https://jacek.kwasniewski.org.pl

Jacek Kwaśniewski

Gusteve Doré, Dante i Wergiliusz otoczeni przez demony w ósmym kręgu piekła

[5] Lęk przed śmiercią jako siła kulturotwórcza - mapa zjawiska i zapytania do literatury

Metodologia. Rola matematyka w budowie rdzeni semantycznych

2026

Wprowadzenie do serii

Poniższy zestaw ośmiu tekstów przedstawia projekt mapowania kulturotwórczej roli lęku przed śmiercią. Moją intencją nie jest wyjaśnianie, szukanie przyczyn czy odkrywanie mechanizmów. Cel jest o wiele skromniejszy - zrobić możliwie dokładną mapę tego, w jakich obszarach kultury i poprzez jakie kanały lęk przed śmiercią działa kulturotwórczo - oraz przygotować narzędzie, które pozwoli to sprawdzić w literaturze naukowej.

Mapa powstała przez stopniową dekompozycję tezy ogólnej i zwiększanie rozdzielczości opisu: od ujęcia makro, przez podgrupy, po rdzenie semantyczne, które mogą zostać, przy pomocy narzędzi AI, zamienione na embeddingi i użyte do semantycznego przeszukiwania korpusu literatury. W tym sensie mapa staje się zbiorem zapytań zarzucanych na literaturę jak sieć. Chcemy zobaczyć, co zostało już opisane, co pominięte, czemu poświęcano dużo uwagi, a czemu mało.

Dwa teksty mają charakter metodologiczny. Opisują zasady budowy rdzeni oraz rolę analizy matematycznej w kontroli jakości mapy (kolizje, luki, nadmierne skupienia). Całość traktujemy jako hipotezę roboczą do weryfikacji, nie jako rozstrzygnięcie.

Zainteresowanym udostępniam model Excela ze szczegółową analizą statystyczną rdzeni semantycznych.

Teksty:

[1] Rozpisanie tezy, że lęk przed śmiercią jest kulturotwórczy, na siedemnaście makro twierdzeń (grup). Pierwszy etap dekompozycji **1 → 17** → 55 → 265

[2] Rozpisanie siedemnastu makro twierdzeń na pięćdziesiąt pięć podgrup. Drugi etap dekompozycji 1 → **17 → 55** → 265. Uzasadnienie i schemat opisu podgrup

[3] Od 55 podgrup do 265 rdzeni semantycznych. Ostatni etap dekompozycji 1 → 17 → **55 → 265**. Jakie cechy muszą mieć rdzenie, aby przeszukiwanie literatury dawało ogląd kultury szeroki, głęboki i wielostronny

[4] Metodologia. Zasady budowy rdzeni semantycznych dla modelu embeddingowego

[5] Metodologia. Rola matematyka w budowie rdzeni semantycznych

Załącznik do tekstu [2]. Pięćdziesiąt pięć podgrup (lista opisów)

Załącznik do tekstu [4]. Tabelaryczny opis grup (17), podgrup (55) i rdzeni (265)

Załącznik do tekstu [4]. Opis strukturalny 265 rdzeni semantycznych

Spis treści

Do czego potrzebny jest matematyk przy projektowaniu zapytań do literatury z wykorzystaniem AI 3
Jak matematyk bada, czy rdzenie pasują do chunków i które są nietrafne. 4
Kolizje rdzeni 6
Pokrycie chunków rdzeniami 10
Obciążenie rdzeni chunkami — diagnostyka list wyników 13
Diagnostyka geometryczna - podsumowanie 16

Do czego potrzebny jest matematyk przy projektowaniu zapytań do literatury z wykorzystaniem AI

Ten tekst wyjaśnia, do czego w moim projekcie potrzebny jest matematyk / informatyk od systemów wyszukiwania semantycznego. Jego zadaniem nie jest „ocena sensu" rdzeni semantycznych, tylko diagnoza, czy jako wektory tworzą układ, który działa przewidywalnie: bez kolizji, bez dziur pokrycia i bez rdzeni martwych lub zbyt ogólnych.

Rdzeń semantyczny jest kilkunastowyrazowym tekstem zaprojektowanym w języku naturalnym według pewnych reguł (kotwica → proces → przedmiot → pole/kontekst). Niesie określoną treść semantyczną (sens, znaczenie). Po embeddingowaniu staje się jednym punktem w przestrzeni wektorowej[1] i zaczyna działać jak zapytanie: ma stabilnie przyciągać chunki o najbliższej treści semantycznej.

Kiedy rdzenie semantyczne są już przygotowane, zostają użyte do przeszukania korpusu literatury o lęku przed śmiercią i jego kulturotwórczej roli. Mam dwieście czterdzieści rdzeni, które pokrywają według mnie pole problemowe tego tematu. Skieruję je do literatury (30 000 stron), aby do każdego rdzenia „przylepiły się" fragmenty (chunki)[2] prezentujące i analizujące dokładnie to zagadnienie, które wskazałem w rdzeniu.

Wyszukiwanie odpowiednich chunków opiera się na dwóch elementach: modelu embeddingowym (ME) i programie wyszukującym (np. FAISS). ME zamienia rdzenie i chunki na wektory (embeddingi). FAISS porównuje wektory i dla każdego rdzenia zwraca najbardziej podobne wektorowo chunki.

Model embeddingowy (ME) działa według geometrii, nie hermeneutyki: nie „interpretuje" tekstu jak człowiek, tylko porównuje położenia wektorów. Skutkiem ubocznym jest to, że rdzeń może zwracać wyniki „tematycznie bliskie", ale operacyjnie nietrafne — na przykład, jeśli jego konstrukcja nie daje wektorowi jednoznacznego kierunku.

Rdzeń poprawny językowo i metodologicznie, nie przesądza, czy jest „zdrowy wektorowo": czy nie pokrywa się z innymi rdzeniami, czy nie jest odizolowany i czy realnie trafia w istniejące w korpusie tekstów skupiska treści.

Dlatego potrzebny jest równoległy poziom kontroli: matematyczna diagnostyka relacji rdzeń–rdzeń i rdzeń–chunki. Autor merytoryczny projektuje treść rdzeni; matematyk wskazuje, które rdzenie i które obszary korpusu są podejrzane z punktu widzenia geometrii wektorów.

Zadanie matematyka (w praktyce: inżyniera wyszukiwania wektorowego) polega na diagnostyce, jak pakiet rdzeni zachowuje się jako układ punktów w przestrzeni wektorowej i czy ten układ „pracuje" poprawnie na korpusie chunków. W uproszczeniu sprawdza się trzy rzeczy:

- Kolizje rdzeni – czy część rdzeni nie leży tak blisko siebie, że w praktyce „robią to samo", ściągają te same lub bardzo podobne chunki, przez co się niepotrzebnie dublują.

[1] Rdzeń (oraz chunk) jest zarazem punktem i wektorem. Punktem w przestrzeni i wektorem od początku układu współrzędnych do tego punktu

[2] Chunki – model embeddingowy nie powinien zamieniać na wektory zbyt długich tekstów (ponad 200-300 słów). Strony tekstów trzeba zatem dzielić na mniejsze fragmenty – chunki. W moim projekcie, liczącym 30 000 stron, zakładam dzielenie jednej strony na trzy chunki. Mam więc 90 000 chunków.

- Pokrycie korpusu tekstów rdzeniami – czy istnieją obszary korpusu tekstów, które są daleko od wszystkich rdzeni (dziury pokrycia), oraz czy istnieją rdzenie, które nie mają w pobliżu sensownych chunków (rdzenie „puste" albo źle osadzone).
- Obciążenie rdzeni – czy kilka rdzeni nie działa jak „magnesy" (przyciągają zbyt dużo fragmentów, m.in., bo są zbyt ogólnie), podczas gdy inne rdzenie są marginalne lub puste.

