Przetwarzanie języka naturalnego
Pracownia 4c

Wszystkie zadania są ważne do końca semestru, zadania oznaczone literką **E** można oddawać również podczas dodatkowego terminu przed egzaminem.

Zadania w większości są z gwiazdką (która nie jest zaznaczana, niegwiazdkowych zadań jest za 10 punktów). Część zadań dotyczy materiału, który pojawi się na kolejnych wykładach (co będzie wyraźnie opisane). Na tej liście znajdą się również punkty za zadania z rekonstrukcją (permutacje, ogonki i małe/wielkie litery) oraz nowe dane do zadania ze zgadywaniem autorstwa tekstu (wraz z informacją o punktacji).

Osoba, która rozwiąże zadania z literką **E** za co najmniej 10 punktów może uzyskać zwolnienie z egzaminu i przepisanie oceny z ćwiczeń, jeżeli ta ocena jest co najmniej 4. Jeśli chodzi o zaliczanie ćwiczeń, to zadania z E działają jak normalne zadania.

Uwaga: Na liście pojawiając się pojęcia z wykładu 12 (PCFG) i 13 (Gramatyki zależnościowe)

Zadanie 1. (10p, E) Stwórz możliwie prostą, a jednocześnie nietrywialną gramatykę generującą zdania w języku polskim (zdania powinny być poprawne gramatycznie, możliwie naturalne, możesz korzystać z typów słów ze Składnicy, na SKOSie pojawi się też lista czasowników tranzytywnych[1]). Możesz stosować pomysły i kod z innych zadań. Do generowania słów możesz wykorzystać plik polomorfologik.txt.

Do nadawania sensu generowanym zdaniom wykorzystaj następujący mechanizm:

1. Wejściem do generatora powina być liczba N (liczba wyrazów w zdaniu, traktowana jako „wskazówka odnośnie wielkości zdania") oraz zbiór słów, stanowiących: *osnowę historii.* Przykładowe osnowy to: malina-koszyk-zazdrość-morderstwo, programowanie-błąd-zmienna-deklaracja, lotniskowiec-łódź-podwodny-tonąć-atak-torpeda-ocean.

2. Pierwszym etapem generowania zdania jest generowanie zdania poprawnego gramatycznie (bez związku z osnową)

3. Następnie dla każdego wyrazu, który jest rzeczownikiem, przymiotnikiem, imiesłowem, czasownikiem, przysłówkiem generujemy kilkaset jego wariantów i wybieramy taki, który pasuje najlepiej do **jakiegoś** słowa z osnowy. Wykorzystujemy tu zarówno zanurzenia słów i/lub form bazowych.

4. Proces powtarzamy wiele razy, wybierając zdanie, które wydaje się najlepsze (na przykład wysokie jest wzajemne PMI poszczególnych par słów oraz czy wszystkie słowa z osnowy zmieściły się w zdaniu)

Planem minimum jest zapewnienie, żeby było łatwo zgadnąć, którą z trzech przykładowych osnów realizuje wygenerowane zdanie. Możesz to przetestować na sobie, generując na przykład 30 zdań (dla każdego losując wcześniej jedną z trzech osnów, zapisując losowanie do pliku i sprawdzając, czy jesteś w stanie odtworzyć ten plik patrząc tylko na wyniki losowań)

Zadanie 2. W tym zadaniu będziemy rozważać probabilistyczne gramatyki bezkontekstowe (PCFG) utworzone ze Składnicy. Została przygotowana wersja składnicy w formacie akceptowanym przez NLTK, zawierającym informacje o parametrze *head* dla każdej frazy. Programik `skladnica_demo.py` może stanowić pomoc w przetwarzaniu takich drzew. Analizując ten bank drzew utwórz gamatykę PCFG, którą następnie wykorzystamy do generowania zdań. Rozważamy następujące warianty:

a) Wariant w pełni zleksykalizowany: symbolem nieterminalnym jest symbol nieterminalny z wszystkimi parametrami oraz z parametrem head, przykładowo: `ff:poj:2:[np[bier]]|wypijesz`. Zauważ, że powtarzanie pewnych produkcji tej gramatyki powoduje tworzenie nienaturalnych zdań, takich jak:

