Dataset netejat porterstemmer

A green and blue rectangular bars

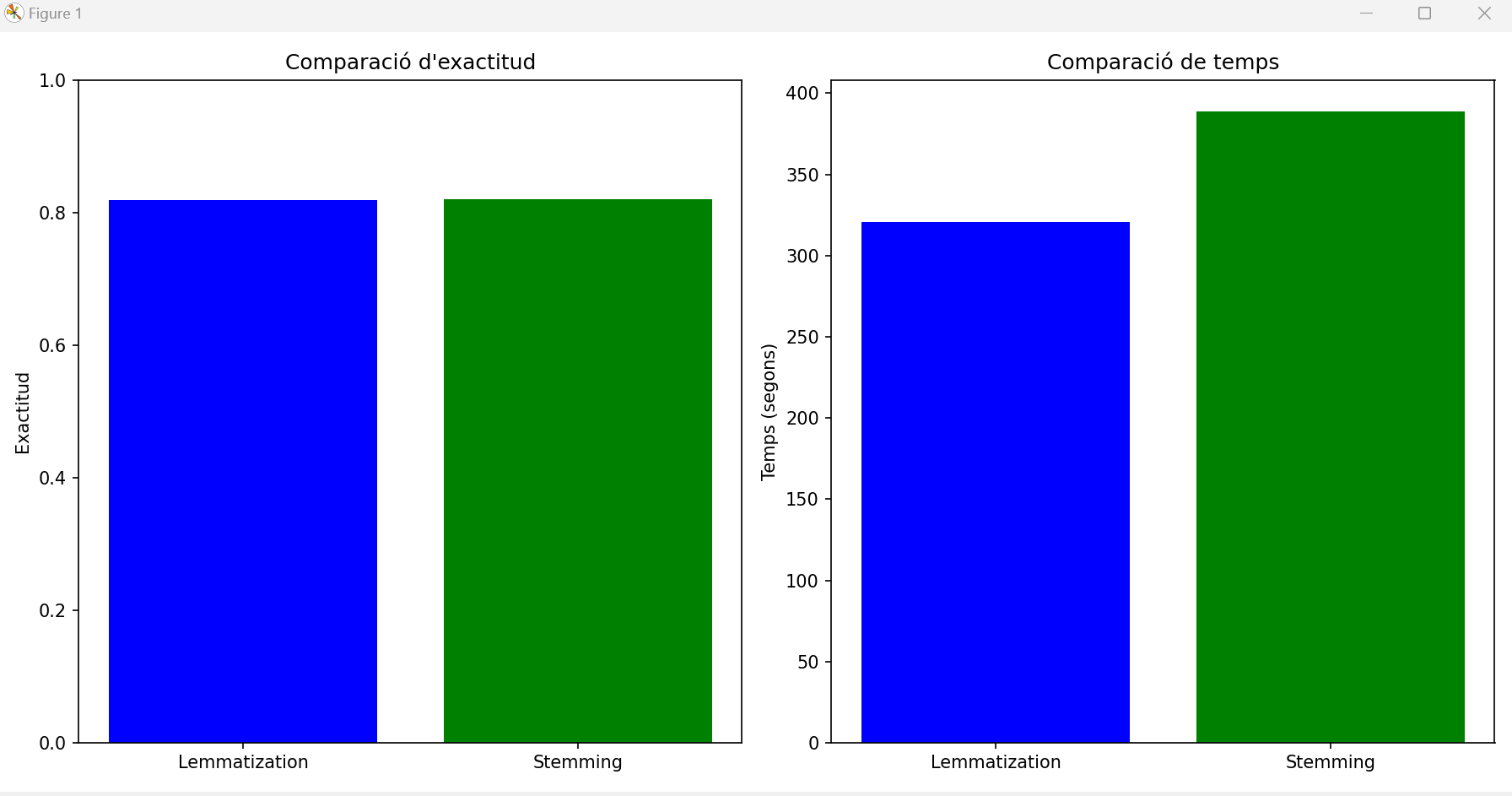
Description automatically generated

Exactitud amb lematització: 81.77%

Exactitud amb stemming: 81.76%

Temps amb lematització: 279.78 segons

Temps amb stemming: 420.30 segons

Dataset OG porterstemmer  
  
Exactitud amb lematització: 81.92%

Exactitud amb stemming: 82.00%

Temps amb lematització: 320.39 segons

Temps amb stemming: 388.65 segons

Comparació diversos Stemmings i Lema data set net:  
A green and blue bars

Description automatically generated

Resultats:

