DESIGNING A SEARCH SYSTEM
Lab Session 7

Cristina Aguilera, Jesús Antoñanzas, Alex Carrillo
Cerca i Anàlisi de la Informació, Ciència i Enginyeria de Dades (UPC)

November 18, 2019

Contents

1 Introducció

Proposem un sistema de cerca, emmagatzematge i recomanació de publicacions científiques. Aquest sistema recopila informació dels *hubs* científics més importants, creant així un directori des d'on accedir a qualsevol document, autor o tòpic. A més, recopilem diferents mètriques com els tòpics importants a cert moment, relacions entre ells i hem fet l'entorn *user-friendly* de tal manera que usuaris no especialitzats poden adquirir informació sobre nous descobriments, resums de diferents temes, curiositats o fets destacables.

2 Funcionalitats

El nostre sistema de cerca inclourà diferents funcionalitats que es repartiran entre els següents apartats:

- **Perfils d'autors**. Els usuaris que siguin investigadors científics podran crear els seus propis perfils. A més, es crearan automàticament perfils per aquells autors que no l'hagin creat ells mateixos a partir que el nostre sistema reconeixi més de dos papers escrits per ell. La informació s'extraurà segons els nostres algorismes i per tant, la informació que es reculli pot no ser del tot exacta. Els camps que s'inclouran són els següents:

    1. Dades personals. Nom i cognoms de l'autor, una foto personal, centre de recerca i/o universitat en els quals treballa, pàgina web personal.
    2. Dades relacionades amb el seu treball. Llista de temes en els que treballa l'autor, a partir de conèixer els autors als que ell més ha referenciat i els que més l'han referenciat a ell (influenciades per la tasca realitzada pel pagerank) podrem extreure quins són els autors més relacionats amb la seva feina, llista d'autors amb els que ha treballat, tots els papers en els que ha treballat i en els que ha estat co-autor, diverses estadístiques d'influència (número de publicacions cada any, de quin tipus, quants cops ha estat citat...).

- **Pàgina d'inici**. En accedir al cercador, el primer que ens apareixerà serà una pàgina en el que trobarem informacions varies.

    1. Trending topics. Apareixeran llistats els temes científics i els papers que més gent està buscant en un moment en concret.
    2. Fets destacats d'un dia. Es destacaran aquells papers que es van publicar el mateix dia que l'usuari està fent la cerca i van ser molt importants.
    3. Paper destacat. Cada dia es referenciarà un paper que està sent important pels científics en l'actualitat.

- **Buscador de papers**.

    1. Filtratge de documents per temes. El sistema de cerca inclourà la possibilitat de cercar per diferents temes científics i veure quins són els paper més rellevants de cadascuns d'ells o veure'ls ordenats per data de publicació. També permetrà buscar més d'un tema en la mateixa cerca i realitzarà la intersecció del documents que tracten els dos.
    2. Filtratge d'autors per temes. De manera semblant al que hem explicat anteriorment, serà possible trobar els autors que treballen en un cert camp o tema.
    3. Informació clau. Un cop feta la cerca i retornats els resultats, es mostrarà un petit text extret del paper relacionat amb la cerca que s'ha realitzat. Per això, caldrà fer un *scraping* per extreure les cites i referències.

- **Sistema recomanador**. Es tindrà un apartat de recomanacions en el que cada usuari podrà veure un petit nombre de papers en funció dels seus gustos. Es recomanaran documents relacionats amb els temes de recerca sobre els quals estigui investigant actualment, però tenint en compte que no sempre buscarà el mateix tipus de documents sinó que els temes poden anar variant i el sistema doncs haurà d'estar al dia. També es voldrà que el sistema recomanador no sigui sempre determinista i per tant voldrem que algun cop recomani papers que no estiguin del tot relacionats amb les seves cerques, de manera que l'usuari pugui descobrir nous camps de recerca que li resultin interessants.

