SZTUCZNA INTELIGENCJA W ROLI EKSPERTA OD BAZY WIEDZY W PRZEDSIĘBIORSTWIE. WYKORZYSTANIE RAG

// RAG – po angielsku Retrieval Augmented Generation – przetłumaczone na nasz ojczysty język jako *generowanie wspomagane wyszukiwaniem*, nadal brzmi dość tajemniczo. Postaram się wytłumaczyć co kryje się za tymi trzema słowami.

Autor // PIOTR KMITA

Inżynier mechanik, pasjonat zagadnień związanych ze sztuczną inteligencją i jej wykorzystaniu w szeroko rozumianym temacie predictive maintenance, czyli, najogólniej mówiąc, przewidywaniu serwisu maszyn, zanim nastąpi awaria. piotrkm@gmail.com

AI W PRZEDSIĘBIORSTWIE

Sztuczna inteligencja dziś najpowszechniej kojarzona jest z modelami językowymi LLM (Large Language Models), takimi jak popularny ChatGPT, Gemini, Perplexity, Grok lub rodzime rozwiązania, jak Bielik, PLLuM czy Qra.

Zaprzęgając AI w przedsiębiorstwie w formie RAG możemy osiągnąć wymierne korzyści. W zależności od rodzaju wdrożenia może to poprawić dostępność do wiedzy wewnątrz przedsiębiorstwa, ułatwić dostęp do informacji dla klientów, pozwolić na skuteczniejsze wdrożenie nowych pracowników, poprzez szybkie i wydajne zapoznawanie się z branżową firmową wiedzą, dotyczącą procedur opisanych w wewnętrznej dokumentacji lub informacji o produktach.

Należy pamiętać, że LLM-y to zaledwie jeden z typów modeli AI. Sztuczna inteligencja to bardzo szeroka dziedzina, która znajduje zastosowanie w wielu aspektach życia codziennego. Ze względu na temat niniejszego artykułu skupimy się właśnie na modelach językowych, gdyż są one nierozłączne z techniką RAG.

Załóżmy, że chcemy zasięgnąć wiedzy od ulubionego LLM-a na pewien szczególny temat. Zazwyczaj otrzymamy odpowiedź, niejednokrotnie rzeczową i na temat. Jeśli jednak zakres wiedzy, o który pytamy, jest na tyle specjalistyczny, że model nie posiadał żadnych wartościowych informacji, aby móc się o nie oprzeć, to jego odpowiedź może być rozczarowująca. W najlepszym przypadku będzie nieprecyzyjna i mało odkrywcza, a w najgorszym – nieprawdziwa. Możemy dostać przekonujący i poprawny językowo esej, będący czystą konfabulacją. Takie zachowanie modelu nazywa się fachowo halucynacją. Z tego powodu zalecana jest daleko idąca ostrożność co do tego, co komunikują LLM-y.

Funkcjonowanie firmy prowadzi do powstania pewnego zbioru wiedzy. Są to dokumenty związane z umowami i kontraktami, instrukcje obsługi urządzeń, historia korespondencji z klientami wraz z udzielonymi odpowiedziami, kontakty z serwisem zainstalowanych w naszym przedsiębiorstwie urządzeń, zawierające informacje o metodach usuwania usterek. To również przepisy prawne dotyczące naszej działalności. Słowem – obszerna baza wiedzy spisanej w formie tekstowej. Istnieją także narzędzia pozwalające na włączenie doń innych typów danych, jak np. obrazy, nagrania dźwiękowe, filmy itp.

Treścią takiej bazy danych można „nakarmić" ulubionego LLM-a. Można by przekazać ją wraz z samym zapytaniem, lecz limit maksymalnej długości zapytania uniemożliwia przesłanie jednorazowo większej ilości danych. Rozwiązanie RAG pozwala na wykorzystanie zgromadzonej wcześniej wiedzy i użycie jej do wygenerowania odpowiedzi przez model językowy.

