Data Mining

Martina Evangelisti

7/11/2021

Contents

1 Search Engine

The code is divided into 3 parts:

- web crawler: file *simple_crawler.py* announcement saved in *jobs.tsv*
- index construction: file *'index_constructor.py'* that pre-processes the documents and builds the inverted index and writes it in the file *'index_file.tsv'*
- query processing part: file *'query_processing.py'*that uses the index in the file to answer the queries

To test the code you first have to launch `python3 index_constructor.py` and then `python3 query_processing.py`.

1.1 Crawler

I designed a simple crawler for downloading and parsing job advertisements in "Informatica/Grafica/Web" from the kijiji web site.
I Used the package Requests for downloading the web pages and the package Beautiful Soup to parse the HTML pages.
I inspected the web page from Chrome to retrieve the name of the classes I was interested In, retrieved the number of pages of the results and iterated on them taking the information needed.
The output is saved on the file "jobs.tsv".

1.2 Index construction

The file *'jobs.tsv'* is structured with a document per row with the following columns: title, description, location, date, url.
I read the file line per line taking only the textual fields I considered useful (title,description,location).
I preprocessed the text removing the HTML in the description, transformed all the sentences to lower case and applied stemming to the tokens.

From them I built the inverted Index, storing for each term in each posting the DocID and the term frequency.
Then I pre-computed all the lengths of the documents that will be useful later to compute cosine scores.
The tf-idf for each term,document is computed as it follows:

$$W_{t,d} = log_{10}(1 + tf_{t,d}) \cdot log_{10}(N/df_t) \tag{1}$$

Where t is the term, d is the document and $tf_{t,d}$ is the term frequency of term t in document d, N is the total number of documents, df_t is the document frequency of term t in the collection, so N/df_t is the idf, the tf-idf

score is weighted with the logarithm.
The length of the document is the square root of the sum of all his components $(W_{t,d})^2$.
I built the inverted index as a dictionary containing for each term the list of the postings: each posting is [docID,termfrequency].

After building the inverted index and computing the lengths of the documents, I Stored in the file *'index_file.tsv'* the total number of documents in the first row, the length of the documents in the second and the inverted index from row 3 as it follows: one term per row, and for each term the document frequency in the second column and all the postings in the following columns.

The structure of the file can be seen in the following image:

3159					
lengths	1.9754917556190086	5.608521251781702	4.365562556717266	4.005575179195399	3.557458425208839
aaa	1	666,1			
aad	1	807,1			
aaddett	1	2678,1			
aai	4	930,2	936,1	951,3	985,2
aba	1	230,1			
abap	4	477,1	604,1	836,2	3142,1
abb	1	2933,2			
abbast	1	3029,1			
abbiam	30	48,1	71,1	539,1	695,1
abbigl	5	352,1	621,1	1480,2	2970,1
abbin	1	1964,1			

1.3 Query Processing

As first thing I load the index from the file obtaining: the index, the array of document lengths and the number of documents.

I let the user insert a query and then, for each query, I preprocess it (as in the previous part I did for the documents) and I then search for all the documents containing at least one query term (if parameter is False), or all the documents with ALL query terms (if parameter is True [1]) and store it in *docs*.
Then I compute the cosine similarity between the documents in *docs* and return the Top-10 documents using a min-heap and show the results from the most relevant.

[1]Boolean search is an optional implementation if the parameter of the function search is changed, otherwise if you run 'query_processing.py' is set to False

