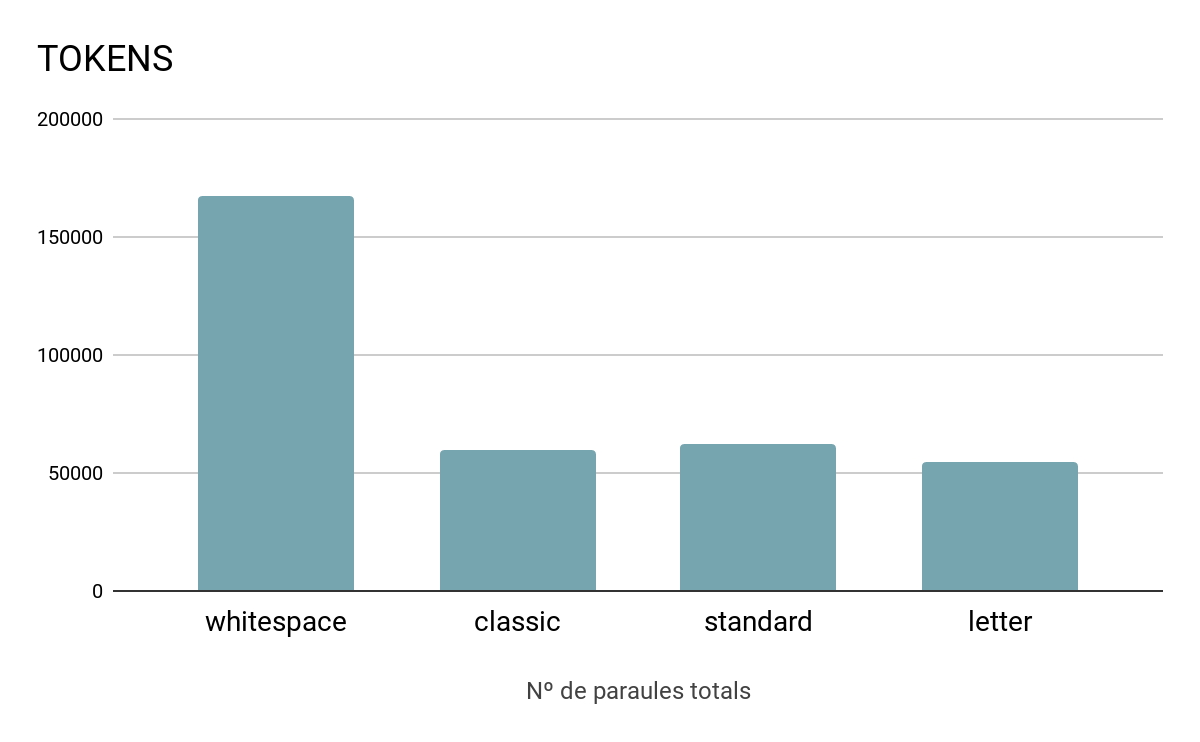
# Pràctica 2: Programming with ElasticSearch

### Primera part (TOKENS i FILTRES)

En aquesta primera part de la pràctica hem treballat els tokens i filtres. Hem començat estudiant els diferents tipus de tokens que es poden aplicar. (Totes les dades són aplicades sobre els documents de novel·les)

