Kontrola generatywnej sztucznej inteligencji

Charakterystyka rozwiązania NVIDIA NeMo Guardrails

Duże modele językowe zaskakują swoimi zdolnościami konwersacyjnymi i coraz częściej stają się integralną częścią systemów sztucznej inteligencji. Jednak wraz z ich rosnącą popularnością pojawia się pytanie: jak nad nimi zapanować? Modele te potrafią generować fałszywe informacje, a użytkownicy nierzadko ufają im bezkrytycznie. W niniejszym artykule przedstawiamy jedno z narzędzi, które pozwala skutecznie kontrolować ich działanie – NVIDIA NeMo Guardrails.

KONIECZNOŚĆ KONTROLI MODELI JĘZYKOWYCH

Duże modele językowe (Large Language Models, LLMs) podbiły publiczną wyobraźnię i są aktywnie wdrażane w wielu sferach aplikacji. Stanowią elementy rozwiązań wspierających programistów, do obsługi klienta czy jako systemy rekomendujące. Na łamach „Programisty" pojawiła się już seria artykułów, w tym opisujących korzystanie z tych rozwiązań z wykorzystaniem technik RAG czy dotyczące ich kognitywnych aspektów pracy albo wdrożeń w bezpiecznym środowisku [1], [2], [3].

Niniejszy artykuł suplementuje te wcześniejsze opracowania. Upowszechnienie bowiem LLM-ów i poszukiwanie ich miejsca w ekosystemie biznesowych rozwiązań wiąże się z rosnącą świadomością ich ograniczeń i koniecznością suplementacji za pomocą dodatkowych narzędzi. Wspomniany już RAG poprzez odwołanie się do źródłowych dokumentów pozwala zwiększyć skuteczność modeli w zadaniach z kategorii wyszukiwania wiadomości i udzielania odpowiedzi, a także prezentacji bardziej wiarygodnych uzasadnień. Podobnie wygląda zagadnienie umożliwiania korzystania modelom z zewnętrznych narzędzi (ang. *tools*). Na przykład popularna biblioteka LangChain zawiera szereg rozwiązań umożliwiających wywoływanie LLM-om takich wspomagających podsystemów, jak wyszukiwarki internetowe czy interpretery kodu. Anthropic opracowało standard MCP (Model Context Protocol) w celu umożliwienia integracji systemów sztucznej inteligencji z zasobami w rodzaju źródeł danych czy narzędzi, zaadaptowany również przez inne przedsiębiorstwa.

Wraz z postępującymi wdrożeniami rośnie konieczność odpowiedniej moderacji treści i nadzoru nad tym, w jaki sposób te są prezentowane użytkownikowi. Modele podatne są na specyficzne ataki i mogą generować szkodliwe treści. Na podstawie [4] można tutaj wymienić *jailbreaking* (omijanie ich kontrolek bezpieczeństwa), udostępnianie przez nie informacji umożliwiających identyfikację osób (PII, *personally identifiable information*) czy publikację niezmoderowanych treści. Dobry chatbot oparty o LLM-y powinien być konkretny i nie popadać w dygresje, toteż pożądane jest ograniczenie konwersacji do predefiniowanych tematów, a ze strony biznesowej – limitowanie odniesień do produktów konkurencji w generowanych treściach. Wreszcie, istotne jest ograniczenie halucynacji. Ponadto można wspomnieć również o ataku prompt injection („wstrzykiwania zapytań"), mającym na celu realizację poleceń niezamierzonych przez twórców danego chatbota poprzez przekazanie odpowiednio przygotowanych danych wejściowych.

Szyny bezpieczeństwa (ang. *guardrails*) to rozwiązanie służące do filtrowania, monitorowania i kierowania konwersacją prowadzoną z wykorzystaniem dużych modeli. Poprzez wprowadzenie mechanizmów kierowania konwersacją i zewnętrznych narzędzi służących ewaluacji generowanych treści umożliwiają ograniczanie niekorzystnych zjawisk wymienionych w poprzedzającym akapicie. W opracowaniu omówiona zostanie architektura rozwiązania implementującego tę architekturę, w postaci NVIDIA NeMo Guardrails [6] i jego potencjalne aplikacje. Doświadczenie w pracy z nim pozwoli także na wskazanie niedostatków i paru praktycznych wskazówek ułatwiających jego wykorzystanie.