1.4 Results: query examples

The first example is a one-term query: 'abituale' and we can see from the Index file that the only document containing that term (stemmed is 'abitual') is doc_ID:991 as the search engine returns.

```
 _____________________________
|                             |
| Eva's simple search engine |
|_____________________________|
This is a simple search engine for kijiji jobs announcements,
 feel free to insert queries or type 'q' if you want to quit
 Enjoy! =)

What are you looking for? abituale
1 results found, showing the TOP-10:
-------------------------------------
1. doc_ID: 991 score:0.207
Web Master Senior       Nielsen Communication, azienda leader nel settore della comunicazione dâ€™impresa, a comple
tamento del proprio organico ricerca un Web Master Senior. Il candidato ideale ha unâ€™età compresa tra i 25 ed i 3
5 anni e ha pluriennale esperienza nella creazione e gestione dei siti web. Sono richieste competenze nellâ€™analis
i , nella gestione periodica e nellâ€™incremento dellâ€™efficacia dei canali digitali per la promozione delle attiv
ità e del business aziendale. Sono altrettanto gradite competenze e conoscenze nellâ€™utilizzo efficace delle piatt
aforme social per il business e competenze nelle soluzioni personalizzate per la comunicazione (SEO, SEM, ottimizza
zione dei motori di ricerca, eccâ€¦). Inoltre, costituirà titolo preferenziale lâ€™abituale utilizzo dei principali
 strumenti per le riprese video (fotocamere, telecamere, droni) e la capacità di realizzazione e montaggio di video
clip. Inquadramento e retribuzione saranno in linea con lâ€™effettiva esperienza maturata e le competenze acquisite
. Indispensabile residenza o domicilio in zona Verona. Per candidarsi, inviare cv e portfolio personale specificand
o il rif. WMS.  Verona  11 ottobre, 09:47       https://www.kijiji.it/annunci/offerta/verona-annunci-verona/web-mas
ter-senior/165535647

-------------------------------------
```

Another example is a query with 3 terms: 'abituale abitudine able' that returns the 3 documents containing that words (each term appears only in one document).
As we can see from the inverted index file, the posting list of the query terms are:

$$\text{abitual} \rightarrow 991{,}1$$
$$\text{abitudin} \rightarrow 1965{,}1$$
$$\text{able} \rightarrow 1151{,}1$$

```
---------------------------------------
What are you looking for? abituale abitudine able
3 results found, showing the TOP-10:
---------------------------------------
1. doc_ID: 991 score:0.119
Web Master Senior       Nielsen Communication, azienda leader nel settore della comunicazione dâ€™impresa, a comple
tamento del proprio organico ricerca un Web Master Senior. Il candidato ideale ha unâ€™età compresa tra i 25 ed i 3
5 anni e ha pluriennale esperienza nella creazione e gestione dei siti web. Sono richieste competenze nellâ€™analis
i , nella gestione periodica e nellâ€™incremento dellâ€™efficacia dei canali digitali per la promozione delle attiv
ità e del business aziendale. Sono altrettanto gradite competenze e conoscenze nellâ€™utilizzo efficace delle piatt
aforme social per il business e competenze nelle soluzioni personalizzate per la comunicazione (SEO, SEM, ottimizza
zione dei motori di ricerca, eccâ€¦). Inoltre, costituirà titolo preferenziale lâ€™abituale utilizzo dei principali
 strumenti per le riprese video (fotocamere, telecamere, droni) e la capacità di realizzazione e montaggio di video