Istnieje wiele komercyjnych narzędzi typu RAG, dostępne są także darmowe. Można użyć ich, w pewnym zakresie, do stworzenia swojego RAG-a. Po przesłaniu dokumentów (w postaci plików tekstowych oraz .doc/.docx i PDF), tworzących wewnętrzną bazę wiedzy, system przekonwertuje je i umożliwi sięgnięcie do ich zawartości. Odpowiedzi LLM-a będą wzbogacane treścią pochodzącą z naszych zasobów i opierać się będą o dostarczoną mu rzetelną wiedzę. Niesie to jednak za sobą pewne ryzyka. Nasza cenna wiedza wychodzi poza firmę, nie wiadomo, kto inny będzie z niej korzystał. Jeśli chcemy uniknąć takiego wycieku danych, konieczne będzie stworzenie RAG-a we własnym zakresie.

REALIZACJA ROZWIĄZANIA RAG

Istnieje kilka możliwości realizacji takiego rozwiązania. Pierwszy został opisany powyżej. W nim wszystko dzieje się „na zewnątrz", tj. poza naszą infrastrukturą IT. Serwer jest wirtualny – wynajęty lub pracuje w chmurze, dostęp do niego odbywa się przez internet. Można go kontrolować poprzez mechanizmy logowania, autoryzacji lub dostępu dozwolonego tylko ze wskazanych adresów sieciowych IP.

Zgoła odmiennym sposobem jest uruchomienie wszystkiego wewnątrz firmy. Wiąże się to jednak z koniecznością zakupu wydajnego komputera wraz z kartą graficzną GPU (Graphics Processing Unit) wspierającą technologię CUDA i odpowiednią ilością pamięci VRAM (VideoRAM). W takim rozwiązaniu zarówno model językowy, jak i baza danych mogą (a nawet powinny) być odizolowane od dostępu z zewnątrz.

Można wyróżnić dwa sposoby korzystania z modeli językowych. Z komercyjnymi modelami, takimi jak ChatGPT lub Gemini, można komunikować się poprzez ich tzw. API (Application Programming Interface). Wiąże się to jednak z opłatami, najczęściej powiązanymi z intensywnością użytkowania konkretnego modelu i jego wersji. Wykupuje się określoną ilość tzw. tokenów, które zużywane są podczas każdego z zapytań. Rozwiązanie to nie musi jednak oznaczać dużych kosztów. Ceny użycia słabszych i starszych modeli mogą wynosić kilka USD za 1 milion tokenów. Jedno zapytanie do LLM-a może zużyć, w zależności od wielkości zapytania, od kilkuset do kilku tysięcy tokenów. Jeśli taki system będzie wykorzystywany sporadycznie, to koszty będą relatywnie niskie, czyniąc takie rozwiązanie optymalnym. Jednak nasze dane „idą w świat" i nie mamy kontroli ani wiedzy, co się z nimi dalej dzieje.

Drugi sposób to uruchomienie modelu językowego wewnątrz firmy. Jak wspomnieliśmy powyżej, wiąże się z zakupem komputera, jego obsługą przez dział IT, osobą implementującą RAG – lub usługą zewnętrznej firmy, która to wszystko wdroży, ale samo korzystanie z LLM-a będzie się odbywać bez dodatkowych kosztów. Modele niekomercyjne, takie jak np. wspomniany Bielik, które można pobrać z internetu i zainstalować u siebie, nie zawsze są podobnie wydajne i równie dobrze wytrenowane jak te komercyjne. Toteż jakość generowanych przez nie treści może odbiegać od odpowiedzi pochodzących z komercyjnych LLM-ów.

Co do samego Bielika – jest to model, który był trenowany głównie w języku polskim, co w pewnych okolicznościach może stanowić jego przewagę nad innymi, uniwersalnymi wielojęzycznymi modelami.

Odrębną sprawą jest lokalizacja zasobów z wiedzą. Tu też są dwie możliwości: lokalna i zdalna. Istnieją wyspecjalizowane zdalne bazy, służące do przechowywania danych do systemów RAG. Zapewniają one szybkie, wydajne przeszukiwanie rekordów i mogą pomieścić znaczne ilości informacji.

Jest też możliwość rozwiązania mieszanego – wewnętrzna baza danych znajduje się lokalnie, w zasobach informatycznych przedsiębiorstwa, zaś model językowy pozostaje na zewnątrz. Schemat ten można – jeśli to uzasadnione – odwrócić. Wybór sposobu realizacji zależy od potrzeb i konkretnych oczekiwań co do jakości działania systemu. Powinien też uwzględniać kwestie związane z bezpieczeństwem danych.

