Laboratori de CAIM

Pràctica 3

Walter J.Troiani

Prof: Ignasi Gómez Sebastià

24/9/2023 - 2023/24 Q1

1 Pseudo-Relevance Feedback

En aquesta pràctica l'objectiu primordial serà implementar un mecanisme de "Relevance Feedback" per refinar les queries de l'usuari i també una sèrie d'experiments o mètodes per avaluar la qualitat d'aquest mecanisme. L'algorisme escollit per aquesta fita serà la repetició iterada de la llei de Rocchio.

$$\mathbf{q} = \alpha \cdot \mathbf{q}_0 + \beta \cdot \frac{1}{|D_r|} \sum_{\mathbf{d} \in D_r} \mathbf{d} - \gamma \cdot \frac{1}{|D_{nr}|} \sum_{\mathbf{d} \in D_{nr}} \mathbf{d} \tag{1}$$

Per a facilitar la implementació i evitar preguntar a l'usuari que doni la seva opinió dels documents donats, farem una sèrie d'assumpcions inicials:

- Com no hi ha manera de conèixer el conjunt de documents desitjats $\mathbf{D_r}$, caldrà assolir que els $\mathbf{K}$ documents més rellevants són realment els $\mathbf{K}$ més rellevants per la query $\mathbf{q}$.
- Com tampoc hi haurá manera de saber quins documents no són desitjats, evitarem usar la tercera part de l'equació relacionada amb el terme γ
- Sabem per resultats empírics que els guanys després de les poques iteracions són minúsculs, aleshores $\mathbf{N}_{rounds}$ serà un nombre finit i petit.

Com a petit detall, analitzem el cost computacional de la fusió de k vectors utilitzant dues estructures de dades diferents. Amb llistes ordenades, l'enfocament eficient és una fusió en forma d'arbre, amb una complexitat O(n log k), on n és el nombre total d'elements. Amb diccionaris (taules de dispersió), la fusió té una complexitat O(n), ja que es recorre cada element només una vegada. En resum, la fusió amb diccionaris és més eficient en termes de temps computacional.

2 Experimentació

Per l'experimentació hem fet ús dels índexs generats a partir de novel·les (news) i d'articles científics. Hi ha un conjunt de 5 variables amb les quals

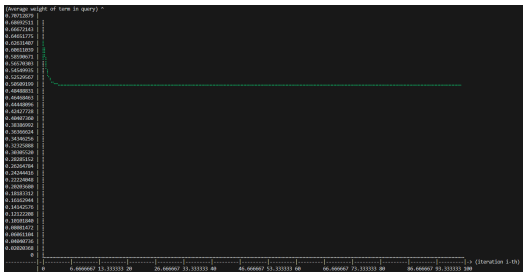
podem experimentar per optimitzar els resultats del mecanisme que hem ideat:

- $\mathbf{N}_{rounds}$: nombre d'iteracions de la llei de Rocchio
- **R**: nombre de termes més rellevants de cada document a incorporar en la nova query
- **k**: nombre de documents més rellevants per a qualsevol cerca
- α, β: les ponderacions de la llei de Rocchio

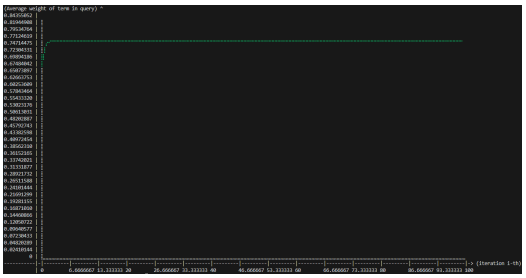
Distingirem l'estudi en dues seccions ben diferenciades per tractar d'avaluar des de tots els angles si aquest mètode d'expansió de consultes té cap impacte significatiu. Per a fer un estudi més extens i precís caldria conèixer a la perfecció el conjunt de documents, aleshores podríem usar mètriques com precisió, recall, F1-score, NDCG, Mean Average Precision... Amb les que podríem comparar fàcilment les diverses models de cerca, com el tf-idf sense rocchio vs amb rocchio i veure si hi ha millores significatives (Podríem també jugar amb els paràmetres per veure com és de millor).