clip. Inquadramento e retribuzione saranno in linea con lâ€™effettiva esperienza maturata e le competenze acquisite
. Indispensabile residenza o domicilio in zona Verona. Per candidarsi, inviare cv e portfolio personale specificand
o il rif. WMS.  Verona  11 ottobre, 09:47       https://www.kijiji.it/annunci/offerta/verona-annunci-verona/web-mas
ter-senior/165535647

---------------------------------------
---------------------------------------
2. doc_ID: 1965 score:0.077
Junior Sales Executive in ambito ICT    Beta80 è una società ICT presente sul mercato da più di 30 anni. Oggi conti
amo circa 550 professionisti dedicati a implementare le soluzioni IT più idonee per contribuire al vantaggio compet
itivo dei nostri clienti. Ci distinguiamo - grazie alle competenze acquisite su clienti di primaria importanza - su
 diversi temi: alcuni molto verticali, legati all'Emergency & Crisis Management e Supply Chain Management, altri ch
e attraversano i capitoli più significativi oggi della Digital Transformation.  In particolare per la BU ICT Servic
es, desideriamo inserire  Junior Sales Executive in ambito ICT:  Cerchiamo candidati che abbiano l'aspirazione e la
 determinazione di voler fare il passaggio dal mondo Inside Sales a quello delle vendite e portino con sé in dote:
 -      Metodo di contatto dei clienti attraverso strumenti digital (Linkedin, hubspot, etc.)  -      Abitudine a l
avorare con la funzione marketing per la profilazione dei deal e per la vendita a valore  -       Attitudine a lavo
rare su new business  -        Consuetudine a lavorare per obiettivi  -  Passione per il settore dell'ICT e per le
tematiche di digital transformation (Cloud Enablement, Data Analysis, etc.)  Investiamo in affiancamento e formazio
ne affinché il candidato   -       diventi autonomo nella gestione del portafoglio d'offerta della nostra Business
Unit dedicata alla System Integration su clienti nuovi  -    impari un approccio strategico alla vendita di servizi
, guadagnando nel tempo un proprio portafoglio clienti  -       acquisisca un metodo per creare ed eseguire efficac
emente piani di account strategici per indirizzare l'offerta a lui assegnata, gestendo e mantenendo un'accurata pip
eline di vendita attraverso i sistemi aziendali.  E' richiesta:  -        tenacia e piacere nella vendita  -
 famigliarità con il mondo digital  -        approccio metodico e strutturato  -        desiderio di mettersi in gi
oco e cogliere una sfida  -        ottime abilità relazionali   Titoli preferenziali sono:  Laurea di tipo economic
o  Certificazioni di settore    Milano  24 settembre, 10:57     https://www.kijiji.it/annunci/offerta/milano-annunc
i-milano/junior-sales-executive-in-ambito-ict/166628335

---------------------------------------
---------------------------------------
3. doc_ID: 1151 score:0.040
IT Help-Desk    Www.patriziapepe.com  IT Help-Desk:  IT Help-Desk   Patrizia Pepe is currently looking for an IT He
lp-Desk. In this role, you will be responsible for offering support and technical assistance to internal users rega
rding software, hardware, or other computer systems. The scope of the role will include, but is not limited to, the
 setup of new/replacement laptops, phones, end-user technical support, and retail infrastructure support.  The Role
:   -Serving as the first point of contact for internal users seeking technical assistance over the phone or email
```

