Clústering de Documentos basados en Big Data

Ana María Bedoya Hernández, *abedoy19@eafit.edu.co, ,Universidad EAFIT*
Esteban Salazar Montoya, *esalaza7@eafit.edu.co, Universidad EAFIT*
Jeniffer Maria Palacio Sandoval, *jmpalaci@eafit.edu.co, Universidad EAFIT*

***Abstract*—Document Clustering is a widely studied problem in text mining. It is the process of partitioning or grouping a given set of documents into different clusters where documents in the same cluster are similar. K-means, one of the algorithms used in this cases, solves the clustering problem following a simple way to classify a given data set through a certain number of clusters (we can suppose k clusters). The principal idea is to define k centroids, one for each cluster. This clustering algorithm uses an iterative procedure which converges to one of the many local minimum. At the same time, we have found that these iterative techniques are very sensitive to initial starting conditions of the centroid of each cluster and the more the distance among the cluster centroid the better the clustering performance.**

***Keywords*—Big data, algoritmo, Tf-idf, K-means, clúster, minería de textos, Spark, Python.**

I. Resumen

En este documento se puede encontrar una solución al problema que se plantea en el proyecto 4 de la materia Tópicos Especiales en Telemática el cual consiste en la implementación de un algoritmo distribuido en Big Data que permite la agrupación de documentos similares en diferentes clústers. Esto fue trabajado con el algoritmo de k-means y con la medida numérica tf-idf. Ambos proporcionan ayuda en la clasificación de documentos que se encuentren relacionados entre sí, teniendo en cuenta el grado de similitud que puedan tener dichos documentos.

II. Introducción

La minería de texto (text mining), es una de las técnicas de análisis de textos que ha permitido implementar una serie de aplicaciones muy novedosas hoy en día. Buscadores en la web (Google, Facebook, Amazon, Spotify, Netflix, entre otros), sistemas de recomendación, procesamiento natural del lenguaje, son algunas de las aplicaciones. Las técnicas de agrupamiento de documentos (clustering) permiten relacionar un documento con otros parecidos de acuerdo a alguna métrica de similaridad. Esto es muy usado en diferentes aplicaciones como: clasificación de nuevos documentos entrantes al dataset, búsqueda y recuperación de documentos, ya que cuando se encuentra un documento seleccionado de acuerdo al criterio de búsqueda, el contar con un grupo de documentos relacionados, permite ofrecerle al usuario otros documentos que potencialmente son de interés para él.

III. Marco teórico

Encontramos varios documentos en un dataset, los cuales están desordenados y requieren ser organizados en diferentes clústers, pero se debe de tener en cuenta la similitud que tengan entre los documentos. Lo primero que se debe de realizar es determinar qué tan similares son los documentos entre sí, para lo cual hay diferentes algoritmos que nos ayudan a trabajar en este tema, de los cuales podemos encontrar: Coeficiente de Jaccard, Coeficiente de correlacin de Pearson, la Distancia Euclidiana, tf-idf.

Posteriormente de encontrar la similitud que se tiene entre los documentos, debemos de utilizar un algoritmo que nos ayude al agrupamiento de los documentos en los diferentes clusters, como lo es K-means.

Las técnicas de agrupamiento de documentos (clústering) permiten relacionar un documento con otros parecidos de acuerdo a alguna métrica de similaridad. Esto es muy usado en diferentes aplicaciones como: clasificación de nuevos documentos entrantes al dataset, búsqueda y recuperación de documentos, ya que cuando se encuentra un documento seleccionado de acuerdo al criterio de búsqueda, el contar con un grupo de documentos relacionados, permite ofrecerle al usuario otros documentos que potencialmente son de interés para él.

Antes de agrupar, se debe determinar una medida de similitud o distancia. La medida refleja el grado de cercanía o separación de los objetos de destino y debe corresponder a las características que se cree que distinguen los grupos dentro de los datos. En muchos casos, estas características dependen de los datos o del contexto del problema, y no hay medida que sea universalmente mejor para todos los tipos de problemas de agrupamiento.

