Faculty of Computer Science
Master of Computational Optimization, 2nd year

January 12, 2021

DETECTING TEXT SIMILARITIES AMONG MEDICAL ARTICLES

PYSPARK PROJECT 5

Madalina-Alina Racovita
Sergiu-Andrei Dinu
Cristian Vintur
Stefan Strugari

Faculty of Computer Science
Master of Computational Optimization, 2nd year

DETECTING TEXT SIMILARITIES AMONG MEDICAL ARTICLES

TODAY'S AGENDA

1. INTRODUCTION. ABOUT THE PROJECT.
2. PUBMED DATASET
3. PARSING THE DATA AND GATHERING SOME USEFUL FEATURES
4. STATISTICS
5. MIN-HASHING ALGORITHM
6. KMEANS WITH TF-IDF
7. EVERYTHING INSIDE A GC CLUSTER
8. CONCLUSIONS. Q&A

Faculty of Computer Science
Master of Computational Optimization, 2nd year

January 12, 2021

DETECTING TEXT SIMILARITIES AMONG MEDICAL ARTICLES

PROJECT'S DESCRIPTION

BRIEF INTRODUCTION

Provide statistics for **sub-domain specific publications** (e.g. Malaria research and not only) on structural properties of the text. Run some Machine Learning algorithms to establish similarities between articles. Configure everything to run inside a cloud environment.

PROJECT OBJECTIVES

EXPECTATIONS AND OUTCOMES

To deal with a large amount of data and to provide some relevant statistics.

To train some Machine Learning algorithms for text similarities and obtain some pair of articles with some similarity percentage scores.

To learn how to configure properly an environment able to solve such tasks inside a cloud provider.

OUR DEVELOPMENT PLAN

01 DONE → Environment configuration, Github repository

02 DONE → Parsing the dataset, gathering all files together in a pyspark rdd, finding relevant XML fields

03 DONE → Since the small samples dataset is not available anymore, we are going to build our own testing dataset, with a smaller size comparing to the initial one

04 DONE → Provide statistics / visualization and insights about the data

05 DONE → Research on the algorithms used for text similarity detection on datasets of large dimensions (ideas: Min-hashing)

06 DONE → Discuss and transform the initial dataset into a set of feature vectors for training the algorithms

07 DONE → Min-hashing implementation: return most similar pairs of articles and their similarity percentage

08 DONE → Cluster the texts using K-means

09 DONE → Inside each cluster returned by K-means apply a text similarity algorithm (like TF-IDF or Gensim or Spacy) are return most similar pairs of articles and their similarity percentage

10 DONE → Configure the project to run inside Google Cloud, using Spark jobs inside a cluster

11 ON GOING → **Presentation**

Faculty of Computer Science
Master of Computational Optimization, 2nd year

January 12, 2021

DETECTING TEXT SIMILARITIES AMONG MEDICAL ARTICLES

PUBMED DATASET

This dataset is a collection of gzipped xmls gathered in each day from 1st of January 2016 to 31st of December 2016.

The dataset has a directory structure that we needed to change: i.e. multiple directories, each directory represents the data collected in a given day from 2016. In each directory you can find zero or more gzipped xmls.

CHALLENGES THAT WE FACED

How are we going to solve them?

1 **BRAINSTORMING**

We needed to find a library capable of loading these gzipped XML just by giving as an input the directory with multiple files of this type

2 **BRAINSTORMING**

This library should've been also able to parse the information from the XMLs without to much effort from our side, because the files could very large, and pasing them even if with a REGEX approach would've been implied a lot of time.

3 **BRAINSTORMING**

Do all XML files have the same structure? Because if the structure is not the same it is very likely that when we'll parse the data for getting the relevant information, we'll get errors if some tags are not present in that file.

SPARK XML

A library for parsing and querying XML data with Apache Spark, for Spark SQL and DataFrames.