3 Arquitectura d'alt nivell

- **URL server**. Zona d'emmagatzematge de diferents enllaços de pàgines web. Es farà servir una estructura de dades del tipus cua o *queue*. Al principi contindrà els enllaços de les pàgines web que posem com a llavors, és a dir, aquelles des de les quals començarem a analitzar amb els crawlers. Un cop inicialitzat aquest procés, que és offline com explicarem a continuació, tornarem a introduir a la cua de l'URL server aquells enllaços que encara no haguem visitat retornats pel procés **URL resolver** per tornar a iterar el procés fins que ens quedem sense pàgines a visitar o ens allunyem massa del punt d'origen, ja que començariem a trobar webs que no fossin d'utilitat pel nostre projecte. En resum, com que aquest mòdul va plenament lligat al procés de *crawl* es farà servir bàsicament en tasques offline.

- **Crawlers**. Utilitzem diferents procesos offline de *crawl* per tal que el temps d'execució sigui factible. Aquests seràn robusts, eficients i *polite*.

  Com volem que el nostre motor de cerca sigui un repositori de la màxima quantitat de publicacions científiques diferents i sabem que aquestes no estan publicades de manera molt dispersa, és a dir, estaran en pàgines importants científiques, perfils d'investigadors i a pocs llocs més, predeterminem unes pàgines llavor a mà. A més de les pàgines llavor, tenim el procés ***URL resolver*** que a més d'alimentar els diferents *crawlers* amb URL's per visitar, evitant les repeticions, és capaç d'identificar si aquests pertanyen a dominis d'interès o si ho són en sí. Per tant, això permet al sistema controlar que els *crawlers* només analitzin pàgines d'interés. A més, visitaran pàgines importants periòdicament per tal de mantenir el sistema al dia.

  Els dominis predefinits poden ser, per exemple, `https://arxiv.org/` o `https://scholar.google.com/`.

  Cada *crawler* analitzarà la pàgina en la que està, enviant al procés ***store server*** el *doc id* del URL i el corresponent *raw text*. Si aquesta pàgina conté HTML, extreure el *raw text* és fàcil. Si l'URL, en canvi, conté un arxiu PDF molt probablement aquest serà una publicació. Aquestes, si són recents, seràn fàcils de realitzar *parsing* sobre elles, però si són antigues i, per tant, la qualitat no serà molt alta s'hauran de tractar amb els *crawlers* mitjançant *software* d'OCR (*Optical Character Recognition*).

  Com en molts casos de *web scraping*, els *crawlers* han de saber quines pàgines no indexar. Ja hem dit que es centraran en pàgines de dominis del nostre interès, però també s'hauran de saltar pàgines duplicades (publicacions repetides d'articles) i spam. El primer problema el podríem tractar amb *k-shingles*: aquest mètode consisteix en mirar bàsicament totes els possibles conjunts de paraules consecutives de llargada k i després comparar-los entre documents, ja que si els conjunts obtinguts per cadascun són els mateixos el més probable és que haguem trobat dos documents repetits. El segon pensem que hauria de ser tractat més endavant, ja que no pensem que en la comunitat científica es trobin massa d'aquests casos.

- **Indexer**. En paral·lel a l'execució dels crawlers, el procés ***store server***, com hem explicat abans, s'encarrega d'emmagatzemar tuples de (*docid*, *raw text*) a una base de dades que anomenem ***doc repository***. D'aquesta manera, el procés ***Preprocessing*** pot anar agafant text cru de la base de dades a un ritme constant i 'dividint' els termes que són paraules dels que són enllaços en dues bases de dades: *Anchors* i *Raw texts*, respectivament. Amb aquest procés d'extracció, ***Anchors*** rep tuples (*docid*, {*links*}) i ***Raw text*** rep les paraules dels documents preprocessades (sense enllaços i amb algun tipus de *stemming* o *stopwords*).