Koncepcja szyn bezpieczeństwa wpisuje się w szereg istniejących rozwiązań służących do moderacji i sterowania wykonaniem LLM-ów. Łączy się także z nurtem badawczym dotyczącym wyjaśnialności i kontroli nad systemami AI, obecnie w znacznym stopniu stanowiących „czarne skrzynki" – działają w sposób nieprzejrzysty, utrudniając zrozumienie i kontrolę ich decyzji. Również Akt w sprawie sztucznej inteligencji [7] – w art. 13 (wymogi w zakresie transparentności), 14 (ludzki nadzór) czy 86 (prawo do uzyskania wyjaśnienia) – wskazuje na konieczność wprowadzenia odpowiednich rozwiązań technicznych. Za pomocą szyn bezpieczeństwa użytkownik ma zatem możliwość zdefiniowania ścieżek dialogowych i filtrować konwersacje zgodnie z wcześniej zaprogramowanymi regułami. Niektórzy czytelnicy mogą mieć zresztą już doświadczenie w pracy z systemami regułowymi – np. Drools. Do rozwiązań mogących służyć moderacji w aplikacjach *stricte* związanych z LLM-ami służy z kolei np. Perspective API – umożliwiające ocenę, jaki wpływ na konwersację miałaby dana wypowiedź (np. obraźliwa czy mająca charakter groźby) – albo Azure AI content Safety API (również dokonujący ocen, czy dana treść jest np. obraźliwa). W tym miejscu można też wspomnieć na rodzinę modeli językowych wytrenowanych (ang. *instruction-tuned*) do klasyfikacji treści, również dostosowana do zagadnień związanych z moderacją – Llama Guard.

Wykrywać podatności w LLM-ach może rozwiązanie NVIDIA garak – zawierające zresztą moduł integracji z NeMo Guardrails.

W porównaniu do powyższych rozwiązań, w swej istocie architektura szyn bezpieczeństwa pozwala na jawne wyspecyfikowanie zachowań i kategorii treści, które byłyby niepożądane. Niniejszy artykuł uzupełnia pewną lukę, jako że brakuje opisów tego rozwiązania, a dokumentacja również pozostawia miejscami sporo do życzenia (w artykule, między innymi, zmodyfikujemy i szerzej omówimy niektóre z jej przykładów [6]). Będziemy starali się w sposób uporządkowany i systematyczny wprowadzać poszczególne konstrukcje i terminologię stosowaną w bibliotece NeMo Guardrails. Pokażemy także, w jakich okolicznościach i w jaki sposób biblioteka zadaje zapytania modelom językowym, gdyż znacząco ułatwia to zrozumienie jej działania.

W opracowaniu korzystaliśmy z biblioteki w najnowszej dostępnej w chwili pisania wersji 0.14.1. Dziedzinowy język (DSL, domain-specific language) Colang służący do definiowania szyn bezpieczeństwa prezentujemy w jego wersji 2.0. Jakkolwiek wersja ta nie jest domyślna (*gros* dostępnych w dokumentacji przykładów oparty jest na wcześniejszej 1.0) i pozostaje w fazie beta, to zaprezentowanie najnowszych odmian jest bardziej przyszłościowe w perspektywie zastąpienia wersji 1.0. Praktyka wskazuje również, że ta nowsza wersja jest łatwiejsza w odbiorze dzięki wykorzystaniu wcześniejszych doświadczeń jej developerów. W nowszej wersji składnię upodobniono do Pythona, dzięki czemu zmniejszony jest narzut kognitywny związany z nauką nowego języka dziedzinowego. W niniejszym artykule skupimy się wyłącznie na rozwiązaniu NVIDII, unikając tym samym opisów zewnętrznych zależności, a te wprowadzając tylko w niezbędnym zakresie. W tym miejscu wspomnimy jedynie, że omawiane rozwiązanie integruje się z innymi przeznaczonymi dla pracy z dużymi modelami, jak Huggingface czy Langchain. W artykule opisywać będziemy wyłącznie pracę na danych tekstowych.