This package supports to process format-free XML files in a distributed way.

xml_file_structure.txt

```
root
 |-- _corrupt_record: string (nullable = true)
 |-- abstract: struct (nullable = true)
 |    |-- _abstract-type: string (nullable = true)
 |    |-- _id: string (nullable = true)
 |    |-- p: string (nullable = true)
 |    |-- sec: array (nullable = true)
 |    |    |-- element: struct (containsNull = true)
 |    |    |    |-- _id: string (nullable = true)
 |    |    |    |-- p: string (nullable = true)
 |    |    |    |-- styled-content: array (nullable = true)
 |    |    |    |    |-- element: struct (containsNull = true)
 |    |    |    |    |    |-- _VALUE: string (nullable = true)
 |    |    |    |    |    |-- _style: string (nullable = true)
 |    |    |    |-- title: string (nullable = true)
 |    |-- sub: array (nullable = true)
 |    |    |-- element: long (containsNull = true)
 |-- back: struct (nullable = true)
 |    |-- ref-list: struct (nullable = true)
 |    |    |-- ref: array (nullable = true)
 |    |    |    |-- element: struct (containsNull = true)
 |    |    |    |    |-- _id: string (nullable = true)
 |    |    |    |    |-- label: long (nullable = true)
 |    |    |    |    |-- mixed-citation: struct (nullable = true)
 |    |    |    |    |    |-- _publication-type: string (nullable = true)
 |    |    |    |    |    |-- article-title: string (nullable = true)
 |    |    |    |    |    |-- collab: string (nullable = true)
 |    |    |    |    |    |-- fpage: long (nullable = true)
 |    |    |    |    |    |-- lpage: long (nullable = true)
 |    |    |    |    |    |-- name: array (nullable = true)
 |    |    |    |    |    |    |-- element: struct (containsNull = true)
 |    |    |    |    |    |    |    |-- given-names: string (nullable = true)
 |    |    |    |    |    |    |    |-- surname: string (nullable = true)
 |    |    |    |    |    |-- pub-id: array (nullable = true)
 |    |    |    |    |    |    |-- element: struct (containsNull = true)
 |    |    |    |    |    |    |    |-- _VALUE: string (nullable = true)
 |    |    |    |    |    |    |    |-- _pub-id-type: string (nullable = true)
 |    |    |    |    |    |-- source: string (nullable = true)
 |    |    |    |    |    |-- year: long (nullable = true)
 |    |    |    |    |-- person-group: array (nullable = true)
 |    |    |    |    |    |-- element: struct (containsNull = true)
 |    |    |    |    |    |    |-- _person-group-type: string (nullable = true)
 |    |    |    |    |    |    |-- name: array (nullable = true)
 |    |    |    |    |    |    |    |-- element: struct (containsNull = true)
 |    |    |    |    |    |    |    |    |-- given-names: string (nullable = true)
 |    |    |    |    |    |    |    |    |-- surname: string (nullable = true)
 |    |    |    |    |    |    |-- string-name: string (nullable = true)
 |    |    |    |    |-- pub-id: array (nullable = true)
 |    |    |    |    |    |-- element: struct (containsNull = true)
 |    |    |    |    |    |    |-- _VALUE: string (nullable = true)
 |    |    |    |    |    |    |-- _pub-id-type: string (nullable = true)
 |    |    |    |    |-- source: string (nullable = true)
 |    |    |    |    |-- year: long (nullable = true)
 |    |    |    |-- _VALUE: string (nullable = true)
 |    |    |    |-- _ref-type: string (nullable = true)
 |    |    |    |-- _rid: string (nullable = true)
 |    |    |    |-- sup: long (nullable = true)
```

How big is this structure file? This big, and here are just tag names.

CHALLENGES THAT WE FACED

How are we going to solve them?

1. **SOLVED**

We needed to find a library capable of loading these gzipped XML just by giving as an input the directory with multiple files of this type

2. **SOLVED**

This library should've been also able to parse the information from the XMLs without to much effort from our side, because the files could very large, and pasing them even if with a REGEX approach would've been implied a lot of time.

3. **STILL NO SOLUTION**

Do all XML files have the same structure? Because if the structure is not the same it is very likely that when we'll parse the data for getting the relevant information, we'll get errors if some tags are not present in that file.