  D'una banda, el ***URL resolver*** s'alimenta dels enllaços de *Anchors*. D'altra banda, només després que s'hagi arribat a aquest punt del preprocessat, és a dir, quan es disposi de la base de dades *Raw text*, es poden començar a indexar els documents. Per realitzar aquesta tasca, ens ajudem del sistema de ElasticSearch, el qual genera automàticament un *forward index* i es pot configurar perquè emmagatzemi els termes en un *Inverted Index* com a base de dades, que és el que realment ens interessa pel nostre sistema per tal de poder respondre les queries de forma més ràpida i eficient.

- **Page ranker**. Es farà servir la tècnica del pagerank sensible a temes, ja que els papers es poden dividir clarament en diferents àmbits de recerca. Aquest es divideix en dues parts: la primera offline i la segona online. Per la primera part, el primer pas que haurem de fer serà establir els diferents temes, alguns dels quals poden ser: ciències de la computació, biologia i ciències naturals, ciències socials, medicina, arquitectura... A continuació, cal assignar un subconjunt de pàgines web a cada tema. Per aquest propòsit utilitzarem el tf-idf de cada documents i la cosinus similaritat entre documents. Suposarem que documents amb una similaritat alta pertanyen al mateix tema, ja que faran servir aproximadment les mateixes paraules. Per exemple, un document que parli sobre algun tema de biologia repetirà molts cops la parula virus i un altre que parli sobre computació en farà servir *software*. Aquest concepte no ha estat explicat a classe, però creiem que podria ser una bona manera d'implementar-ho. A continuació, es calcularà per cada tema el pagerank amb cadascun dels nodes. La segona part consisteix en que cada cop que l'usuari enviï una *query* el sistema internament haurà de calcular el *ranking* de puntuació per les diferents pàgines donada aquesta *query*, tenint en compte el pagerank prèviament calculat i la similaritat de la *query* amb cadascun dels temes seleccionats. Més concretament, la puntuació final assignada a cada pàgina serà el producte dels termes *pagerank sensible a temes* i *similaritat de la query de l'usuari* amb els documents del nostre índex, amb una ponderació d'ambdues parts que doni més pes al terme del pagerank (el qual donara una relevància més global sobre el que s'està buscant).

- **Searcher**. Aquest procés consisteix en agrupar tota la informació que hem creat fins ara (inverted file i page ranks) amb la *query* que afegeix l'usuari per tal d'obtenir una resposta amb les pàgines que li puguin interessar, intentant maximitzar tant la precisió com el recall d'aquesta. Per això, aplicarem un algorisme iteratiu de *pseudorelevance feedback* fent servir la regla de Rocchio. No preguntarem a l'usuari quins documents li resulten interessant, ja que la seva paciència és limitada i doncs confiarem en que els primer documents retornats són els rellevants per tal d'iterar. D'aquesta manera, la query que aplicarem al sistema serà una derivació de la inicial amb noves paraules i així, aconseguirem obtenir documents que no continguin la query inicial però que també siguin rellevants per la tasca de l'usuari (augment del recall).

- **Formatter**. El que aconseguirem amb aquest procés és combinar diferents bases de dades extretes arrel de la combinació de tots els processos explicats anteriorment, per donar un format adequat a les respostes de les diferents queries. El principal mòdul que farà servir és el **doc index** que conté el text essencial de cada document i ens permetrà mostrar un petit resum de les idees claus del paper. També es podran incloure altres informacions com els

papers al quals referencia, la data de publicació del paper, els autors del document (amb un enllaç als seus perfils) i un *snippet* amb un fragment del text en que apareix la *query* (o les paraules claus) que ha fet l'usuari. Aquesta informació s'extreu directament de les dades del HTML recuperat en els processos anteriors.

