Przetwarzanie języka naturalnego
Pracownia 3
Zajęcia 7 i 8

Można wydłużyć termin wybranego zadania. Grupa poniedziałkowa, ze względu na to, że lista pojawiła się dość późno, może wydłużyć termin dwóch zadań.

Zadanie 1. (5p) W zadaniu tym masz napisać system, który bierze na wejściu (ztokenizowany) tekst w języku polskim, pozbawiony wielkich liter oraz polskich znaków diakrytycznych i wypisuje na wyjściu poprawny tekst w języku polskim. Zakładamy, że literka „ź" na wejściu jest reprezentowana przez „z" (a nie „x"). Liczymy dwie miary dokładności:

a) Dokładność polskawa, czyli liczba słów poprawnie zrekonstruowanych (modulo wielkość liter, której nie uwzględniamy w tej mierze) podzielona przez liczbę słów w ogóle

b) Dokładność pełna, czyli liczba słów poprawnie zrekonstruowanych podzielona przez liczbę słów (tu uwzględniamy zarówno ogonki jak i wielkość liter).

Ostatecznym wynikiem będzie średnia geometryczna tych liczb. W tym zadaniu sprawdzany jest poziom `basic`, to znaczy że prezentowane rozwiązanie powinno:

- rekonstruować stokenizowany tekst,
- wykorzystywać dane dotyczące unigramów z części uczącej korpusu,
- w jakiś sposób (dowolny sensowny) uwzględniać informacje o dłuższych ciągach słów.

Zadanie 2. (3 + Xp) W tym zadaniu rozwiązać należy dokładnie ten sam problem, co w poprzednim zadaniu. Żeby zadanie było uznane za zrobione poprawnie, wynik Twojego programu (na zbiorze ewaluacyjnym), musi być wyższy niż K. Dodatkowo, jeżeli wynik R Twojego programu będzie większy niż Y, to za zadanie dostaniesz $4 \times \frac{R-Y}{1-Y}$[1]. Dodatkowa premia to 4 punkty za najlepszy program, 3 punkty za drugie miejsce, 2 punkty za trzecie i 1 punkt za czwarte (liczone w obu grupach). Dozwolone jest korzystanie z korpusu PolEval, N-gramów NKJP oraz Morfologika. Zbiór testowy zostanie podany w przyszłym tygodniu, będzie on zawierał między innymi podzbiór części testowej PolEvala.

Zadanie 3. (4p) W zadaniu tym zajmiemy się omawianym na wykładzie ukrytym łańcuchem Markowa, na przykładzie nieuczciwego krupiera rzucającego kością. Przypominam zasady:

1. Krupier ma dwie kości, uczciwą i oszukaną.
2. Kość oszukana daje 6 oczek z $p = \frac{1}{2}$, a pozostałe wyniki z $p = \frac{1}{10}$
3. Krupier zmienia kość uczciwą na nieuczciwą z $p_1 = 0.04$, a nieuczciwą na uczciwą z $p_2 = 0.05$
4. Zaczynamy od uczciwej kości.

Napisz program, który dla danego ciągu rzutów (który musisz sam wygenerować) wypisuje ciąg stanów (`u` – kość uczciwa, `n` – kość nieuczciwa, długość rzędu 10000), w sposób maksymalizujący liczbę prawidłowo zgadniętych stanów. Rozwiąż to zadanie na dwa sposoby:

- Proponując heurystyczny algorytm decydujący na podstawie *badania skupisk szóstek*
- Implementując poprawny algorytm, bazujący na zmiennych α oraz β (zobacz wykład o HMM).

Wykonując eksperymenty, oszacuj poprawność działania obu algorytmów, mierzoną liczbą poprawnie zgadniętych stanów (podzieloną przez długość ciągu).

[1] Wartości K i R zostaną podane wkrótce.

Zadanie 4. (4p) W tym zadaniu powinieneś zrekonstruować „parametry" krupiera. Mamy dwie sześcienne kości o nieznanych rozkładach (każdy rozkład to 6 liczb dodatnich, sumujących się do jedynki), zaczynamy od losowo wybranej kości (z prawdopodobieństem π oraz $1-\pi$). Podobnie jak w poprzednim zadaniu p_1 i p_2 to prawdopodobieństwa zmiany kości. Na SKOSIe znajdziesz zestaw 20000 obserwacji (wyników rzutów kością), poczynionych dla tego modelu (ale do testów możesz też używać danych wygenerowanych w poprzednim zadaniu). Masz zrekonstruować model, uruchom Twój program dla kilku prefiksów dostępnych danych i porównaj wyniki.

Zastanów się, jak zainicjować model. Czy rozpoczynanie od równych prawdopodobieństw to dobry pomysł?