OUR DATASETS AFTER WE GATHERED THE DATA

The initial dataset

2016_all_data

2016_testing_df

01

02

03

Faculty of Computer Science
Master of Computational Optimization, 2nd year

January 12, 2021

DETECTING TEXT SIMILARITIES AMONG MEDICAL ARTICLES

...AND WE PARSED, AND PARSED, AND PARSED

The problem was that the functions we wrote worked for the current testing dataframe, if we decided the to change the dataframe we encountered a lot of errors due to the different structure of the XMLs.

```python
def get_bodys_list(spark: SparkSession, input_dir):
    df = spark.read \
                .format('com.databricks.spark.xml') \
                .options(rowTag='record')\
                .options(rowTag='body')\
                .load(input_dir)
    return df.select("sec", "p")\
            .rdd\
            .map(lambda row: build_body(row['sec'], row["p"]))\
            .zipWithIndex()\
            .map(lambda record: (record[1], record[0]))
```

```python
def get_abstract_list(spark: SparkSession, input_dir):
    df = spark.read \
        .format('com.databricks.spark.xml') \
        .options(rowTag='record') \
        .options(rowTag='abstract') \
        .load(input_dir)
    return df.select("sec", "p") \
            .rdd \
            .map(lambda row: build_body(row['sec'], row["p"])) \
            .zipWithIndex() \
            .map(lambda record: (record[1], record[0]))
```

● ● ● And a lot of similar functions for parsing.

Faculty of Computer Science
Master of Computational Optimization, 2nd year

January 12, 2021

STRATEGY CHANGED

Let's have something functional!

	abstract	paragraphs	sections	body	categories	title	pages count	authors	affiliations	figures count
0	Expression of the oncogenic transcription fact...	[We hypothesized that MYC-dependent metabolic ...	[[For U-]]	We hypothesized that MYC-dependent metabolic d...	[Article]	Inhibition of fatty acid oxidation as a therap...	Not specified	[Roman Camarda, Zhou Zhou, Rebecca A. Kohnz, S...	[Not specified]	Not specified
1	Membrane proteins are of outstanding importanc...	[A major challenge is maintaining membrane pro...	[[Membrane proteins are encoded by approx. 30%...	A major challenge is maintaining membrane prot...	[Article]	A novel lipoprotein nanoparticle system for me...	Not specified	[Jens Frauenfeld, Robin Löving, Jean-Paul Arma...	[Not specified]	Not specified
2	Epidemiological and experimental data implicat...	[PGC-1α in skeletal muscle induces broad genet...	[[C2C12 cells were grown in 10 cm dishes until...	PGC-1α in skeletal muscle induces broad geneti...	[Article]	A branched chain amino acid metabolite drives ...	Not specified	[Cholsoon Jang, Sungwhan F Oh, Shogo Wada, Gle...	[Not specified]	Not specified
3	Development of multicellular organisms is comm...	[Morphogenetic patterns observed in experiment...	[[The greatest manifestation of biological dev...	Morphogenetic patterns observed in experimenta...	[Article]	Scaling of morphogenetic patterns in reaction-...	Not specified	[Manan'larivo Rasolonjanahary, Bakhtier Vasiev]	[Not specified]	Not specified
4		[The heritability of muscle strength and power...	[[Environmental and genetic factors influence ...	The heritability of muscle strength and power ...	[Original Paper]	Association analysis of	Not specified	[Not specified]	[Not specified]	Not specified
5	The performance of professional strength and p...	[Excessive body weight gain because of an incr...	[[Regular physical activity has significant be...	Excessive body weight gain because of an incre...	[Review Paper]	Genetic variants influencing effectiveness of ...	Not specified	[A Leońska-Duniec, Il Ahmetov, P Zmijewski]	[Not specified]	Not specified
6	Frequent and regular physical activity has sig...	[There is growing evidence linking T to motiva...	[[In sport, the testosterone (T) contribution ...	There is growing evidence linking T to motivat...	[Original Paper]	Temporal associations between individual chang...	Not specified	[BT Crewther, J Carruthers, LP Kilduff, CE San...	[Not specified]	Not specified
7	To advance our understanding of the hormonal c...	[Nevertheless, the mechanism responsible for t...	[[Effective energy metabolism and transport of...	Nevertheless, the mechanism responsible for th...	[Original Paper]	The effect of the competitive season in profes...	Not specified	[A Dzedzej, W Ignatiuk, J Jaworska, T Grzywacz...	[Not specified]	Not specified

