Projecte Algorismia FIB 2024-2025 Q2 Grup 15

Mario Rodrigo Ramos, Diego Moreno Vera
J.Oriol Ventura Cebada, Albert Aulet
10/3/2025

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

1 Introducció

1.1 Descripció de l'objectiu

L'objectiu d'aquest projecte és la validació experimental per a diferents algoritmes de hash per detectar la similitud entre documents.

1.2 Tasca realitzada

Per assolir aquest objectiu, hem implementat i comparat tres algoritmes de detecció de similitud: Similitud de Jaccard, MinHash i LSH. Cada algoritme ha estat provat amb diferents configuracions segons la mida dels k-shingles. Els experiments s'han realitzat sobre documents generats a partir de permutacions d'un mateix text per observar com responen els algoritmes en diferents graus de variació.

1.3 Conclusions esperades

El que esperem que succeeixi per als nostres tres algoritmes:

Similitud de Jaccard: Pensem que serà el més precís, però amb un cost computacional elevat per a conjunts grans donat que és l'algoritme que compara més exhaustivament.

MinHash: Esperem que també s'aproximi fortament a la similitud de Jaccard i que tingui un cost computacional inferior respecte Jaccard, donat que és bastant menys exhaustiu pel que fa a shingles, però segueix comparant amb tots els documents.

LSH: Preveiem que sigui el més eficient, ja que és menys exhaustiu en quant a shingles i limita les comparacions amb documents, però creiem que en alguns casos no trobarà el document que coincideix més.

2 Disseny experimental

2.1 Descripció del algoritme

Similitud de Jaccard:

Aquest mètode compara directament els k-shingles de dos documents per calcular la seva similitud.

- Generem k-shingles per als dos documents, que son cadenes consecutives de caràcters que tenen longitud k.
- Per a cada document creem un conjunt d'aquestes cadenes, que en el nostre cas serà de caràcters però podrien ser també de paraules.
- Fem el càlcul de la similitud de Jaccard, que es defineix com la intersecció entre els shingles compartits dividit entre la unió dels shingles d'ambdós conjunts.

$$J(A,B) = |A \cap B| \,/\, |A \cup B|$$

La similitud de Jaccard es dóna com a un espectre entre 0 i 1 sent 0 completament diferent i 1 identic.

Avantatges: Precisió alta perquè compara tots els *shingles*.
Inconvenients: Cost computacional elevat, especialment per conjunts grans de documents.

MinHash:

Proporciona una estimació eficient de la similitud de Jaccard utilitzant funcions de hash generades amb la funció hash definida per c++. A partir d'aquesta funció generem tantes funcions de hash com necessitem amb la fórmula: HashC++(shingle)*a + b.

- Generem els k-shingles igual que a Jaccard.
- Apliquem diverses funcions de hash als shingles per obtenir valors numèrics.
- Calculem la signatura MinHash guardant per cada funció de hash el valor més petit de cada document, creant així una signatura de longitud n.
- Estimació de la similitud de Jaccard, que és el percentatge de valors coincidents entre les dues signatures MinHash.

Avantatges: Redueix la complexitat en comparació amb Jaccard, mantenint una bona precisió.
Inconvenients: Encara requereix comparacions entre totes les signatures, fent-lo menys escalable per conjunts grans.

LSH:

LSH redueix dràsticament el nombre de comparacions agrupant documents similars abans de fer càlculs detallats.

- Càlcul de la signatura MinHash per cada document la qual representa la seva similitud relativa amb altres documents.
- Dividir en b bandes cadascuna d'**r** valors cadascuna on **n** serà igual a **(b * r)**, i cada banda es tracta com una "mini signatura".
- Aplicar una funció de hash per banda, donada la sortida d'aquesta funció els documents amb la mateixa sortida es consideren candidats.
- Comparació únicament amb candidats utilitzant MinHash

Avantatges: Redueix enormement el nombre de comparacions, millorant l'eficiència en conjunts grans.
Inconvenients: Pot ometre alguns documents similars si no comparteixen bandes al mateix bucket.

2.2 Descripció de les entrades experimentals

Generació de dades sintètiques

Per provar els algoritmes, hem generat permutacions controlades d'un mateix document original, aplicant modificacions com:

- Substitució de paraules per sinònims.
- Eliminació o afegit de frases.
- Reordenació de paràgrafs.
- Canvi d'ordre de paraules.
- Aquest mètode ens permet avaluar com reaccionen els algoritmes a diferents nivells d'alteració d'un text.

2.3 Descripció dels experiments realitzats

S'han dut a terme dos experiments amb diferències fonamentals en la seva metodologia i enfocament. A continuació, es descriu el primer experiment:

EXPERIMENT 1:

En aquest experiment, es va treballar amb conjunts de documents generats mitjançant permutacions aleatòries d'un document base. L'objectiu principal era analitzar i estudiar com es comporten les tècniques de comparació de documents basades en k-shingles, Jaccard, minhash i Locality-Sensitive Hashing (LSH).

Característiques principals:

Dades:

Document Base: Es va utilitzar un document de text base que contenia almenys 50 paraules diferents. Aquest document va servir com a punt de partida per generar els conjunts de documents permutats.
Generació de Documents Permutats: Es van crear almenys 20 documents diferents mitjançant permutacions aleatòries del document base. Aquestes permutacions van alterar l'ordre de les paraules o frases del text original, però sense modificar el contingut de les paraules individuals.

Generació de k-shingles:

Per a cada document, es va crear un conjunt de k-shingles, on els k-shingles són subconjunts de k lletres consecutives del text. Aquests k-shingles van ser la base per a la comparació de documents.

Generació de signatures hash:

Es va aplicar la tècnica de minhash per generar signatures hash a partir dels k-shingles. Aquestes signatures permeten comparar documents de manera eficient en termes de similitud.

Paràmetres estudiats:

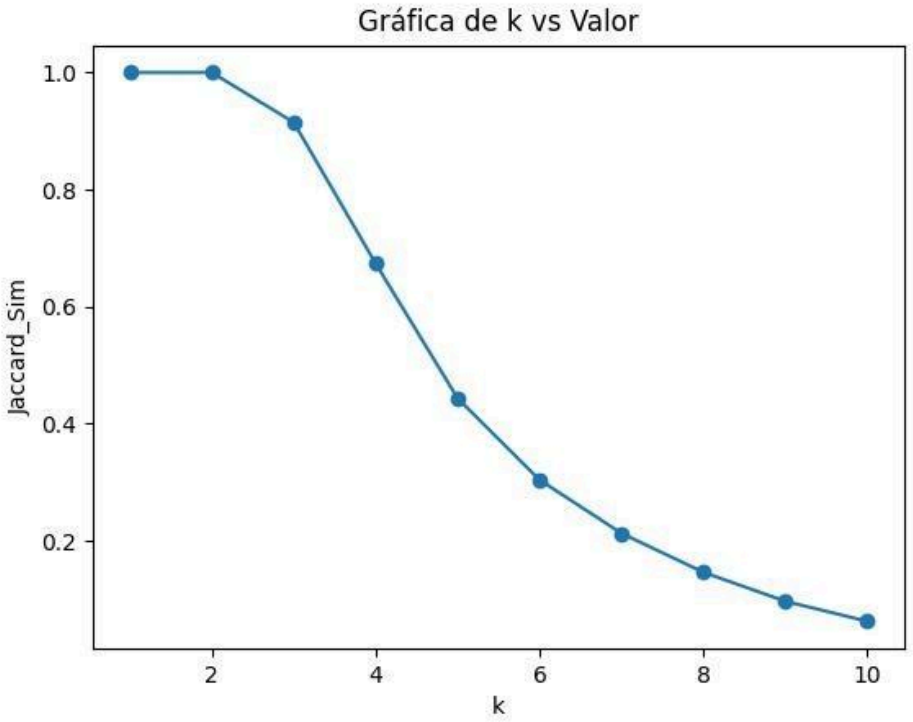
Respecte a k: Es van analitzar les diferències en la similitud de Jaccard en funció del valor de k (nombre de lletres consecutives en els k-shingles).
Respecte a les funcions de hash: Es van comparar les diferències entre la similitud calculada amb minhash i la similitud de Jaccard directa.
Respecte a les bandes del LSH: Es va estudiar com el nombre de bandes en la tècnica de Locality-Sensitive Hashing (LSH) afecta el nombre de documents potencialment semblants, així com la precisió i el recall.

