractivist models. It finds that legal compliance and attention to community norms (licensing, attribution, and data-collection practices) strengthen legitimacy and acceptability. However, their still-lacklustre performance currently limits their ability to make an impact in the programming assistant market.

Keywords

Code LLM, open model, copyright, FLOSS, StarCoder.

1 Introduction

Le terme « *IA générative* » désigne « *des techniques informatiques capables de générer à partir de données d'entraînement des contenus apparemment nouveaux et significatifs* » (p.111) [4]. Trois niveaux sont distingués [4]. Le premier concerne le modèle, par exemple le *Large Langage Model* (LLM) comme GPT-5.5. Le second est relatif au système technique, par exemple l'agent conversationnel tel que ChatGPT, permettant d'interagir avec le modèle. Le troisième est le niveau applicatif relatif aux problèmes résolus avec l'IA générative comme, dans le cas qui nous intéresse davantage, la production d'un code source fonctionnel. La diffusion des modèles d'IA générative s'est accompagnée de nombreux conflits juridiques, qu'il s'agisse de modèles de diffusion ou de modèles de langage [5]. Citons en particulier le litige « *DOE 1 et al v. GitHub, Inc. et al* » portant notamment sur la réutilisation de codes sources sous licences libres pour l'entraînement du modèle CODEX.

Les LLM peuvent donc, comme les Code LLM, être spécialisés. Les « Code LLM » sont notamment capables de transformer en code source des spécifications en langage naturel [7, 8]. Les « Code LLM » ont suscité un intérêt croissant du fait, d'une part, des gains de productivité espérés dans les activités de développement [13], d'autre part, de la démocratisation de la programmation grâce à des approches *low / no code* telles que le *vibe coding* [16]. Parmi les outils les plus connus, citons GitHub Copilot, présenté en juin 2021, puis intégré à la plateforme GitHub, basé sur le « Code LLM » CODEX produit par OpenAI [13]. D'autres ont suivi comme Amazon CodeWhisperer ou Star Coder [7]. Ces modèles sont entraînés sur de vastes corpus de données, notamment composés de codes sources, généralement collectés massivement sur le Web [7]. Dans ce contexte, le modèle Star Coder se distingue par l'attention portée au respect des droits d'auteur et des licences logicielles, tentant ainsi d'échapper à la critique de l'extractivisme des *bigtechs* [2].

Notre communication portera dès lors sur la question de recherche suivante : « *Le choix de publier des « Code LLM » respectueux des licences libres et open-sources constitue-t-il une stratégie viable de différenciation face aux modèles extractivistes ?* ». Notre recherche sera découpée en trois sections. Dans une première section, nous reviendrons sur les bases juridiques liées à la protection des œuvres de l'esprit, en particulier les logiciels, incluant les logiciels libres et open-sources. Dans une seconde section, nous présenterons les griefs portant sur les producteurs de LLM puis, plus spécifiquement de « Code LLM ». Sur cette base, dans une troisième section dédiée à une discussion, et avant de conclure, nous discuterons notre question de recherche au regard des éléments précédents.

2 Propriété intellectuelle

Toute œuvre de l'esprit (un roman, un article scientifique, une photographie, etc.) est automatiquement protégée par droit d'auteur dès lors qu'elle découle d'une activité humaine, a été mise en forme et répond à des exigences minimales d'originalité [9, 12]. L'application du droit d'auteur fait l'objet d'un effort d'homogénéisation à l'aide de traités internationaux comme la Convention de Berne [1], Certains détails diffèrent cependant entre le droit d'auteur étasunien (*copyright*) et le droit d'auteur continental. Le *copyright* reconnaît ainsi le *fair use*, une exception créée par la jurisprudence. Quatre facteurs non

exclusifs sont pris en compte pour déterminer si un usage particulier est équitable [1] : (1) le but et la nature de l'usage (commercial ou non), (2) la nature du matériel protégé, (3) la quantité et l'importance du matériel utilisé et (4) les effets de l'utilisation sur le marché potentiel ou la valeur du travail protégé. Ce dernier critère peut d'ailleurs être rapproché du triple test de la Convention de Berne. Côté continental, cette exception n'existe pas. Cependant, dans le cas particulier des activités de *Text and Data Mining* (TDM), le *scraping* de ressources en ligne est autorisé dans un cadre commercial dès lors que l'*opt out*, mis par exemple en œuvre avec le protocole d'exclusion des robots, est respecté [17].

