Pràctica de Programació Funcional + Orientada a Objectes
Similitud entre documents (Part II)

Programació Declarativa, Aplicacions.

October 25, 2024

Introducció

Us proposem fer un recomanador basat en l'algoritme PageRank (`PR`) proposat per Google i aplicar-ho amb un subconjunt de la Vikipèdia. L'algoritme consisteix a assignar una puntuació a les pàgines web de manera que les pàgines amb puntuacions més altes són considerades més importants. El recomanador ha de ser capaç de llistar les pàgines per ordre d'importància donada una *query* no buida però de longitud indeterminada, com ara: "Guerra" o "segona guerra".

Per entendre com funciona el `PR` suposem que tenim quatre documents de la Vikipèdia: A, B, C i D. Els enllaços d'una pàgina cap a ella mateixa s'ignoren. Els enllaços múltiples d'una pàgina a una altra es tracten com un sol enllaç. El `PR` s'inicialitza amb el mateix valor per a totes les pàgines i la suma total del `PR` de totes les pàgines es 1.0. Així, el valor inicial de cada pàgina en aquest exemple és 0.25.

Ara suposem que la pàgina B té un enllaç a les pàgines C i A, la pàgina C té un enllaç a la pàgina A, i la pàgina D té enllaços a les tres pàgines.

$$[(B,\ [C, A]),\ (C,\ [A]),\ (D,\ [A,\ B, C])]$$

Així, en la primera iteració, la pàgina B transferirà la meitat del seu valor actual (0.125) a la pàgina A i l'altra meitat (0.125) a la pàgina C. La pàgina C transferirà tot el seu valor actual (0.25) a la pàgina A. Com que D té tres enllaços a altres pàgines, transferirà un terç del seu valor actual, aproximadament 0.083 a A. Finalment, la pàgina A tindrà un `PR` d'aproximadament 0.458. Fixeu-vos que el `PR` d'una pagina es igual a la suma dels `PR` de les pagines que el referencien dividit entre el nombre de pagines que referencien, L:

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}$$

En el cas general, el valor de `PR` per a qualsevol pàgina p_i es pot expressar com:

$$PR(p_i) = \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)},$$

on $M(p_i)$ és el conjunt que conté totes les pàgines que enllacen a la pàgina p_i.

Intuïtivament, el `PR` calcula la importància d'una pàgina en funció de les referències que rep i de la rellevància de les pàgines que la referencien (COMPTE: més endavant expliquem com es poden extreure les referències d'un document de la Viquipèdia). El càlcul del `PR` es basa en la idea que els usuaris fan clics aleatoris en els enllaços de les pàgines que visiten, però eventualment es cansen i surten del sistema. La teoria del `PR` afirma que un navegant que fa clics aleatoris té, en qualsevol moment, una probabilitat de deixar de fer-ho amb un factor de frenada, *d*. La probabilitat que, en lloc de seguir navegant, salti a una pàgina aleatòria és *1 - d*. Diversos estudis han provat diferents valors per aquest factor, però habitualment es fixa al voltant de 0,85. Així doncs, per calcular el `PR`, tindrem en compte el factor de frenada d mitjançant un algoritme de punt fix:

- En l'instant $t = 0$, es pressuposa una distribució inicial de probabilitats:

$$PR(p_i; 0) = \frac{1}{N},$$

  on N és el nombre total de pàgines, i $p_i; 0$ és la pàgina i en el moment 0.

- A cada pas de temps, el càlcul dóna com a resultat:

$$PR(p_i; t+1) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j; t)}{L(p_j)},$$

- El procés continua fins que:

$$|PR(t+1) - PR(t)| < \epsilon,$$

  on $\epsilon = 1e - 3$.

IMPORTANT: Podria ser que, en alguns casos, la convergència trigui massa. En aquestes situacions, seria recomanable establir un nombre màxim d'intents per aconseguir la convergència, per exemple, $steps = 5$.