2.1 Estudi de N_{rounds}

Hem volgut investigar aquest paràmetre ja que augmenta molt la complexitat temporal de l'algorisme i tenim la sospita de que no són necessàries moltes iteracions per produir l'extensió de la query. Fixant els paràmetres $\alpha = 2, \beta = 1, k = 10, R = 5$:



(a) "Detroit"



(b) "Ohio"

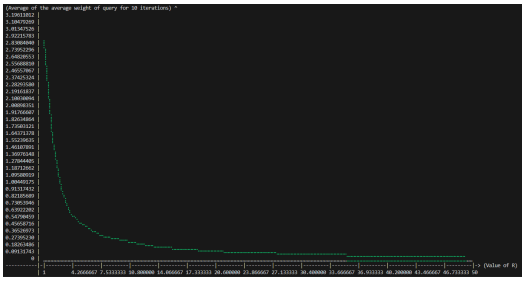
L'experimentació anterior ha estat feta per a 10 queries diferents i en totes s'observa un comportament idèntic, augmenti o disminueixi, la query pateix

una variació de pesos dels seus termes (en mitjana) en les primeres iteracions i després s'estabilitza. Aleshores té sentit fer servir valors de $N_{rounds} \leq 10$, fins i tot $N_{rounds} \leq 5$.

2.2 Estudi de R

Primerament, per analitzar l'efecte de R en el pes mitjà de les queries, hem fixat els paràmetres $\alpha = 2, \beta = 1, k = 10, N_{rounds} = 10$. Això ens serà útil per veure en mitjana com varia aquest pes promig de les paraules que conformen la query estesa.

(a) "Nintendo"

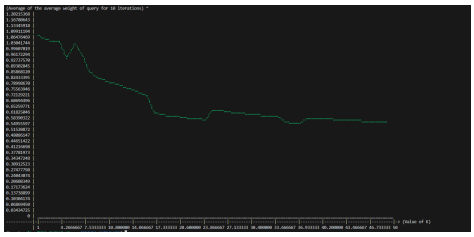


(b) "Nintendo" and "Gameboy"

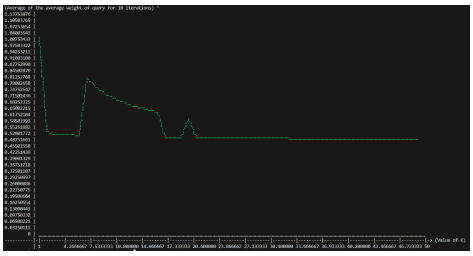
S'ha dut a terme aquesta prova per a 10 queries diferents (la que s'observa al gràfic es "nintendo" and "gameboy")i per a totes hem observat un comportament idèntic. El pes promig comença a baixar en picat, que és provocat per la contínua expansió de la query (Evidentment, nous termes menys rellevants tindran menys pes i farà baixar el promig). Però és que a més el pes dels termes que si són rellevants es cada cop menor a causa de la normalització. També ens hem adonat que el nombre de documents trobat disminuïa per valors de R molt grossos. Aleshores, com el nostre objectiu no és pas prioritzar el "recall" sinó la "precision", ens interessaria emprar valors petits de $R \in \{1, 2, ..., 5\}$. Aquesta troballa ens permetria millorar la implementació inicial i encomptes de quedarnos amb les R paraules més importants hauríem de obtenir les $|query_0| + R$ més rellevants per assegurar-se de un mínim d'expansió.

2.3 Estudi de k

Com podem imaginar, el paràmetre k, que representa el nombre de documents més rellevants a tenir en compte, té una estreta relació amb R. Amb paràmetres fixats $\alpha = 2, \beta = 1, R = 5, N_{rounds} = 10$ hem obtingut els seguents resultats per a dues queries diferents:



(a) "Japan"

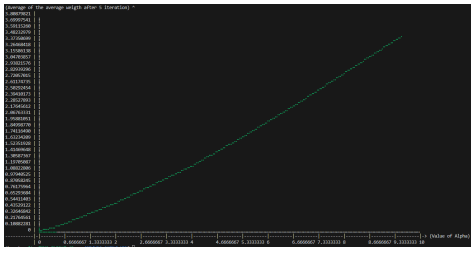


(b) "Italy"

