Modele językowe
Ćwiczenia 5
ostatnie zajęcia przed sesją

Każde zadanie warte jest 1 punkt (chyba, że napisano inaczej).

Zadanie 1. Załóżmy, że dysponujemy dużym, n-gramowym modelem M_{ng} działającym na tokenach. Zaproponuj dwa scenariusze **treningu** modelu typu GPT, który wykorzystuje M_{ng} (model M_{ng} może być również później wykorzystywany w inferencji, czyli w generowaniu tekstu)

Zadanie 2. Znajdź informacje o wielkości modelu GPT-3. Przyjmijmy założenie, że cyfra=token, mnożenie jest realizowane na sekwencyjnym, jednowątkowym procesorze, a do mnożenia macierzy używamy szkolnego algorytmu $O(N^3)$. Jak wiele operacji mnożenia musi wykonać transformer GPT-3, żeby pomnożyć dwie liczby trzycyfrowe, większe od 400. Wynik nie musi być dokładny, możesz pomylić się o rząd, czy dwa.

Zadanie 3. Na jakich zbiorach danych trenuje się modele wymienione na stronie HuggingFace w grupie Table Question Answering. W zadaniu należy się posłużyć informacjami, jakie dają twórcy najbardziej popularnych modeli w danej klasie.

Zadanie 4. Na jakich zbiorach danych trenuje się modele wymienione na stronie HuggingFace w grupie Zero-Shot Classification.

Zadanie 5. Wybierz jakąś grę planszową, w której rzucane są kości (na przykład chińczyka czy tryktraka, możesz posiłkować się też listą `https://en.wikipedia.org/wiki/List_of_dice_games`) Jak napisać agenta, grającego w tę grę, używając autoregresywnego modelu językowego. Czy użycie kości wprowadza tu jakąś komplikację, a jeżeli tak, to jak ją najlepiej rozwiązać?

Zadanie 6. Wyobraźmy sobie, że tworzysz model autoregresywny, który gra w grę taką jak szachy. Dysponujesz dużym zbiorem rozgrywek toczonych przez dobrze grających ludzi, jak również kilkoma agentami $(A_1, \ldots, A_K)$, znajdującymi dobre ruchy znajdywane za pomocą klasycznych, deterministycznych algorytmów AI. Twoim celem jest jak najlepsze modelowanie ruchów (czyli klasyczny cel modelu językowego). Zaproponuj sposób tworzenia modelu, który w pewnych sytuacjach wykorzystuje agenty A_i. Twoje rozwiązanie powinno spełniać następujące warunki:

a) Powinno zacząć od wytrenowania modelu M_1 przewidującego ruchy niekorzystającego z żadnego A_i

b) Wzbogacać korpus o miejsca, w których korzystne jest wywołanie któregoś agenta

c) Trenować model M_2 na wzbogaconym korpusie

d) Podczas rozgrywki kontrolować ilość wywołań agentów (bo są one kosztowne).

Zadanie 7. Wróćmy do pisania „wierszyków", będących czterema wierszami po 8 sylab każdy, takich jak:

> W Pacanowie kozy kują,
> więc koziołek mądra głowa,
> błąka się po całym świecie,
> żeby dojść do Pacanowa.

Załóżmy, że mamy wytrenowany model języka polskiego typu BERT w którym tokenami są sylaby (niektóre występujące w dwóch wariantach: wariant na początku słowa i wariant w środku (dla uproszczenia pominiemy słowa zerosylabowe i interpunkcję), i dla naszego koziołka pierwsze linijka wygląda tak: `[^wpa, ca, no, wie, ^ko, zy, ^ku, ją]`. Taka struktura pomaga w modelowaniu języka na potrzeby wierszyków, bo umożliwia kontrolę rytmu, akcentów i (łatwiejszą) kontrolę rymów. Wyjaśnij, dlaczego.

Opisz wykorzystanie modelu BERT do generacji wierszyków. Procedura generacji powinna zakładać:

a) potencjalne wielokrotne zmiany każdej sylaby,

b) pilnowanie, że sylaby na pewnych pozycjach **muszą** rozpoczynać słowa, a na pewnych **nie mogą** go rozpoczynać,

c) pilnowanie albo przynajmniej promowanie rymów.

Zadanie 8. W kodzie do dodawania liczb, bazującym na MinGPT, w klasie AddDataset jest fragment kodu:

```
def __len__(self):
    return 10000 # ...
```

którego autor nie skomentował, choć nosił się z takim zamiarem. Popatrz jak używana jest ta klasa podczas treningu i zastanów się, jaki wpływ miałaby na proces treningu zamiana liczby `10000` na jakąś istotnie różną w tym konkretnie zadaniu.

Zadanie 9. (0.5-1.5p) Wracamy do znanego nam Niefrasobliwego Programisty, który (ponownie) wytrenował duży model językowy dla pewnego języka naturalnego. Problem jest taki, że źle podzielił ztokenizowany korpus na część uczącą i testową i pewna liczba istotnych tokenów do części treningowej nie trafiła. Chce to jakoś naprawić, ale nie wie jak. Poszedł więc do Bardziej Biegłej Koleżanki, która mu powiedziała:

> Jest kilka możliwości. Po pierwsze zostawić model taki jaki jest, a jedynie zmodyfikować tokenizator. Jak? To proste: (1). W sumie można by też dostrajać model na pominiętym podkorpusie, zamroziwszy wszystkie wagi oprócz pewnych wierszy jednej z macierzy. Której? No przecież (2). Widzę też sposób na wykorzystanie tu zwykłego word2vec-a, choć to trochę bardziej skomplikowane. Mianowicie (3)

Punktacja zależy od tego, jak wiele części wypowiedzi Koleżanki jesteś w stanie zrekonstruować.

Zadanie 10. ⋆ Powiedzmy, że chcesz stworzyć bazującego na modelu językowym czatbota z „osobowością" (czyli takiego, który w dłuższej konwersacji udaje spójną, ciekawą osobę). Zaprojektuj proces tworzenia takiego czatbota, przyjmując, że dysponujesz bardzo dużym budżetem.

Zadanie 11. ⋆ Jaką widzisz przyszłość dla modeli językowych i dużych modeli językowych.