ARCHITEKTURA SZYN BEZPIECZEŃSTWA W BIBLIOTECE NEMO GUARDRAILS

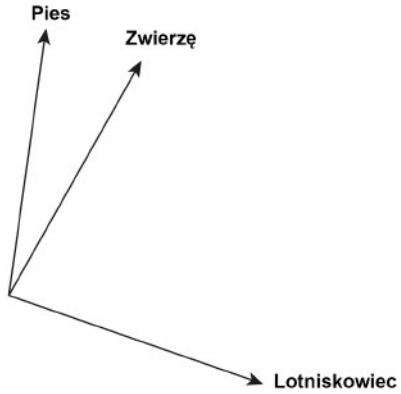
Omawiane rozwiązanie, obok alternatywnej implementacji w postaci Guardrails AI [5], stanowi dodatkową warstwę uzupełniającą wdrażane systemy oparte o duże modele językowe. Do systemu sztucznej inteligencji wprowadzają dodatkowy moduł – zestaw reguł w formie listy wytycznych, które służą do kierowania konwersacją. Stanowią one warstwą pośredniczącą, analizującą dane wejściowe i wyjściowe dla modelu oraz pozwalają na przejęcie konwersacji i wprowadzenie do niej predefiniowalnych i programowalnych wypowiedzi, zgodnie z intencjami wdrażającej system sztucznej inteligencji organizacji. Dzięki temu możliwe jest, między innymi, moderowanie treści, ograniczenie tematyki konwersacji czy wprowadzenie mechanizmów umożliwiających sprawdzanie faktologiczności wypowiedzi.

Rozwiązanie przedsiębiorstwa NVIDIA definiuje pięć kategorii szyn bezpieczeństwa:

a. Wejściowe (input) – moderujące treści podane przez użytkownika.
b. Wyjściowe (output) – moderacja treści generowanych przez model.
c. Dialogowe (dialog) – sterowanie konwersacją, by podążała zgodnie z predefiniowanym kierunkiem.
d. Pobierania (retrieval) – wywoływane w celu zmoderowania danych pobranych w scenariuszu RAG.
e. Wykonawcze (execution) – moderacja narzędzi wspomagających LLM.

Generalnie komunikacja z użyciem szyn bezpieczeństwa zakłada, że różne kategorie treści konwersacji (jak wypowiedzi użytkownika czy te wygenerowane przez LLM) zostają zmoderowane przez narzędzie umożliwiające ich ocenę. W zależności od wyniku tej oceny, przeprowadzanej w ramach poszczególnych szyn bezpieczeństwa, konwersacja może zostać przekierowana na inny temat lub wygenerowany może zostać odpowiedni komunikat diagnostyczny. Oprogramowanie analizuje przebieg rozmowy i w przypadku napotkania wzorców odpowiadających tym wstępnie zaprogramowanym podejmuje odpowiednie działanie.

W NeMo Guardrails mechanizm samego dopasowania może polegać na (a) prostym sprawdzeniu tożsamości sprawdzanego tekstu ze wzorcem bądź wykryciu podobieństwa na poziomie semantycznym z wykorzystaniem (b) podobieństwa wektorowego lub (c) dużego modelu językowego. W przypadku zastosowania trybu (b) teksty, których podobieństwo oceniamy, zapisywane są w wysokowymiarowej przestrzeni wektorowej i wykonywane jest porównanie orientacji tych wektorów. Generalnie wektory reprezentujące zbliżone semantycznie frazy są w tej przestrzeni zbliżone do siebie (Rysunek 1). Ten mechanizm jest wykorzystywany przez wektorowe bazy danych, takie jak Chroma czy pythonowa VectorDB, z którymi zapoznani są czytelnicy mający doświadczenie z aplikacjami typu NLP. W trybie (c) generowane jest zapytanie (ang. *prompt*) dla dużego modelu językowego, który ma za zadanie sklasyfikować dane wejściowe. Rozwiązanie to, poza narzutem obliczeniowym i niedeterminizmem, wiąże się również z ryzykiem wystąpienia halucynacji. Istotne jest więc, jaki model LLM zostanie użyty oraz w jaki sposób zostanie dostosowany do danej aplikacji.