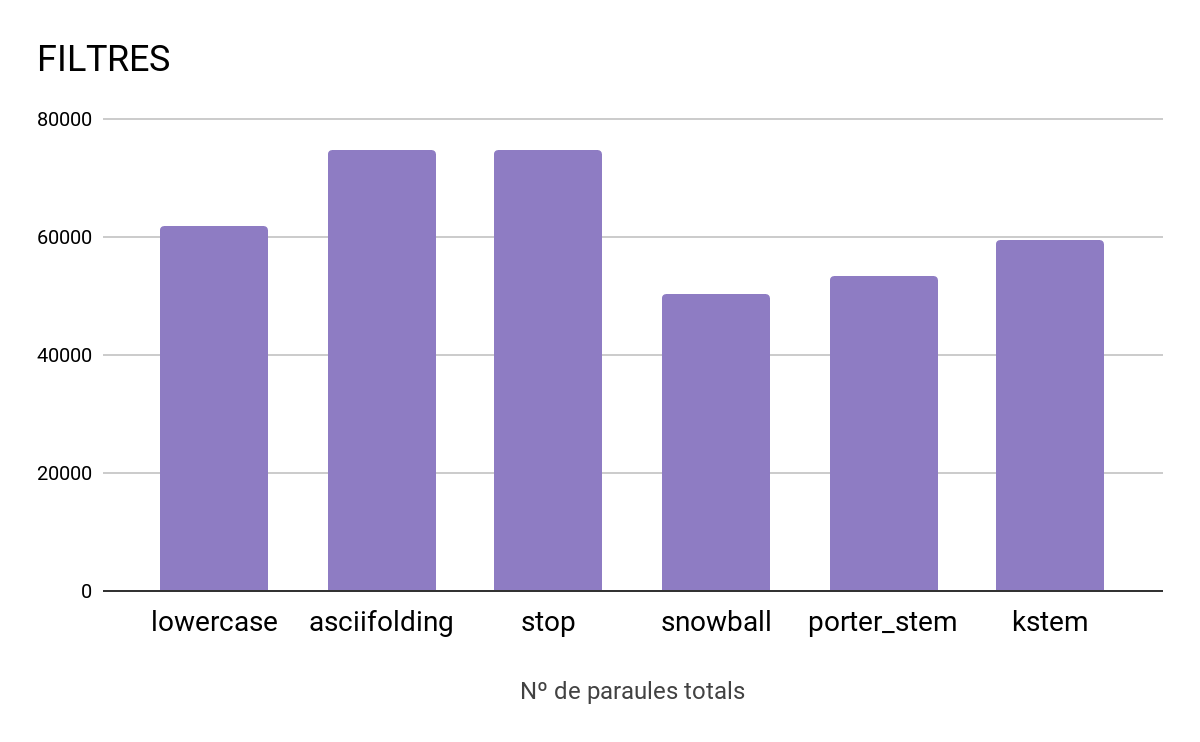
Breu explicació dels diferents tokenizers:

* **whitespace:** un tokenizador d’espai talla el text quan es troba un espai.
* **classic:** el tokenizador clàssic està basat en la gramàtica natural del llenguatge.
* **standard:** el tokenizador estàndard divideix el text en termes com es defineix la segmentació Unicode. Elimina la majoria de símbols de puntuació.
* **letter:** El tokenizador de lletres divideix el text en termes quan es troba un caràcter que no és una lletra.



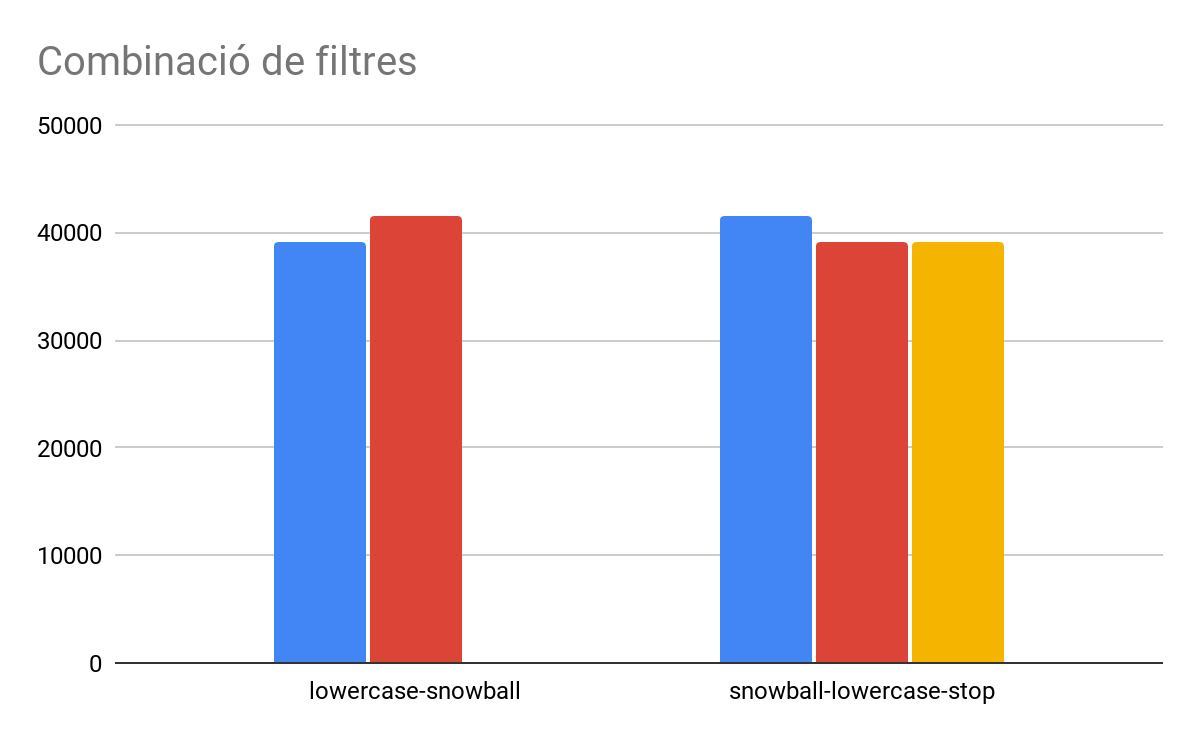
A partir d’aquestes dades es pot concloure que el tokenizador més estricte es el de letter.