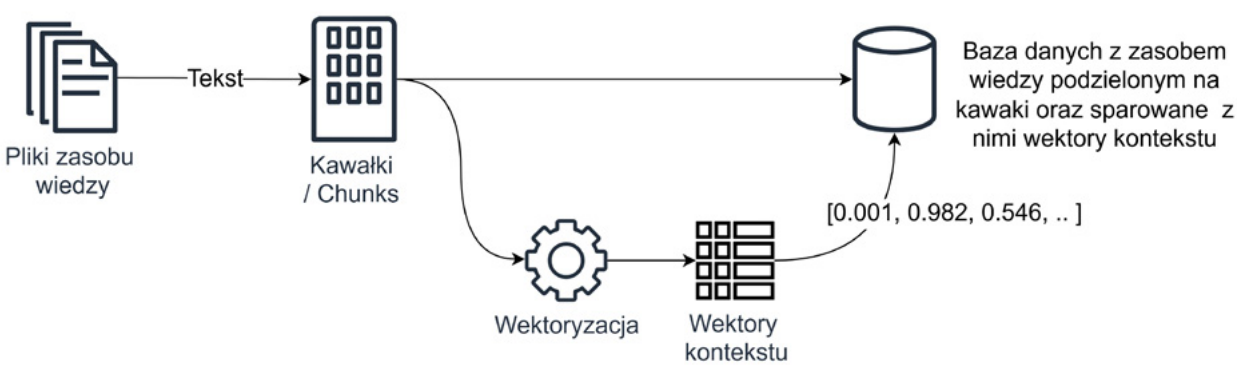
ZASADA DZIAŁANIA

Przyjrzyjmy się zasadzie działania systemu RAG. Najogólniej mówiąc, do modelu językowego wysyłany jest tzw. prompt, którego częścią jest bezpośrednie zapytanie. Prompt w systemach RAG zawiera wycinki naszego zasobu wiedzy oraz dodatkowe parametry. W zależności od modelu prompt posiada pola, np. „system:", w którym przekazuje się instrukcje, w jaki sposób model ma odpowiadać. Może to być instrukcja „Udziel odpowiedzi jak profesjonalny prawnik" albo „Udziel odpowiedzi w punktach, rozważając przesłanki za i przeciw". Inne pole to „user:", które zawiera treść naszego bezpośredniego zapytania. Pole „assistant:", w którym zawarta jest cała nasza dotychczas przeprowadzona konwersja, jest niezbędne, aby model językowy mógł odnosić się do wcześniej udzielonych odpowiedzi. Pole „context:" zawiera fragmenty źródła naszego zasobu wiedzy wyodrębnione przez RAG. Dodatkowymi polami są „temperature:" – zmiana tego parametru ma wpływ na „twórczą dowolność" modelu. Kiedy jest ustawiona na małe wartości – poniżej 1, model będzie ściśle trzymał się dostarczonego tekstu. Gdy zwiększymy tę wartość, odpowiedzi będą bardziej dowolne – twórcze, co jednak może zwiększyć skłonność modelu do konfabulacji i wspomnianych wcześniej halucynacji. Na podstawie tak skonstruowanego promptu model LLM może wygenerować zwięzłą, logiczną i poprawną językowo odpowiedź, opartą merytorycznie na dostarczonych informacjach źródłowych.