Además, elegir una medida de similitud apropiada también es crucial para el análisis de clústers, especialmente para un tipo particular de algoritmos de agrupamiento. Por ejemplo, los algoritmos de agrupamiento basados en densidad, como DBScan, dependen en gran medida del clculo de similitud. La agrupacin basada en la densidad encuentra clústers como áreas densas en el conjunto de datos, y la densidad de un punto dado se estima a su vez como la cercanía del correspondiente objeto de datos a sus objetos vecinos. Esa cercanía se cuantifica como el valor de distancia / similitud, podemos ver la gran cantidad de cálculos de distancia / similitud que son necesarios para encontrar áreas densas y estimar la asignación de clústers de nuevos objetos de datos. Por lo tanto, entender la efectividad de diferentes medidas es de gran importancia para ayudar a elegir la mejor.

Diferentes medidas de similitud no solo dan como resultado diferentes particiones finales, tambin impone diferentes requisitos para el mismo agrupamiento del algoritmo.

Métricas: Ya que no todas las medidas para la distancia pueden ser consideradas métricas, estas deben cumplir 4 condiciones: Sean x ,y los dos objetos en un conjunto y d(x, y) la distancia entre x,y.

- La distancia entre dos puntos cualquiera debe ser no negativa.
- La distancia entre dos objetos debe ser cero si y solo si los dos objetos son iguales.
- La distancia debe ser simétrica, es decir, la distancia desde x a y es lo mismo que la distancia de y a x.
- La medida debe satisfacer la desigualdad del triángulo.

A continuación se discuten cinco medidas de similitud que pueden emplearse para la agrupación de documentos:

Apache Spark: Es un framework de computación de clúster con código abierto. Spark proporciona una interfaz para programar clusters completo son paralelismo de datos implícitos y de tolerancia a fallas.

Distancia euclidiana: Es una métrica estándar para problemas geométricos. Esta mide la distancia ordinaria entre dos puntos y se puede medir fácilmente con una regla en dos o tres dimensiones, es ampliamente utilizada en problemas de agrupacin. Además,es el predeterminado para la medida de distancia utilizada con el algoritmo K-means.

$$D_E(\overrightarrow{t_a}, \overrightarrow{t_b}) = (\sum_{t=1}^{m} |w_{t,a} - w_{t,b}|^2)^{1/2},$$

En donde ta y tb representan los vectores de términos de dos documentos da y db respectivamente.

Similitud coseno: Cuando los documentos se representan como vectores de términos, la similitud de dos documentos corresponde a la correlación entre los vectores. Esto se cuantifica como el coseno del ángulo entre vectores, es decir, la llamada similitud coseno. Dados dos documentos (ta y tb), su similitud coseno está dada por la expresión:

$$SIM_C(\overrightarrow{t_a}, \overrightarrow{t_b}) = \frac{\overrightarrow{t_a} \cdot \overrightarrow{t_b}}{|\overrightarrow{t_a}| \times |\overrightarrow{t_b}|},$$

En donde ta y tb son vectores de dimensión m en el conjunto de términos T = t1,. . . , tm. Cada dimensión representa un término con su peso en el documento, que no es negativo. Como resultado, la similitud del coseno es no negativa y limitada entre [0,1]. Cabe resaltar que la medida de similitud empleando coseno es independiente de la longitud del documento. Por ejemplo, combinando dos copias idnticas de un documento d para obtener un nuevo pseudo documento d', la similitud del coseno entre d y d' es 1, lo cual significa que ambos documentos se consideran idénticos.

Coeficiente de Jaccard: A veces se conoce como el Coeficiente de Tanimoto, mide la similitud como la intersección dividida la unión de los objetos. Para el documento de texto, el coeficiente de Jaccard compara la suma de peso de los términos compartidos al peso de la suma de los términos que están presentes en cualquiera de los dos documentos, pero no son los términos compartidos.

$$SIM_J(\overrightarrow{t_a}, \overrightarrow{t_b}) = \frac{\overrightarrow{t_a} \cdot \overrightarrow{t_b}}{|\overrightarrow{t_a}|^2 + |\overrightarrow{t_b}|^2 - \overrightarrow{t_a} \cdot \overrightarrow{t_b}}.$$

El coeficiente de Jaccard es una medida de similitud y cuyo rango va entre 0 y 1. Es 1 cuando ambos objetos son iguales y 0 cuando son disjuntos.