Ara anem a estudiar el comportament dels diferents filtres. Com que en aquest cas s’explica com funcionen els filtres a l’enunciat de la pràctica no els descriurem aquí. Els resultats obtinguts són:



Per tant es pot concloure que el filtre snowball que treballa la derivació de les paraules és el que més redueix.

Un cop hem vist els diferents tokens i filtres per separat i hem vist quins són els més agressius anem a comprovar algunes combinacions de filtres i tokens.



Els diferents colors del gràfic mostren ordres diferents de la implementació dels filtres. Com es pot veure s’obtenen diferents resultats. Per exemple, el primer cas: en el cual l’ordre canvia de 39067 a 41574. És obvi però, que quan s'aplica el filtre de lowercase abans que snowball resultaran menys paraules que a l’inrevés ja que primerament convertim les paraules en majúscula a minúscula i després s’aplicarà snowball aquest es carregarà alguns noms propis o acrònims que es considerin eliminables per aquest.

### Segona part (EXPERIMENTACIÓ TFIDF i CONCLUSIONS)

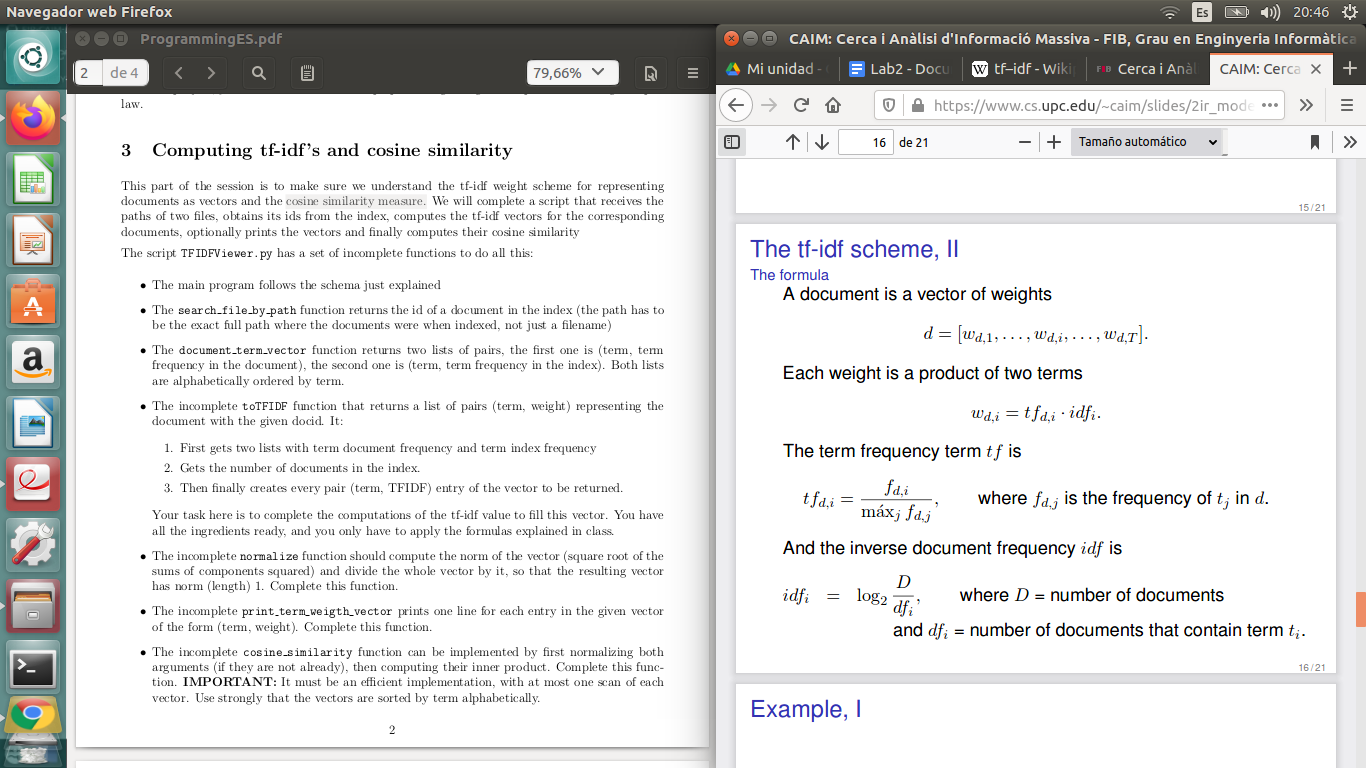
Per aquesta segona part de la pràctica hem modificat el fitxer *TFIDFViewer.py* per tal d’implementar el càlcul del pes tf-idf i la similitud entre diferents documents. Per això hem modificat diversos mètodes del codi, que s’expliquen a continuació:

**Funció *toTFIDF***

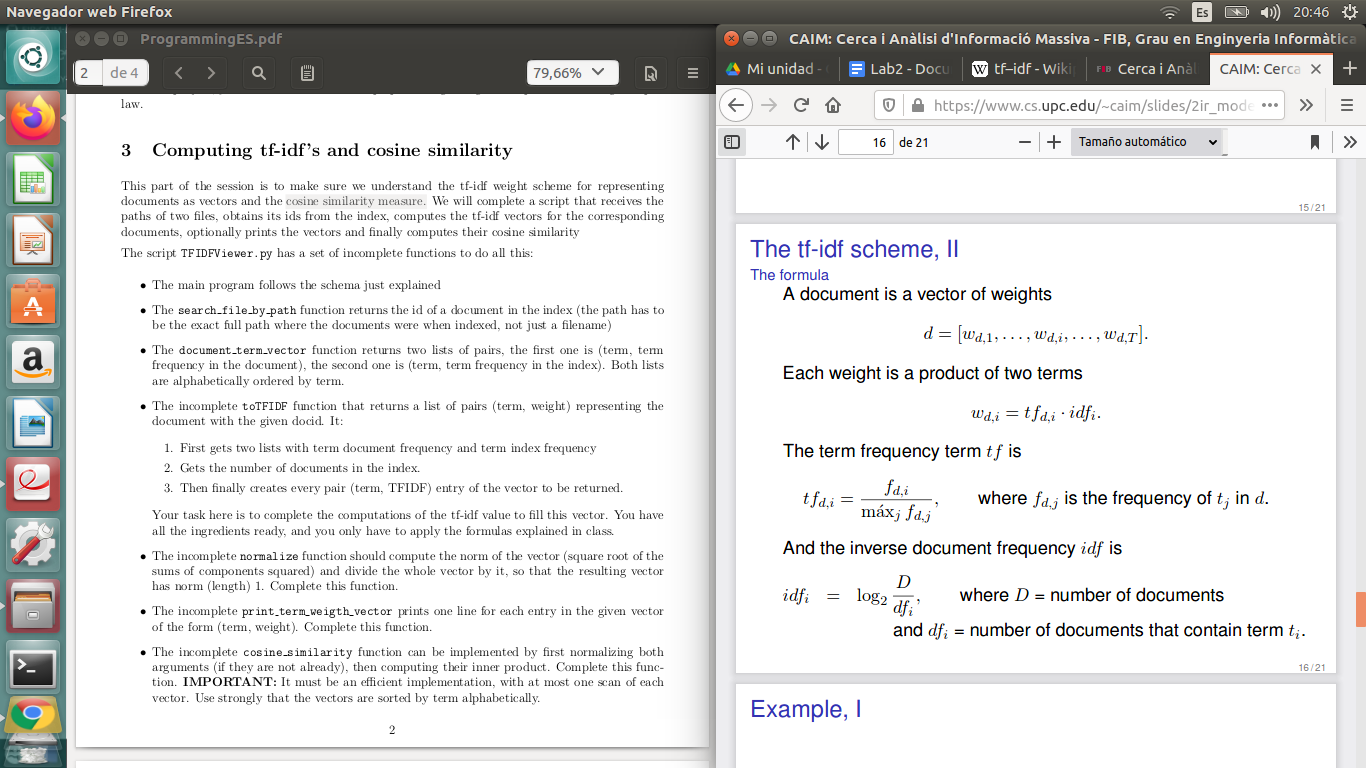
Aquesta funció s’encarrega de aplicar la formula TF-IDF i retornar el vector resultant. És a dir el vector de pesos.