Anàlisi de k:

Procediment: Es vol recollir la similitud de Jaccard del document base amb les seves permutacions per a diferents valors de k un total de 10 vegades, els valors obtinguts per cada k resultaran de la mitja de les 10 execucions del experiment.

Resultats esperats: Esperem que la similitud disminueixi amb l'augment de la variable k degut a que el tamany del shingle creixerà i farà que trobar coincidències al conjunt de shingles sigui més complex.

Resultat:



Gràfica 1.1

Anàlisi de resultat:

Per avaluar la variació de la similitud de Jaccard en funció del paràmetre, observem la Gràfica 1.1. En aquesta, es veu clarament que és com la similitud disminueix a mesura que augmenta el valor de k. Aquest resultat coincideix amb la nostra

hipòtesi: quan k és petita (per exemple, 1 o 2), els shingles corresponen a lletres o síl·labes, fet que facilita la coincidència entre textos. En canvi, quan augmenta i els shingles representen paraules senceres o frases curtes, es fa més difícil que dos textos tinguin molts shingles en comú i, per tant, la similitud de Jaccard es redueix.

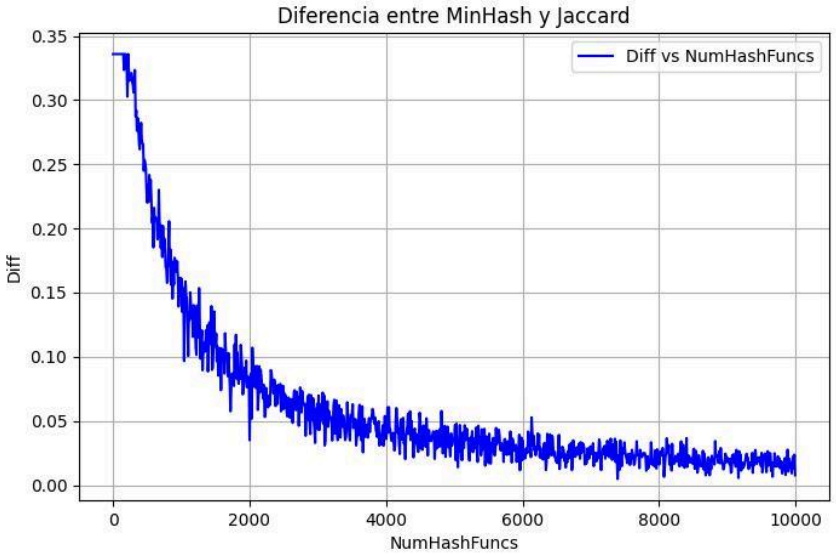
A partir d'aquest experiment en endavant treballarem amb un sol valor de k que serà 3. Amb aquest valor i pels documents que hem seleccionat veurem com és el més eficient en quant a resultats i també pel que fa a temps d'execució.

Anàlisi del número de funcions de hash:

Procediment: Per trobar l'efecte en la qualitat de solucions de treballar amb minhash en comptes de Jaccard calcularem la diferència en el resultat per cada nombre de funcions de hash possible. Caldrà tenir en compte que l'augment de nombre de funcions hash tindrà un efecte negatiu en el cost computacional del procés. Per aquest experiment també executarem el codi 10 vegades per cada valor i calcularem el resultat mitjà.

Resultats esperats: Podem esperar que la diferència de minhashing cap a la similitud de Jaccard tendeixi cap a 0 amb l'augment del nombre de funcions hash, encara així mai podrem garantir la precisió del resultat.

Resultat:



Gràfica 1.2

Anàlisi de resultats: Observem una gràfica que demostra la nostra teoria de tendència cap al 0 amb un interval de resultats de (0, 0.35). Hem de tenir en compte que els resultats de les aproximacions van de [0, 1] i, per tant, un error de 0.35 en pot canviar en gran mesura el resultat. S'observa a més molta fluctuació de resultats encara després de calcular els resultats 10 cops.

Podem extreure certes conclusions sobre els resultats obtinguts que ens ajudaran a determinar un valor de nombre de funcions de hash de cara als següents experiments. Primerament, veiem que no és viable el treball amb un valor menor a 2000, ja que podem tenir una variància en un 0.1 del resultat correcte, per tant, per tal de no augmentar massa el cost computacional del programa i mantenir l'eficiència en resultats, a partir d'aquest punt utilitzarem 4000 funcions de hash.

Anàlisi del número de bandes en LSH:

Procediment: Per aquesta part de l'experiment voldrem analitzar 3 aspectes del treball amb LSH. Sabem que els resultats de l'aplicació d'aquest algorisme venen definits pel nombre de bandes **b** i, en conseqüència, la mida de les bandes **r**. El primer que voldrem analitzar és la quantitat de resultats que ens retorna el LSH al comparar un sol document amb tot el conjunt en funció del nombre de bandes sense tenir en compte la correctesa del resultat. Els següents valors que voldrem calcular són el **recall** i la **precisió** per tal de minimitzar els falsos positius i maximitzar els positius reals retornats, paràmetres els quals volem maximitzar i venen donats per les següents fòrmules:

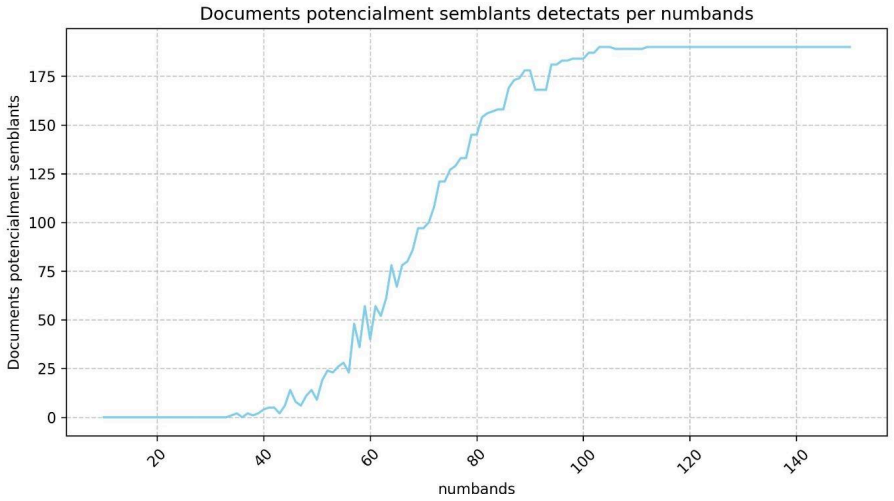
$$Precisió = \frac{\#\ Parelles\ realment\ semblants\ detectades}{\#\ Parelles\ semblants\ detectades} \quad recall = \frac{\#\ Parelles\ semblants\ reals\ detectades}{\#\ Parelles\ semblants\ existents}$$

Per fer aquesta última part farem una petita variació de l'experiment amb el qual veníem treballant, fins ara hem utilitzat un document base per crear un cert nombre de permutacions però aquestes permutacions sempre seran semblants. Per arreglar aquest problema utilitzarem 6 documents dels quals farem les permutacions i compararem un sol document amb tot el nou conjunt.