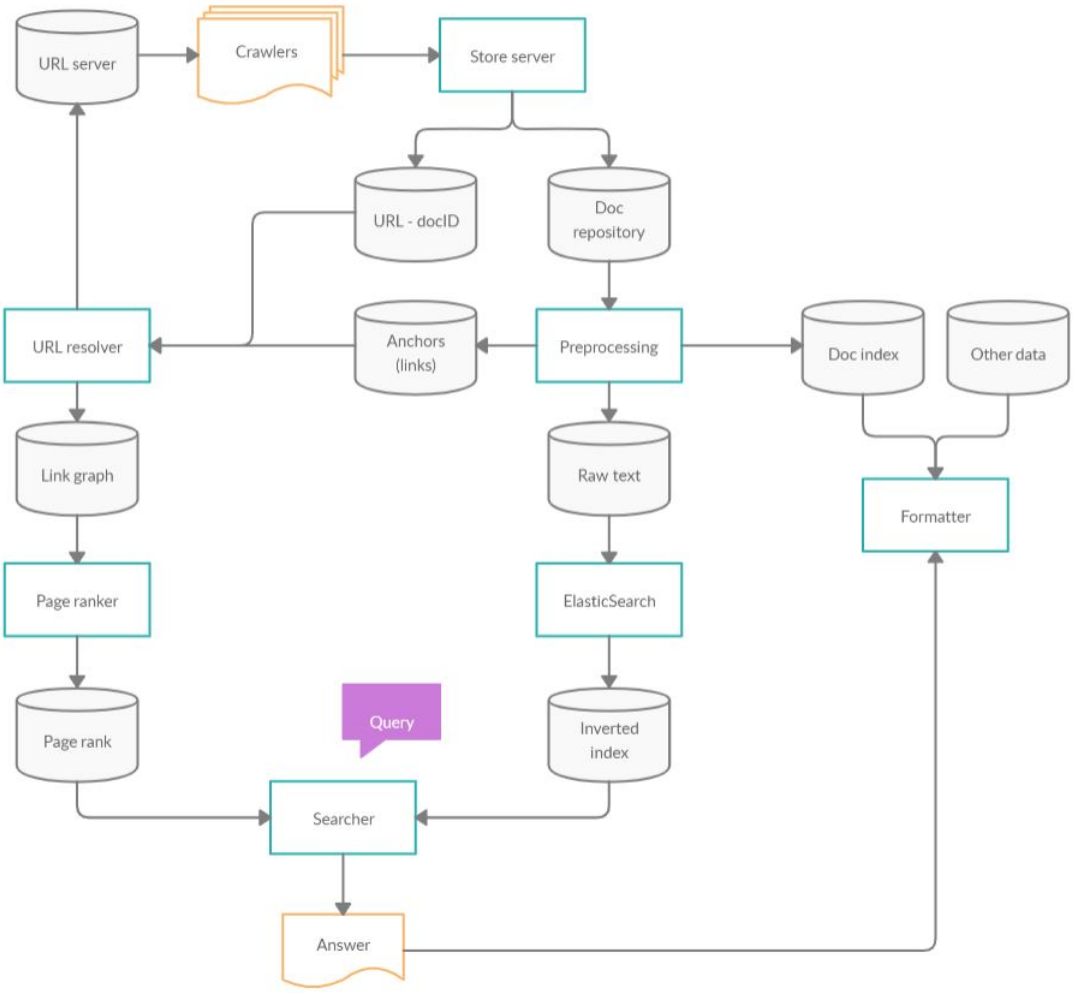

Figure 1: Diagrama de blocs amb els diferents mòduls i quins processos els uneixen.

- **On guardar les dades:** Pel tipus de dades que guardem i els objectius que tenim en ment, optem per guardar totes dades a un *data lake* de manera distribuïda, sobre el qual extraurem les dades necessàries per diferents funcionalitats. Una bona opció és l'utilització de HDFS (*Hadoop Distributed File System*), el qual permet una gran escalabilitat horitzontal.

4 Implementacions

A la introducció del nostre sistema de cerca hem mencionat diverses funcionalitats que creiem que estaria bé implementar per tal de tenir un producte diferent d'altres que trobem al mercat. Veiem com implementaríem aquestes, a partir de quines dades i amb quin objectiu.