Les logiciels constituent aussi des œuvres de l'esprit, par défaut protégées par le droit d'auteur. Le caractère libre ou propriétaire d'un logiciel découle de la licence qui encadre les usages autorisés [11]. Les logiciels propriétaires relèvent du régime le plus courant : l'éditeur conserve un contrôle étroit sur l'exploitation du programme afin d'en tirer des revenus via la concession de licences. Ces licences prévoient généralement des restrictions, telles qu'un usage soumis à redevance, l'interdiction de copier, l'absence d'accès au code source, ainsi que l'interdiction de modifier et de redistribuer le logiciel. À l'inverse, les logiciels libres se définissent par les libertés garanties à l'utilisateur : la liberté d'utiliser le logiciel, la liberté d'accéder au code source pour l'étudier ou le modifier, et la liberté de redistribuer des copies, gratuitement ou non (FSF). Certaines licences libres (p. ex. GPL) intègrent un mécanisme de *copyleft* visant à empêcher l'appropriation, en imposant lors de la redistribution la mise à disposition du code source et le maintien de la même licence. À l'inverse, les licences permissives (p. ex. BSD) autorisent plus facilement la réutilisation et l'intégration dans des logiciels propriétaires, sans obligation de réciprocité. Enfin, les licences hybrides combinent des clauses inspirées du libre et du propriétaire : elles peuvent autoriser l'usage, la copie et l'accès au code, tout en limitant certaines modalités de distribution. Elles ne satisfont donc pas pleinement aux critères du libre, sans relever entièrement du propriétaire.

3 Conflits juridiques et éthiques

La diffusion des LLM s'est, d'une manière générale, accompagné de conflits judiciaires autour de l'utilisation de données massivement collectées sur le Web pour l'entraînement des modèles. Les modèles de la famille GPT sont ainsi substantiellement entraînés à partir de données du Common Crawl, une archive du Web collectée par l'association éponyme, mais aussi sur des données scrapées par les producteurs eux-mêmes [17]. En ont très rapidement résulté des procès entre producteurs de modèles et éditeurs à l'image de l'action opposant OpenAI et The New York Times [1]. La plupart des litiges sont actuellement localisés aux USA. Les producteurs y justifient leurs actions par le *fair use* tandis que les auteurs ou éditeurs critiquent cette interprétation, en plus de relever de fréquents problèmes de régurgitation de données d'entraînement, ce qui constitue une violation des droits d'auteur. Côté européen, les producteurs sont protégés par l'exception TDM dès lors que l'*opt-out* est respecté. Les régurgitations excèdent cependant cette exception dès lors qu'elles suggèrent une mémorisation durable des données d'entraînement au sein du modèle.

Dans ce contexte tendu, les éditeurs ont progressivement établi un rapport de force en interdisant l'accès à leurs sites. Pour ce faire, des dispositifs de blocage actif ou passif ont été massivement déployés, à l'image du protocole d'exclusion des robots reconnu comme mécanisme valide d'*opt-out* [17]. Les tensions s'aplanissent cependant progressivement grâce au développement d'un marché de la licence permettant aux producteurs de payer pour disposer d'un accès légal à des données d'entraînement ou de contexte [14].

Dans le cas particulier des logiciels libres, d'autres griefs sont venus s'ajouter. Premièrement, en plus de la polémique sur la violation du droit d'auteur dans les activités de collecte puis d'entraînement, certains responsables de projets libres ont dénoncé les pratiques de collecte agressive (« *agressive crawling* ») conduisant à drainer les ressources des serveurs des projets voire conduisant à l'indisponibilité des dépôts Git autohébergés[1]. Ce phénomène s'explique notamment par des dispositifs tels que Mellowtel, permettant la monétisation des extensions de navigateurs en transformant ces derniers en *crawlers* discrets difficilement détectables. Deuxièmement, certains projets se sont plaints de la croissance de l'« *AI slop* », c'est-à-dire de contenus de faible qualité produits par IA générative. Le projet curl, par exemple, a ainsi vu le nombre de contributions augmenter mais celui des propositions exploitables se réduire, ce qui provoque une dégradation du rapport signal sur bruit dans les contributions au projet, source d'épuisement des mainteneurs [16]. Troisièmement, les capacités d'un « Code LLM » peuvent pénaliser le modèle d'affaires des éditeurs, comme dans le cas de Tailwind CSS (forte popularité mais réduction du chiffre d'affaires de 80 % et licenciement de 75 % des ingénieurs[2]). Ces constats crédibilisent la critique de l'extractivisme des *bigtechs* [2], c'est-à-dire la collecte de données, sans considération pour les ressources liées à leur hébergement ni contribution utile.