Coeficiente de correlación Pearson: También mide el grado en que dos vectores están relacionados. A diferenciaa de otras medidas de similitud, su rango va de +1 a -1 y el resultado es 1 cuando ambos objetos son iguales. Hay diferentes formas de calcular el coeficiente de correlación, una de las más utilizadas es:

$$SIM_P(\overrightarrow{t_a}, \overrightarrow{t_b}) = \frac{m \sum_{t=1}^{m} w_{t,a} \times w_{t,b} - TF_a \times TF_b}{\sqrt{[m \sum_{t=1}^{m} w_{t,a}^2 - TF_a^2][m \sum_{t=1}^{m} w_{t,b}^2 - TF_b^2]}}$$

where $TF_a = \sum_{t=1}^{m} w_{t,a}$ and $TF_b = \sum_{t=1}^{m} w_{t,b}$.

IV. Análisis mediante ETL

Para poder generar el diseo del algoritmo distribuido utilizamos la técnica ETL (Extract, Transform, Load), cuyo objetivo principal es facilitar el movimiento de los datos y la transformacin de los mismos, para poder analizarlos y utilizarlos con un fin específico.

En la extracción: Extraemos los datos desde uno o varios sistemas válidos que contengan un dataset origen, con una ruta local desde el HDFS.

En la transformación: Es la transformación de los datos, es decir la posibilidad de reformatear y poder limpiar los datos cuando se necesiten.

En la carga: Es la carga de los datos que ya se tienen transformados en otro lugar o base de datos, para analizarlos y poder guardas estos datos.

V. Implementación

La solución del problema se dividió en dos subproblemas (como fue sugerido). De acuerdo a las recomendaciones del profesor del curso, para entender y familiarizarse con el problema se comenzó a realizarel algoritmo distribuido; luego se procedió a paralelizar este.

A continuación, enunciaremos las librerías y las funciones que se requirieron para la implementación del algoritmo distribuido. Se debe de tener en cuenta que se utilizó Apache Spark

Entre las librerías que se utilizan son:

IDF: Permite filtrar los términos que no están apareciendo en un número de documentos. Ayuda para computar la frecuencia inversa entre los documentos.

K-Means: Es el método de agrupamiento, el cual su objetivo es la partición de un conjunto de K grupos, cuyo valor es más cercano.

KMeansModel: Es un modelo de agrupamiento para K-means. Cada documento pertenece al clúster con la similitud más cercana.

SparkContext: Es el punto de entrada principal para la funcionalidad de Spark. Nos ayuda a la conexión a un clúster. Carga la configuración desde las propiedades del sistema.

Rendint: Es el modelo que sirve para generar el número de los random.

HashingTF: Mapea un conjunto de términos para poder obtener la frecuencia de los términos.

Es importante tener en cuenta que se debe de identificar cual es la ruta en la cual se va a guardar los resultados que se obtengan

ruta = sys.argv[1]

El segundo parámetro que se debe de tener en cuenta es el K, que nos indican cuantos androides son en los que se van a trabajar

k = int(sys.argv[2])

El tercer parámetro es definir la cantidad máxima de iteraciones que queremos que sean con las que se trabajan, la cual es un parámetro que nos obliga a tener presente Spark.

maximoIter = int(sys.argv[3])

Se empieza creando el contexto de Spark, para la cual se utiliza la función que la misma ofrece.

sc = SparkContext(appName="Proyecto04")

Se leen todos los archivos de la carpeta de la carpeta ingresada como parámetro

documentos = sc.wholeTextFiles(ruta)

Para calcular los tf podemos utilizar la fórmula, que nos indicas. Esta fórmula lo que explica es la forma de hallar el termino t que está en el documento d.

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

hashingTF = HashingTF()

tf = hashingTF.transform(docs)

Separarmos por palabras que sean similares los documentos en los diferentes dataset

docs = documentos.values().map(lambda doc: doc.split(" "))

Para medir que tan relevantes son los términos en el cluster y saber cuál es el resultado de la multiplicación entre los hashingTF y los tf.

idf = IDF().fit(tf)

tfidf = idf.transform(tf)

Para la construcción del modelo y la agrupación de los datos con K-Means. Nos ayuda a obtener el modelo que vamos a trabajar, nos muestra la lista de la agrupación de los resultados y se crea un diccionario con dos listas.

clusters = KMeans.train(tfidf,k,maxIterations=maximoIter)

clustersid = clusters.predict(tfidf).collect()

diccionario = dict(zip(nombreDocumentos, clustersid))