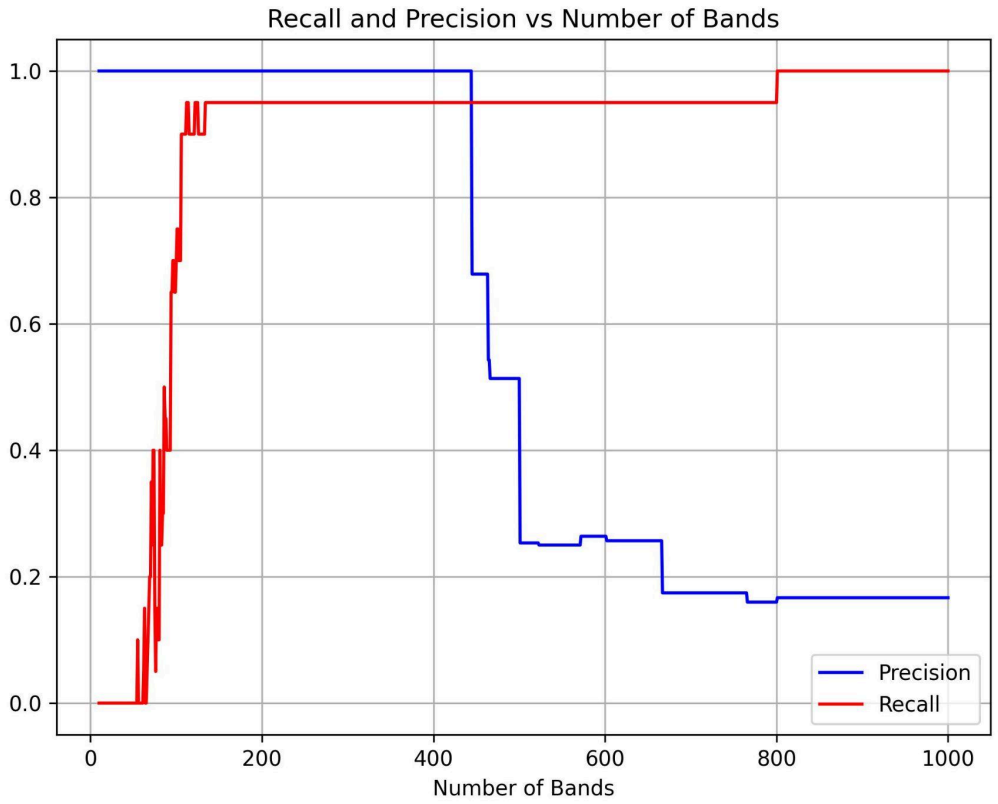
Resultats esperats: El comportament del LSH és bastant predictible, ja que sabem que com més bandes utilitzi, el seu tamany per banda serà més petit i en conseqüència menys restrictiu, per tant, el nombre de documents començarà a 0 i finalment retornarà totes les parelles possibles.

Per la segona part sabem que el document que estem comparant és realment similar només a les seves permutacions, les quals representen una sisena part del total de documents, per tant, la precisió és molt probable que sigui del 100% per baixos números de bandes, ja que són les més restrictives i el més probable és que només detectin positius reals i que per grans números de bandes baixi fins a ⅙, pel fet que retornarà com a possible similitud el 100% dels documents. El recall podem assegurar que inicialment serà 0%, ja que no retornarà cap possible parella i que arribarà fins al 100% quan el nombre de bandes sigui prou lax.

Resultats:



Gràfica 1.3



Gràfica 1.4

Anàlisi de resultats: En la primera gràfica observem que amb un nombre de bandes baix i molt restrictiu costa molt trobar documents semblants i que a mesura es va relaxant la restricció de les bandes l'algorisme té més facilitat per detectar nous documents potencialment similars fins a acabar donant com a possibles candidats a similars el conjunt total de documents amb els quals estem treballant.

La segona gràfica demostra la nostra hipòtesi tant per la precisió com pel recall. Aquesta primera arriba fins al valor predit de ⅛ pel percentatge de documents similars entre si. Al recall també es reafirmen les nostres hipòtesis, però podem veure que fins a gairebé al final no hem arribat al 100%. Aquest fet sembla degut al fet que alguna de les permutacions del document que utilitzem com a emparellador justament tenia una firma minhash especialment diferent i ha sigut difícil de detectar per l'algorisme lsh. Veiem que aproximadament entre les 200 i 400 bandes minhash tenim una zona òptima per treballar, ja que tant el recall com la precisió són pràcticament màximes.

EXPERIMENT 2:

En aquest experiment, es va treballar amb subconjunts de k-shingles extrets d'un únic document base. L'objectiu principal era analitzar com es comporten les tècniques de comparació de documents basades en k-shingles, la similitud de Jaccard, MinHash, Locality-Sensitive Hashing (LSH) i la similitud teòrica esperada, variant els paràmetres k i el percentatge d'extracció dels shingles respecte al document base.

Característiques principals

Dades:

Document Base: Es va utilitzar un únic document de text base, del qual es van extreure els seus k-shingles. Aquest document contenia més de 100 paraules úniques, excloent stopwords.

Generació de subconjunts: Es van crear subconjunts seleccionant un percentatge determinat dels *k-shingles* originals.

1. Es va definir el nombre de subconjunts i el percentatge, el qual indica la quantitat de shingles que contindrà del document base
    a. Nombre de subconjunts: 20
    b. Percentatges: 30%, 50% i 70%
2. Un cop definit el nombre de subconjunts i el percentatge, per cada subconjunt es duia a terme una selecció aleatòria dels k-shingles del document base tenint en compte el percentatge

Generació de *k-shingles*:

Per a cada subconjunt generat, es va obtenir un conjunt de *k-shingles*, on els *k-shingles* son cadenes consecutives de *k* lletres del text. Aquests *k-shingles* van ser la base per a la comparació entre subconjunts.

Generació de signatures hash:

Es va aplicar la tècnica de minhash per generar signatures hash a partir dels k-shingles. Aquestes signatures permeten comparar documents de manera eficient en termes de similitud.

Càlcul de similitud:

Similitud teòrica: Es va calcular segons la proporció esperada de *k-shingles* compartits entre subconjunts.

Similitud de Jaccard: Es va mesurar directament la similitud Jaccard entre els subconjunts de *k-shingles*.

Similitud mitjançant MinHash: Es va aplicar la tècnica de MinHash per generar signatures hash i estimar la similitud entre subconjunts.

Paràmetres estudiats:

Respecte a *k*: Es van comparar les variacions en la similitud de Jaccard i MinHash en funció del valor de *k* (nombre de lletres consecutives en els *k-shingles*).

Respecte a la diferència entre MinHash, Jaccard i la similitud teòrica: Es va analitzar com varia l'error entre la similitud estimada per MinHash i la similitud Jaccard real en comparació amb la teòrica.

Respecte a les bandes del LSH: Es va estudiar com el nombre de bandes en la tècnica de Locality-Sensitive Hashing (LSH) afecta el nombre de documents potencialment semblants, així com la precisió i el recall.

Tots els paràmetres mencionats anteriorment, van ser estudiats amb tres subconjunts de documents virtuals diferents. La diferència entre els subconjunts era el percentatge d'extracció dels shingles respecte al document base. L'objectiu era veure com afecta la selecció parcial de k-shingles als diferents paràmetres. En el nostre experiment els percentatges escollits han sigut 30%, 50% i 70%. La tria d'aquests valors s'ha fet per tal d'abastar un ventall ampli de solapament entre documents, sent el percentatge de 30% el que simula documents amb poc solapament de contingut, el de 50% amb solapament moderat i el de 70% documents molt similars. Aquesta tria ens permet veure com els diferents paràmetres es comporten en escenaris diferents.