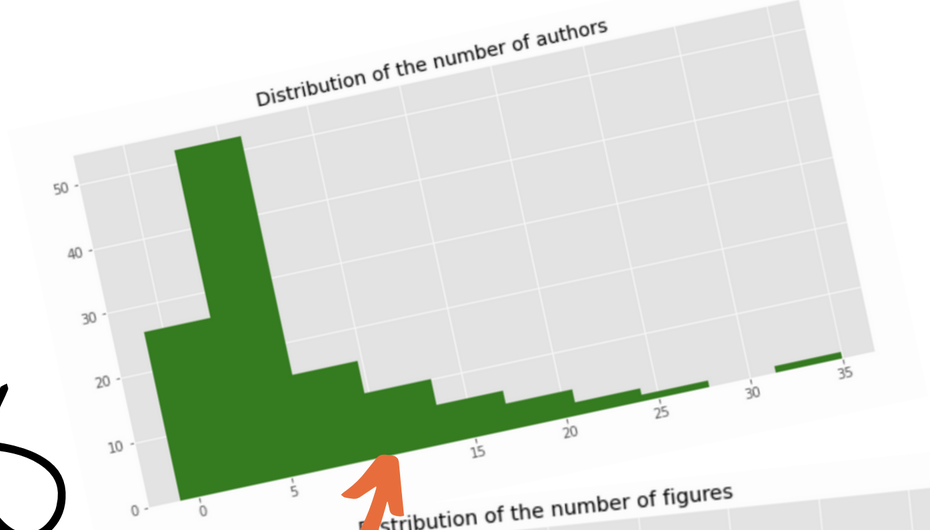
Faculty of Computer Science
Master of Computational Optimization, 2nd year

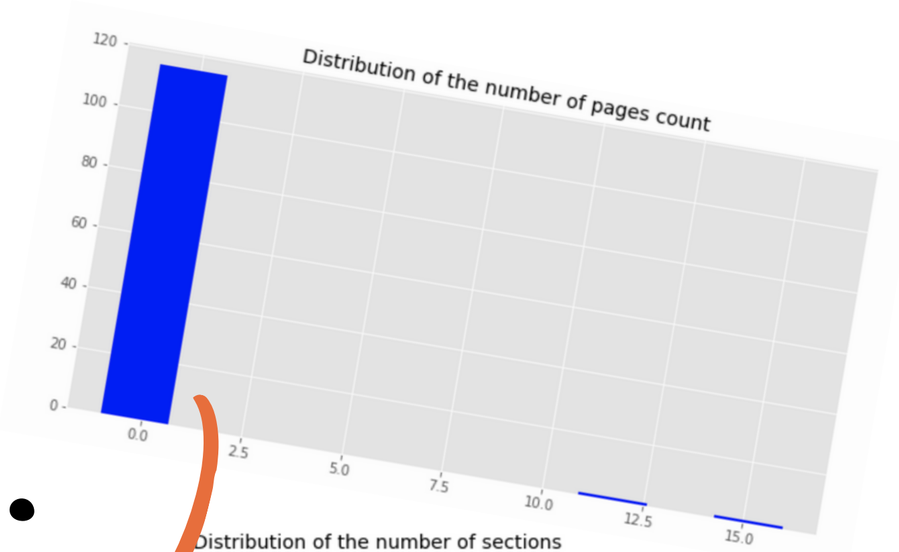
January 12, 2021

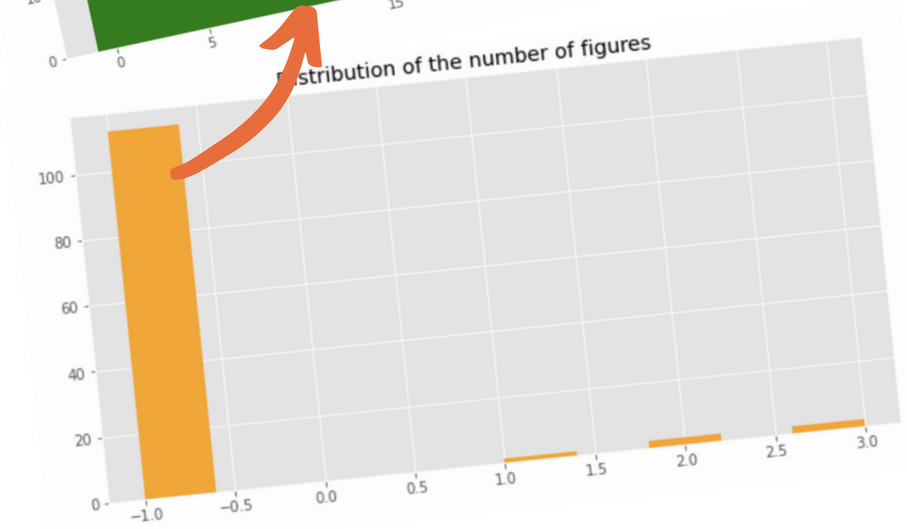
DETECTING TEXT SIMILARITIES AMONG MEDICAL ARTICLES