W praktyce rdzenie, po embeddingowaniu, przestają być zdaniami „do czytania", a stają się wektorami wysokowymiarowymi. Dopasowanie rdzeń↔chunk jest wtedy operacją geometryczną (kąty, odległości, sąsiedztwa, gęstości), a nie interpretacją treści. To tworzy naturalny podział kompetencji:

- Autor merytoryczny kontroluje sens i konstrukcję rdzenia: kotwicę, kierunek relacji (co na co działa) i poprawność obiektu działania/efektu.
- Matematyk kontroluje stabilność i geometrię układu: kolizje, dziury pokrycia, degeneracje (magnesy / puste rdzenie), oraz artefakty danych (np. techniczne fragmenty tekstów).

Celem matematyka nie jest „ulepszanie semantyki" rdzeni, tylko diagnoza i ustalenie priorytetów zmian: które rdzenie poprawiać w pierwszej kolejności, gdzie są dziury pokrycia i jak ustawić parametry raportowania (top-K, próg τ)[3], gdy zacznie działać program wyszukiwawczy (np. FAISS).

Wynik tej diagnostyki jest operacyjny: daje listę rdzeni do przebudowy (kolizyjne / puste / magnesy) i wskazówki, gdzie dopisać nowe rdzenie (dziury pokrycia). Dopiero na tym etapie praca merytoryczna nad treścią rdzeni jest dobrze ukierunkowana.

Zastosowane metody zależą od wielkości projektu. W moim projekcie (ok. 90 000 chunków i 240 rdzeni) możliwa jest pełna, „offline'owa" diagnostyka liczona na wszystkich parach rdzeń↔chunk, bo to jest rząd wielkości dziesiątków miliardów elementarnych operacji (a nie bilionów) – jest to zwykle wykonalne jako jednorazowa analiza kontrolna. W większych projektach przechodzi się natomiast od liczenia „wszystkiego ze wszystkim" do pracy na próbkach i do indeksów przybliżonych (próby istotne statystycznie).

Jak matematyk bada, czy rdzenie pasują do chunków i które są nietrafne.

W tym celu należy zbadać:

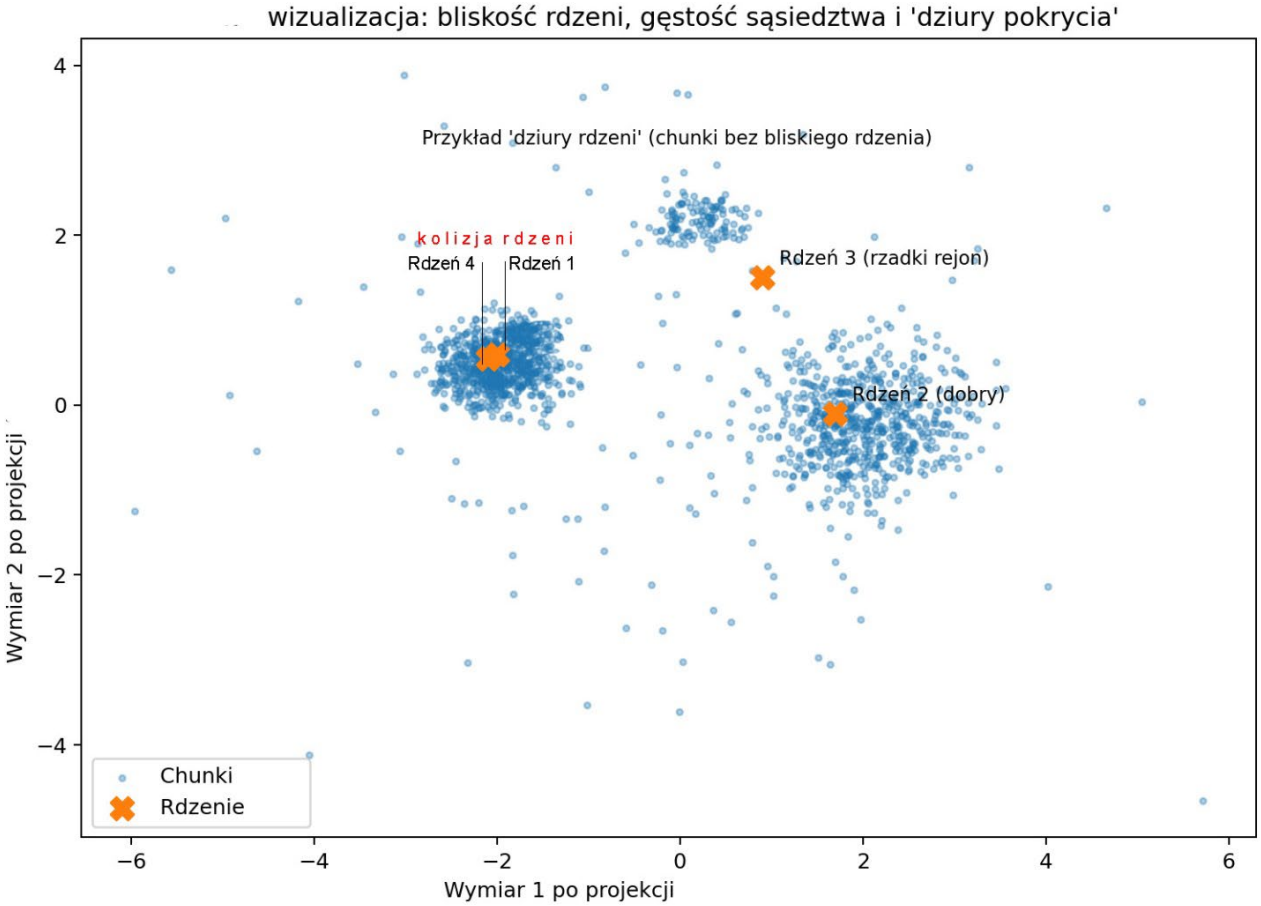
- Kolizje rdzeni
- Pokrycie chunków rdzeniami
- Obciążenie rdzeni chunkami

Wszystkie te kwestie można poglądowo przedstawić graficznie. Służy do tego narzędzie wizualizacji PCA/UMAP, które wykonuje dwu i trzywymiarowe odwzorowania wektorów

[3] Programowi FAISS, który wyszukuje chunki najbardziej zbliżone do kolejnych rdzeni ustawia się m.in. dwa parametry raportowania. Top-K to liczba najbardziej zbliżonych do danego rdzenia chunków, które autor merytoryczny chce zobaczyć. Może to być top-1, top-5, top-20 itd. Próg τ (tau) to miara podobieństwa między rdzeniem a chunkiem. Zwykle jest nią cosinus kąta porównywanych wektorów (od -1 do +1). Im cosinus większy a kąt mniejszy, tym bardziej wektory są do siebie zbliżone pod względem semantycznym. Autor merytoryczny może ustalić minimalny, akceptowany próg podobieństwa rdzenia i chunka, w praktyce od 0,5 do 1,0, poniżej którego chunki nie będą pokazywane

rdzeni i wektorów chunków istniejących w przestrzeni wielowymiarowej, 1536-wymiarowej lub 3072-wymiarowej, zależnie od ME (modelu embeddingowego). PCA/UMAP jest tylko wizualizacją i nie zastępuje diagnozy.