Anàlisi de k:

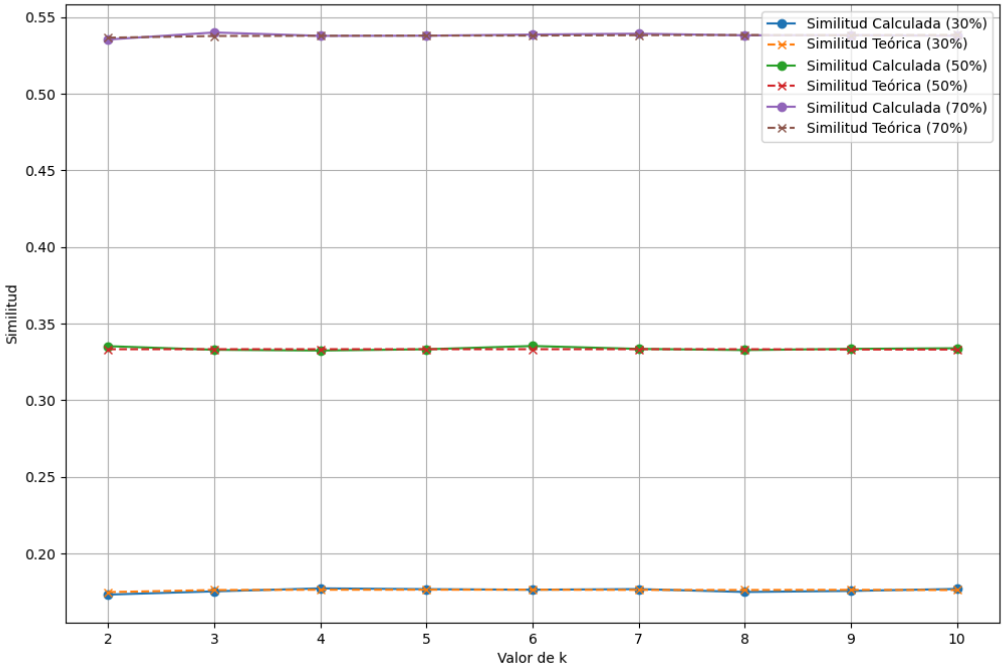
Objectiu: Analitzar com la mida dels k-shingles afecta a la similitud entre un subconjunt de documents virtuals generats a partir d'un percentatge d'extracció respecte a un document base. També es busca validar si la similitud calculada, en aquest cas mitjançant Jaccard, es comporta d'acord amb la similitud teòrica.

Procediment: Es vol recollir la similitud de Jaccard d'un conjunt de documents virtuals generats a partir d'un document base. Com s'ha mencionat en les característiques principals de l'experiment, el document base conté més de 100 paraules úniques, excloent stopwords, i es generaran 20 documents virtuals. Els

valors obtinguts per cada k resultaran de la mitja de les 10 execucions de l'experiment.

Resultats esperats: Esperem que la similitud es mantingui constant independentment del valor de la k, ja que tots els documents tenen el mateix percentatge d'extracció i aquesta és aleatòria, fet que assegura una distribució uniforme.

Resultat:



Gràfica 2.1

Anàlisi resultat:

Per avaluar la variació de similitud de Jaccard en funció del paràmetre k i la diferència d'aquesta amb la similitud teòrica observem la gràfica 2,1. Com podem veure, independentment del percentatge, la similitud es manté constant i és, aproximadament, igual que la teòrica. El resultat obtingut recolza la hipòtesi formulada abans de dur a terme l'experiment, ja que tots els documents virtuals tenen el mateix percentatge de shingles extrets respecte al document base i sent l'extracció aleàtoria ens permet assegurar-nos que la distribució sigui uniforme i que cada shingle tingui la mateixa probabilitat de ser seleccionat. És per això que la similitud es manté constant independentment del valor de la k.

Per altra banda, observem que la variació del percentatge d'extracció no afecta en les conclusions extretes, simplement per cada percentatge el valor de la similitud, tant teòrica com calculada, serà diferent, però el comportament d'aquesta serà el mateix independentment del valor del percentatge.

A partir d'aquest experiment en endavant treballarem amb un valor fix de k que serà 3. Això és degut al fet que els resultats previs extrets mostren que la similitud no depèn de k i el valor de 3 és un valor moderat entre cost i qualitat dels resultats.

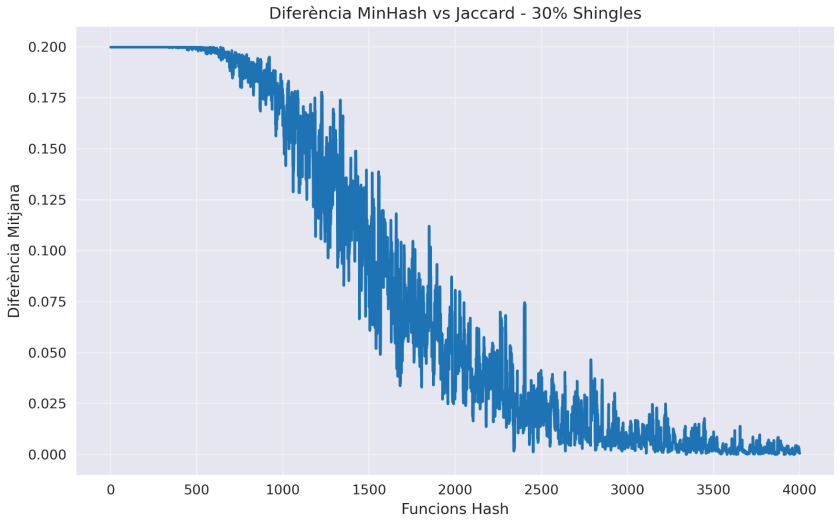
Anàlisi del número de funcions de hash:

Objectiu: Analitzar com el nombre de funcions de hash afecta a la precisió i eficàcia en l'estimació de la similitud de Jaccard utilitzant minhash.

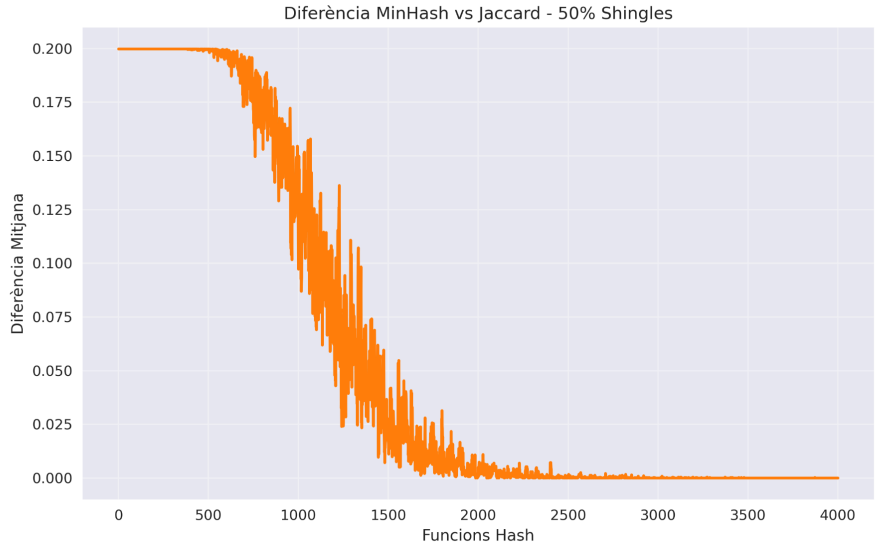
Procediment: Es vol recollir la precisió en l'aproximació de la similitud de Jaccard mitjançant l'ús de signatures minhash. Per tal de trobar la precisió i eficàcia a l'hora de treballar amb signatures minhash, calcularem la diferència entre la similitud teòrica i la similitud obtinguda a través de signatures minhash per cada nombre de funcions hash possible. Tot i això, caldrà tenir en compte que l'augment de nombre de funcions de hash implicarà un augment en el cost computacional del nostre programa. Com en l'anàlisi anterior, també durem a terme l'experiment 10 cops per tal d'obtenir uns resultats més robustos.

Resultats esperats: Esperem que, a mesura que augmentem el nombre de funcions de hash, la diferència disminueixi considerablement tendint així cap a 0. Per altra banda, esperem que, com més alt sigui el percentatge, la diferència entre la similitud calculada a través de signatures minhash i la teòrica disminueixi de forma més ràpida.