Lorsque la licence est mentionnée, son respect peut être exigé. Ainsi une *class action* est menée par des développeurs contre la plateforme GitHub (« *DOE 1 et al v. GitHub, Inc. et al, No. 4:2022cv06823 - Document 195 (N.D. Cal. 2024)* ») ainsi que contre OpenAI, producteur du modèle CODEX, et Microsoft, producteur de Copilot et propriétaire de la plateforme GitHub depuis 2018 [3]. Le premier grief concerne la vente supposée de code source par GitHub malgré les conditions générales d'utilisation (*Term of Use*, ToS) de la plateforme. Le second grief concerne le non-respect de la licence open-source car, d'une part, le producteur réalise une reproduction du code source (entraînement), d'autre part, l'IA générative est parfois sujette aux régurgitations de code source (génération), et ce, sans attribution (paternité) ni mention

[1] Cf. https://www.theregister.com/2025/03/18/ai_crawlers_sourcehut.

[2] Cf. https://www.indiatoday.in/ (...) -project-2849230-2026-01-09.

correcte de la licence [3]. En pratique, seul un des plaignants s'est révélé capable de démontrer qu'un *prompt* conduit à une réponse plagiant son code source tandis que seuls deux autres (sur les cinq) se sont révélés capables de démontrer la similitude [6]. Or, l'application du DMCA suppose une reproduction à l'identique [3, 6]. Finalement, seule la violation de la licence open-source a été conservée par le juge.

Reste que la mention de la licence n'est pas systématique, même sur GitHub. Ainsi, en pratique, l'affectation d'une licence aux logiciels publiés en ligne n'est pas systématique. Par exemple, lors d'une étude réalisée en 2013, seul 14,9 % des répertoires de code source sur GitHub présentait un fichier de licence à la racine du projet [10]. Parmi ceux-ci, près d'un tiers (28 %) mentionnait la licence dans le fichier README plutôt que dans un fichier LICENSE ou COPYING, comme recommandé. La plupart du code publié sur GitHub n'était donc pas open-source. Une étude menée en 2024 sur des plateformes de gestion de *package* a montré que l'absence de licence restait endémique dans certains environnements comme PyPi mais s'améliorait sur d'autres comme RubyGems [19]. Dans le cas des deux études, les licences permissives comme MIT et Apache sont privilégiées par les développeurs. Or, un code publié, en l'absence de licence explicite, est soumis au droit d'auteur et doit donc être considéré comme un code source propriétaire. Vendome confirme la forte diffusion de la licence permissive Apache sur GitHub [15].

4 Vers des « Code LLM » plus éthiques ?

Le respect de la paternité et des licences d'un logiciel est cependant au cœur de certains projets d'intelligence artificielle. C'est notamment le cas du jeu de données The Stack v2 [8]. Mené dans le cadre du BigCode Project, The Stack v2 comprend : (1) du code source provenant de l'archive de Software Heritage, avec un filtrage par licence pour ne garder que les codes sous licence permissive (donc rien sans licence ni *copyleft*) après détection via ScanCode[3] ; (2) des *issues* GitHub reprises de GHArchive ; (3) des *pull requests* de GitHub reprises de GHArchive ; des Notebooks de Jupyter extraits de l'archive de Software Heritage ou publiés par Kaggle ; (4) des *crawls* de documentation, incluant Read The Docs pour sa clarté ; (5) des extraits de StackOverflow et de RedPijama. Cette diversité de données est notamment nécessaire pour pouvoir relier des consignes de codage aux codes appropriés. Ce jeu de données est par exemple utilisé par le modèle StarCoder 2.

L'immixtion de l'IA générative dans les projets de logiciels libres s'est exposée à de fortes résistances comme l'attestent les exemples des projets curl ou Gentoo [16]. L'exposé des principaux griefs a par exemple été synthétisé au sein de l'« *AI Policy* » de Gentoo Linux. Ces griefs concernent, premièrement, la problématique du respect du droit d'auteur des réponses (« *copyright concerns* »), deuxièmement, la médiocrité des codes sources générés (« *quality concerns* ») et, troisièmement, diverses problématiques éthiques (« *ethical concerns* »). Ces dernières incluent à nouveau la problématique du droit d'auteur, cette fois-ci pour le volet de la provenance des données d'entraînement (et plus le manque de traçabilité des contributions en lien avec le risque de régurgitation). Cette préoccupation ressort de nombreuses discussions où la GenAI est perçue comme une sorte de « *copyright laundering machine* » imposée par les *bigtechs*.