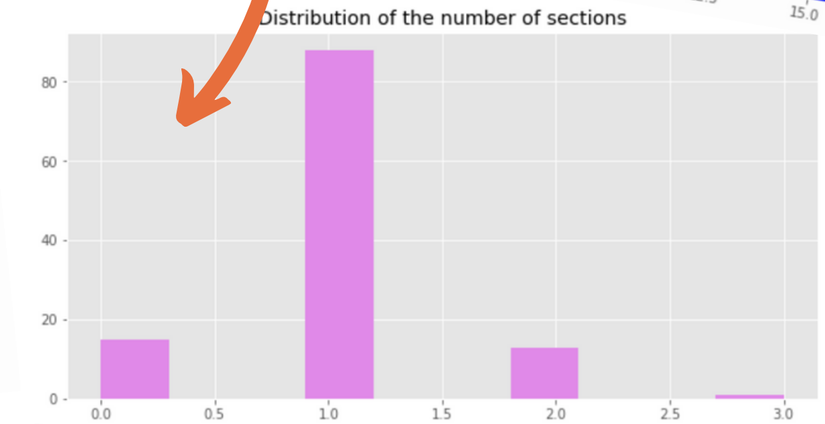
WE BUILT STATISTI-CAL PLOTS

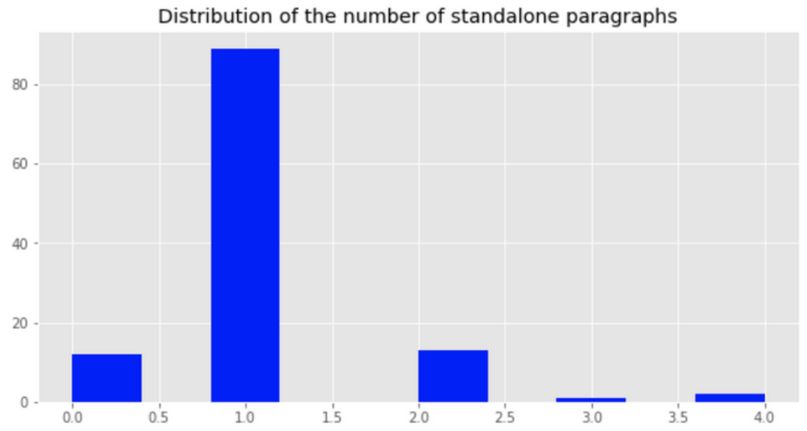
For different fields from the collected data