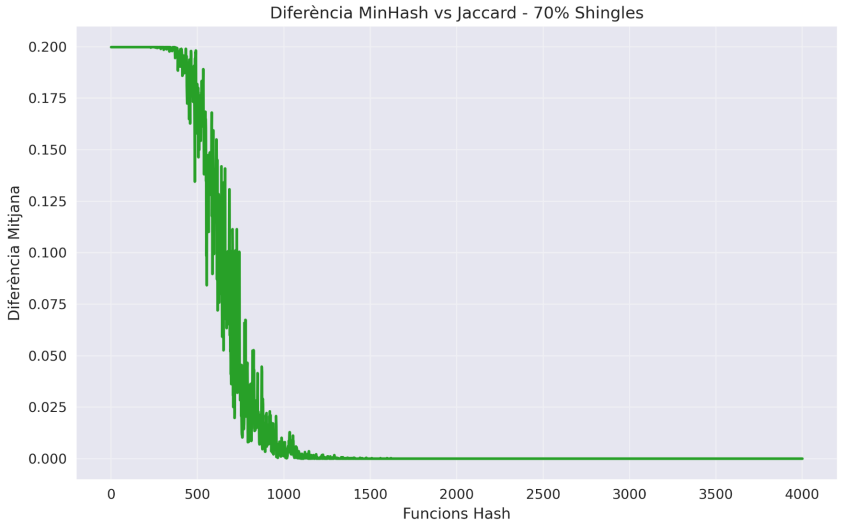
Resultat:



Gràfica 2.2



Gràfica 2.3



Gràfica 2.4

Anàlisi del resultat:

Per avaluar la precisió i l'eficiència a l'hora de calcular la similitud utilitzant signatures minhash observem les gràfiques 2.2, 2.3 i 2.4, cada una corresponent a un subconjunt de documents amb un percentatge determinat d'extracció, 30%, 50% i 70% respectivament. Com podem observar en les tres gràfiques la diferència entre la similitud teòrica i la similitud mitjançant minhash tendeix cap a 0, recolzant així la nostra hipòtesi inicial que com més elevat fos el nombre de funcions de hash, més precís seria el càlcul de la similitud, fent per això més elevat el cost computacional a l'hora de fer el càlcul. Si ens fixem en la tendència cap a 0, podem observar com en el cas dels subconjunts amb un 30% d'extracció fan falta, aproximadament, 3000 funcions de hash per tal que la diferència sigui inferior a 0.025. Per altra banda, en

el cas dels subconjunts amb 50% d'extracció, veiem que per tal que la diferència sigui inferior a 0.025 són necessàries, aproximadament, més de 1750 funcions de hash, necessitant així 1250 funcions menys que en el cas de 30%. Per últim, tenim el cas del subconjunt de 70%, on seguint la nostra hipòtesi inicial, el nombre de funcions de hash necessàries per arribar a una diferència inferior a 0.025 haurien de ser inferiors que en el cas anterior. Fet que podem confirmar observant la gràfica 2.4, on observem que fan falta una mica menys de 1000 funcions per tal d'assolir-ho. Per altra part, veiem que a partir de les 1300 funcions, ja tendeix completament a 0, mentre que en el cas del 30% i 50% es necessita més de 4000 i 3000 funcions respectivament. Aquest fet reforça la nostra altra hipòtesi inicial, la qual indicava que com més elevat fos el percentatge la diferència entre la similitud teòrica i la similitud de minhash més ràpid tendiria cap a 0.

En el següent experiment ens serà necessari utilitzar funcions de hash i, basant-nos en les gràfiques obtingudes, el valor que farem servir seran 3000 funcions de hash. Això és degut al fet que, tot i que en el cas dels subconjunts del 50% i 70% serien necessàries menys de 2000 i 1000 funcions hash respectivament, en el cas de 30% la diferència comença a considerar-se viable per treballar en el cas de les 3000 funcions de hash. Per tant, a partir d'aquest punt treballarem amb 3000 funcions de hash.

Anàlisi del número de bandes en LSH

Objectiu: Analitzar com afecta el número de bandes a la precisió a l'hora de detectar documents similars i en la reducció de falsos positius i negatius.

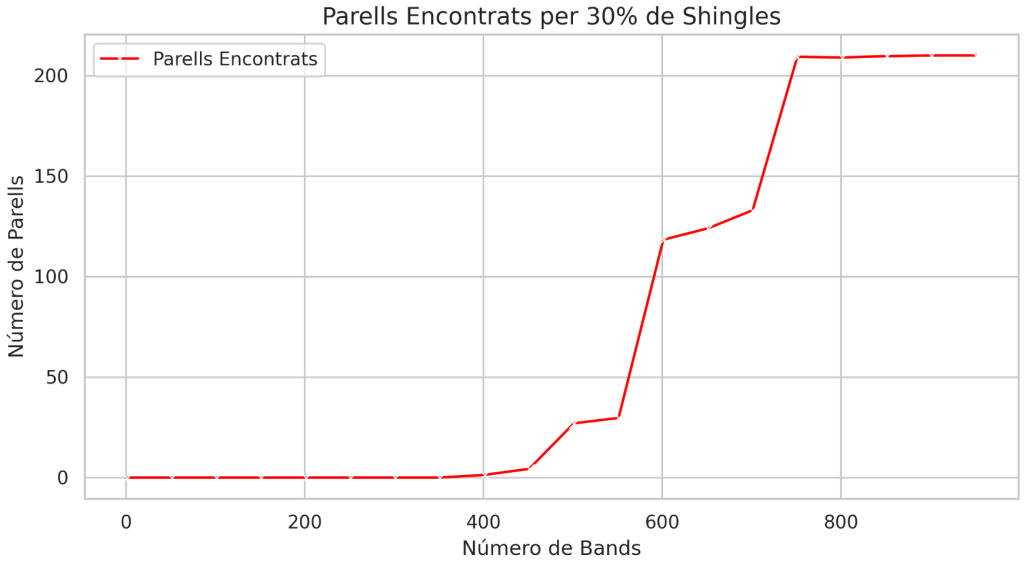
Procediment: Es vol recollir com afecta el número de bandes i, per tant, també la seva mida, a la precisió i eficàcia a l'hora de treballar amb LSH. Per tal de fer-ho, primer de tot analitzarem la quantitat de resultats que ens retorna el LSH al comparar els documents virtuals dels subconjunts en funció del nombre de bandes, sense tenir en compte la correctesa del resultat retornat. A continuació, analitzarem el que s'anomena recall, que és la fracció de parells similars reals detectats, i la precisió, fracció de parells candidats que són realment semblants. L'anàlisi d'aquests dos valors ens ajudarà a veure la correctesa dels resultats.

Resultats esperats: Esperem que a l'inici el nombre de parells similars que detectarà sigui 0, això passa perquè, com menys bandes utilitzi més gran serà la seva mida i, per tant, més restrictiu serà. En canvi, a mesura que augmentem el nombre de bandes, esperem que detecti més parells similars, pel mateix motiu. Finalment, hi haurà un punt on ens retornarà totes les parelles possibles. Per altra

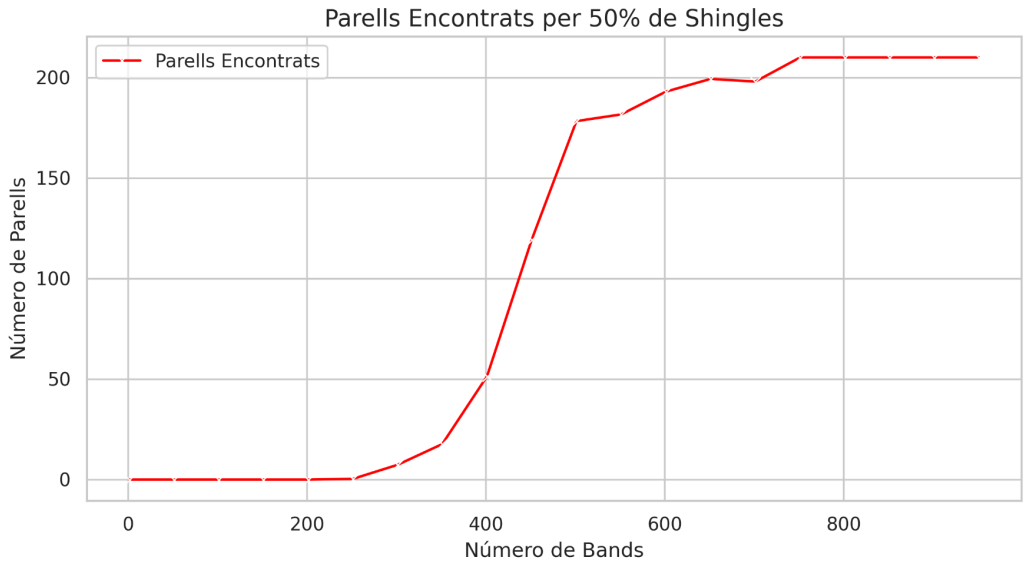
banda, sabem que tots els documents del subconjunt tenen, aproximadament, la mateixa similitud. Per tant, esperem que, a partir d'un cert nombre de bandes, el nombre de parells detectats creixi de forma sobtada. Aquest creixement sobtat, en el cas del subconjunt de 30% esperem que necessiti més nombre de bandes en comparació amb el subconjunt de 50% i aquest al mateix temps, del subconjunt de 70%. Això és degut al fet que, com més elevat sigui el percentatge, més components coincidents tindran les signatures minhash i, per tant, augmentarà la probabilitat que dues signatures minhash coincideixin en una banda.