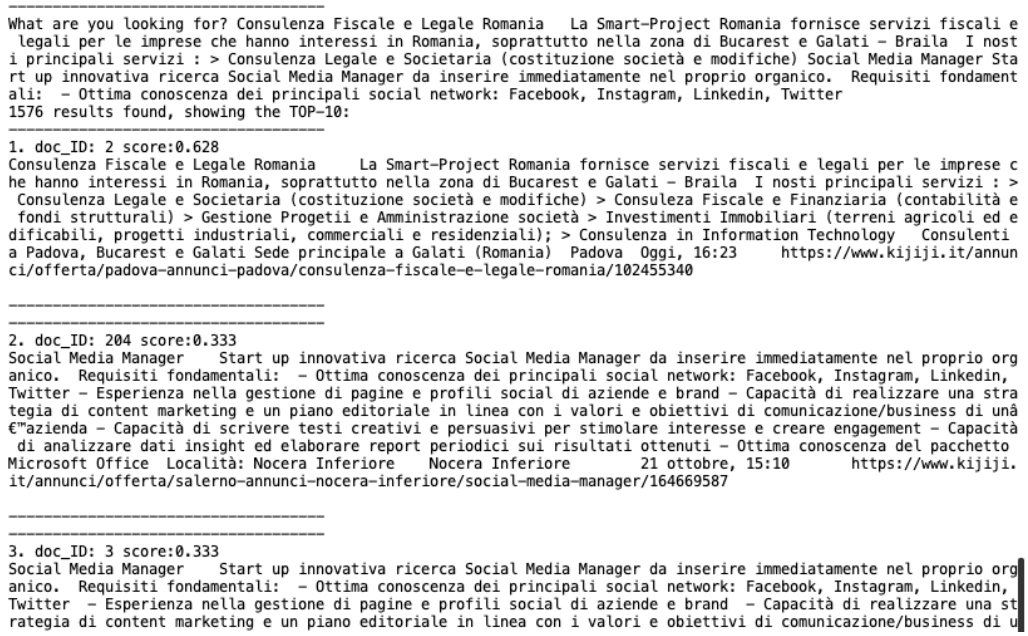
Then I show an example of a long query that corresponds to the entire title description and location of Document 2, that in Fact is the first result returned with cosine similarity score=1.

```
----------------------------------------
What are you looking for? Consulenza Fiscale e Legale Romania   La Smart-Project Romania fornisce servizi fiscali e
 legali per le imprese che hanno interessi in Romania, soprattutto nella zona di Bucarest e Galati - Braila  I nost
i principali servizi : > Consulenza Legale e Societaria (costituzione società e modifiche) > Consuleza Fiscale e Fi
nanziaria (contabilità e fondi strutturali) > Gestione Progetii e Amministrazione società > Investimenti Immobiliar
i (terreni agricoli ed edificabili, progetti industriali, commerciali e residenziali); > Consulenza in Information
Technology   Consulenti a Padova, Bucarest e Galati Sede principale a Galati (Romania)  Padova
1512 results found, showing the TOP-10:
----------------------------------------
1. doc_ID: 2 score:1.000
Consulenza Fiscale e Legale Romania     La Smart-Project Romania fornisce servizi fiscali e legali per le imprese c
he hanno interessi in Romania, soprattutto nella zona di Bucarest e Galati - Braila  I nosti principali servizi : >
 Consulenza Legale e Societaria (costituzione società e modifiche) > Consuleza Fiscale e Finanziaria (contabilità e
 fondi strutturali) > Gestione Progetii e Amministrazione società > Investimenti Immobiliari (terreni agricoli ed e
dificabili, progetti industriali, commerciali e residenziali); > Consulenza in Information Technology   Consulenti
a Padova, Bucarest e Galati Sede principale a Galati (Romania)  Padova  Oggi, 16:23     https://www.kijiji.it/annun
ci/offerta/padova-annunci-padova/consulenza-fiscale-e-legale-romania/102455340

----------------------------------------
----------------------------------------
2. doc_ID: 2007 score:0.147
Amministratore di reti, servizi e server in cloud       Clicca sul link sottostante "sito web" per inviarci la tua
candidatura.    Padova  24 settembre, 04:03     https://www.kijiji.it/annunci/offerta/padova-annunci-padova/amminis
tratore-di-reti-servizi-e-server-in-cloud/166610532

----------------------------------------
----------------------------------------
3. doc_ID: 1644 score:0.141
Operatore call center part time padova  Clicca sul link sottostante "sito web" per inviarci la tua candidatura. Pad
ova     30 settembre, 04:05     https://www.kijiji.it/annunci/offerta/padova-annunci-padova/operatore-call-center-p
art-time-padova/166773362

----------------------------------------
----------------------------------------
4. doc_ID: 1189 score:0.133
Centralinista part time padova  Clicca sul link sottostante "sito web" per inviarci la tua candidatura. Padova  8 o
```