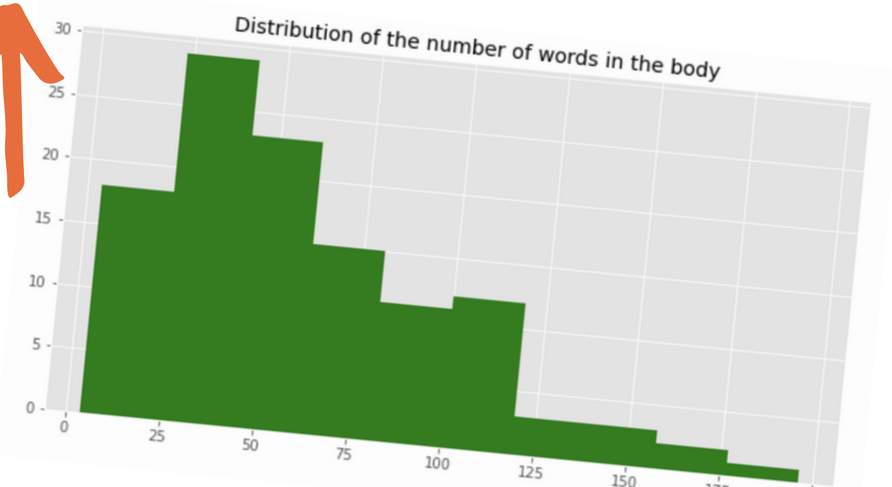












Faculty of Computer Science
Master of Computational Optimization, 2nd year

January 12, 2021

DETECTING TEXT SIMILARITIES AMONG MEDICAL ARTICLES

Faculty of Computer Science
Master of Computational Optimization, 2nd year

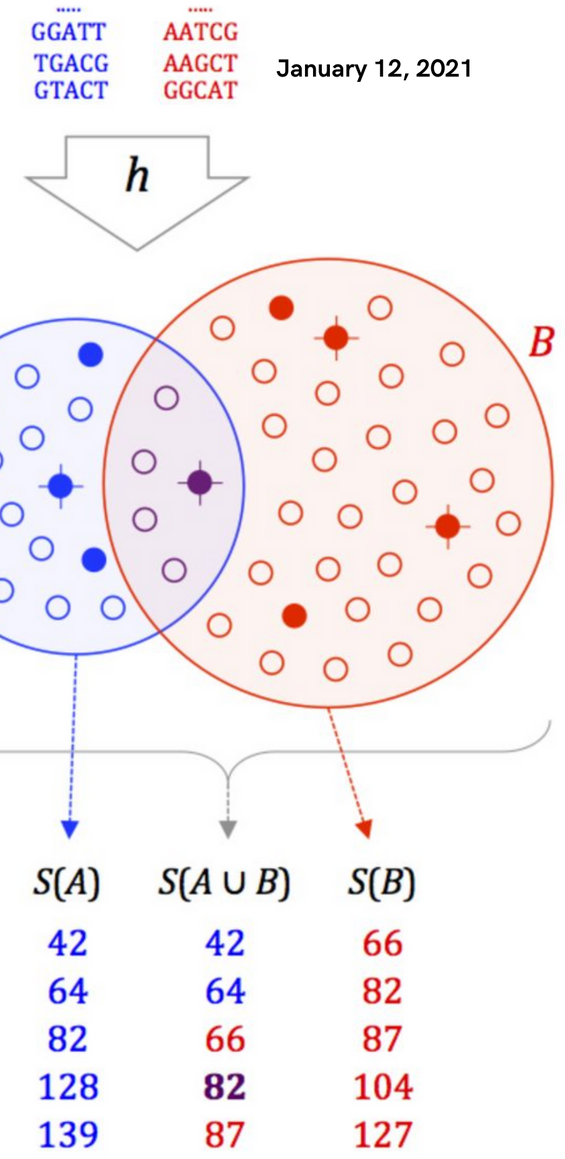
January 12, 2021

MIN-HASHING ALGORITHM

Overview

In computer science and data mining, MinHash (or the min-wise independent permutations locality sensitive hashing scheme) is a technique for quickly estimating how similar two sets are.

```
def getArticlesFeaturesFromBody(articlesDataFrame, numFeatures = 20) :
    regexTokenizer = RegexTokenizer(inputCol = "body", outputCol = "words", pattern = "\\W")
    remover = StopWordsRemover(inputCol = "words", outputCol= "filtered")
    hashingTF = HashingTF(inputCol = "filtered", outputCol = "rawFeatures", numFeatures = numFeatures)
    idf = IDF(inputCol = "rawFeatures", outputCol = "features")
```



$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \approx \frac{|S(A \cup B) \cap S(A) \cap S(B)|}{|S(A \cup B)|}$$

MIN-HASHING ALGORITHM

Jaccard similarity and minimum hash values

Jaccard similarity and minimum hash values
The Jaccard similarity coefficient is a commonly used indicator of the similarity between two sets. Let U be a set and A and B be subsets of U, then the Jaccard index is defined to be the ratio of the number of elements of their intersection and the number of elements of their union:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$

This value is 0 when the two sets are disjoint, 1 when they are equal, and strictly between 0 and 1 otherwise. Two sets are more similar (i.e. have relatively more members in common) when their Jaccard index is closer to 1.