Funcionalitats fonamentals

1. **Identificació d'autors** El sistema de *parsing* podria saber identificar quan es menciona un autor, categoritzant-los amb etiquetes. S'utilitzaria una base de dades de noms i cognoms

predeterminada la qual s'aniria actualitzant automàticament i/o amb la intervenció dels usuaris o autors a l'hora de crear-se un perfil en la plataforma.

2. **Autoria de publicacions** Cada publicació estarà associada al set d'autors corresponents a través d'un *logbook* que pot servir per llançar *queries* del tipus: `authors of Introduction to Kernel Methods`, és a dir, trobar autors a partir d'una publicació mitjançant preguntes del llenguatge natural.

3. **Creació de perfils** Si el sistema troba autors recurrents, es podria crear un perfil automàticament mitjançant el mateix plantejament que en l'apartat d'identificació d'autors: amb el sistema de detecció del nostre sistema o amb la creació manual de perfils.

4. **Sistema de referències** Al saber quins son els autors de cada publicació i tenir perfils creats, es pot referenciar als perfils d'aquests quan es miri la publicació. També es pot mirar publicacions del mateix autor, etc. En altres paraules, el *logbook* que s'utilitza també inclou la informació relativa a les referències de cada autor per a una publicació donada.

5. **Detecció d'aliances, *spam* i duplicats** Els autors fraudulents que facin *cross-referencing* per tal de guanyar popularitat es detectaran al graf de hyperlinks: són comunitats tancades. Es marcaran aquests autors com a 'possiblement fraudulents', així com les seves publicacions associades. Els papers duplicats es detectaran amb la implementació de l'algoritme de *Locality Sensitive Hashing*.

Funcionalitats addicionals

1. **Etiquetat de perfils per temes** Havent etiquetat cada publicació per tòpic segons el context de les paraules que hi apareixen, és inmediat proposar una cerca per tòpics de l'estil `publications about Machine Learning`.

2. **Publicacions del dia** Amb l'ajuda del *logbook* proposat anteriorment i de la informació relativa a la data de publicació de cada paper, es pot mostrar un parell de publicacions destacades que van sortir el mateix dia que la data d'accés a la plataforma.

3. **Recomanació** Es basarà en una combinació de recomanacions *user-to-user* i *item-to-item*. Es calcularan els *nearest neighbours* amb lsh per reforçar l'eficiència per a cada usuari en funció del papers que ha llegit i basat en això es crearan les prediccions per cada item per calcular quin d'ells li pot resultar interessant.

4. **Fils de discussió** Per a cada publicació, es pot tenir una secció on l'autor pot respondre públicament a investigadors interessats en ella a mode de fòrum o safata de comentaris.

5. **Visualització de la popularitat** L'ús del *pagerank* permet saber quines són les publicacions més referenciades. Voldríem poder veure la popularitat, doncs, de publicacions en cert moment. Això creiem que es pot fer analitzant quines tenen un increment major en el nombre de referències en una finestra de temps de longitud n.

6. **Ranking de perfils (h-score)** Cada autor té una puntuació. La *h-score* és un sistema que combina la productivitat de cada autor amb l'impacte de les seves publicacions. El primer es pot mesurar identificant l'any de les publicacions, i el segon mirant el numero de mencions a la publicació.

5 Sobre les limitacions

Portar a terme i mantenir la implementació d'un sistema de cerca requereix molts recursos, tant computacionals com humans.

5.1 Tècniques

1. Computacionals
   Les possibles dificultats que ens trobaríem a causa dels recursos computacionals serien causades per diverses coses. Seguint l'ordre natural de la implementació del sistema, primer de tot ens trobem com explorar tantes publicacions i pàgines, cosa que hem dit que faríem mitjançant la distribució de *crawlers* en diferents màquines. Seguidament, s'hauran de processar els documents, i fer *scraping* de les pàgines web. Aquest procés es pot fer perfectament de forma distribuïda, com els *crawlers*.

