

Cassandra Crossing 666- Debito Tecnologico e Collasso dello Slopware



Miniatura contenuta nell'*Apocalisse di Bamberg*, codice miniato dell'undicesimo secolo.

(666) - *Gli LLM ci stanno conducendo verso destinazioni tanto evidenti quanto preoccupanti. E quella di una catastrofe software potrebbe essere la più vicina a noi.*

13 maggio 2026 - Per festeggiare questo “diabolico” numero di Cassandra Crossing, la vostra profetessa preferita riaffilerà le sue armi per un problema che le è particolarmente caro e chiaro, dato il suo percorso “professionale”. Si tratta degli effetti a lungo termine dell’uso di LLM.

Non vi parlerà dei soliti argomenti, dei nuovi stuoli di disoccupati, dei cervelli putrefatti e di Skynet che scatena robot cromati ed assassini.

Vi racconterà invece, nuovamente e semplicemente, di un debito che si accumula, di un mucchio di polvere sotto il tappeto che cresce a dismisura e ci seppellirà. Di qualcosa di reale, conosciuto e misurabile, cioè della qualità del software che ci circonda.

La vostra profetessa preferita ringrazia tutti i lettori, sia i 24 indomiti che quelli occasionali, per averle permesso questo piccolo sfogo. Riesce infatti a sopportare città date alle fiamme ed abitanti passati a fil di spada, ma il mondo del software che va a schifo proprio no.

Intendiamoci, l’intera industria della produzione del software ha sempre fatto abbastanza schifo, e le cose hanno sempre continuato a peggiorare. D’altra parte in una situazione in cui i primi a fregarsene sono quelli che lo producono e lo vendono, con l’unico scopo di guadagnare, cos’altro mai sarebbe potuto andare storto?

Ma fintanto che sono stati degli umani a scrivere software schifoso, lo hanno fatto con i limiti delle loro incompetenza e della loro produttività, che sono, per fortuna, finite, e non illimitate.

In questo modo i problemi che si manifestano sono occasionali e circoscritti; qualche apparato per radioterapia perforerà un paziente, un aereo o due cascheranno dal cielo, una sonda interplanetaria scaverà l'ennesimo cratere in un mondo lontano, e forse domani un asteroide giudicato innocuo dal sistema di difesa planetaria finirà il lavoro che il suo predecessore ha iniziato nello Yucatan.

In attesa di questo, vediamo di capire cosa può peggiorare nel software che ci circonda e ci fa sopravvivere. Visto che la quantità di software da cui dipendiamo continua costantemente a crescere e la sua qualità a peggiorare, è evidente che i problemi aumenteranno. Ma di quanto?

Se davvero la maggioranza del nuovo software sarà scritto usando degli LLM, il problema della qualità diventerà ancora più insidioso e pericoloso.

Sì, perché se da una parte tutti sanno che gli LLM sono dei generatori di testi apparentemente raffinati ma solo “**plausibili**”, forse non tutti hanno pensato che **il software scritto dagli LLM possiede esattamente le stesse caratteristiche**. E' un software “**plausibile**” scritto elegantemente, ben formattato e commentato; basta sistemarlo in un paio di punti ed è pronto per andare in produzione, poiché passa tutti i test di prova; anche questi certamente preparati usando gli LLM.

Ma è tutto falso come una banconota da 6 Euro; [Denis Stetskov](#) lo definisce, in maniera efficacissima “**una illusione di esattezza**”.

Infatti anche il software viene creato dagli LLM su base statistica, non di comprensione del problema, “ispirandosi liberamente” alla marea di software con cui è stato allenato, che in grande quantità risiede oggi su Github ed altri repository pubblici.

Lo fa nello stesso modo con cui genera testi con riferimenti e citazioni inesistenti, cioè prendendo pezzi di quello con cui è stato allenato e mettendoli insieme.

E questo è un problema.

Cosa c'è nei progetti su Github? Per il 99% sono progetti appena abbozzati, o copiati e modificati da altri progetti, magari lasciati a mezzo ed infine spesso abbandonati.

Per un progetto software ben scritto e mantenuto, chissà, OpenSSL od il kernel Linux, ce ne sono 99 che fanno pena, e che certo non vorreste controllassero l'aereo che vi sta trasportando.

Ma c'è di peggio; quando un grave bug, come il recente [CopyFail](#), viene trovato e corretto nel repository principale, in quanti dei progetti abbandonati od usati per prove questo verrà effettivamente corretto, ed in quanti invece rimarrà?

E statisticamente, a cosa si “ispireranno” gli LLM? Quale software produrranno visto che i progetti non aggiornati od abbandonati sono molti di più di quelli mantenuti? I peggiori bug diventeranno eterni, riemergendo periodicamente dagli LLM come i [Grandi Antichi di R'lyeh](#), garantendone la loro conservazione e moltiplicazione.

Gli LLM si comporteranno quindi come generatori di grande efficienza **non di codice, ma di bug**, che aumenteranno continuamente di numero e soprattutto di insidiosità in tutto il software prodotto tramite LLM.

Trascuriamo per brevità altri problemi, come la complessità crescente e non gestita del software così prodotto, che lo rende non più mantenibile dagli umani.

Software che infatti nessuno più controlla; “*Gira, è così bello, e poi il mio capo mi misura sulla produttività, non sulla qualità*”.

Quindi nel software inevitabilmente si accumulerà, o meglio si sta già accumulando, un “**debito tecnologico**”, fatto di bug e malfunzionamenti ineliminabili, perché con l’impiego degli LLM li abbiamo dotati della capacità di “riprodursi”.

Al peggio tuttavia, non c’è mai fine, perché l’“aiuto” degli LLM, anche quando utilizzati con moderazione e dove opportuno, potrebbe venir meno proprio quando coloro che sanno programmare solo usando gli LLM diventeranno la maggioranza dei programmatori.

E’ noto dal 2023 che i modelli linguistici, quando addestrati con testi prodotti da altri modelli linguistici, sono “[soggetti a collasso](#)”. Perdono la loro capacità di fornire risposte “credibili” e cominciano a fornirne di inutilizzabili.

Il fenomeno del “[Collasso dei Modelli](#)”, già scientificamente verificato nel caso della generazione di testi, si ripresenterà inevitabilmente anche per la generazione di software, visto che il software “slop” prodotto dagli LLM popolerà, in maniera perlopiù nascosta e difficilmente rilevabile, tutti i repository software che vengono utilizzati per il loro addestramento.

E’ quindi possibile che, proprio quando i programmatori esperti di oggi cominceranno a sparire, **anche gli LLM perderanno le loro capacità di generare software**.

Venir fuori dal pantano del **debito tecnologico accumulato** potrebbe allora essere molto, ma molto difficile.

[Scrivere a Cassandra](#) - [Mastodon](#) - [Buttondown.com](#)

[Videorubrica “Quattro chiacchiere con Cassandra”](#)

[Lo Slog \(Static Blog\) di Cassandra](#)

[L’archivio di Cassandra: scuola, formazione e pensiero](#)

[Cassandra per i posteri: l’archivio](#) su [Internet Archive](#)

Licenza d’utilizzo: i contenuti di questo articolo, dove non diversamente indicato, sono sotto licenza Creative Commons Attribuzione—Condividi allo stesso modo 4.0 Internazionale (CC BY-SA 4.0), tutte le informazioni di utilizzo del materiale sono disponibili a [questo link](#).