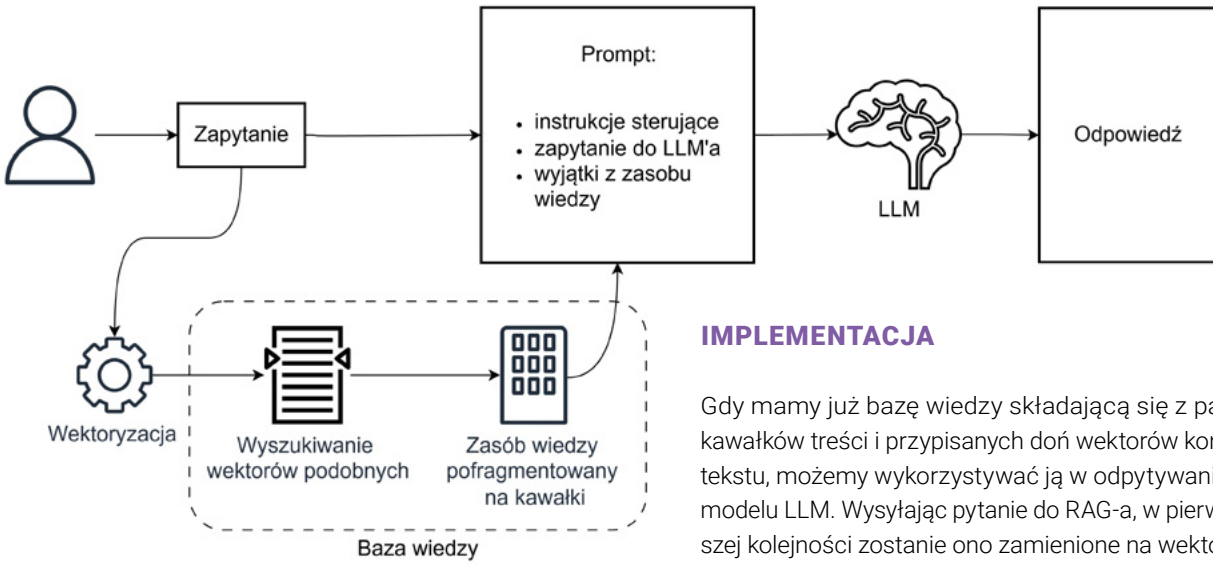
Jak już wspomnieliśmy, do promptu trafiają wycinki naszego zasobu wiedzy. To, które to będą wycinki, jest właśnie zasługą funkcjonalności systemu RAG.

Zanim system zacznie funkcjonować, konieczne jest odpowiednie przygotowanie danych z naszego zbioru. Najlepiej gdy nasze zasoby są w postaci czystego tekstu – plików tekstowych. Wprawdzie możliwe jest pozyskanie treści z plików PDF czy .doc, ale pamiętać należy, że mogą one zawierać zbędne informacje, takie jak stopki, nagłówki czy znaki sterujące. Tekst sformatowany w kolumnach nie gwarantuje kontroli nad sposobem, w jaki zostanie odczytany przez funkcję wyodrębniającą. Toteż zalecane jest świadome, ręczne zapisanie treści jako plik tekstowy – wtedy mamy pewność, co się w nim znajduje. Tak przygotowane treści naszego zasobu wiedzy należy podzielić na mniejsze kawałki – ang. chunks. Dzieje się to w sposób zautomatyzowany. Wielkość kawałka nie jest odgórnie określona, standardowo przyjmuje się 1000 znaków. Równie dobrze może to być np. 10 zdań rozdzielonych kropkami. Zazwyczaj tekst dzieli się na tzw. zakładkę (overlapping), a jej standardowa wielkość to 20% całości. Pierwszy kawałek będzie zawierał 1000 znaków od 1 do 1000, następny od 800 do 1800, kolejny od 1600 do 2600 itd. Technika dzielenia z zakładkami ma na celu zachowanie kontekstu.

Gdy nasz zasób wiedzy zostanie podzielony, kolejny etap to wektoryzacja. Polega na stworzeniu tzw. wektorów kontekstu. Tu ujawnia się „magia" sztucznej inteligencji. Wektor kontekstu to lista liczb zmiennoprzecinkowych, zawierająca – w zależności od użytej funkcji wektoryzującej – najczęściej od kilkuset do ponad tysiąca pozycji. W owym wektorze, zwanym po angielsku *embedding*, zawarty jest kontekst – czyli znaczenie treści pojedynczego kawałka tekstu – w postaci liczbowej. Technika jego generowania jest dość zawiła i wykracza daleko poza ramy tego artykułu. Model wektorujący tworzy go na zasadzie podobieństw semantycznych poszczególnych słów.