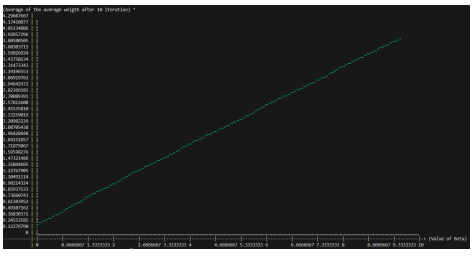
Tenint en compte que R = 5, és fàcil adonar-se que k més gran explora més documents (Quan es produeix un pic en el gràfic cap amunt, és perquè ha escollit alguns nous termes amb més pes que els anteriors possiblement) i aleshores tindrem un recall més gran, però també baixa la precisió i en el sentit pràctic, l'usuari te molts més resultats que consultar, augmentant la complexitat de la cerca.

2.4 Estudi de α i β

Finalment arribem als 2 paràmetres més rellevants, els coeficients de Rocchio. Primerament, hem volgut investigar com varia la mitjana de pesos quan varien els paràmetres β, α. Fixem els paràmetres per defecte $\alpha = 1, \beta = 1, k = 10, N_{rounds} = 10, R = 5$:



(a) Δw_i en funció d'α. q: "war"



(b) Δw_i en funció d'β. q: "peace"

És fàcil deduir que a mesura que ambdós constants augmenten, també ho fa la mitjana de manera lineal. Cal fixar-se en el fet que el que és important és la ràtio entre $\frac{\alpha}{\beta}$ i no tant els valors absoluts. La diferència rau en el pes que s'atribueix a les paraules originals i les noves (en cas que $\frac{\alpha}{\beta} > 1$, molta fins i tot al punt de donar molt poc pes a nous termes; en cas de $\frac{\alpha}{\beta} < 1$, molt poca, fins i tot al punt que poden desaparèixer les paraules originals). Per evitar l'escenari anterior, és evident que necessitarem que aquesta ràtio no sigui petita i, a més, $\alpha > \beta$, ja que ens interessa expandir amb nous termes sense perdre precisió pel camí.

2.5 Estudi qualitatiu

Si provem amb una query qualsevol com "Nintendo", amb els paràmetres $\alpha = 2, \beta = 1, k = 5, N_{rounds} = 10, R = 4$ obtenim els següents resultats:

$$[nintendo^{0.707}, tetri^{0.444}, castlevania^{0.515}, game^{0.206}]$$

La query original ha estat millorada, amb termes relacionats i que solen estar presents en articles de Nintendo, millorant la similitud "SCORE" de la query original amb els documents, però a cost de afegir paraules que potser no tenien res a veure amb el que l'usuari vol veure. Si modifiquem $\alpha = 0.5$:

$$[nintendo^{0.240}, nemesi^{0.280}, castlevania^{0.298}, tetri^{0.257}]$$

Com bé es pot apreciar ara , la query original te menys importància i fins i tot els nous termes arriben quasi a superar-ho. Si probem amb un nombre d'iteracions $N_{rounds} = 100$, restablint l'alfa a 2:

$$[tetri^{0.258}, nintendo^{0.240}, castlevania^{0.298}, nemesi^{0.281}]$$

Com era d'esperar els pesos no han variat molt, ja que amb 10 iteracions casi havien arribat al punt de convergència observat en el estudi de N_{rounds}. Ara seria convenient modificar el paràmetre k per observar l'impacte que te sobre els pesos si l'augmentem o el disminuim. Probarem $k = 2$ i $k = 20$:

$$k = 2[epilepsi^{0.214}, nintendo^{1.306}, supermario^{0.217}, zapper^{0.217}]$$