VI. Implementación en clúster Spark en producción

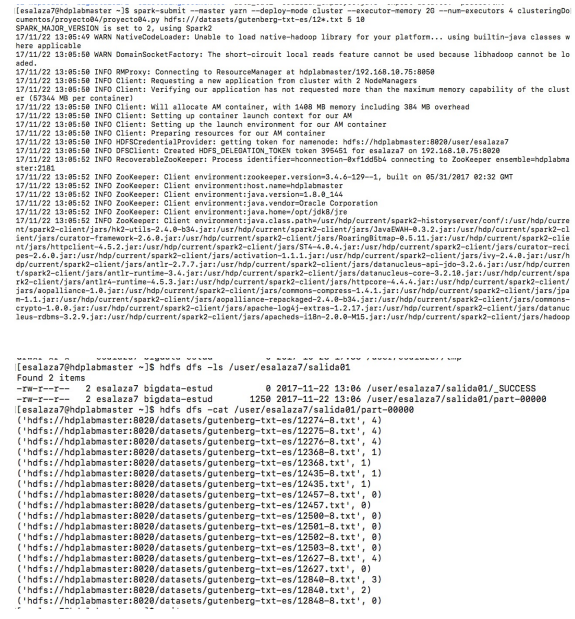
Se realiza la clonación del repositorio, cambiando la ruta de donde se va ha extraer el dataset.

d = sc.parallelize(diccionario.items())

Luego lo ejecutamos

d.coalesce(1).saveAsTextFile("hdfs:///user /abedoy19/salida/salida")

VII. Análisis de resultados



```
[esalaza7@hdplabmaster ~]$ spark-submit --master yarn --deploy-mode cluster --executor-memory 2G --num-executors 4 clusteringDo
cumentos/proyecto04/proyecto04.py hdfs:///datasets/gutenberg-txt-es/12*.txt 5 10
SPARK_MAJOR_VERSION is set to 2, using Spark2
17/11/22 13:05:49 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes w
here applicable
17/11/22 13:05:50 WARN DomainSocketFactory: The short-circuit local reads feature cannot be used because libhadoop cannot be lo
aded.
17/11/22 13:05:50 INFO RMProxy: Connecting to ResourceManager at hdplabmaster/192.168.10.75:8050
17/11/22 13:05:50 INFO Client: Requesting a new application from cluster with 2 NodeManagers
17/11/22 13:05:50 INFO Client: Verifying our application has not requested more than the maximum memory capability of the clust
er (67344 MB per container)
17/11/22 13:05:50 INFO Client: Will allocate AM container, with 1408 MB memory including 384 MB overhead
17/11/22 13:05:50 INFO Client: Setting up container launch context for our AM
17/11/22 13:05:50 INFO Client: Setting up the launch environment for our AM container
17/11/22 13:05:50 INFO Client: Preparing resources for our AM container
17/11/22 13:05:50 INFO HDFSCredentialProvider: getting token for namenode: hdfs://hdplabmaster:8020/user/esalaza7
17/11/22 13:05:50 INFO DFSClient: Created HDFS_DELEGATION_TOKEN token 395461 for esalaza7 on 192.168.10.75:8020
17/11/22 13:05:52 INFO RecoverableZooKeeper: Process identifier=hconnection-0xf1dd5b4 connecting to ZooKeeper ensemble=hdplabma
ster:2181
17/11/22 13:05:52 INFO ZooKeeper: Client environment:zookeeper.version=3.4.6-129--1, built on 05/31/2017 02:32 GMT
17/11/22 13:05:52 INFO ZooKeeper: Client environment:host.name=hdplabmaster
17/11/22 13:05:52 INFO ZooKeeper: Client environment:java.version=1.8.0_144
17/11/22 13:05:52 INFO ZooKeeper: Client environment:java.vendor=Oracle Corporation
17/11/22 13:05:52 INFO ZooKeeper: Client environment:java.home=/opt/jdk8/jre
17/11/22 13:05:52 INFO ZooKeeper: Client environment:java.class.path=/usr/hdp/current/spark2-historyserver/conf/:/usr/hdp/curre
nt/spark2-client/jars/hk2-utils-2.4.0-b34.jar:/usr/hdp/current/spark2-client/jars/JavaEWAH-0.3.2.jar:/usr/hdp/current/spark2-cl
ient/jars/curator-framework-2.6.0.jar:/usr/hdp/current/spark2-client/jars/RoaringBitmap-0.5.11.jar:/usr/hdp/current/spark2-clie
nt/jars/httpclient-4.5.2.jar:/usr/hdp/current/spark2-client/jars/ST4-4.0.4.jar:/usr/hdp/current/spark2-client/jars/curator-reci
pes-2.6.0.jar:/usr/hdp/current/spark2-client/jars/activation-1.1.1.jar:/usr/hdp/current/spark2-client/jars/ivy-2.4.0.jar:/usr/h
dp/current/spark2-client/jars/antlr-2.7.7.jar:/usr/hdp/current/spark2-client/jars/datanucleus-api-jdo-3.2.6.jar:/usr/hdp/curren
t/spark2-client/jars/antlr-runtime-3.4.jar:/usr/hdp/current/spark2-client/jars/datanucleus-core-3.2.10.jar:/usr/hdp/current/spa
rk2-client/jars/antlr4-runtime-4.5.3.jar:/usr/hdp/current/spark2-client/jars/httpcore-4.4.4.jar:/usr/hdp/current/spark2-client/
jars/aopalliance-1.0.jar:/usr/hdp/current/spark2-client/jars/commons-compress-1.4.1.jar:/usr/hdp/current/spark2-client/jars/jpa
n-1.1.jar:/usr/hdp/current/spark2-client/jars/aopalliance-repackaged-2.4.0-b34.jar:/usr/hdp/current/spark2-client/jars/commons-
crypto-1.0.0.jar:/usr/hdp/current/spark2-client/jars/apache-log4j-extras-1.2.17.jar:/usr/hdp/current/spark2-client/jars/datanuc
leus-rdbms-3.2.9.jar:/usr/hdp/current/spark2-client/jars/apacheds-i18n-2.0.0-M15.jar:/usr/hdp/current/spark2-client/jars/hadoop
```

```
[esalaza7@hdplabmaster ~]$ hdfs dfs -ls /user/esalaza7/salida01
Found 2 items
-rw-r--r--   2 esalaza7 bigdata-estud          0 2017-11-22 13:06 /user/esalaza7/salida01/_SUCCESS
-rw-r--r--   2 esalaza7 bigdata-estud       1250 2017-11-22 13:06 /user/esalaza7/salida01/part-00000
[esalaza7@hdplabmaster ~]$ hdfs dfs -cat /user/esalaza7/salida01/part-00000
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12274-8.txt', 4)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12275-8.txt', 4)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12276-8.txt', 4)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12368-8.txt', 1)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12368.txt', 1)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12435-8.txt', 1)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12435.txt', 1)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12457-8.txt', 0)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12457.txt', 0)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12500-8.txt', 0)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12501-8.txt', 0)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12502-8.txt', 0)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12503-8.txt', 0)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12627-8.txt', 4)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12627.txt', 0)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12840-8.txt', 3)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12840.txt', 2)
('hdfs://hdplabmaster:8020/datasets/gutenberg-txt-es/12848-8.txt', 0)
```