Rysunek pokazuje możliwe relacje między rdzeniami i korpusem rdzeni. Rdzenie 1 i 4 leżą zbyt blisko siebie i zbierają te same chunki. Chmura chunków wokół rdzenia 2 jest dobrze



osadzona, rdzeń ją identyfikuje. Rdzeń 3 jest rdzeniem odstającym (outlier), bo jest położony daleko od reszty, nie ma wokół niego „gęstej chmury" chunków, dopasowania robią się przypadkowe. Jest za ogólny, wieloznaczny albo opisuje rzadkie lub niespotykane w korpusie tekstów zjawisko. Chmura chunków na lewo od rdzenia 3, nie ma pokrycia, żaden rdzeń nie kieruje uwagi na zjawiska opisane w tej części korpusu.

Na podstawie tego rysunku można wskazać, co widzi, mający kompetencje merytoryczne, autor projektu przeszukiwania korpusu literatury, a co widzi matematyk. I jak ich kompetencje mogą się uzupełniać.

Poniższa tabela jest odpowiedzią na pytanie, „po co matematyk"

Problem	Co widzi autor merytoryczny	Co widzi matematyk
Redundancje (kolizje) rdzeni	„To są różne zdania"	Rdzenie leżą bardzo blisko (mały kąt/odległość) → kolizja semantyczna

Problem	Co widzi autor merytoryczny	Co widzi matematyk
„Puste” rdzenie	„Rdzeń ma sens”	Brak stabilnego sąsiedztwa[4] chunków; niska gęstość wokół rdzenia
Rdzenie „za szerokie”	„To ważny temat”	Rdzeń przyciąga za dużo, brak separacji klastrów[5]
„Dziury semantyczne”	Autorowi trudno zauważyć dziury bez mapy[6]	Regiony chunków bez odpowiadających rdzeni (brak pokrycia przestrzeni)
Dryf / zmiana modelu	„Rdzenie są te same”	Wektory się przesuwają, zmieniają sąsiedztwa; spada replikowalność[7]

Kolizje rdzeni

Jeśli dwa rdzenie leżą bardzo blisko w przestrzeni embeddingów, to operacyjnie „robią to samo”: mają bardzo podobne sąsiedztwa w korpusie i będą ściągać bardzo podobne listy chunków.

Taką sytuację nazywamy kolizją (redundancją rdzeni): dwa rdzenie konkurują o te same fragmenty literatury (chunki), więc jeden z nich bywa zbędny albo oba wymagają wyraźniejszego rozdzielenia zakresu.

Kolizje bada się przez podobieństwa rdzeń↔rdzeń, liczone jako cosine similarity (cosinus kąta między wektorami). Im cosinus większy (kąt mniejszy), tym rdzenie są bardziej podobne semantycznie.

Przy 240 rdzeniach liczy się macierz 240×240, czyli 57 600 wartości podobieństwa (w tym porównania rdzenia z samym sobą). Jeśli interesują nas tylko unikalne pary różnych rdzeni, jest ich 28 680[8].

[4] Rdzeń ma listę top-K chunków (np. top-10). Jeśli za każdym uruchomieniem / przy minimalnych zmianach parametrów top-10 mocno się tasuje, to „sąsiedztwo” jest niestabilne.
„Niska gęstość” = w pobliżu rdzenia nie ma wyraźnego skupiska chunków; rdzeń nie trafia w „zagęszczony rejon” korpusu tekstów

[5] „Brak separacji klastrów” - „Klastry” to grupy punktów (np. chunków) w przestrzeni embeddingów. Brak separacji znaczy, że grupy są wymieszane: granice między „rodzinami tematycznymi/procesowymi” są rozmyte, a rdzenie nie „wpadają” w wyraźnie odróżnialne obszary. Efekt: rdzeń przyciąga fragmenty z sąsiednich tematów, bo one są geometrycznie blisko.

[6] Trudne dla humanisty jest zauważenie braków pokrycia: że istnieją rejony korpusu tekstów (chunki o jakimś typie zjawiska), które nie mają żadnego rdzenia blisko siebie. „Mapa” to dowolna globalna wizualizacja/diagnoza przestrzeni: np. PCA/UMAP rdzeni i chunków, albo statystyka „dla każdego chunku: najbliższy rdzeń i jego similarity”. Bez tego widzi się tylko lokalnie: „ten rdzeń działa / nie działa”, ale nie widzi się „obszarów, gdzie nie ma rdzeni”.

[7] Oznacza to, że powtórzenie tego samego procesu (embeddingowanie + FAISS + te same parametry), może dać wyniki top-K i ich podobieństwa mniej powtarzalne w czasie albo między środowiskami. Najczęstsza przyczyna w to inne wersje embeddingów, albo zmiany indeksu/parametrów ANN.

[8] [(240 x 240) – 240] / 2 = 28 680

Następnie dla każdego rdzenia wyszukuje się jego najbliższych sąsiadów wśród rdzeni (kNN na rdzeniach[9]). To jest proste „kto jest najbliżej kogo" w zbiorze 240 punktów — nie ma tu żadnych „pokręteł" typu nprobe/nlist[10], bo ten test można policzyć dokładnie.

Wypisuje się pary o podobieństwie „podejrzanie wysokim", tzn. wyraźnie odstającym od typowego podobieństwa tego rdzenia do reszty rdzeni.

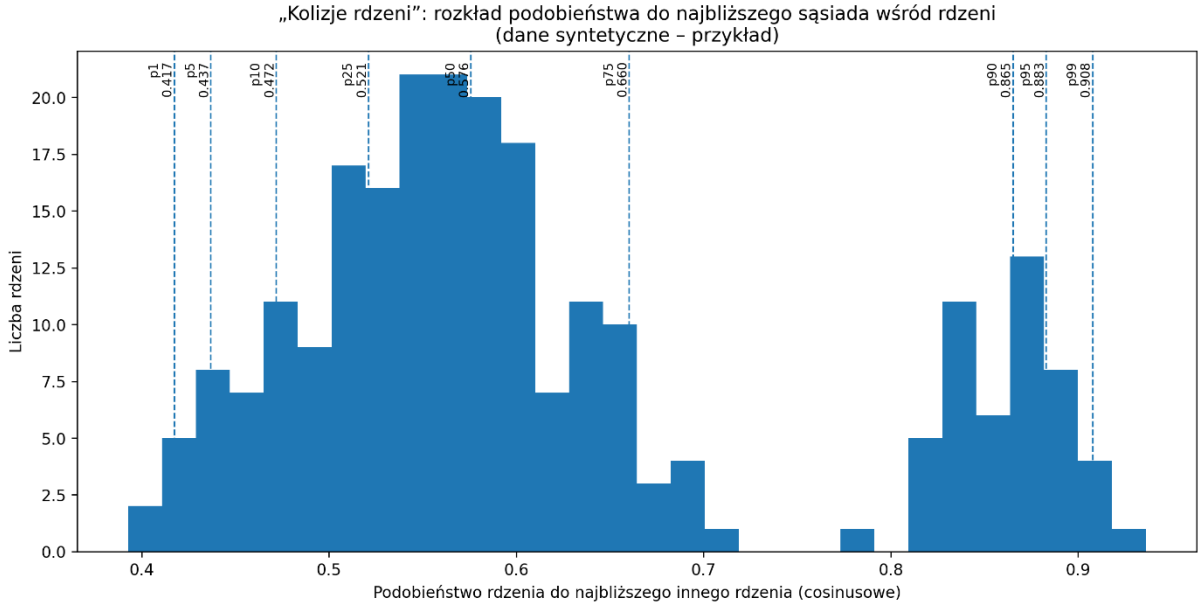
Sama bliskość wektorów nie wystarcza, bo rdzenie mogą być po prostu tematycznie pokrewne. Dlatego kluczowy jest drugi test: podobieństwo list wyników.