MIN-HASHING ALGORITHM

Jaccard similarity scores

	hashes	ids	titles
0	[[111046356.0], [989858842.0]]	[8589934594, 111669149701, 120259084291, 12884...	[An epidemiological study of rates of illness ...
1	[[111046356.0], [989993821.0]]	[51539607555, 94489280514, 77309411331, 146028...	[Bitpacking techniques for indexing genomes: I...
2	[[154535185.0], [990398758.0]]	[111669149697]	[Efficacy of Intra-aortic Balloon Pump before ...
3	[[154535185.0], [989858842.0]]	[0, 17179869188, 146028888067, 77309411330, 34...	[The Lung Screen Uptake Trial (LSUT): protocol...
4	[[154535185.0], [990533737.0]]	[128849018882]	[Possible Mechanism of Therapeutic Effect of 3...
5	[[241512843.0], [989858842.0]]	[163208757251]	[Validation of a symphysis-fundal height chart...
6	[[804900880.0], [989858842.0]]	[111669149698, 120259084292]	[MiR-106b-5p Inhibits Tumor Necrosis Factor-α-...
7	[[198024014.0], [989858842.0]]	[154618822661, 34359738372, 51539607556]	[Mood, anxiety, and alcohol use disorders and ...
8	[[111046356.0], [990263779.0]]	[128849018885, 111669149699]	[Association between Carotid Intima-media Thic...
9	[[111046356.0], [990128800.0]]	[94489280512, 128849018883, 94489280513]	[Regeneration of the lung: Lung stem cells and...
10	[[154535185.0], [989993821.0]]	[8589934596, 85899345923]	[Lung functions among patients with pulmonary ...
11	[[154535185.0], [990128800.0]]	[5, 25769803777]	[Deregulation of , Combination of single quant...
12	[[457973618.0], [989858842.0]]	[103079215104, 120259084289]	[Improving chemical similarity ensemble approa...
13	[[457973618.0], [989993821.0]]	[77309411329, 8589934592]	[Multiplex SNaPshot—a new simple and efficient...

	id1	id2	jaccardDistance
0	154618822658	94489280515	0.350000
1	163208757250	77309411328	0.350000
2	42949672960	103079215105	0.350000
3	137438953475	163208757250	0.350000
4	137438953477	154618822661	0.350000
...	...	...	...
9165	154618822660	60129542144	0.125000
9166	103079215104	120259084289	0.214286
9167	120259084289	103079215104	0.214286
9168	128849018885	137438953473	0.454545
9169	137438953473	128849018885	0.454545

Faculty of Computer Science
Master of Computational Optimization, 2nd year

January 12, 2021

DETECTING TEXT SIMILARITIES AMONG MEDICAL ARTICLES

KMEANS & TFIDF

Overview

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

KMEANS & TFIDF

Implementation

In [3]:
```
def create_tfidf_features(sdf, num_features=100):
    sdf.registerTempTable('sdf')
    new_sdf = sqlContext \
        .sql("SELECT CONCAT(abstract, ' ', body, ' ', title) as text, categories as category from sdf")

    tokenizer = Tokenizer(inputCol="text", outputCol="words")
    wordsData = tokenizer.transform(new_sdf)

    hashingTF = HashingTF(inputCol="words", outputCol="rawFeatures", numFeatures=num_features)
    featurizedData = hashingTF.transform(wordsData)
    # alternatively, CountVectorizer can also be used to get term frequency vectors

    idf = IDF(inputCol="rawFeatures", outputCol="features")
    idfModel = idf.fit(featurizedData)
    rescaledData = idfModel.transform(featurizedData)

    return rescaledData.select('category', 'features')
```

In [5]:
```
from pyspark.ml.clustering import KMeans

def run_k_means(data, k=3):
    kmeans = KMeans(k=5)
    model = kmeans.fit(data.select('features'))
    return model.transform(data)
```

In [46]:
```
transformed_full = run_k_means(data, k=3)
```

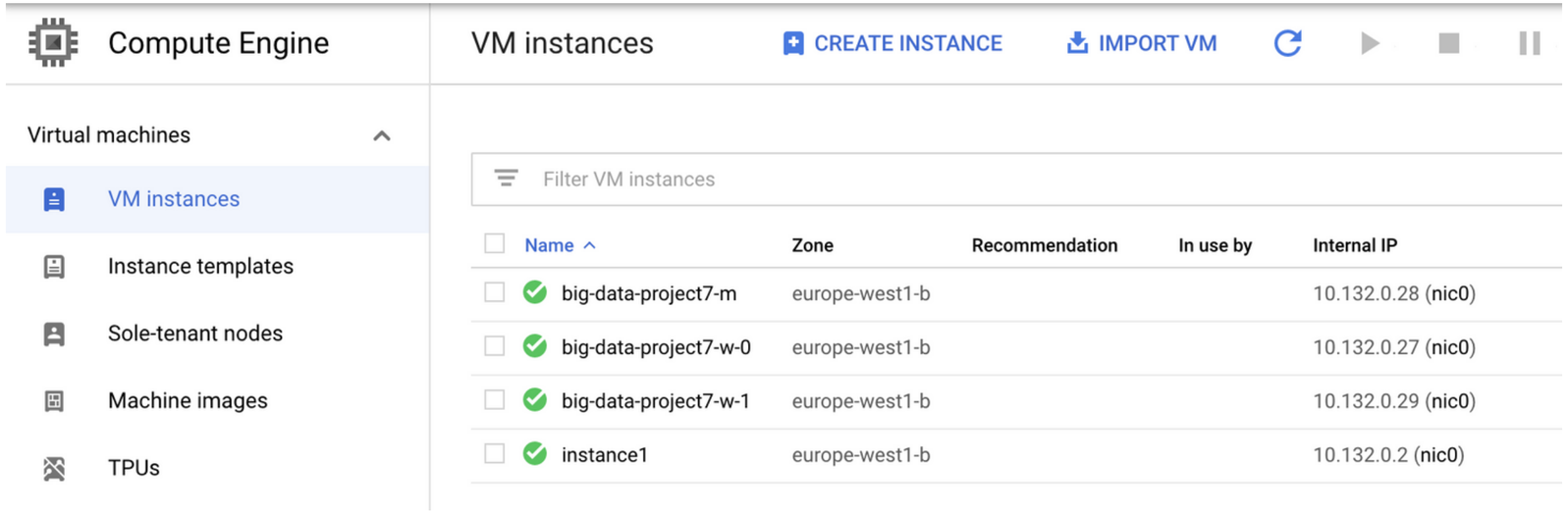
Faculty of Computer Science
Master of Computational Optimization, 2nd year