Se explican los resultados que se obtuvieron en la ejecución del algoritmo.

El nombre del archivo que está en el clúster

El número del clúster en el que se encuentra el documento

VIII. Conclusiones

- Con este trabajo se aprendieron cosas que eran desconocidas para nosotros como lo era Apache Spark, para lo cual fue de mucha utilidad el curso del Mooc que se realizó.

- Se logró tener un acercamiento a los problemas reales que soluciona el campo de Big Data.

- Poder entender el proceso de ETL, cuando se está utilizando para el diseo y análisis de los algoritmos distribuidos. Es un proceso de mucha importancia cuando se está trabajando con volúmenes muy altos de datos y se necesita mejorar el rendimiento, nos ayuda a dividir los documentos y hacer un análisis mucho más rápido y eficiente, utilizando el paralelismo.

- Se observa la diferencia al realizar un algoritmo distribuido y uno de paralelo, su ejecución en tiempo es significativa.

- Se encontró que la paralelización es la mejor alternativa para solucionar problemas que requieran un tiempo de espera alto, pues esta técnica permite ejecutar procesos a la vez en un menor tiempo posible.

References

[1] Anna Huang. Similarity measures for text document clustering. Proceedings of the Sixth New Zealand, (April):4956, 2008.

[2] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. Introduction to Infor- mation Retrieval. Cambridge University Press, 2008.

[3] Renienj. Implementacin del mtodo de Jaccard. Available at: https://gist.github.com/arieznca hyo/7fa9c0a88b474a1b5f3b72e4d9650292. 29 Sept. 2017.

[4] Bistaumanga. Implementación de K-means. Available at: https://gist.github.com/bistaumanga/6023692. 29 Sept. 2017.

[5] