Program de Curățare Lexicală

~ Natural Language Processing ~

Această lucrare are ca scop dezvoltarea unui program codat în limbajul de programare Python de detectare și modificare a limbajului vulgar într-un text dat, folosind mai multe metode de Procesarea Limbajului Natural. Obiectivul final este de a avea posibilitatea de a insera un fișier de intrare .txt în program și a obține un fișier de ieșire care să cenzureze parțial toate cuvintele cu conotație obscenă sau licențioasă, într-un mod în care să se păstreze intenția scriitorului acelui text.

Treptat, se vor aborda toate posibilele probleme principale în procesul dezvoltării unui astfel de program, pentru a înțelege mai bine intenția din spatele deciziilor luate.

O primă întrebare, și posibil cea mai ușor de rezolvat, este „Cum putem detecta cuvintele vulgare?”. Pentru această problemă am ales variantei unei liste de „cuvinte-cheie”, anume o selecție din cele mai folosite cuvinte vulgare în limba engleză. Într-o situație reală, această listă poate conține o multitudine de expresii și cuvinte considerare indecente.

Odată ce aceste cuvinte sunt detectate, se ridică problema metodei de cenzurare. Așa cum am menționat în paragraful introductiv, am ales să recurg la o cenzurare parțială a cuvintelor detectate, întrucât această variantă evidențiază vulgaritatea acestora dar păstrează posibilitatea înțelegerii contextuale a folosirii acelui cuvânt. Modalitatea aleasă implică un sistem relativ simplu, dar folositor:

- Se va ține minte numărul de „*” care trebuie inserate în variabila *numar_cenzuri.*
- Pentru primele 4 litere ale unui cuvânt (frecvența cea mai mare de cuvinte vulgare folosite în limba engleză este de cuvinte cu patru litere) se va acorda o unitate.
- Dacă un punct depășește limita de 4 litere, pentru fiecare 3 litere adiționale se va mai adăuga un punct.
- Odată calculat numărul de cenzuri, se va porni de la a doua literă a cuvântului și se va insera un asterisc în locul fiecărei litere, până la epuizarea numărului alocat.

Un bun exemplu ar fi o conversație dintre doi oameni, unde se folosește cuvântul „shit”. O cenzurare totală ar putea duce la confuzia cititorului, prin asocierea înlocuitorului „****” cu alte cuvinte. Modalitatea aleasă generează sintagma „s*it”, care cenzurează cuvântul păstrând înțelesul acestuia.

Mai departe, se ridică problema integrării în formatul original prezent în text a unei versiuni modificate a cuvintelor. Este relevant ca semnele de punctuație și alte simboluri prezente în text, precum și capitalizarea cuvântului, să rămână neschimbată, pentru a păstra înțelesul în contextul dat. Pentru a rezolva această situație, odată ce un cuvânt este detectat ca fiind vulgar, se vor face toate calculele pe variante „stripped” (fără capitalizare și delimitatori) a cuvântului, urmând ca aceasta să fie adăugată în secvența inițială prin metoda *.replace( )* din Python. Astfel, se asigură cenzurarea fără modificarea formatării originale.

În continuare, se ridică problema posibilității scriitorului de a ignora această metodă de interzicere prin înlocuirea unei litere cu alta, fapt care previne funcționalitatea detectării unui cuvânt cenzurabil. Această dilemă va fi rezolvată prin introducerea funcției *test_similitudine()*, care testează toate cuvintele unui text contra cuvintelor vulgare din lista de cuvinte-cheie care au același număr de litere. Se va itera prin pozițiile corespunzătoare fiecărei litere din cuvânt, iar dacă se găsește o egalitate între cele două, se va incrementa variabila *count_litere_similare*. Dacă la sfârșitul parsării, această variabilă conține o valoare cu 1 mai mică decât numărul de litere, cuvântul va fi semnalat ca fiind o încercare de ocolire a sistemului, rezultând într-o cenzura a acestuia.
Spre exemplu, cuvântul „fvck” ar putea fi folosit pentru a imite cuvântul „fuck”, într-un context dat. Algoritmul va detecta că 3 din cele 4 litere ale cuvântului sunt pe aceeași poziție și au aceeași valoare, așa că va fi semnalat ca fiind cenzurabil.

Însă, acest sistem are anumite sincope care trebuie abordate. Spre exemplu, ce se întâmplă dacă în text se folosește cuvântul „duck”? După regulile date, acesta ar trebui semnalat ca fiind vulgar, deci cenzurat. Pentru acest caz se va folosit API-ul WordsAPI pentru a consulta dicționarul explicativ al limbii engleze. Se va face un HTTP GET request către API, iar obținerea unui rezultat afirmativ va confirma existența cuvântului considerat „similar” în dicționar. Dacă acesta este într-adevăr un cuvânt real, se va semnala acest lucru în consolă, iar șirul de caractere va fi considerat acceptabil.

Un ultim caz abordat în analiză este acela în care se întâmplă ca unele cuvinte vulgare, prin schimbarea uneia dintre litere, să formeze cuvinte reale dar foarte rar utilizate. Un exemplu adesea folosit este „dyck”, care în teorie este un cuvânt real din dicționarul limbii engleze, dar de cele mai multe ori este folosit cu o conotație vulgară prin modificarea literei „i” în „y”. Pentru a include această situație, se va modifica apelul făcut către API pentru a include frecvența de apariție a unui cuvânt dat. În cazul în care cuvântul este unul simplu, nu format din două cuvinte alăturate (de exemplu „sandcastle”), dacă frecvența acestuia este mai mică de 1 per milion de cuvinte parsate într-o scriere în limba engleză, atunci va fi semnalat ca fiind cuvânt „rar”. Pentru acest exemplu didactic, cuvintele rare sunt cenzurate în același mod ca cele vulgare.

În concluzie, procesul de curățare al unei lucrări scrise este unul laborios, care implică mult cazuri adiționale și multe variabile de considerat, însă este un proces interesant de discutat pe subiectul înțelegerii modului în care un om își exprimă frustrarea, supărarea sau alte sentimente și trăiri asociate unui astfel de limbaj neacademic.