Pel que fa a la precisió i recall, aquesta primera esperem que sigui del 100% amb números de bandes petits, ja que és quan és més restrictiu i, doncs, només detectarà parells que siguin positius reals, i a mesura que vagin augmentant el nombre de bandes la precisió disminuirà sútilment, pel fet que en tenir tots els documents la mateixa similitud, aproximadament, tots els parells seran positius reals. Respecte al recall, esperem que a l'inici sigui 0 fins a arribar al 100% perquè el nombre de bandes farà que sigui menys restrictiu.

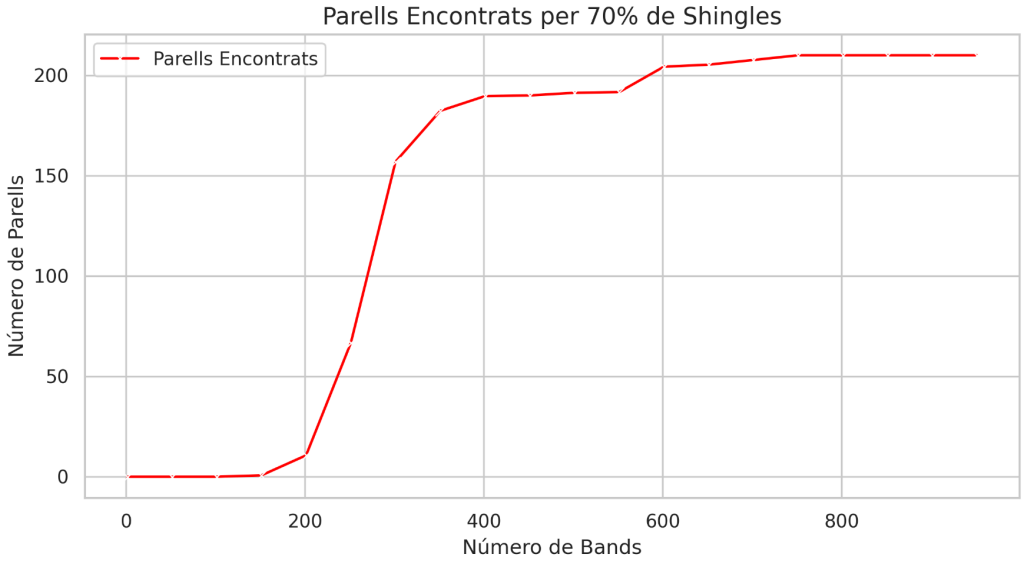
Resultats parells similars detectats:



Gràfica 2.5



Gràfica 2.6



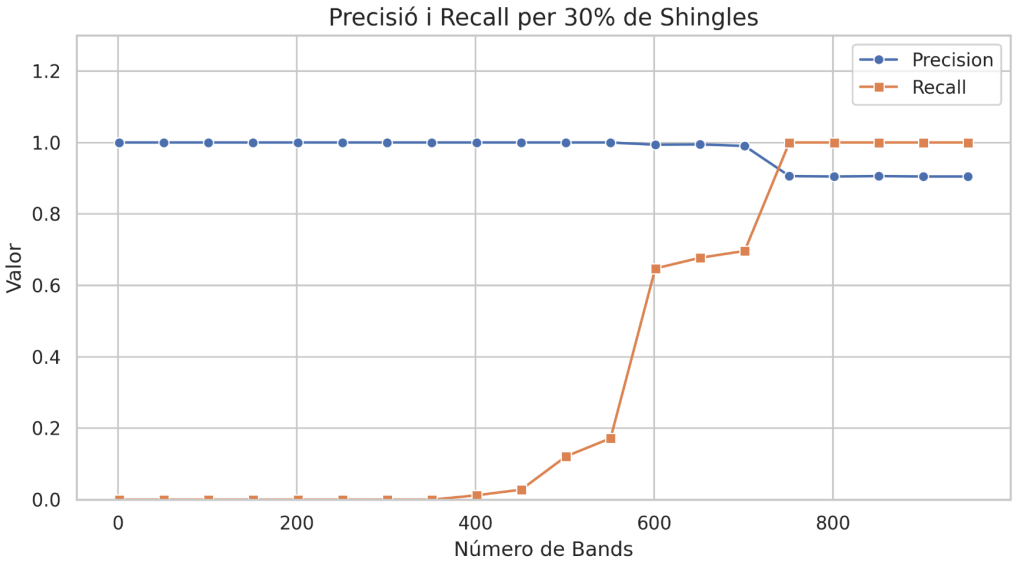
Gràfica 2.7

Anàlisi resultat número de parells detectats:

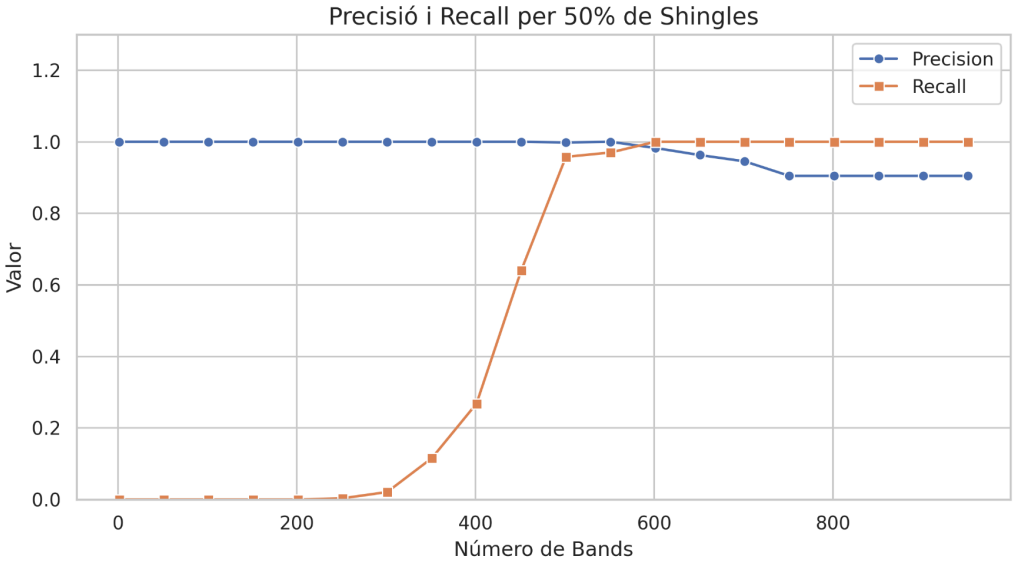
Per avaluar com afecta el nombre de bandes a la quantitat de parells de documents similars detectats, observem les gràfiques 2.5, 2.6 i 2.7, corresponents als subconjunts de 30%, 50% i 70% respectivament. Com podem observar, a l'inici el nombre de parells detectats és zero, ja que en haver-hi un nombre petit de bandes, aquestes són més grans i, per tant, més restrictives. Com podem observar, a mesura que es va augmentant el nombre de bandes, cada cop és menys restrictiu i, en conseqüència, detecta més parells de documents. Fins que arriba a un cert nombre de funcions on, com tots els documents tenen la mateixa similitud, ja detecta tots i, aleshores, es manté constant. Per altra banda, podem veure com, quant més elevat sigui el percentatge, menys nombre de bandes seran necessàries

per tal de detectar els parells similars, ja que com més elevat sigui el percentatge, més components coincidents tindran les signatures minhash.