$k = 20[nintendo^{1.306}, castlevania^{0.101}, super^{0.088}, game^{0.116}]$

Es pot apreciar que amb una k menor, la query estesa està conformada per termes diferents dels quals teníem originalment i a més a més com a observació addicional aquesta query després de la primera iteració es incapaç de trobar documents, ja que els termes més rellevants que la conformen són molt restrictius. En el cas de $k = 20$ observem que ha aconseguit termes més generals com game i que ha donat menys pes a aquests termes que no pas a la propia query original, la qual atribueix molta importància. D'aquí es pot extreure la conclusió que explora més amb valors més grans i els resultats són més coherents. Llavors és preferent augmentar el valor inicial de $k = 5$ que no pas disminuir-lo. Pel següent experiment provarem de posar un valor major de $R = 8$, restablint els altres paràmetres a la configuració base. No tindria sentit probar R's menors propers a 1, ja que aleshores a penes s'expandirà la query i a penes explorarà:

$[game^{0.089}, epilepsi^{0.083}, gameboy^{0.076}, nintendo^{1.249}, tetri^{0.091}, zapper^{0.084}, supermario^{0.084}, castlevania^{0.105}]$

Per la incorporació de tants nous termes la query s'ha tornat massa restrictiva i a penes dona espai a trobar nous documents, ràpidament s'exhaureixen. Caldria augmentar k proporcionalment amb R per evitar això, encara que en algun punt segur que s'exhaureixen els resultats quan R és prou gran. Finalment, farem una petita prova més, pel paràmetre $\beta = 3$:

$[nintendo^{0.402}, tetri^{0.432}, nemesi^{0.471}, castlevania^{0.500}]$

Evidentment, dona resultats similars que quan vam provar de baixar l'alfa, proporcionalment els nous termes guanyen més rellevància que la query original. Efectivament com he esmentat anteriorment la ràtio $\frac{\alpha}{\beta}$ és més rellevant que no pas els valors de les constants.

3 Conclusions i Reptes

Els únics problemes que vaig afrontar van ser, com sempre, la manca de temps i un error d'implementació quan s'usaven majúscules en queries, arreglat fàcilment passant a lowercase tota l'entrada. Però sense dubte el repte més gran va ser pensar com plasmaria els resultats i treure unes conclusions

significatives. Vaig pensar que analitzant tots els paràmetres un a un, podria fer una anàlisi més quantitativa i finalment una qualitativa basant-me en l'opinió personal (a pesar del possible esbiaix que això pot comportar). Si tingués més temps faria jo el meu propi conjunt de dades on llavors podria tenir coneixement del conjunt de documents rellevants donada una consulta en concret.

L'anàlisi quantitativa podria haver sigut multivariable i podria haver considerat altres mesures més, com distància de Levenshtein per comparar resultats i fins i tot es podria haver aprofundit calculant correlacions. Un altre cop l'enemic ha estat la manca de temps i recursos.

Les conclusions dels diversos paràmetres es pot trobar a la secció anterior, però de manera molt resumida podem recalcar diverses troballes: Es més rellevant la ràtio dels 2 coeficients α, β que no pas els valors, però sempre ens interessaran valors $\alpha\ \beta$, probablement 2 cops majors serà suficient per obtenir resultats decents. De lo contrari els nous termes poden arribar a tenir molt pes, fins a fer desaparèixer la query incial i donar resultats que no tinguin cap mena de sentit (Exemple: troba documents de la Generalitat en cercar: "Pokémon"). També ens adonem que els paràmetres k i R permeten expandir l'exploració que es realitza, sobretot augmentant k i hem vist que valors grans de R pot estendre massa la query i tornar-la restrictiva de manera que deixa de trobar documents. Valors grans de k exploren un subconjunt major de documents inicialment, el que fa variar molt com és la query estesa a quant la k és més petita en comparació. També observem que el nombre d'iteracions pot ser petit, ja que la variació de pesos a les poques iteracions passarà a ser nul·la, aleshores no es tan costos temporalment com es va imaginar inicialment.

Una conclusió lògica després de l'experimentació de l'aplicació de la llei de Rocchio, es que aquest mètode pot empitjorar o millorar força la query original, llavors cal conèixer les diferents variables que entren en joc i aplicar els valors més adients dins dels rangs esmentats, per tal d'aconseguir millorar la precisió sense arruïnar el recall. Però concluim que val la pena, ja que la queries esteses, amb els paràmetres adients es capaç d'oferir millors resultats i amb més sentit que no pas la query original, limitada al coneixement/desconeixement de l'usuari del cercador.