WordNetLemmatizer: Accuracy = 81.77%, Time = 221.53 seconds

PorterStemmer: Accuracy = 81.76%, Time = 422.64 seconds

SnowballStemmer: Accuracy = 81.95%, Time = 314.27 seconds

RegexpStemmer: Accuracy = 81.80%, Time = 160.26 seconds

Comparacio dataset OG:

A comparison of blue and green bars

Description automatically generated  
Resultats:

WordNetLemmatizer: Accuracy = 81.92%, Time = 268.46 seconds

PorterStemmer: Accuracy = 82.00%, Time = 419.95 seconds

SnowballStemmer: Accuracy = 81.83%, Time = 319.75 seconds

RegexpStemmer: Accuracy = 81.71%, Time = 164.10 seconds

Conclusions extretes:

El PorterStemmer utilitza un conjunt de regles iteratives per truncar sufixos i identificar l’arrel d’una paraula, cosa que pot implicar més càlculs, especialment en textos amb paraules complexes o sufixos llargs. Això fa que sigui significativament més lent que altres alternatives, com el RegexpStemmer, que és consistentment el més ràpid, tant amb el dataset original com amb el dataset netejat, gràcies al seu algorisme simple basat en expressions regulars senzilles. El RegexpStemmer només elimina sufixos específics definits per l'usuari, cosa que el fa computacionalment més lleuger i eficient. Tot i això, pot truncar paraules de manera massa agressiva, la qual cosa pot implicar la pèrdua de certa informació rellevant, sobretot en textos amb riquesa semàntica. Malgrat aquest risc, l’impacte d’aquesta pèrdua sembla ser mínim en aquest experiment, ja que les exactituds obtingudes amb el RegexpStemmer són molt similars a les d’altres tècniques més robustes.

En comparació, el WordNetLemmatizer consulta una base de dades semàntica preoptimitzada (WordNet) per identificar les formes base de les paraules. Això podria fer pensar que seria més lent, però en el teu codi, el lemmatizer mostra un millor rendiment que el PorterStemmer gràcies a l’eficiència de WordNet en gestionar paraules senzilles, com les dels datasets utilitzats. A més, la lematització conserva el significat semàntic de les paraules, cosa que fa que sigui més adequat en problemes on la semàntica és crucial. El SnowballStemmer, d’altra banda, ofereix un punt intermedi, amb regles més optimitzades que el PorterStemmer i una millor precisió en contextos variats.

Pel que fa als datasets, treballar amb el dataset netejat, on el text és més homogeni i conté menys soroll (sense enllaços, mencions ni símbols), redueix el temps de preprocessament i la variabilitat en les paraules processades. Això afavoreix tècniques senzilles i ràpides, com el RegexpStemmer o fins i tot el PorterStemmer. En canvi, amb el dataset original, que conté textos més complexos i amb més variabilitat, tècniques com el WordNetLemmatizer o el SnowballStemmer poden gestionar millor les variacions semàntiques i morfològiques, tot i requerir més temps de càlcul.

Conclusió Final

El RegexpStemmer és la millor opció en termes d'eficiència computacional i velocitat, gràcies a la seva simplicitat i rendiment consistent en ambdós datasets. Amb el dataset netejat, la seva simplicitat és ideal, ja que no hi ha gaire informació addicional que es pugui perdre en el truncament. Tanmateix, en el dataset original, l'ús del RegexpStemmer pot ser arriscat en problemes on la preservació semàntica és crítica. En aquests casos, tècniques com el WordNetLemmatizer o el SnowballStemmer, tot i ser més lentes, ofereixen un balanç entre rendiment i preservació de la informació semàntica.

En resum, si l’objectiu és maximitzar la velocitat mantenint una precisió competitiva, el RegexpStemmer és la millor elecció. No obstant, per a problemes que requereixin una gestió semàntica més rica, el WordNetLemmatizer o el SnowballStemmer són opcions més adequades, encara que a costa d’un temps d’execució més elevat, tot i que com mostra a les gràfiques la diferènicia d’exactitud no val la pena.