Another long query example can be done copying Document 5 and the result is the same as before with doc_ID:5 as first result.

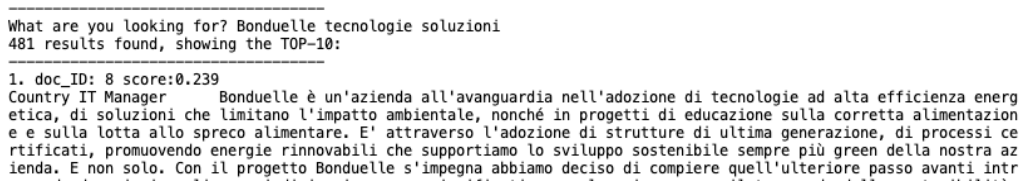
```
----------------------------------------
What are you looking for? Graphic Designer 3D   Descrizione: creazioni grafiche 3D.  Caratteristiche: padronanza so
ftware per modellazione 3D, creatività e cura del dettaglio.  Verrà valutato esclusivamente il CV caricato sul "Lav
ora con noi" di Moka Adv.  Requisiti: il CV deve necessariamente essere corredato da foto. Non verranno presi in co
nsiderazione profili che non abbiano inviato portfolio o show reel.  1 - I candidati selezionati faranno un video c
olloquio. 2 - A seguire un colloquio frontale.  Catania
2930 results found, showing the TOP-10:
----------------------------------------
1. doc_ID: 5 score:1.000
Graphic Designer 3D     Descrizione: creazioni grafiche 3D.  Caratteristiche: padronanza software per modellazione
3D, creatività e cura del dettaglio.  Verrà valutato esclusivamente il CV caricato sul "Lavora con noi" di Moka Adv
.  Requisiti: il CV deve necessariamente essere corredato da foto. Non verranno presi in considerazione profili che
 non abbiano inviato portfolio o show reel.  1 - I candidati selezionati faranno un video colloquio. 2 - A seguire
un colloquio frontale.  Catania Oggi, 11:36     https://www.kijiji.it/annunci/offerta/catania-annunci-catania/graph
ic-designer-3d/167354986

----------------------------------------
----------------------------------------
2. doc_ID: 2037 score:0.424
Sviluppatore Backend    Caratteristiche: laurea in Ingegneria Informatica o in Informatica, PHP (logica MVC), NodeJ
S, MySql, HTML, CSS. Si lavorerà nel contesto di progetti internazionali riguardanti la realtà aumentata e la realt
à virtuale. Verrà presa in considerazione la conoscenza di iOS per poter supportare altre attività aziendali.  Verr
à valutato esclusivamente il CV caricato sul "Lavora con noi" di VITECO.  Requisiti: il CV deve necessariamente ess
ere corredato da foto.  1 - I candidati selezionati faranno un video colloquio. 2 - A seguire un colloquio frontale
.       Catania 23 settembre, 15:21     https://www.kijiji.it/annunci/offerta/catania-annunci-catania/sviluppatore-
backend/166600514

----------------------------------------
----------------------------------------
3. doc_ID: 2035 score:0.391
Front-end Developer     Caratteristiche: laurea in Ingegneria Informatica o in Informatica, CSS, JavaScript, HTML,
framework Bootstrap, PHP. Si lavorerà nel contesto di progetti internazionali riguardanti lâ€™intelligenza artifici
ale e il machine learning. Verrà presa in considerazione la conoscenza di iOS per poter collaborare su altre attivi
```

Another example is built putting in the query part of the text from document 2 and part of the text of document 3.

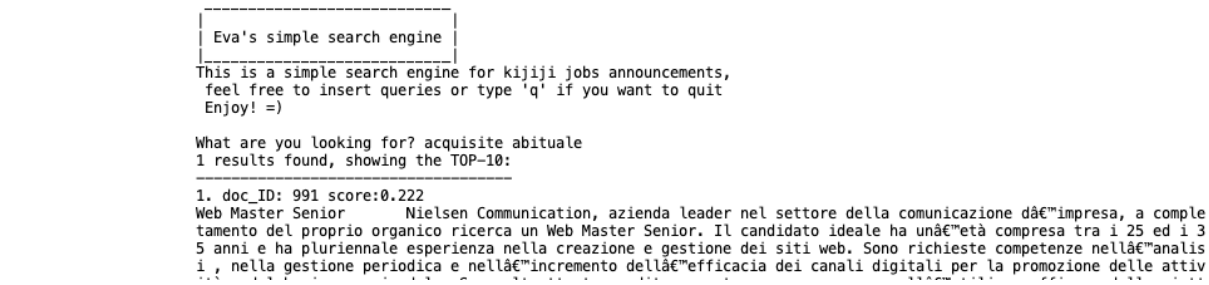


And we can see from the results that the Top1 is Document 2 and then we have doc 204 and doc 3 (Note that doc 204 has the same content of doc 3, and in fact it has the same score).