Jeśli dwa rdzenie są blisko wektorowo i dodatkowo ich listy rdzeń→top-K chunków w dużej części się pokrywają, to jest to mocny sygnał kolizji. Ale jeśli dwa rdzenie są blisko siebie, ale ich top-K prawie się nie pokrywa, to praktycznie nie kolidują.

Praktyczna zasada brzmi: bliskość rdzeni + podobieństwo list top-K = najlepszy test kolizji.

Wynik tej diagnozy nie mówi „który rdzeń jest merytorycznie lepszy", tylko wskazuje: „tu prawdopodobnie dublujesz zapytanie", więc warto te rdzenie przejrzeć i rozdzielić albo scalić.

Na poniższym wykresie przedstawiam histogram pokazujący kolizję rdzeni



Oś X – CoreNN, dla każdego z 240 rdzeni pokazane podobieństwo do najbliższego innego rdzenia. Dla każdego rdzenia jedna liczba: cosinusowe podobieństwo do najbliższego innego rdzenia (k=1).

Oś Y - liczba rdzeni, których wartość CoreNN wpada do danego przedziału (bina) na osi

Interpretacja wykresu

Lewa część rozkładu (niższe wartości): rdzenie są względnie od siebie odseparowane — mało ryzyka, że „robią to samo".

[9] kNN (k nearest neighbors) to standardowa, klasyczna operacja wyszukiwania. W moim projekcie znaczy "znajdź k najbliższych sąsiadów" wśród rdzeni. Dla 240 rdzeni można policzyć macierz podobieństw (bliskości) i dla każdego rdzenia posortować sąsiadów - wziąć np. 5 najbliższych.

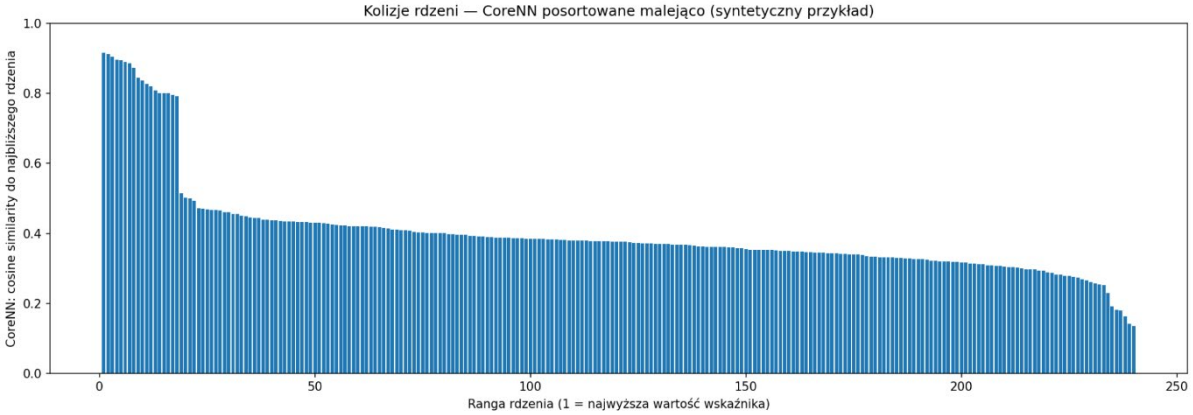
[10] W wielokrotnie większych zbiorach od mojego stosuje się inny wariant FAISS. Nie bada się wszystkich punktów (chunków) przestrzeni wektorowej. Jest ona dzielona na n części (nlist, np. 1000). Dla każdego rdzenia FAISS bada się tylko określoną liczbę tych części (nprobe wyznacza ile, np. 20), które są najbliżej rdzenia.

Prawa strona rozkładu / skupienie blisko 1.0: rdzenie mają bardzo bliskich sąsiadów → kandydaci do kolizji (redundancja, dublowanie sensu lub zbyt podobna konstrukcja).

Podane percentyle[11] (np. P95/P99): dają prostą regułę operacyjną: „sprawdź merytorycznie rdzenie z górnych 5% / 1% najwyższych podobieństw do najbliższego sąsiada".

Bliskość rdzenia do najbliższego innego rdzenia, czyli potencjalną kolizję, można też zbadać sortując rdzenie według ich podobieństwa do najbliższego sąsiada.

Rdzeń↔rdzeń (CoreNN jako rozkład)



ranking: rdzenie posortowane po CoreNN od najwyższych

CoreNN(rdzeń) – „bliskość rdzenia do najbliższego innego rdzenia"

Definicja: cosine similarity rdzenia do jego najbliższego sąsiada wśród rdzeni (kNN na rdzeniach, k=1).

Oś X: ranga rdzenia (1 = najwyższe CoreNN).

Oś Y: CoreNN (cosinus do najbliższego rdzenia).

Interpretacja (krótko):

Początek rankingu (lewa strona, najwyższe CoreNN) to rdzenie, które są najbliżej innych rdzeni. To nie jest jeszcze dowód kolizji w sensie działania systemu, ale to jest lista „podejrzanych par". Jeśli rozkład ma wyraźny, wysoki lewy szczyt, to znaczy, że istnieje grupa rdzeni, które mają „bliźniaków" w geometrii ME (modelu embeddingowego). Jeżeli widzisz stromy spadek, to znaczy, że masz kilka „bardzo ciasnych" par oraz resztę pakietu bardziej rozproszoną. „Ciasne pary" to kandydaci do sprawdzenia.

Ważne: sam wykres mówi o geometrii rdzeń↔rdzeń. Żeby potwierdzić kolizję „w praktyce", należy zrobić test OverlapK(rdzeń), który bada: czy dwa podejrzane rdzenie mają mocno podobne listy top-K chunków.

OverlapK(rdzeń) – kolizja operacyjna na wynikach Top-K

Test polega na badaniu listy top-K wyników rdzeni (chunków) i ich najbliższych sąsiadów. Mierzy się, jaki odsetek fragmentów (chunków) jest wspólny w obu listach. Test odróżnia

[11] Percentyl to miara, która określa, jaki procent danych znajduje się poniżej danej wartości. Przykładowo, dwudziesty piąty centyl (p25) oznacza wartość poniżej której znajduje się 25% danych, siedemdziesiąty piąty centyl (p75) oznacza wartość poniżej której znajduje się 75% danych.

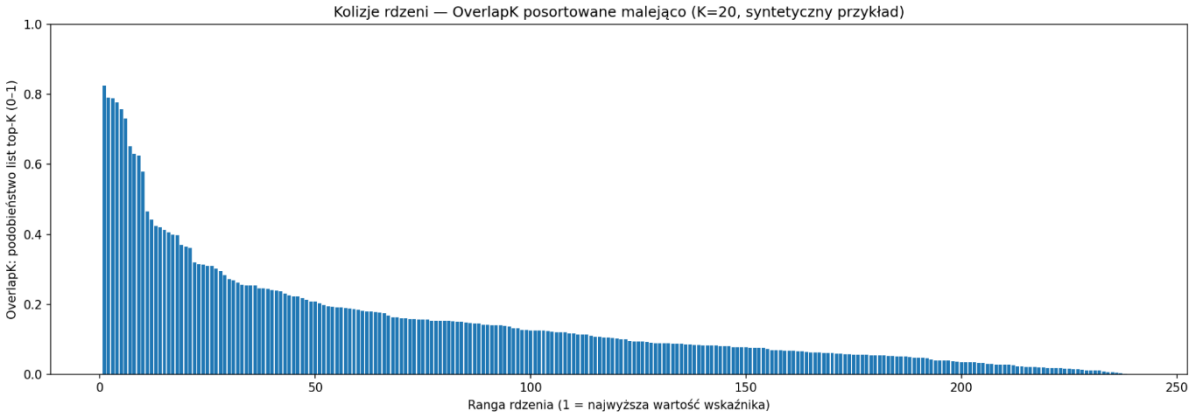
bliskość geometryczną rdzeni od realnego problemu, dwóch rdzeni, które ściągają te same fragmenty i przez to dublują pracę.