> który w szczecinie załatwia ważne problemy , istotne dla sądu dla wsi dla użytkowników

[1] Czyli takich, które „obsługują" dopełnienie w bierniku i dopełniaczu w wersji zanegowanej (czyła książki, nie czytał gazet)

człowiek staje się wyleniałym wyleniałym wyleniałym wyleniałym wyleniałym tygrysem
proponuję przyjąć decyzję o wyrwanej wyrwanej kartce z kolei z historii

Postaraj się jakoś temu zapobiec.

b) **(3p)** W tym wariancie powinieneś zrezygnować z leksykalizacji (czyli symbolem nieterminalnym będzie, przykładowo, `ff:poj:2:[np[bier]]`. Dodatkowym parametrem powinna być liczba N, mówiąca jak długie (w przybliżeniu) powinno być zdanie. Dodatkowo powinieneś premiować zdania „rozłożyste", czyli takie, w których drzewo rozbioru jest niezbyt głębokie. Oczywiście najprościej zrealizować to losując wielokrotnie i wybierając takie zdanie, które najlepiej odpowiada kryteriom.

c) **(3p)** Sporządź listę typów używanych przez Składnicę (czyli przypisanie słowom *napisów w nawiasach kwadratowych*, takich jak `[np[bier]]` dla słowa *wypijesz*). Zmodyfikuj losowanie z poprzednich dwóch punktów w ten sposób, by w miejscu słowa o określonym typie mogło pojawić się inne słowo o tym samym typie. Dodatkowo w ostatecznym zdaniu słowa, dla których użyto typu pustego (`[]`) powinny być oznaczone gwiazdką.

d) **(3p, E)** Wzbogać losowanie zdań o zamianę słów z gwiazdką z poprzedniego podpunktu na słowa ze słownika o takim samym tagu (użyj pliku supertags.txt). W losowaniu powinieneś premiować słowa „pasujące" (pasowanie możesz zdefiniować dowolnie, za pomocą bigramów, PMI, zanurzeń słów, etc).

Zadanie 3. (**7+X, Ep**) Wykorzystaj wersję zależnościową Składnicy ze strony Universal Deppendencies, aby generować zdania. Twój mechanizm powinien:

a) Umożliwiać podstawianie za liście słów o podobnych gramatycznie.

b) Uwzględniać typy wyrazów (czyli to, jakie tagi mają ich dzieci, kolejność dzieci i ich położenie względem rodzica)

c) Generować zróżnicowane zdania

d) Generować poprawne gramatycznie zdania.

e) Uwzględniać fakt, że niektóre typy są „uniwersalne", czyli możliwe jest ich stosowanie do dowolnych wyrazów o określonym tagu.

Szczegóły pozostawione są do dopracowania przez Studenta. Wartość X jest nieujemna, zależy od złożoności rozwiązania. Domyślnie jest równa 0.

Zadanie 4. (**10, Ep**) Generuj poezję za pomocą gramatyk PCFG. Poezja może wyglądać jak Pan Tadeusz, albo jak piosenka Tanie Dranie. Konieczne jest zadbanie o poprawność rytmiczną i rymy oraz o poprawność gramatyczną. Należy również w jakiś sposób premiować zdania sensowne treściowo.

Zadanie 5. (**10, Ep**) Zapowiedź: generuj poezję za pomocą gramatyk zależnościowych. Poezja może wyglądać jak Pan Tadeusz, albo jak piosenka Tanie Dranie. Konieczne jest zadbanie o poprawność rytmiczną i rymy oraz o poprawność gramatyczną. Należy również w jakiś sposób premiować zdania sensowne treściowo.

Zadanie 6. (**15+, Ep**) Zapowiedź: generuj poezję korzystając z rozszerzonego banku wzorców zdań z PolEvala (utworzenie jego jest częścią zadania). Rozszerzenie polega na tym, że zdanie wejściowe nie musi być poprawne rytmicznie, ale powinno być możliwe jego „urytmicznienie" (to znaczy zamiana pewnych słów na inne, o innej liczbie sylab, przy częściowym zachowaniu sensu zdania).

Dodatkową daną dla generatora powinna być *osnowa* wiersza(zobacz zadanie 6). Uwaga: to może być dość trudne zadanie!