Rysunek 1. Koncepcyjna prezentacja bliskości semantycznej słów „pies" i „zwierzę". Słowo „lotniskowiec" jest znaczeniowo oddalone od obu pozostałych

APLIKACJE SAMOCHODOWE ANDROID AUTO. SENSORY I STEROWANIE GŁOSOWE · Z ARCHIWUM CVE

INDEX: 285358

www.programistamag.pl

Magazyn programistów i liderów zespołów IT

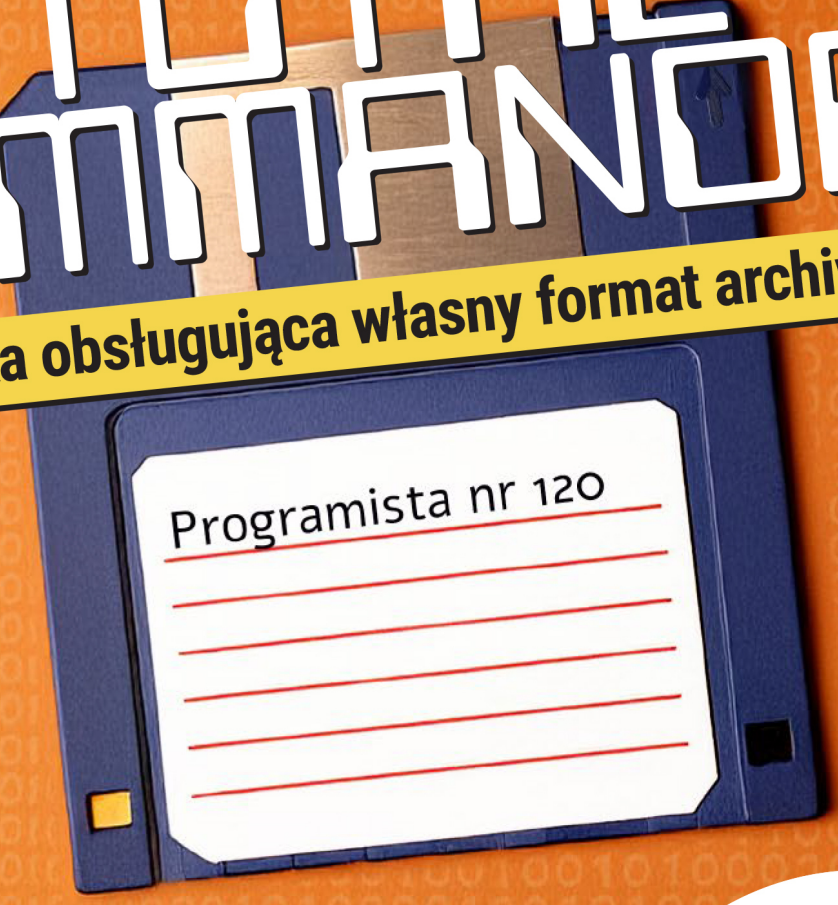
programista

5/2025 (120)

Cena 32.90 zł (w tym VAT 8%)

TOTAL COMMANDER

Wtyczka obsługująca własny format archiwum



ISSN 2084-9400

05

9 772084 940503

- MERA-400. Sequel, który się udał
- Piszemy programy na zegarki Garmin
- Kontrola generatywnej sztucznej inteligen
- Droga przez mękę, czyli testowanie budowanych

SECURITY
LICENSING
ION IN PROTECTION

Programista nr 120 w Empikach