Interpretacja

- wysokie OverlapK → rdzenie kolidują operacyjnie (w praktyce przynoszą podobny materiał); te rdzenie są kandydatami do scalenia lub doprecyzowania rozdzielającego,
- niskie OverlapK → nawet jeśli rdzenie są blisko geometrycznie, operacyjnie rozdzielają materiał (często można je zostawić).

Uwaga praktyczna: OverlapK zależy od K (większe K zwykle zwiększa nakładanie), więc w porównaniach między iteracjami K musi być stałe.

Kolizja operacyjna (OverlapK — czy rdzenie ściągają te same chunki)



Ranking: pary z największym OverlapK w top-K

Oś X: ranga rdzenia (1 = największe nakładanie list wyników).

Oś Y: miara „ile wspólnych chunków mają dwa rdzenie w top-K” (operacyjna kolizja). OverlapK w skali 0–1

Decyzje / produkty pracy matematyka – lewy szczyt to:

- listy par rdzeni do sprawdzenia (z CoreNN),
- listy par rdzeni kolidujących operacyjnie (z OverlapK),
- do każdej pary: krótka paczka dowodów (np. 5–10 wspólnych fragmentów z top-K) do oceny przez zespół merytoryczny.

Pułapki interpretacyjne (żeby nie wyciągać fałszywych wniosków)

W praktyce trzy rzeczy najczęściej „psują” interpretację kolizji:

- Zależność od K. OverlapK rośnie, gdy zwiększasz K – to normalne, bo przy większym K listy mają więcej okazji do nakładania się. Dlatego K w diagnostyce trzeba traktować jak stały parametr eksperymentu: porównujesz iteracje tylko wtedy, gdy K jest takie samo.
- Wpływ progu τ. Jeśli odcinasz wyniki poniżej τ, to listy top-K mogą się „sztucznie” skracać albo zmieniać skład ogona. To potrafi obniżyć OverlapK nie dlatego, że rdzenie się rozdzieliły, tylko dlatego, że odciąłeś wspólny szum.
- Huby/boilerplate. Są to głównie elementy techniczne (spisy treści, stopki, nagłówki, szum z OCR itp.) Jeżeli w danych są huby/boilerplate, to sztucznie zawyżają

OverlapK. Dlatego przy wysokim OverlapK zawsze warto spojrzeć na kilka wspólnych chunków: jeśli wyglądają jak strukturalny powtarzalny tekst, to kolizja jest wtórna wobec brudu danych (huby).

Dopiero po odfiltrowaniu tych trzech czynników decyzje są sensowne: czy rdzenie scalić, czy doprecyzować, czy zostawić jako dwa bliskie, ale jednak różne narzędzia badawcze.

Podsumowując miary CoreNN i OverlapK, są to dwie niezależne osie tego samego problemu. CoreNN: „jak blisko stoją rdzenie w przestrzeni" oraz OverlapK: „jak bardzo ich listy trafień są podobne". One nie muszą korelować 1:1, ale korelacja jest widoczna w górnym ogonie. To znaczy:

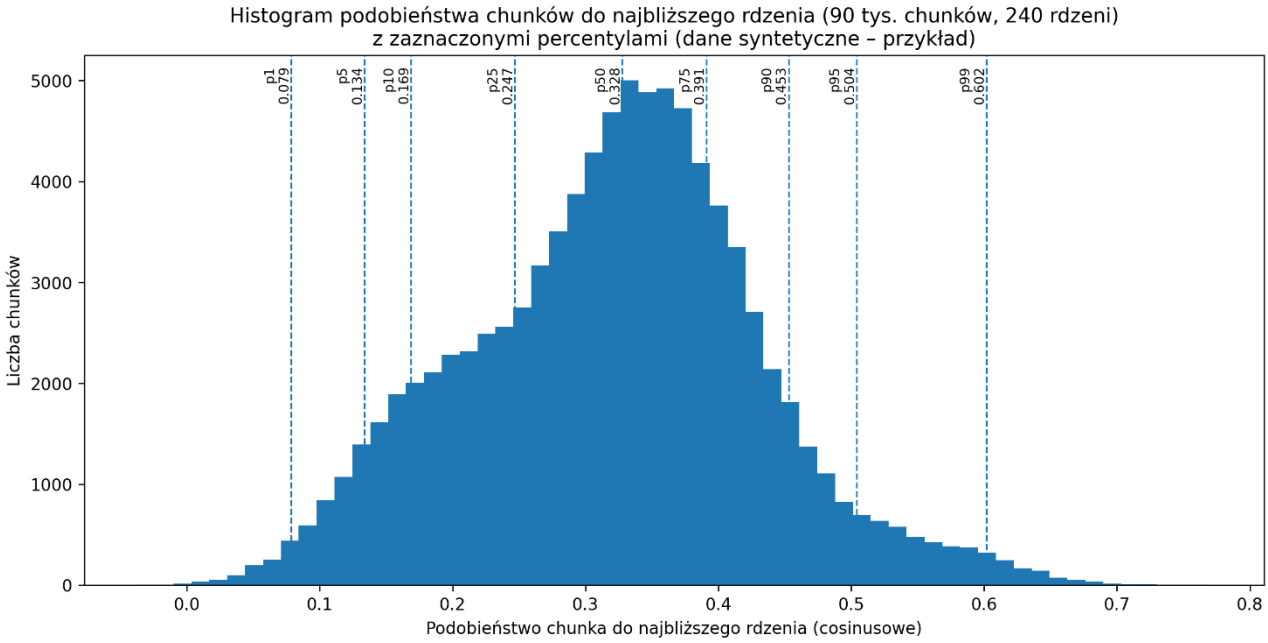
- są pary rdzeni geometrycznie bliskie,
- które jednocześnie łowią te same fragmenty tekstu.
- to jest twardy dowód na redundancję konceptualną, a nie „subiektywne wrażenie"

Pokrycie chunków rdzeniami

Drugie pytanie brzmi: czy pakiet 240 rdzeni pokrywa geometrię korpusu, czy istnieją obszary chunków daleko od wszystkich rdzeni.

Sytuacja byłaby idealna, gdyby każdy fragment tekstu (chunk) był względnie blisko któregoś z rdzeni – to oznaczałoby, że dany rdzeń trafiałby na ten fragment, czyli tematyka każdego chunku mieściłaby się w zakresie któregoś z rdzeni. Jeśli tak nie jest, mamy do czynienia z dziurą pokrycia: część korpusu leży daleko od wszystkich rdzeni, co sugeruje brak odpowiedniego rdzenia dla tych treści.

Przykładowy histogram pokrycia przedstawiony jest poniżej.



Oś X – wielkość podobieństwa chunka do najbliższego rdzenia (cosine similarity). Od 0 (podobieństwa nie ma) do 0,8 (podobieństwo wysokie)

Oś Y – ile chunków ma takie podobieństwo

Wykres pokazuje rozkład wartości podobieństwa kosinusowego między każdym chunkiem a jego najbliższym rdzeniem (czyli takim rdzeniem, do którego dany chunk jest wektorowo najbliżej). Każdy chunk wnosi do histogramu jedną wartość: similarity do najbliższego rdzenia.