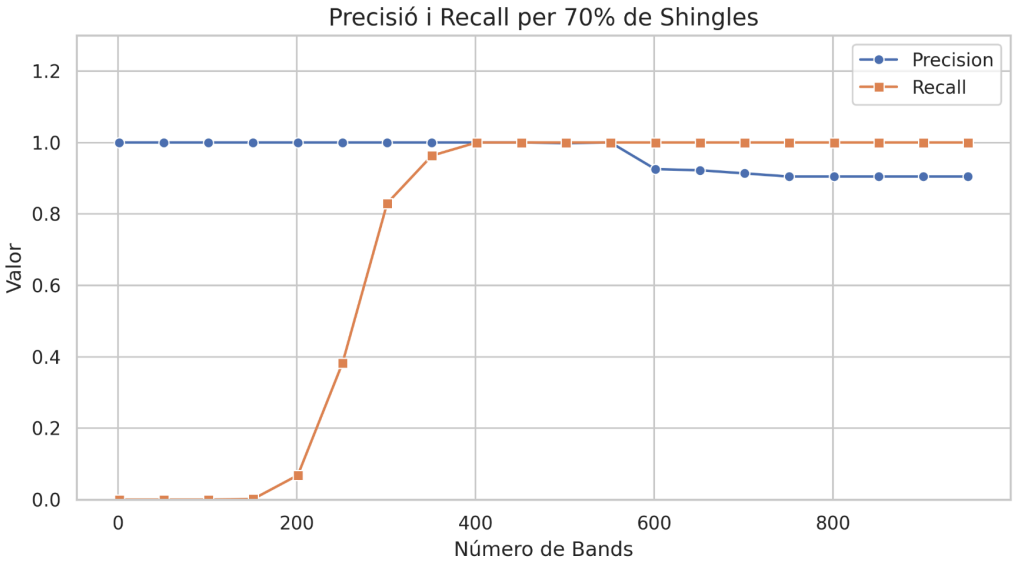
Resultats recall i precisió:



Gràfica 2.8



Gràfica 2.9



Gràfica 2.10

Anàlisi resultats recall i precisió:

Per avaluar com afecta el nombre de bandes a la precisió i recall observem les gràfiques 2.8, 2.9 i 2.10, corresponents al subconjunt de 30%, 50% i 70% d'extracció respectivament. Com podem observar en els gràfics, es confirma la nostra hipòtesi inicial que, quan el nombre de bandes és petit la precisió és del 100%, ja que és més restrictiu i només detectarà els parells que són positius reals i a mesura que el nombre de bandes vagi augmentant, farà que sigui menys restrictiu i, per tant, la precisió baixarà perquè hi haurà falsos positius, i el recall pujarà fins a arribar al 100%. Per altra banda, observem que com més elevat sigui el percentatge més ràpid arribarà el moment en el qual detecta tots els parells de documents. Això és degut al fet que, com tots els documents tenen el mateix percentatge d'extracció, hi haurà un moment en què el nombre de bandes serà prou per a detectar tots els documents amb aquella similitud.

2.4 Entorn experimental

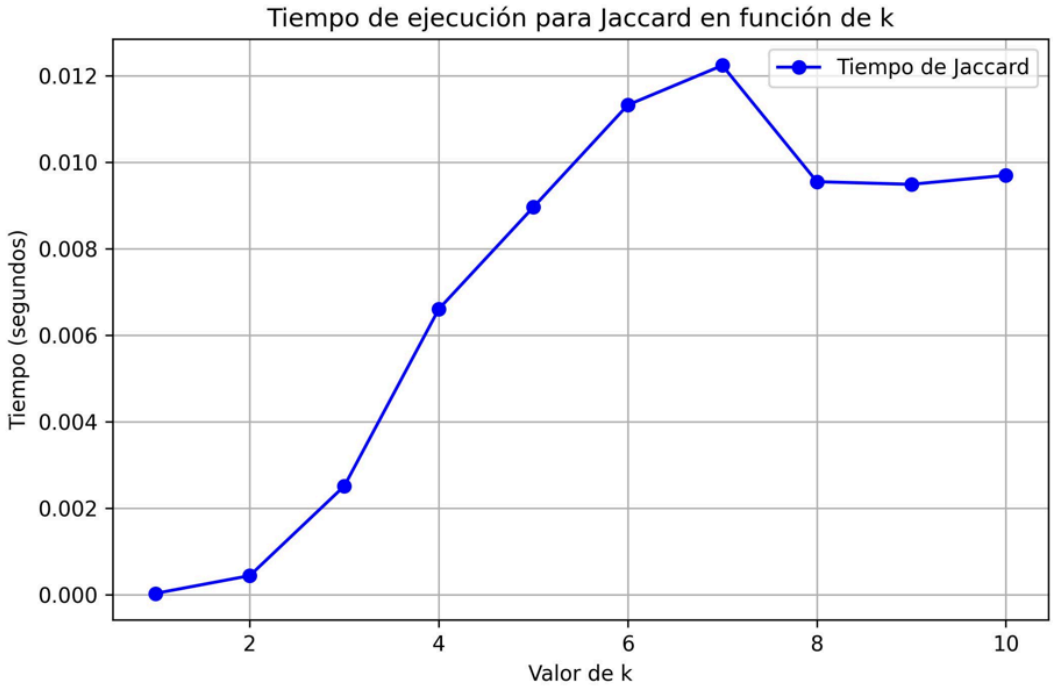
Tots els experiments han sigut executats en una CPU: AMD Ryzen 5 7600 i en Windows 11.

3. Anàlisi temporal

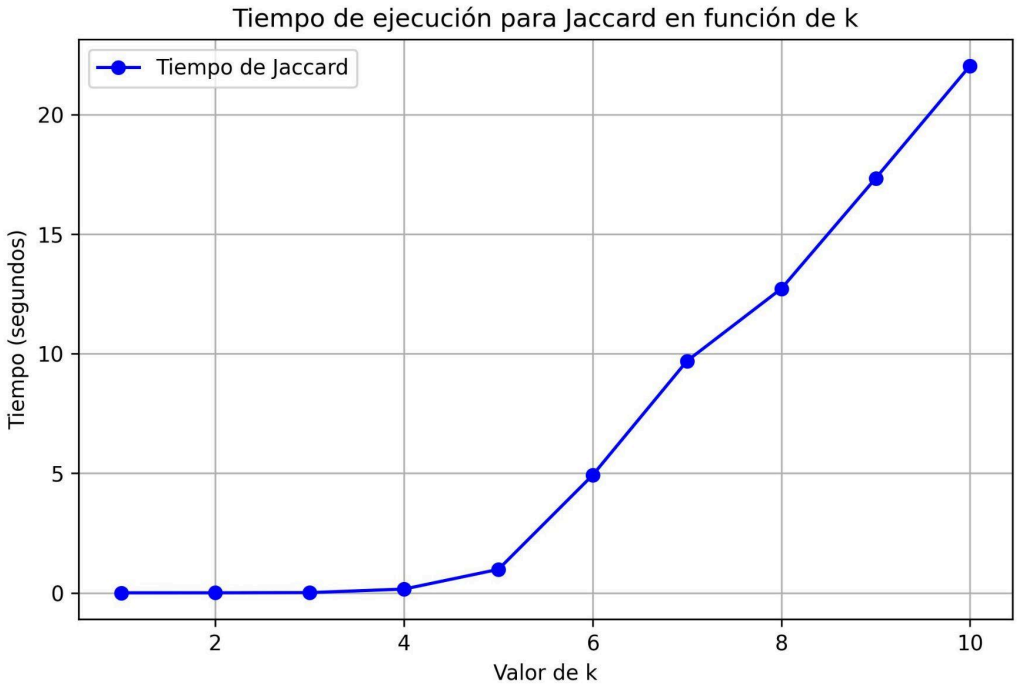
Un cop explorades les conseqüències de les diferents variables del codi volem saber quin cost temporal tenen aquestes variables sobre l'execució del nostre programa. Analitzarem l'efecte de k i el número de funcions de hash.