   No obstant, cal destacar que el preprocessament del text present en cadascun dels documents o webs és una tasca que es pot convertir en el coll d'ampolla del sistema. No per la tasca en sí, sinó pel conjunt de documents que necessiten ser processats usant software OCR. Aquest tipus d'eines augmenten el temps de processament i són, en general, més lentes que el simple *parsing* de les paraules.

   Un altre punt és la indexació dels papers la qual, mitjançant la distribució dels documents en diferents clusters de la mateixa manera que feiem amb els *crawlers*, no suposa un problema. Això és degut al fet que l'ús del sistema ElasticSearch facilita la tasca i garantitza, a grans trets, que tant el *forward index* com el *Inverted Index* (el que utilitzem) siguin el més eficient possible.

   Per últim, cal notar quin és el verdader impacte de les consultes demanades pels usuaris en quant a cost computacional. Tal i com es pot veure en l'esquema de disseny del sistema i en l'explicació del pagerank, cada cop que l'usuari enviï una *query* el sistema haurà de calcular internament el *ranking* de puntuació per les diferents pàgines donada aquesta *query*. Aquesta tasca es realitzaria en un cluster concret, deixant altres clusters lliures per l'accés a altres usuaris. És clar que, si el nombre de consultes d'usuaris diferents en un instant és molt gran (en general, més gran que el nombre de clusters disponibles), això pot provocar una cua d'espera entre els usuaris per rebre una resposta a la seva *query* o, fins i tot, pot generar una fallida del sistema.

2. Emmagatzematge
   Un cop fet el *scraping*, hem de guardar els papers indexats. Actualment hi han, aproximadament de 7 a 8 milions de personal d'investigació, i es publiquen més de 3 milions d'articles cada any [1] i el volum va en augment. Aquest nombre, encara que no sempre ha sigut així d'alt, comporta un repte significatiu, ja que implica que haurem d'analitzar, indexar i emmagatzemar de l'ordre de decenes de milions de documents.

Ens agradaria comentar que, encara que és veritat que es necessitin molts recursos computacionals, el fet de que el nostre sistema de cerca estigui centrat en un subconjunt de la web fa que el nostre problema sigui bastant més accessible que si fóssim a oferir un sistema de cerca d'Internet sencer. Per tant, i com no necessitem que els processos *offline* es facin de manera molt ràpida estimem que amb una quantitat de l'ordre de 100 *commodity class* PC's serà suficient.

5.2 Legals

Encara que indexem les publicacions, no podem publicar cap sense el consentiment dels autors. De fet, cal tenir present que hi poden haver certs documents publicats a la web que disposin d'una llicència que expressa explícitament la no difusió del propi document. Per tant, s'han d'evitar per tal d'acatar la llei vigent.

[1] 2018 - An overview of scientific and scholarly journal publishing, R. Johnson et. al. STN.

Es per aquesta raó que només indexem per tal de retornar els resultats de *queries* amb el corresponent *link* a la pàgina web on es troba la publicació original. A més, molts cops, publicacions de revistes de subscripció es publiquen de forma il·legítima a altres portals. Per evitar problemes legals, doncs, nosaltres no alberguem publicacions ni les emmagatzemem en cap moment, sinó que referenciem al domini web (enllaç URL) en el qual el/els propietari/s les van publicar originalment.

5.3 Aspectes d'emprenedoria

El mercat dels portals de coneixement, específicament, de publicacions científiques ja n'és un bastant treballat. Podem trobar molts d'ells bastant establerts (una senzilla cerca a Google ens dóna exemples com CORE, ScienceOpen, DOAJ, ERIC, Google Scholar, Arxiv i molts més). A més, encara que sempre hi han coses a millorar, aquest segment de mercat és un de molt específic: la popularitat d'aquestes pàgines està quasi totalment lligada al personal d'investigació. Doncs, no és gens fàcil introduir-se al mercat. Per aquestes raons, pensem que hi ha projectes amb més probabilitat d'èxit als que dedicar-hi temps i recursos.