Interpretacja histogramu

Im bardziej rozkład przesunięty w prawo (wyższe similarity), tym lepsze pokrycie korpusu rdzeniami: tym częściej chunki mają bliskie dopasowanie do jakiegoś rdzenia w przestrzeni embeddingów (ale warto dodatkowo sprawdzić, czy rdzenie nie są zbyt ogólne)

Im więcej masy po lewej (niskie similarity), tym większe ryzyko dziur pokrycia: istnieją obszary korpusu, dla których rdzenie nie stanowią dobrych „punktów zaczepienia".

Percentyle są praktycznym sposobem streszczenia wykresu bez „wrażenia na oko":

P90 to taka wartość similarity, że 90% chunków ma podobieństwo do najbliższego rdzenia nie większe niż P90 (a 10% ma większe). Innymi słowy: to próg, powyżej którego znajdują się tylko najlepiej pokryte fragmenty korpusu.

P99 analogicznie: tylko 1% chunków ma similarity do najbliższego rdzenia większe niż P99. To „górny ogon" — ekstremalnie dobrze dopasowane chunki.

Dane te mogą posłużyć do następującej analizy:

Dolne percentyle (np. P1, P5, P10) wskazują najgorzej pokryte fragmenty korpusu. Przegląd kilkudziesięciu–kilkuset chunków z tego ogona pozwala rozróżnić dwie przyczyny:

- realne braki rdzeni (trzeba dopisać/rozszczepić rdzenie),
- problemy danych (OCR, boilerplate[12]), które „psują geometrię".

Górne percentyle (P90–P99) pokazują, jak wygląda górna granica jakości dopasowania: czy istnieją w korpusie fragmenty mocno „trafiające" w rdzenie, czy też nawet najlepsze dopasowania są umiarkowane.

To jest diagnoza pokrycia: mówi, jak blisko chunki mają do najbliższych rdzeni. Nie mówi jeszcze, czy wyniki są merytorycznie trafne — to wymaga osobnej oceny treści na próbie rdzeni.

Dziury pokrycia (obszary korpusu słabo przypisane do rdzeni)

Jeśli dużo chunków ma bardzo niskie similarity do swojego najbliższego rdzenia, jest to sygnał dziur pokrycia: rdzenie nie mają kontaktu z istotną częścią korpusu tekstów.

Równolegle liczy się „obciążenie rdzeni": dla każdego rdzenia ile chunków wskazuje go jako najbliższy.

Jeśli kilka rdzeni jest najbliższych ogromnej liczbie chunków, a wiele rdzeni prawie nigdy nie jest najbliższe żadnemu chunkowi, to sygnał, że:

[12] Boilerplate to powtarzalne fragmenty techniczne, które nie są treścią merytoryczną, np.: nagłówki/stopki stron, numery stron, tytuły rozdziałów. Ponieważ one są bardzo podobne w całym korpusie, embeddingi tych fragmentów też są do siebie podobne — i potrafią być „blisko" wielu rdzeni. Efekt: boilerplate staje się hubem-chunkiem, czyli pojawia się w wynikach wielu różnych rdzeni (np. jest w top-20 dla setek rdzeni) mimo że merytorycznie ma małą wartość (bo jest ogólny lub techniczny).

- część rdzeni to „magnesy” (zwykle zbyt ogólne rdzenie - przejmują za dużą część korpusu),
- część rdzeni jest „pusta” (zbyt szczegółowa / źle osadzona / korpus nie ma treści tego typu).

W efekcie „grupy chunków bez rdzeni” wykrywa się przez niskie similarity do najbliższego rdzenia, a „rdzenie bez chunków” przez znikome obciążenie (prawie nigdy nie są najbliższe).

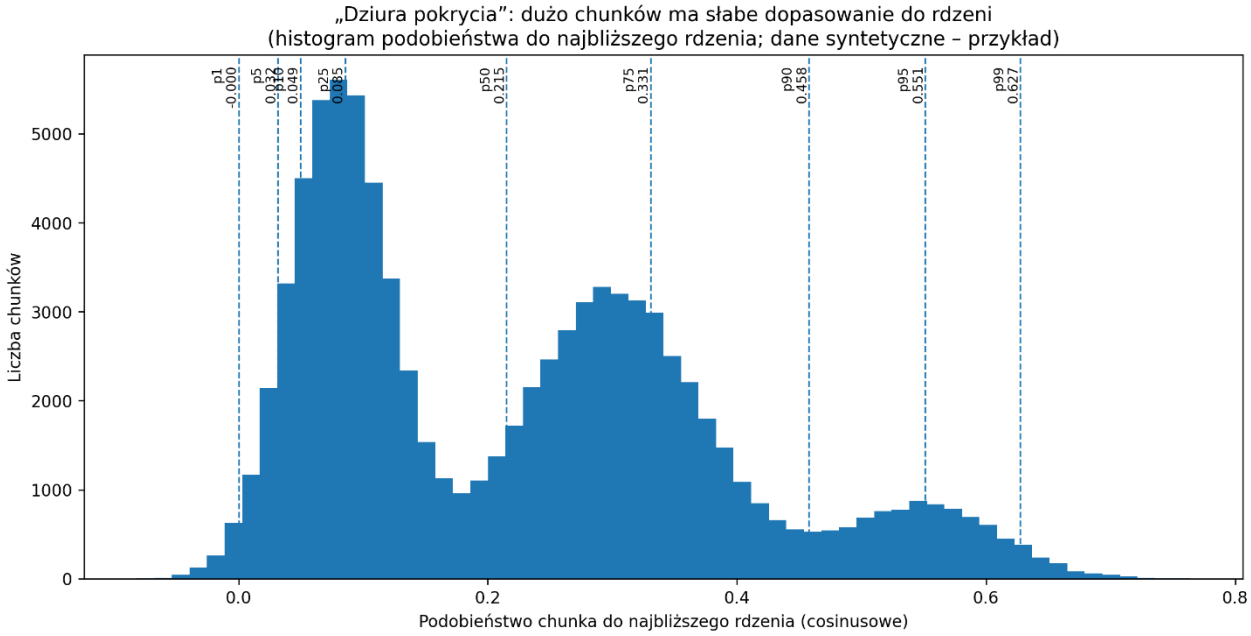
Czy jest tu potrzebna wizualizacja PCA/UMAP? Nie do samej diagnozy. PCA/UMAP jest dobre jako mapa poglądowa (jak na rysunku na str. 3), bo pomaga „zobaczyć” wyspy i dziury. Ale twardą diagnozę robi się liczbami: podobieństwami rdzeń↔rdzeń oraz rozkładem „chunk→najbliższy rdzeń”.

Poniżej prezentowany jest histogram pokazujący dziury pokrycia

Dużo niebieskiego obszaru po lewej stronie (niskie similarity) wskazuje, że wiele chunków ma słabe dopasowanie do jakiegokolwiek rdzenia → dziury pokrycia (rdzenie nie „dotykają” części korpusu).

Jeśli rozkład jest dwugarbny / wielogarbny: jeden „garb” to chunki dobrze pokryte (wyższe similarity, prawa strona), drugi to „populacja” chunków poza zasięgiem rdzeni (niskie similarity).

Percentyle wskazują np. „dolne 10% chunków” (p10) to kandydaci do przeglądu jako materiał, gdzie brakuje rdzeni albo gdzie dane są zaszumione (OCR/boilerplate).



