

Modele językowe

Ćwiczenia 2

Zajęcia 5

Każde zadanie warte jest 1 punkt.

Zadanie 1. Przypomnij, co to jest perplexity? Rozważmy język, w którym słowa są ciągami cyfr, który:

- a) składa się z bloków k -cyfrowych (w bloku są te same cyfry, to znaczy blok ma postać d^n , przykładowo 888888888)
- b) po danym bloku może nastąpić blok z dowolną cyfrą (z tym samym prawdopodobieństwem)

Rozważamy bardzo duży tekst z tego języka. Oblicz dla tego tekstu perplexity dla następujących modeli: unigramowego, bigramowego oraz optymalnego n -gramowego (jakie jest n ?) Podaj wartości liczbowe dla $k = 10$.

Zadanie 2. Pewna firma zaproponowała ministerstwu pewnego państwa utworzenie narzędzia rozpoznającego teksty generowane przez model językowy za pomocą perplexity¹ (to znaczy, firma twierdziła, że generowane teksty mają większe perplexity niż teksty naturalne, stąd wartość perplexity powyżej jakiegoś progu sugeruje nienaturalne pochodzenie tekstu). Zastanów się, dlaczego miałyby to działać w ten sposób? Dlaczego to nie jest rozwiązanie idealne i jak użytkownik takiego modelu jak ChatGPT może kontrolować perplexity wygenerowanego tekstu.

Zadanie 3. W zadaniu tym będziemy myśleć o losowaniu tekstu za pomocą N -gramów (konkretnie 2- i 3-gramów), przy czym wynikowy tekst powinien spełniać jakieś dodatkowe wymagania. Wyjaśnij, dlaczego „metoda naturalna” (czyli normalne losowanie od lewej do prawej tekstu o określonej długości, sprawdzenie warunku, i ewentualne losowanie ponowne) nie sprawdzi się w takich zadaniach. Dla każdego wariantu zaproponuj jakiś algorytm, który daje istotnie większą szansę na losowanie zakończone sukcesem (niż metoda naturalna):

- a) losuj tekst o długości M , który na pozycji k ma określony wyraz,
- b) losuj tekst o długości M , który na pozycjach parzystych ma określone wyrazy (czyli losujesz tylko pozycje nieparzyste, zaczynamy numerację od 0),
- c) losuj niezbyt długi tekst o zadanym pierwszym i ostatnim słowie,
- d) dla listy napisów $[s_1, \dots, s_n]$ losuj tekst o długości n , taki że i -te słowo ma sufix s_i .

Zadanie 4. Wytrenowałeś model językowy (typu GPT). Ale okazało się, że w wyniku błędu wszystkie teksty były czytane od tyłu i model zamiast widzieć: ['Lubię', 'lody', 'malin', 'owe'] widział ['owe', 'malin', 'lody', 'Lubię']. Model się wytrenował, kosztowało to sporo (trening był przeprowadzony w płatnej chmurze). Gdy zauważyłeś błąd, to uruchomiłeś nowy trening (poprawnego modelu), teraz jedyny problem to uzasadnić wydatki na trening pierwszego modelu. Jakich argumentów używałeś, chcąc przekonać Zarząd, że wydatki te istotnie były uzasadnione.

Zadanie 5. Steganografia wg Wikipedii to:

nauka o komunikacji w taki sposób, by obecność komunikatu nie mogła zostać wykryta

Będziemy używać modelu językowego, o dużej jakości (tzn. takiego, który produkuje teksty w zasadzie nieodróżnialne od „naturalnych”), znanego obu stronom komunikacji.

Sytuacja jest następująca: jesteś w więzieniu (niesłusznie skazany), chcesz porozumiewać się z Towarzyszami na Wolności, szczęśliwie możecie wymienić drukowane listy w sposób nieskrempowany, lecz listy te czyta Naczelnik Więzienia – i nic nie powinno go poważnie zaniepokoić, bowiem mogłyby nastąpić Poważne Konsekwencje. W każdym liście chcesz przekazać komunikat (kilkadziesiąt, może trochę więcej bitów), tak by był on jednoznacznie odczytywalny, a jednocześnie całkowicie niewidoczny dla Naczelnika (z różnych względów odpadają jakiegokolwiek fizyczne modyfikacje nośnika, typu sok z cytryny itp). Szczęśliwie w Więziennej Pralni, w której pracujesz, jest

¹To nie jest zmyślona sytuacja.

komputer z modelem językowym i zainstalowanym Pythonem, a Ty masz do niego dobry dostęp i nikt z personelu nie jest nim zainteresowany.

Jak zorganizować niewidoczną dla Naczelnika komunikację między Tobą a Towarzyszami, przy założeniu, że wszystkie szczegóły mogliście omówić wcześniej, a ponadto Towarzysze mają dostęp do idealnie tego samego modelu, wersji Pythona, etc.

Zadanie 6. ★ Sprawdź, jeśli nie wiesz, co to jest Zasada Kerckhoffs (https://pl.wikipedia.org/wiki/Zasada_Kerckhoffs). Przedyskutuj kwestię, czy Twoje rozwiązanie z poprzedniego punktu ją spełnia. Jeżeli nie, to czy można je jakoś naprawić? (za klucz uznajemy model językowy)

Zadanie 7. Załóżmy, że Papuga jest zainstalowana na każdym komputerze, jako część systemu operacyjnego. Twoim zadaniem jest opracować metodę kompresji tekstów polskich wykorzystującą ten model językowy. Postaraj się, by opis był na tyle dokładny, żeby dało się go bez kłopotu zaimplementować (uwaga: zakładamy, że wynikiem kompresji może być ciąg bitów, nie musisz przejmować się tym, jak go upakować w bajty i jak reaguje on na zniekształcenia).

Zadanie 8. W modelu Papuga ważne jest, by prefiks który dajemy modelowi **nie kończył się spacją**. Zaobserwuj na kilku(nastu) przykładach, jak różnią się generacje dla tego samego prefiksu w wariantach ze spacją na końcu i bez spacji. Wyjaśnij, skąd bierze się ten fenomen.

Zadanie 9. Rozważmy wykorzystanie modelu językowego do generacji wierszyków, takich jak poniższy:

Biega, krzyczy pan Hilary:
„Gdzie są moje okulary”
Szuka w spodniach i w surducie,
W prawym bucie, w lewym bucie.

Czym taki tekst się charakteryzuje? Zaproponuj możliwie realistyczny algorytm, który wykorzystuje taki model jak Papuga do generacji wierszyków (oczywiście nie obędzie się bez przeszukiwania, chodzi zatem o to, jak uczynić ten proces możliwie efektywnym).

Zadanie 10. ★ Zaproponuj jakieś zadanie, w którym przewidywanie kolejnego tokenu może dać użyteczną aplikację (lub być w jakiś sposób ciekawe, czy zabawne). Zadanie nie powinno mieć związku z językiem naturalnym (a przynajmniej nie powinno być jedynie związane z językiem naturalnym). Opisz skąd bierzesz (bądź jak generujesz) korpus i jak wygląda tokenizacja.