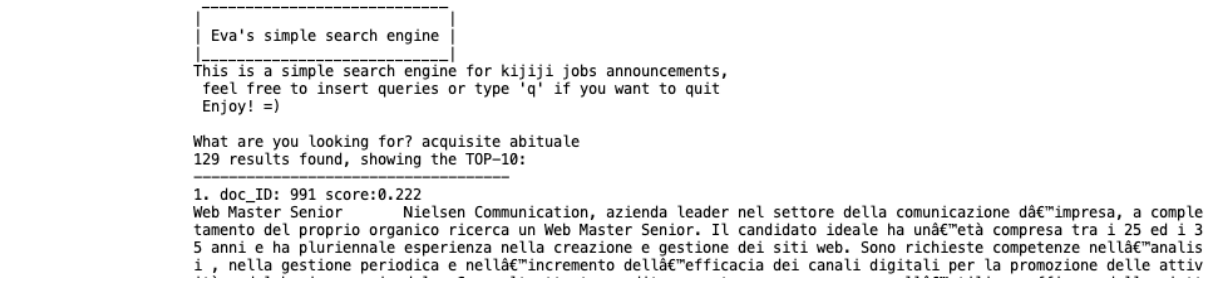
Another example is with the query 'Bonduelle tecnologie soluzioni' and the first document retrieved is doc 8 that is the only document in the collection containing the word 'Bonduel'.



One last example is to show how the search engine works if the search is made with the parameter boolean set to true, that returns the documents containing ALL the query terms.
As an example the query 'acquisite abituale' returns only the document 991 (that from the index file can be seen that is the only one in the intersection of the posting list of the 2 terms).



This result can be compared with the same query as input of the search engine (Non boolean) where we have 129 results found.



2 Nearest Neighbours search

The implementation of nearest neighbor search is split into different files:

- *'Utils.py'* contains util functions
- *'Shingles.py'* the class that given a documents creates the set of hash of the shingles
- *'MinHash.py'* the class that from the hash of the shingles creates the minhash signatures
- *'Lsh.py'* the class that implements LSH finding the candidate pairs
- *'Near_duplicates.py'* finds the nearest neighbours computing the Jaccard similarity of the shingle sets.
- *'p2.py'* the 'main' program that finds the duplicates with LSH, without LSH and shows the number of duplicate pairs found in both cases, the intersection of the candidate pairs and the time required.

To test the code you have to launch `python3 p2.py`.

In order to find near duplicates in the collection of Documents in file *'jobs.tsv* I took the full description of each document.

I considered the shingles of length 10 characters and two documents are considered near duplicates if the Jaccard coefficient is at least 0.8.
The number of hash functions used in LSH to 'permute' is 100.
The parameters are set to b=20 bands and r=5 rows per band, here it is the graph of the probability of becoming candidate as function of the similarity and we can see that at sim=0.8 the probability that the signatures agree in all rows of at least one band is about 1:

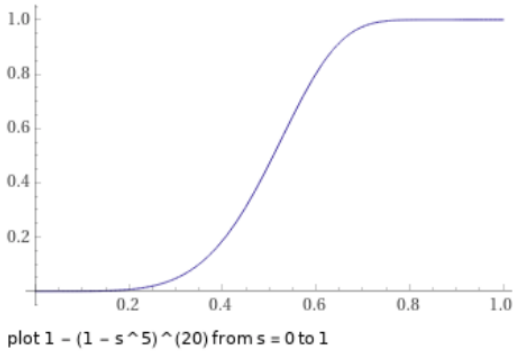


Figure 1: S-curve

In *Shingles.py*, the constructor of the class takes all the k-shingles from the input text, hashes them and adds them in a set of hashed shingles accessible via the function "getShingles".

In *MinHash.py*, I apply 100 different hash functions to each shingle of each Document, keeping the minimum for each document and each hash, constructing and storing the signature matrix as an array of document signatures where the document signature is an array of 100 entries, each entry is the minhash.The signatures are accessible via "getSignatures" function.

In *Lsh.py*, taking the signatures as input, I create the bands and hash them for each document. For each band and hash value I have a different bucket storing the docIDs of near documents. The buckets are stored in a nested dictionary that has has key the band and then as key the hashvalue.
Then for each bucket I take all the candidate pairs (the ones that have the same hash value for at least one band) in a set. That set, containing all the candidate pairs is accessible via "getCandidates" function.

In *Near_duplicates.py*, taking the shingles of the documents as input, I iterate on all the possible pairs of documents, computing the intersection and the union of shingles sets, storing the pairs where the Jaccard similarity is at least 0.8 (intersection/union) in a set accessible via "getDuplicate" function.