I, seguint el fil de la primera part de la pràctica i tenint en compte que ja sabem una manera de comparar documents de text, ens proposem aplicar-ho en aquest entorn. Analitzar la similitud entre pàgines web ens permetria, per exemple, detectar pàgines que s'assemblen molt però que no es referencien, o bé pàgines que es referencien però que no s'assemblen gens.

Per a dur a terme algun d'aquests anàlisis caldrà:

- **Poder comparar la similitud de dues pàgines:** per a calcular la similitud entre dues pàgines farem servir la del cosinus, però enlloc de considerar els vectors de *term frequencies (tf)* (freqüències de paraules), considerarem els vectors *tf_idf* que tenen en compte quant significativa és una paraula en el corpus (conjunt de pàgines) que estem analitzant. Fer servir nomes *tf* implicaria considerar que totes les paraules són igual d'importants. En el nostre cas, per exemple, si considerem pàgines del conjunt de pàgines

que se us proporciona, la paraula "guerra" apareixerà a tots els documents ja que hem filtrat les viquis que contenen "segona guerra mundial".

L'*inverse document frequency* (*idf*) d'un terme t en una col·lecció de documents C és: $idf = log\frac{|C|}{df}$ on $|C|$ és el nombre de documents de C i df és el nombre de documents on apareix el terme t. Fixeu-vos que quant en més documents apareixi, menys rellevant serà un terme per a fer comparacions.

Ara ja podem definir *tf_idf* d'un terme t a un document d dins una col·lecció de documents C com a: $tf\_idf = tf \times idf$ on tf és el nombre de vegades que apareix t al document d i idf és l'inverse document frequency de t per la col·lecció de documents C.

Compte, per a la similitud i per al càlcul dels *tf_idf* considerarem només les paraules que hi hagi a dins dels tags `<text>...</text>` eliminant les stop-words. **Cal eliminar més coses? links?**

- **Identificar les parts de les pàgines que ens són rellevants com ara el títol i el text.** L'aspecte d'un document XML de la Viquipèdia és el següent (en concret el primer de la nostra col·lecció, el `22.xml`):

```
<page>
    <title>Atletisme</title>
    <ns>0</ns>
    <id>22</id>
    <revision>
      <id>14061284</id>
      <parentid>13842377</parentid>
      <timestamp>2014-09-21T14:05:43Z</timestamp>
      <contributor>
        <username>Walden69</username>
        <id>1809</id>
      </contributor>
      <comment>/* Vegeu tambe */</comment>
      <text>{{millorar referencies|data=desembre de 2012}}
[[Fitxer:Athletics competitions.jpg|thumb|400px|Diferents competicions atletiques.]]

El mot atletisme prove de la paraula grega athlon que significa
lluita, competencia, combat o similars.
                        ...
{{ORDENA:Atletisme}}
[[Categoria:Atletisme| ]]
[[Categoria:Articles bons d'esport]]
[[Categoria:Articles bons dels 1000]]</text>
      <sha1>pmodgj07j924qtlj1xz7r7zcns5hg4e</sha1>
      <model>wikitext</model>
```

```
      <format>text/x-wiki</format>
    </revision>
  </page>
```

Ens convé saber identificar dues parts:

- El títol, que està entre els *tags* `<title>`*títol pàgina*`</title>`, i
- el contingut de la pàgina, que està entre els tags `<text>`*contingut*`</text>`.

Necessitarem identificar **com es referencien les pàgines de la Vikipèdia** en general. El mecanisme és utilitzar el títol de la pàgina referenciada entre claus: `[[`*títol pàgina referenciada*`]]`. De totes maneres hi ha molts detalls a considerar:

- quan volem que es vegi un text però que es referencï una pàgina en concret s'afegeix la barra `|` com a separadora, e.g. `[[Bombardeig de Guernica|Guernica]]`, enllaçarà amb la pàgina del bombardeig de Guernica però mostrarà la paraula Guernica com a enllaç.
- Per referenciar una secció de la mateixa pàgina, es fa servir `[[#`*secció*`]]`. També es pot referenciar una secció d'una altra pàgina: `[[`*títol pàgina*`#`*secció*`]]`.
- També hi ha enllaços a pàgines encara no definides: e.g. `[[MG 151 # MG 151/20|MG 151/20]]`. Evidentment no ens interessen.
- Quan es fa referència a un fitxer, com ara una imatge, es fa amb `[[Fitxer:`*nom_del_fitxer i opcions*`]]`. Per exemple `[[Fitxer:Bomber stream.jpg|thumb|300px|right|Part d'un transport ...]]`. Aquestes referències òbviament no les haurem de considerar ja que no referencien altres pàgines.