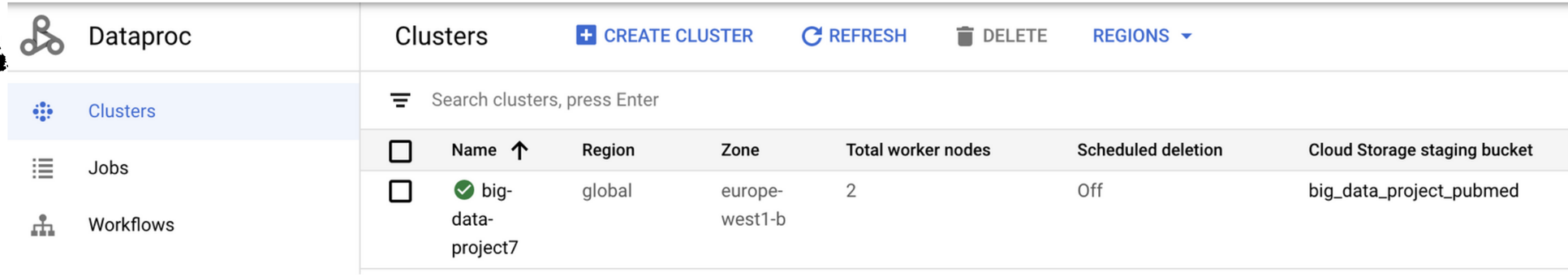
January 12, 2021

DETECTING TEXT SIMILARITIES AMONG MEDICAL ARTICLES

EVERYTHING INSIDE A GOOGLE CLOUD CLUSTER

Google Cloud





Faculty of Computer Science
Master of Computational Optimization, 2nd year

January 12, 2021

DETECTING TEXT SIMILARITIES AMONG MEDICAL ARTICLES

SUMMARY OF TODAY'S PRESENTATION

01 PUBMED DATASET

We saw what this dataset contains and which are the challenges we faced when working with it

02 SPARK XML

We found a library that is capable of parsing gzipped XMLs, but still a lot of manual parsing was needed to be done.

And also not all challenges were solved.

03 STATISTICS

We provide statistics for just a chunk of data, because for more files the parsing wouldn't work anymore.

04 TEXT SIMILARITIES

We find which algorithms can be used for detecting text similarities. We worked with MinHashing and Kmeans.

05 GOOGLE CLOUD

We configured our work to run inside a Google Cloud cluster

DO YOU HAVE

ANY QUESTIONS?

Feel free to make this an open discussion for questions or clarifications.

REFERENCES

- Github repository: https://github.com/RacovitaMadalina/BigData-Project---Text-similarity-over-PubMed-dataset/
- Link to PubMed dataset: https://drive.google.com/drive/folders/0B6LHYB5SN9DEWWdXQUNkS3NVOW8
- Our Google Cloud cluster: https://console.cloud.google.com/access/iam?project=p5-2020-300905
- Spark XML: https://github.com/databricks/spark-xml

THANK YOU!

AND

HAVE A NICE DAY AHEAD!