In *p2.py* I read the file *'jobs.tsv'* line per line, preprocess each line and obtain the set of shingle for each document. Then I get the minhash signature, then perform LSH, keeping track of the time taken and showing the number of candidates obtained. I then compute the near duplicates as implemented in the class *near_duplicates* showing the number of candidates and the time taken. Last, I compare the results showing the number of pairs equals in the two cases (intersection of the results).

2.1 Results

```
---------- Near duplicates search ----------

time to create the shingles: 6.989065170288086 seconds
*--------- Near duplicates with LSH ---------*
[+] MinHashing Done
time to perfom minhashing: 471.17271995544434 seconds
[+] LSH Done
time with LSH: 12.425849914550781 seconds
number of candidate pairs with LSH: 1378747
*------- Near duplicates with Jaccard on shingle set -------*
time without LSH: 212.3667073249817 seconds
number of candidate pairs without LSH: 1374529
number of pairs in the intersection between the results: 1374529
AirdiMartina5:p2 martinaevangelisti$
```

Note that minhashing is the operation that takes more time due to the cryptographic hash function used to hash.

3 Nearest neighbours in Apache Spark

The implementation of nearest neighbor search in Apache spark is in the notebook *'nearest_neighbours.ipynb'*.

To better understand the results I first run the code with only 100 documents obtaining the following results:

Figure 2: number of candidate pairs and running time with LSH

Figure 3: number of candidate pairs, running time without LSH and intersection with LSH results

```
candidate_pairs.sortByKey().take(20)
✓ 0.4s
[(0, 42),
 (0, 21),
 (0, 84),
 (0, 63),
 (18, 20),
 (18, 24),
 (18, 28),
 (18, 32),
 (18, 36),
 (18, 40),
 (18, 48),
 (18, 52),
 (18, 56),
 (18, 60),
 (18, 64),
 (18, 68),
 (18, 72),
 (18, 76),
 (18, 80),
 (18, 88)]
```

```
candidate_pairs2.sortByKey().take(20)
✓ 0.7s
[(0, 42),
 (0, 21),
 (0, 84),
 (0, 63),
 (18, 24),
 (18, 32),
 (18, 40),
 (18, 48),
 (18, 56),
 (18, 64),
 (18, 72),
 (18, 80),
 (18, 88),
 (18, 23),
 (18, 55),
 (18, 71),
 (18, 87),
 (18, 22),
 (18, 38),
 (18, 46)]
```

Figure 4: On the left candidate pairs found with LSH, on the rigth the ones found comparing the set of shingles

As we can see from the length of the intersection and from the cardinalities of the 2 different sets of candidate pairs, with 100 documents LSH found the same duplicates that are found comparing the set of shingles, in this case 1496 duplicate pairs.
We can look for example at the first pair (0,42),and looking at the *'Jobs.tsv'* file line 1 and 43 we can see that corresponds to the same announcement.

The results obtained from the whole Corpus are the following:

```
end = time.time()
lsh_time= end - start
print("time with LSH: "+str(lsh_time)+" seconds")
✓ 0.7s
time with LSH: 493.1639440059662 seconds
```

```
lsh_pairs = candidate_pairs.count()
lsh_pairs
✓ 5.8s
1378779
```

Figure 5: number of candidate pairs and running time with LSH

```
end = time.time()
jac_time= end - start
print("time without lsh: "+str(jac_time)+" seconds")
```
✓ 0.2s

```
time without lsh: 533.6741292476654 seconds
```

```
near_duplicates = candidate_pairs2.count()
near_duplicates
```
✓ 30.7s

```
1374530
```

Figure 6: number of candidate pairs and running time without LSH

Intersection between candidate_pairs found with LSH and candidate_pairs2 found without LSH

```
#intersection
result_intersection=candidate_pairs2.intersection(candidate_pairs)
n_intersect = result_intersection.count()
n_intersect
```
✓ 1m 18.1s

```
1374530
```

Figure 7: intersection of the results

On the notebook is it possible to see the results after each step of the computation with comments of what it's done in each step.

Here we can summarize the results obtained:

```
Number of duplicates found with LSH: 1378779
Time taken by LSH: 493.1639440059662
Number of duplicates found without LSH: 1374530
Time taken without LSH: 533.6741292476654
Number of duplicates in the intersection: 1374530
Number of false positives: 4249
Number of false negatives: 0
```