Anàlisi temporal de la variable k

Procediment: Per analitzar l'efecte de k variarem el seu valor i calcularem el temps total en l'execució de la similitud de Jaccard per un conjunt de documents permutats de 311 mots diferents i, a més, tornarem a executar l'algorisme per un text molt gran, en aquest cas el document serà el text sencer de "El Quixot"



Gràfica 3.1



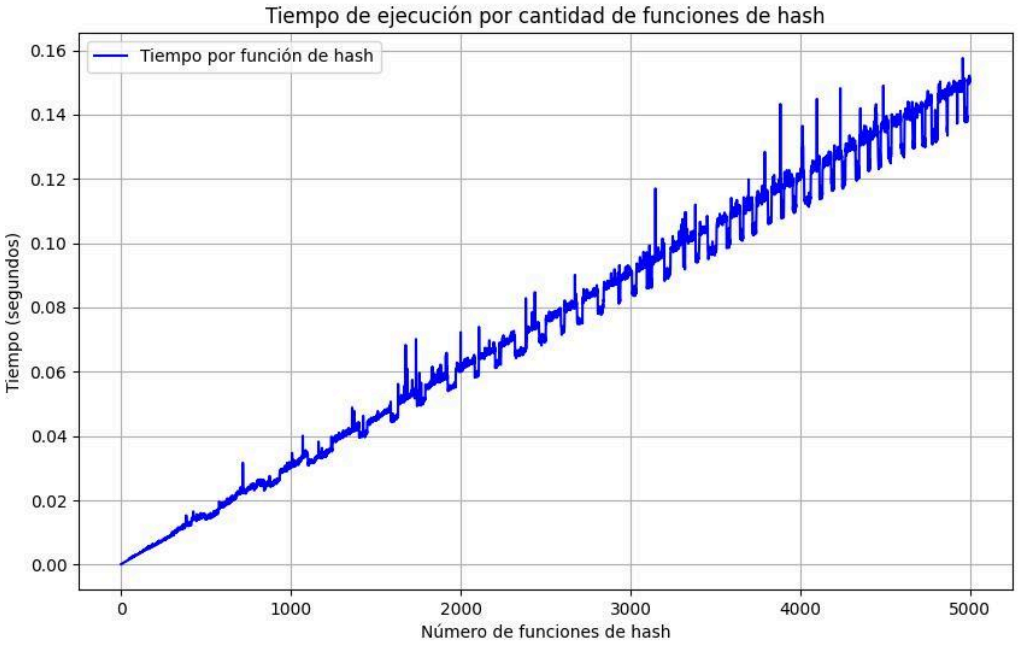
Gràfica 3.2

Anàlisi de resultats: La gràfica 3.1 representa el temps d'execució funció de k pel text de 311 paraules i la gràfica 3.2 pel document de "El Quixot". Podem observar que a 3.1 a partir de k = 8 baixa. Aquest fet es produeix ja que com més gran és la k, menys potencials shingles pot generar el mateix text i per aquest en concret a partir de k = 7 genera menys shingles. Veiem que per un text com "El Quixot" amb gairebé 25000 paraules diferents eliminant stopwords el temps segueix pujant en gran mesura.

Anàlisi temporal de la variable numHashes

Procediment: Per aquest experiment farem una execució simple amb el mateix cas de l'experiment anterior, però en aquest cas variarem el número de funcions de hash per veure el seu efecte temporal. Esperem un augment lineal, ja que com més funcions de hash, més vegades haurem de hashejar els shingles dels documents.

Resultat:



Anàlisi dels resultats: Veiem un clar augment lineal del temps tal i com havíem predit, encara que amb certes irregularitats que poden ser degudes a problemes en els accessos a memòria.

4. Conclusions

4.1 Principals conclusions

A través dels experiments realitzats, hem pogut validar i comparar el comportament de tres algoritmes de detecció de similitud entre documents: Similitud de Jaccard, MinHash i Locality-Sensitive Hashing (LSH). Les conclusions principals són:

1. Similitud de Jaccard:

- És el mètode més precís, però té un cost computacional elevat, especialment per a conjunts grans de documents. Ideal per a conjunts petits on la precisió és crítica.

2. MinHash:

- Ofereix una bona aproximació de la similitud de Jaccard amb un cost computacional inferior. A mesura que augmentem el nombre de funcions de hash, la precisió s'aproxima a la de Jaccard. Adequat per a conjunts de mida mitjana.

3. Locality-Sensitive Hashing (LSH):

- És el mètode més eficient en termes de temps de computació, especialment per a conjunts grans. Tot i que pot perdre alguna precisió, la seva capacitat per reduir el nombre de comparacions el fa ideal per a aplicacions on l'escalabilitat és clau.

4.2 Aspectes millorables

Hi ha alguns aspectes que es podrien millorar en futures investigacions:

1. Optimització de paràmetres:

- Seria interessant explorar com es comporten aquests algoritmes amb valors diferents de k i funcions de hash en conjunts de dades més grans i variats.

2. Ús de datasets reals:

- Provar els algoritmes amb datasets reals, com articles de premsa o textos acadèmics, per veure com es comporten en escenaris més complexos.

4.3 Noves hipòtesis i experiments

A partir dels resultats obtinguts, es podrien plantejar noves hipòtesis:

1. Impacte de la mida del document:

- Investigar com afecta la mida del document (nombre de paraules o shingles) a la precisió i eficiència dels algoritmes.

2. Efecte de la variació de k-shingles:

- Explorar com afecta l'ús de shingles de paraules (en lloc de shingles de caràcters) a la detecció de similitud.

4.4 Valoracions personals

Aquest projecte ha estat una experiència molt enriquidora en termes d'aprenentatge i aplicació pràctica de tècniques de detecció de similitud. Hem après la importància de trobar un equilibri entre precisió i eficiència, especialment en aplicacions amb grans volums de dades.
Un aspecte que ens ha sorprès és que, tot i reduir considerablement el nombre de comparacions, el **Locality-Sensitive Hashing (LSH)** manté una precisió molt alta sempre que s'utilitzi un nombre adequat de bandes. Això ens ha donat una idea clara de com poden funcionar els sistemes de comparació o cerca a gran escala, com els que utilitzen empreses com Google o altres plataformes que gestionen grans volums de documents. Abans d'aquest projecte, sempre havia estat una incògnita per a nosaltres com es podien comparar datasets tan grans de manera eficient. Tot i que no hem pogut explorar-ho en profunditat, aquest experiment ens ha proporcionat una primera intuïció sobre com es poden abordar aquests reptes a gran escala, reduint el cost computacional sense sacrificar excessivament la precisió.

En resum, aquest projecte ha estat una excel·lent oportunitat per explorar i comprendre les tècniques de detecció de similitud entre documents, i esperem que les conclusions i experiències obtingudes puguin ser útils en futures investigacions i aplicacions pràctiques.

5. Bibliografia

xxHash. (s.d.). *xxHash - Extremely fast hash algorithm*. Recuperat de https://xxhash.com

Lumbroso, J. (s.d.). *python-random-hash*. GitHub. Recuperat de https://github.com/jlumbroso/python-random-hash

Alir3z4. (s.d.). *Spanish stopwords list*. GitHub. Recuperat de https://github.com/Alir3z4/stop-words/blob/master/spanish.txt

Panchenko, A., Ruppert, E., & Hagen, M. (2023). *Evaluation of document deduplication algorithms for large text corpora*. Lamarr Institute. Recuperat de https://lamarr-institute.org/publication/evaluation-of-document-deduplication-algorithms-for-large-text-corpora/

MrHasanKthse. (19 de març de 2020). *Minhash and LSH*. Recuperat de https://mrhasankthse.github.io/riz/2020/03/19/Minhash-and-LSH.html

Viquipèdia. (s.d.). *Funció hash*. Recuperat de https://es.wikipedia.org/wiki/Funci%C3%B3n_hash

Andoni, A., Indyk, P., Laarhoven, T., Razenshteyn, I., & Schmidt, L. (2014). *Practical and optimal LSH for angular distance*. arXiv preprint arXiv:1407.4416. Recuperat de https://arxiv.org/abs/1407.4416