Sur le plan de la conformité juridique, un modèle respectueux des droits d'auteurs des contributeurs et des licences (en excluant des données d'entraînement des logiciels couverts par des licences à fort devoir contributif) favorise l'accueil dans le contexte de projets libres puisqu'il répond aux préoccupations légales et éthiques. De même, le soin accordé par BigCode Project aux procédures de collecte respectueuse répond aux préoccupations émergentes liées aux pratiques de collecte agressive. Par contre, ces modèles ne semblent pas aptes, à l'heure actuelle, de répondre au grief relatif à la qualité des réponses. Ainsi, les modèles entraînés sur base de *datasets* plus complets démontrent encore des performances sensiblement supérieures. Jiang et ses co-auteurs mesurent ainsi, pour le test « Pass@1 » (celui-ci représente le pourcentage pour lequel le modèle de langage utilisé une seule fois a pu générer la bonne réponse) un score de 92,0 pour Claude 3.5-Sonet (meilleur modèle propriétaire) contre 90,2 à DeepSeek-Coder-V2-Instruct (meilleur modèle ouvert[4]) mais 72,6, 46,3 et 33,6 pour respectivement StarCoder2-Instruct, StarCoder2 et StarCoder [7]. Le modèle DeepSeek-Coder-V2, bien qu'ouvert, s'appuie sur une collecte de données plus large [20]. Son corpus de pré-entraînement, notamment basé sur GitHub et Common Crawl, comprend ainsi 60 % de code source, 10 % de mathématiques et 30 % de langage naturel. Le filtrage des données collectées est davantage justifié par des critères de qualité que de conformité aux licences sans qu'un audit plus approfondi soit possible, du fait du manque de transparence. L'exigence de conformité semble dès lors moins passer par l'ouverture des modèles que par une meilleure gouvernance des données utilisées pour leur entraînement.

5 Conclusion

Cette étude suggère que la publication de « Code LLM » respectueux des licences ouvre une voie de différenciation principalement fondée sur la légitimité (liée à la conformité, à l'attribution et aux pratiques de collecte) et sur une meilleure acceptabilité auprès des communautés associées aux projets de logiciel libre. Néanmoins, tant que persiste un écart de performance avec les modèles entraînés sur des corpus plus larges et juridiquement ambigus, cette différenciation reste fragile.

[3] Cf. https://scancode-toolkit.readthedocs.io/en/latest/getting-started/home.html. Cette vérification est justifiée du point de vue de l'objectif de conformité étant donné les incohérences dans la documentation juridique des projets sur GitHub [18].

[4] Nous utilisons le terme modèle ouvert (« *open model* ») pour inclure tant les modèle open-sources (au sens de l'Open Source AI Definition) que des modèles open-weights soumis à des licences moins ouvertes comme la licence OpenRAIL couvrant StarCoder2.

La viabilité de cette stratégie dépendra donc de la capacité à réduire cet écart et à inscrire ces modèles dans une chaîne d'outils capable d'articuler garanties de conformité et assistance au codage au sein d'un écosystème de confiance. Dans un contexte de contentieux et d'incertitude réglementaire, cette approche apparaît comme une stratégie de gestion du risque attractive pour les fournisseurs et pour les organisations, en particulier dans les segments de marché sensibles à la conformité (p. ex. secteur public). Cependant, cette différenciation demeure principalement défensive : elle permet de réduire les risques juridiques, réputationnels et d'acceptabilité associés aux modèles extractivistes, sans remettre en cause les mécanismes plus fondamentaux de captation de valeur sur lesquels repose l'économie des « Code LLM ».

6 Références

[1] Al-Busaidi, A. S., Raman, R., ... & Walton, P. (2024). Redefining boundaries in innovation and knowledge domains: Investigating the impact of generative artificial intelligence on copyright and intellectual property rights. Journal of Innovation & Knowledge, 9(4), 100630. https://doi.org/10.1016/j.jik.2024.100630.

[2] Chandrasekhar, R. (2025). Legal frictions for data openness: Reflections from a case-study on re-use of the open web for AI training. Centre Internet et Société - CNRS; Inno3; Open Knowledge Foundation, pp.75. https://dx.doi.org/10.5281/zenodo.15097649.