Oś X – wielkość podobieństwa chunka do najbliższego rdzenia (cosine similarity). Od 0 (podobieństwa nie ma) do 0,8 (podobieństwo wysokie)

Oś Y – ile chunków ma takie podobieństwo

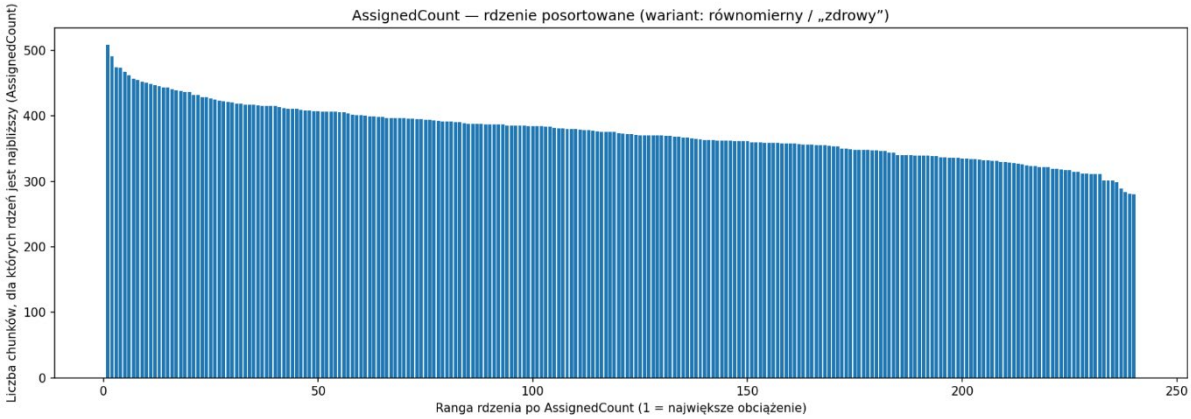
Z samego histogramu nie wiadomo, które rdzenie to naprawią - do tego dochodzi analiza: „które rdzenie są najbliższe tym słabo pokrytym chunkom” + ewentualnie dopisanie/rozszczepienie rdzeni.

Obciążenie rdzeni chunkami - diagnostyka list wyników

Pytanie diagnostyczne brzmi, czy korpus tekstów rozkłada się równomiernie na rdzenie, czy też mniejszość kilka rdzeni „zbiera wszystko”, a część rdzeni prawie nic?

Dobrze sformułowany rdzeń to takie „zapytanie” do literatury, które jako wektor trafia w obszar korpusu opisywany często, obszernie i pod różnymi kątami. W wyszukiwaniu wektorowym nie wystarcza, że rdzeń jest sensowny językowo: musi mieć w korpusie tekstów wyraźne i stabilne sąsiedztwo wektorowe z chunkami.

Wariant zdrowy obłożenia rdzeni chunkami

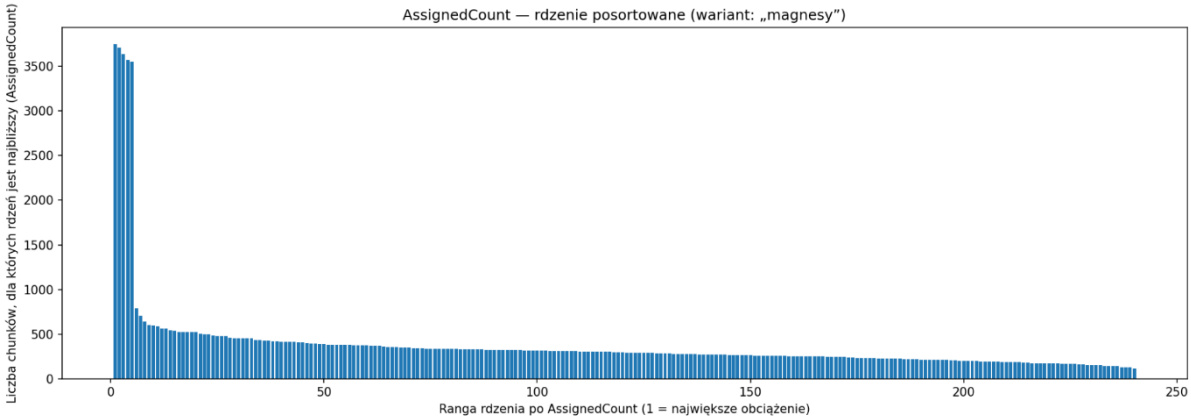


Oś X – Rdzenie posortowane malejąco według liczby chunków, dla których rdzeń jest najbliższy

Oś Y – Liczba chunków dla których rdzeń jest najbliższy

W wariancie zdrowym nie ma ekstremów: brak rdzeni z tysiącami przypisań przy medianie rzędu kilkudziesięciu oraz brak rdzeni z prawie zerem.

Wariant złego obciążenia nr 1 rdzenie-magnesy



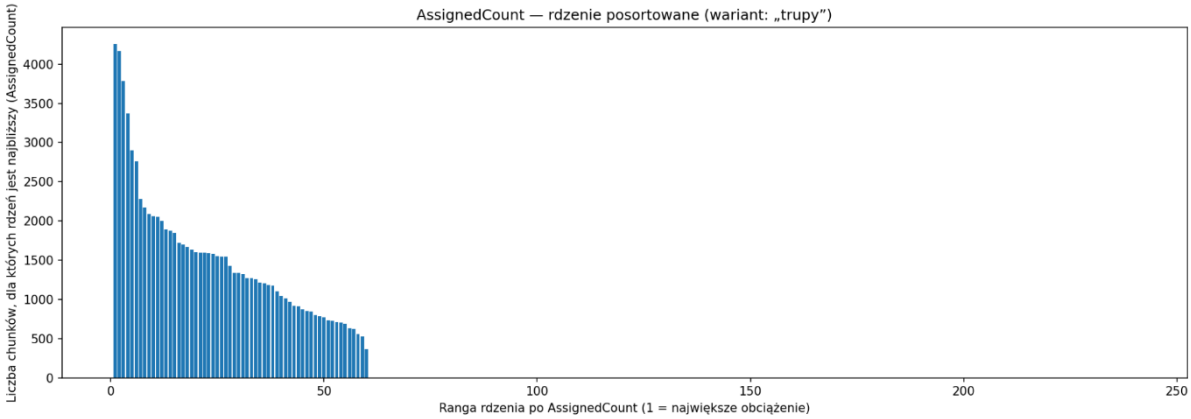
Oś X – Rdzenie posortowane malejąco według liczby chunków, dla których rdzeń jest najbliższy

Oś Y – Liczba chunków dla których rdzeń jest najbliższy

Interpretacja: nieliczne rdzenie są najbliższymi dla wielkiej liczby chunków (lewy szczyt) a większość rdzeni „przyciąga” tylko nieliczne chunki. Rdzenie po lewej stronie nazywane są magnesami. Są zbyt ogólne albo korpus ma powtarzalne fragmenty, które one łatwo

przyciągają. Taki rdzeń wymaga doprecyzowania albo rozbicia na bardziej szczegółowe zapytania.

Wariant złego obciążenia nr 2 rdzenie puste/nieaktywne



Oś X – Rdzenie posortowane malejąco według liczby chunków, dla których rdzeń jest najbliższy

Oś Y – Liczba chunków dla których rdzeń jest najbliższy

Interpretacja: większość rdzeni nie ma żadnego chunka, dla którego jest najbliższa. Są to rdzenie puste, nieaktywne. Traktują o problemie, który jest marginalny w korpusie tekstów albo są tak sformułowane, że nie znajdują zakotwiczenia w danych.

Decyzje / produkty pracy matematyka

- lista rdzeni-magnesów + krótka propozycja działania: doprecyzować / rozbić / sprawdzić, czy nie łapią boilerplate (technicznego szumu),
- lista rdzeni pustych/nieaktywnych + rozróżnienie: (a) temat nieobecny w danych vs (b) rdzeń do przeformułowania,
- do każdej pozycji: kilka pierwszych wyników top-K jako materiał do decyzji zespołu merytorycznego.