Per a aconseguir aquest vector de pesos es cal multiplicar els termes***tf \* idf***:

* El terme de freqüència ***tf*** :



* La inversa de la freqüència del document ***idf*** :



**Funció *print\_term\_weight\_vector***

Aquesta funció només s’encarrega d’imprimir cada terme i pes dels parells existents en el vector d’entrada.

**Funció *normalize***

Aquesta tercera funció rep un array de parells <paraula, pes> y retorna un altre array amb els pesos normalitzats. Això s'aconsegueix dividint cada pes per l'arrel quadrada de la suma de cada pes al quadrat.

**Funció *cosine\_similarity***

La última funció,calcula la similitud de dos vectors de pesos (<paraula, pes>). Aquests vectors es reben com a paràmetre i el càlcul consisteix en anar comparant cada terme dels vectors i, en cas que siguin iguals sumar a l'atribut suma la multiplicació dels pesos en ambdos vectors. Els índexs es van avançant de manera que s'avança l'índex de la paraula més petita.

Experiments realitzats i conclusions:

Un cop completat el programa del càlcul de similitud entre dos fitxers, hem realitzat diversos experiments amb arxius diferents per tal d'observar el comportament de la similitud entre ellsi extreure conclusions al respecte.

Primerament, hem provat el cas més senzill comparant un fitxer amb si mateix. La nostra hipòtesis es que la similitud havia de ser màxima, és a dir, d'1.00 i hem comprovat que així era. Hem utilitzat el fitxer DickensAChristmasCarol.txt

A continuació, hem volgut comparar la similitud entre dos fitxers diferents. Hem escollit l'índex *novels* per començar, i hem comparat fitxers de dos autors diferents Dickens vs Darwin on la similitud ha sigut de 0.00324, i fitxers del mateix autor (en aquest cas Dickens) on la similitud ha sigut més alta, com havíem suposat abans de realitzar l’experiment ja que tot i que el contingut del llibre sigui diferent la forma d’escriure de l’autor no canvia (o ho fa relativament poc), el resultat obtingut és: 0.02136.

Després hem volgut extendre els nostres experiments a diferents índexs. Hem utilitzat l'índex arxiv, el qual conté documents científics de temàtiques diverses. En primer lloc hem escollit dos fitxers que parlessin del mateix topic, quàntica, i el resultat ha estat més satisfactori que en els casos anteriors, ja que la similitud ha sigut de 0.03222. Aquí hem vist reflexat el fet que els documents parlen del mateix tema i, per tant, comparteixen moltes paraules. Per altra banda, hem comparat arxius de temàtiques diferents, quàntica i emulsions i, efectivament, ens hem trobat amb una similitud més reduïda a causa de la diferència de vocabulari (0.00738).

Per últim, hem acabat la nostra experimentació amb els documents dins la carpeta de *20\_newsgroup.* Hem començat indexant totes les carpetes juntes, i escollint dos documents que estiguin relacionats amb esports diferents, hockey vs atletisme, i hem observat que la similitud era baixa (0.02748). Seguidament, hem comparat dos documents d’atletisme per veure quan creixia la similitud, i el resultat obtingut ha sigut gairebé 10 vegades més gran que l’anterior, 0.2418. No hem provat la similitud de dos temes completament oposats (com esports i religió) ja que és obvi que la similitud entre aquests serà baixa.

Un cop plantejats aquests resultats, ens hem preguntat que passaria si reduissim els documents de l’índex i només agafèssim notícies relacionades amb atletisme. La nostra hipòtesis era que la similitud entre els documents testejats baixaria, ja que les paraules que abans havien tingut un pes important en remarcar la similitud dels documents (paraules relacionades amb atletisme, que diferencien clarament als documents d’altres esports que no les contenen), ara perdrien el seu valor al estar contingudes en tots els fitxers de l’índex. Al realitzar la prova, hem obtingut els resultats esperats, i la similitud dels mateixos documents amb els que havíem experimentat abans ha baixat a un 0.17940. D’aquesta manera confirmem la nostra hipòtesi, observant com influeix el fet que una paraula perd pes si està continguda en molts arxius d’un grup.