Hi ha molts més detalls que podeu consultar a: `http://en.wikipedia.org/wiki/Wiki_markup`

Què s'ha de fer?

En concret el que s'us demana és el següent:

1. Calculeu el nombre promig de referències que tenen totes les pàgines.
2. Donada una *query*, fer una recomanació fent ús de l'algoritme `PR`. COMPTE: el càlcul del `PR` no s'ha de fer per a tots els documents, sinó només tenint en compte els documents que continguin la *query* (no es diferenciarà entre majúscules i minúscules).
3. Detectar les pàgines que s'assemblen més però no es referencien mútuament. Podeu considerar que dues pàgines s'assemblen molt si la similitud *tf_idf* (considerant com a corpus com a molt les 100 pàgines més rellevants d'una recomenació feta pel `PR`) és superior a 0.50.

4. Cal que definiu una funció `timeMeasurement[A](···)` que rebi una expressió a calcular i retorni el seu resultat i el temps que ha trigat en fer-lo. Pista: cal que l'expressió no s'avalui en el pas de paràmetres sinó que s'avalui "dins" `timeMeasurement` per tal de poder calcular el temps que ha trigat...

5. Cal que el `MapReduce` el parametritzeu amb un cert nombre de mappers i de reducers. Mostreu el temps que ha trigat el vostre procés sencer considerant 1, 4, 10 i 20 actors. És eficient provar amb més actors? fins quants?

6. Cal definir una funció `MR` que abstragui l'ús del `MapReduce` que hàgiu modificat.

7. **Cal fer ús de `MapReduce` allà on hi veieu que hi pugui ajudar.**

8. Tancar correctament el sistema d'actors (OPCIONAL).

Pistes

- A la modificació del `MapReduce` per tenir un cert nombre de mappers i reducers, us podeu inspirar en com es repartia el comptatge de primers entre els mappers i els reducers a l'exemple vist a classe.

- L'input dels `MapReduce` que han de treballar amb els fitxers hauria de ser una llista dels noms dels fitxers, de manera que les funcions de mapping ja fessin la feina de llegir i processar-los.

- És important que, per fer la recomanació de les pàgines més importants donada una *query*, treballeu amb els documents on, dins del corpus, aparegui aquesta *query*. Per aconseguir-ho, podeu fer ús del mètode `ngrames` que us vam fer implementar a la primera part de la pràctica.

- La recomanació de les **100 pàgines** més rellevants i la **similitud de 0.5** són uns mínims. Com més *queries* complexes i més fitxers pugueu tractar, millor. Podríeu arribar a fer *queries* que apareguin en totes les pàgines?

Lliurament

- El lliurament obligatori te com a data límit el 8 de desembre

- Les pràctiques es poden fer en equips de dos

- Cal lliurar un link i convidar al professor a un repository PRIVAT de GitHub amb el codi ben documentat

- Caldrà lliurar també un document on hi hagi:

  – extractes comentats del codi mostrant les principals funcions d'ordre superior usades (sobretot per la primera part), així com la part de la modificació del `MapReduce`

  - jocs de proves dels resultats obtinguts tant de la primera part com de la segona (llistat de les més importants amb nombre de referències i llistat de pàgines similars sense referències...)
  - com s'han fet tots els `MapReduce` (input, mapping i reducing)
  - quines classes us ha convingut crear i per què
  - taula de rendiment segons nombre d'actors en el `MapReduce`. Cal que especifiqueu la màquina feta servir
  - altres consideracions que volgueu ressaltar (com ara fins a quants documents heu pogut tractar)

- Es farà una presentació de la pràctica presencial/virtual al professor amb els dos membres de l'equip de la pràctica