Diagnoza geometryczna

Realizując polecenie „znajdź top-K", FAISS zwraca dla każdego rdzenia listę K chunków uporządkowanych od najbardziej podobnego do mniej podobnych, wraz z wartością podobieństwa (np. cosine similarity). Dla rdzenia r wygodnie jest zdefiniować konwencję:

s1 = similarity pierwszego, najwyższego wyniku (top-1),

s5 = similarity piątego z kolei wyniku wyniku (top-5),

s10 = similarity 10. z kolei wyniku (top-10),

s20 = similarity 20. z kolei wyniku (top-20),

s50 = similarity 50. z kolei wyniku (top-50).

Ciąg (s1, s5, s10, s20, s50) jest profilem podobieństw rdzenia. Taki profil pozwala odróżnić typowe sytuacje:

1) Wyraźne sąsiedztwo:

s1 i s5 są względnie wysokie, a spadek do s50 jest zauważalny, ale nie przepaścisty.

Interpretacja: rdzeń trafia w gęstszy rejon korpusu i ma więcej niż jeden sensowny „punkt zaczepienia".

2) Pojedynczy bardzo bliski sąsiad + reszta słaba:

s1 jest wysokie, ale s5/s10 spadają ostro.

Interpretacja: rdzeń ma jeden mocny „strzał", a okolica jest rozproszona; bywa to efekt zbyt wąskiego rdzenia, przypadkowego dopasowania albo zanieczyszczeń (np. powtarzalne fragmenty techniczne).

3) Brak kotwicy:

profil jest niski i płaski (s1 ≈ s5 ≈ s10 ≈ …).

Interpretacja: w korpusie nie ma chunków geometrycznie bliskich temu rdzeniowi; temat jest rzadki, rdzeń jest zbyt abstrakcyjny/wieloznaczny albo ME koduje go zbyt ogólnie.

Pojęć „wysoki", „niski" i „ostry spadek" nie da się sprowadzić do jednej, uniwersalnej liczby. To są oceny relacyjne: porównuje się rdzenie między sobą oraz porównuje „przed/po" zmianach ME, chunkowania i parametrów wyszukiwania, przy tej samej normalizacji[13].

Próg τ (tau)

Poza samym K istotny jest próg τ, czyli minimalne podobieństwo chunka do rdzenia, poniżej którego wynik uznaje się za szum i nie pokazuje. Dla chunków ze znormalizowanymi wektorami, cosine similarity mieści się w przedziale [-1, 1]. Wartości ujemne oznaczają wektory skierowane „w przeciwną stronę"; w praktyce dla embeddingów tekstowych i normalizacji L2 są rzadkie i zwykle nie mają wartości użytkowej. Sensowny τ dobiera się empirycznie: tak, aby odciąć „ogon" wyników, które przy ocenie treści okazują się nietrafne.

Diagnoza merytoryczna

Jak ocenić, czy profile podobieństwa rdzeni są merytorycznie trafne

Diagnoza geometryczna nie rozstrzyga, czy wynik jest trafny interpretacyjnie. Wskazuje, które rdzenie wyglądają zdrowo, a które są podejrzane. Ocena merytoryczna polega na ręcznym przejrzeniu wyników zwracanych przez FAISS dla wybranych rdzeni, np. top-10 lub top-20.

W projekcie jest 240 rdzeni, pełny przegląd jest wykonalny, ale jest to duży koszt ludzki (czas czytania wyników). Dlatego w praktyce zaczyna się od próbki: ma ona dać szybki obraz typów awarii i pomóc ustalić sensowne K oraz próg τ.

Dobór próbki można prowadzić dwiema metodami:

Metoda 1 (próba celowa):

dobiera się rdzenie z różnych działów/tematów, krótkie i dłuższe, intuicyjnie „łatwe" i „trudne", oraz takie, które po profilu (s1…sK) wyglądają dobrze i źle. Dobrą praktyką jest wykonanie drugiej, niezależnej próby i sprawdzenie, czy pojawiają się te same typy problemów.

[13] Normalizacja (L2) to sprowadzenie każdego embeddingu do tej samej długości (najczęściej „1"), tak żeby porównanie podobieństwa zależało od kierunku wektora, a nie od tego, jak jest "duży". Ważne: trzeba normalizować w ten sam sposób zarówno embeddingi bazy, jak i embeddingi zapytań; inaczej wyniki będą przekłamane.

Metoda 2 (próba podejrzanych):

najpierw wykonuje się diagnozę geometryczną i wybiera rdzenie z najsłabszymi sygnałami (np. płaskie profile, bardzo niskie s1, podejrzanie wysoka bliskość do innych rdzeni), a dopiero potem sprawdza się je merytorycznie.

Diagnostyka geometryczna - podsumowanie

Diagnoza geometryczna — podsumowanie

Diagnoza geometryczna to zestaw obliczeń na wektorach rdzeni i wektorach chunków (miara: podobieństwo kosinusowe). Jej celem nie jest „rozstrzyganie merytoryczne", tylko wskazanie miejsc ryzyka oraz przygotowanie krótkich próbek do decyzji zespołu merytorycznego.

W praktyce diagnoza odpowiada na cztery pytania:

1. Kolizje rdzeni (rdzeń–rdzeń)
   Czy część rdzeni jest do siebie zbyt podobna (tj. ma wysokie podobieństwo do najbliższego sąsiada), przez co rdzenie mogą opisywać to samo lub silnie się nakładać?
2. Pokrycie korpusu przez rdzenie (chunk–najbliższy rdzeń)
   Czy duża część chunków ma słabe dopasowanie do jakiegokolwiek rdzenia (niska wartość podobieństwa do najbliższego rdzenia), co sugeruje „dziury pokrycia" albo brakujące tematy/nieadekwatne sformułowania rdzeni?
3. Obciążenie rdzeni (AssignedCount)
   Czy rozkład przypisań chunków do rdzeni jest w miarę równomierny, czy też występują:
   - rdzenie-magnesy (bardzo dużo przypisań),
   - rdzenie puste/nieaktywne (prawie brak przypisań).
4. Powtarzalność wyników top-K (sygnał „hubowatości" wyników)
   Czy w wynikach top-K wielu rdzeni często pojawiają się te same chunki (lub bardzo podobne chunki)? To jest sygnał diagnostyczny dotyczący *wyników wyszukiwania*, a nie automatyczna klasyfikacja „to jest boilerplate". O tym, czy to treść techniczna czy merytoryczna, decyduje dopiero szybki przegląd próbek.

Co ma przekazać matematyk zespołowi / autorowi merytorycznemu:

- lista kandydatów do scalenia / rozdzielenia rdzeni na podstawie kolizji (z wartościami podobieństwa i parametrami progów),
- krótka próbka chunków z „dziur pokrycia" (np. 50–100 przykładów z najniższymi wartościami dopasowania do rdzeni),
- lista rdzeni-magnesów i rdzeni pustych/nieaktywnych wraz z kilkoma przykładami top-K dla każdego (materiał do decyzji zespołu),
- lista chunków „często powracających w top-K wielu rdzeni" wyłącznie jako kandydaci do sprawdzenia, bez przesądzania przyczyny.
- Wniosek z diagnozy geometrycznej ma postać krótkiej listy problemów + próbek do oceny, które umożliwiają zespołowi merytorycznemu korektę rdzeni

(doprecyzowanie, rozbicie, scalenie) oraz — jeśli okaże się to zasadne po lekturze próbek — działania po stronie przygotowania korpusu.

wersja 2, 06.01.2026