Ferran Cantariño i Iglesias - 173705
Marc Rabat Pla - 172808

INFORME NLP - Pràctica 1

● Procés d'etiquetat implementat:

Per a dur a terme l'etiquetatge de les paraules el que s'ha fet és primerament llegir el fitxer amb el corpus de paraules per tal de generar el model necessari a partir del qual és possible taggejar les nostres paraules no classificades en els respectius fitxers de test.
Fet això, es genera una estructura de dades que contingui aquest model i després de llegir el fitxer de test s'itera fins arribar a la dimensió de línies d'aquest. Inutïtivament comparem la clau de l'estructura de dades amb la paraula en qüestió de cada línia i se li assigna el tag amb més freqüència si és que la paraula es troba en el model. Altrament s'assigna un dels valors per defecte.
Sempre que hem pogut hem utilitzat estructures amb hashing ja que així l'accés als valors és molt més eficient. En tots els passos hem hagut de fer el decoding i el posterior encoding per tal de garantir la integritat de les dades.
L'estructuració del codi l'hem fet en dos fitxers: tagger.py (que conté les crides principals a les funcions necessàries per a resoldre el problema) i nlp_methods.py (que són tots els mètodes que han estat necessaris per a trobar la solució del problema).

○ Resultats:

En executar el programa els resultats són els següents:

1. **Test:** Com podem veure, pel primer test obtenim un 96.74% de coincidències amb el fitxer de gold_standard, amb un temps de computació aproximat de dos minuts.

Enter the # of test you want to prove:
1

Generating model from training_set...

Loading model from training_set...

Tagging test_1.txt ...

Computing results of results_1.txt vs. gold_standard_1.txt ...

ACCURACY: 96.743697479 %
--- 156.744212151 seconds ---

2. **Test:** En el segon test obtenim un 89.89% de coincidències amb el fitxer de gold_standard, amb un temps de computació aproximat de cinc minuts.

Enter the # of test you want to prove:
2

Generating model from training_set...

Loading model from training_set...

Tagging test_2.txt ...

Computing results of results_2.txt vs. gold_standard_2.txt ...

ACCURACY: 89.8910411622 %
--- 288.167346001 seconds ---

Com es pot observar, els resultats són relativament bons i els temps d'execució són els esperats donada la dimensió dels fitxers d'entrada.

● Dificultats amb el fitxer test_2:

El fet diferencial respecte al fitxer de test_1, és que algunes de les paraules per analitzar i taggejar en el test_2 no es troben disponibles en el model generat a partir del corpus. Això és degut al fet que el training set no és prou gran com per cobrir aquest fet. Altrament, resulta evident que és complicat tenir un corpus d'entrenament el suficientment ampli com per a taggejar qualsevol paraula correctament, amb la qual cosa s'han de trobar nous mètodes per a millorar l'eficiència en cas que aquestes paraules no existeixin en el model, com comentarem a continuació.

○ Solucions possibles:

D'entre les solucions possibles, nosaltres hem optat per fer un recompte d'aquelles categories gramaticals que més apareixien en el model i guardant-les en una llista. En el nostre cas, hem escollit les 5 categories gramaticals amb més freqüència, de forma arbitrària. És clar que el valor del nombre de categories gramaticals retornades es podria adaptar a qualsevol problema amb la finalitat d'obtenir el millor rendiment possible. En el moment que una paraula no es troba en el model, el que fem és assignar-li un tag de forma aleatòria i d'estar de sort amb les probabilitats, podrem obtenir un molt bon rendiment. Una altra opció hauria estat establir directament una categoria com a default i retornar-la (no hem escollit fer-ho així, tot i que possiblement podríem obtenir un lleuger augment en la precisió ja que no ens ha semblat una implementació gaire innovadora). En la última idea que hem tingut, creiem que també ho podríem haver resolt retornant el tag de la paraula amb menys distància en comparació amb la paraula a etiquetar.