Rys. 1. // RAG – wektoryzacja



Rys. 2. // RAG – RETRIEVE

Utworzone w ten sposób wektory mogą być do siebie bardziej lub mniej podobne. Miarę ich podobieństwa określa się za pomocą specjalnych algorytmów. Popularne jest wykorzystanie funkcji podobieństwa cosinusowego, ale używane są również inne funkcje, takie jak: iloczyn skalarny (ang. *dot product*) oraz odległość euklidesowa (ang. *Euclidean distance*).

PRZYKŁAD PORÓWNANIA KONTEKSTU

Wyjaśnijmy to na przykładzie. Załóżmy, że mamy trzy zdania: (1) „Mój kot nazywa się Mruczek", (2) „Pani sąsiadka spod siódemki opiekuje się wszystkimi kotami na osiedlu" oraz (3) „Jak uprawia się ziemniaki?". Dla każdego z tych zdań wygenerowano wektory kontekstu, które zostały do siebie przyrównane. Im bardziej wynik przyrównania jest zbliżony do wartości 1, tym zdania te są do siebie bardziej podobne – w kontekście, którego dotyczą.

Wynik podobieństwa pomiędzy poszczególnymi zdaniami jest następujący:
Zdania (1) i (2) – 0,49
Zdania (1) i (3) – 0,22
Zdania (2) i (3) – 0,23

Inny przykład porównania dwóch zdań: (4) „Boli mnie prawa ręka" oraz (5) „Odczuwam ból w prawej górnej kończynie" dał wynik 0,66. Wyżej wymienione wektory kontekstu zostały wygenerowane przez model *text-embedding-3-small*, dostarczony przez OpenAI.

Zdania (1) i (2) traktują o kotach i pasują do siebie bardziej, niż którekolwiek z nich do pytania o uprawę ziemniaków. Kolejne zdania (4) i (5) dotyczą praktycznie tego samego i choć użyte w nich są zupełnie inne słowa, ich porównanie dało wysokie podobieństwo.

IMPLEMENTACJA

Gdy mamy już bazę wiedzy składającą się z par kawałków treści i przypisanych doń wektorów kontekstu, możemy wykorzystywać ją w odpytywaniu modelu LLM. Wysyłając pytanie do RAG-a, w pierwszej kolejności zostanie ono zamienione na wektor kontekstu. Następnie wektor ten zostanie przyrównany do wszystkich uprzednio utworzonych wektorów dla kawałków bazy wiedzy. Za pomocą funkcji podobieństwa zostanie oszacowany ich poziom podobieństwa. Tam, gdzie wynik podobieństwa będzie najwyższy, wybrane zostaną odpowiednie kawałki zasobu wiedzy, które najbardziej pasują do zapytania. Liczba kawałków może być ustalana na zasadzie statystycznej lub procentowej, wynikającej z rozkładu wyników dla konkretnego przypadku.

Tak przygotowany zestaw, składający się z zapytania oraz dobranych fragmentów zasobu wiedzy, jest wysyłany w postaci rozbudowanego promptu do modelu językowego. Spodziewanym wynikiem będzie odpowiedź poprawna językowo i gramatycznie, oparta na dostarczonych danych.

POTENCJALNE ZASTOSOWANIA

Systemy RAG mogą być pomocne w wielu scenariuszach. Mogą wspierać pracowników w dostępie do zgromadzonej w przedsiębiorstwie wiedzy. Mogą być wystawione na zewnątrz – w postaci chatbota – dla odwiedzających stronę internetową firmy. Potencjalni klienci mogą zapytać go o szczegóły oferowanych produktów, ich ceny oraz wszystko to, co umieszczone zostało w zasobie wiedzy. Takie rozwiązanie może również służyć jako narzędzie pierwszego kontaktu z helpdeskiem.

TESTOWANIE, OPIEKA I ROZWÓJ

Baza danych zasobu wiedzy powinna być administrowana i na bieżąco uzupełniana. Z pewnością system powinien przejść testy. Warto eksperymentować z parametrami. Lepsze efekty może dać zmiana wielkości wektorowanego kawałka treści *chunka* ze standardowego 1000 znaków na większy. Również większa ilość *chunków* zasobu niż standardowe 5-6 może poprawić jakość zwracanych odpowiedzi. //