[3] Farcon, J. F. (2024). Attribution Or Attrition? Doe 1 V. Github, Inc. As A Case For A Robust, Horizontal, Moral Right Of Attribution In Gen AI. https://dx.doi.org/10.2139/ssrn.4946503.

[4] Feuerriegel, S., Hartmann, J., Janiesch, C., & Zschech, P. (2024). Generative AI. Business & Information Systems Engineering, 66(1), 111-126. https://doi.org/10.1007/s12599-023-00834-7.

[5] Freeman, J., Rippe, C., Debenedetti, E., & Andriushchenko, M. (2024). Exploring memorization and copyright violation in frontier LLMs: A study of the New York Times v. OpenAI 2023 lawsuit. arXiv preprint arXiv:2412.06370. https://doi.org/10.48550/arXiv.2412.06370.

[6] Guildea, B. (2025). The problem of pleadings in AI copyright litigation: lessons from Getty v Stability (UK) and Doe v GitHub (US). Journal of Intellectual Property Law and Practice, jpaf022. https://doi.org/10.1093/jiplp/jpaf022.

[7] Jiang, J., Wang, F., Shen, J., Kim, S., & Kim, S. (2024). A survey on Large Language Models for code generation. arXiv preprint arXiv:2406.00515. https://doi.org/10.48550/arXiv.2406.00515.

[8] Lozhkov, A., Li, R., Allal, L. B., Cassano, F., Lamy-Poirier, J., Tazi, N., ... & de Vries, H. (2024). Starcoder 2 and The Stack v2: The next generation. arXiv preprint arXiv:2402.19173. https://doi.org/10.48550/arXiv.2402.19173.

[9] Mattatia, F. (2017), Droit d'auteur & propriété intellectuelle dans le numérique, Paris, Éditions Eyrolles. ISBN : 978-2-212-67426-2.

[10] McAllister, N. (2013). Study: Most projects on GitHub not open source licensed. The Register, 18 avril 2013. https://www.theregister.com/2013/04/18/github_licensing_study/.

[11] Muselli, L. (2008). Le rôle des licences dans les modèles économiques des éditeurs de logiciels open source. Revue Française de Gestion, 181(1), 199-214. https://doi.org/10.3166/rfg.181.199-214.

[12] Novelli, C., Casolari, F., Hacker, P., Spedicato, G., & Floridi, L. (2024). Generative AI in EU law: Liability, privacy, intellectual property, and cybersecurity. Computer Law & Security Review, 55, 106066. https://doi.org/10.1016/j.clsr.2024.106066.

[13] Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). The impact of ai on developer productivity: Evidence from GitHub Copilot. arXiv preprint arXiv:2302.06590. https://doi.org/10.48550/arXiv.2302.06590.

[14] Stratton, M. (2025). Market-Based Licensing for Publishers' Works Is Feasible. Big Tech Agrees. Colum. JL & Arts, 48, 434. https://doi.org/10.52214/jla.v48i4.13925.

[15] Vendome, C. (2015). A large scale study of license usage on GitHub. In 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering (Vol. 2, pp. 772-774). IEEE. https://doi.org/10.1109/ICSE.2015.245.

[16] Viseur, R. (2026). Comment transformer l'« AI slop » en « AI fit » ? Régulation socio-technique des contributions génératives dans les projets libres. Actes de la 31ème conférence de l'AIM, Neuchâtel (Suisse), 20-22 mai 2026.

[17] Viseur, R., & Finet, A. (2025). Dépasser ou respecter la norme? Une analyse de la stratégie de plateformes numériques sous l'angle de la théorie néo-institutionnelle. Innovations, 2025/3 N° 78. https://doi.org/10.3917/inno.pr2.0193.

[18] Wolter, T., Barcomb, A., Riehle, D., & Harutyunyan, N. (2023). Open source license inconsistencies on github. ACM Transactions on Software Engineering and Methodology, 32(5), 1-23. https://doi.org/10.1145/3571852.

[19] Wu, J., Bao, L., Yang, X., Xia, X., & Hu, X. (2024). A large-scale empirical study of open source license usage: Practices and challenges. In Proceedings of the 21st International Conference on Mining Software Repositories (pp. 595-606). https://doi.org/10.1145/3643991.3644900.

[20] Zhu, Q., Guo, D., Shao, Z., Yang, D., Wang, P., Xu, R., ... & Liang, W. (2024). Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence. arXiv preprint arXiv:2406.11931. https://doi.org/10.48550/arXiv.2406.11931.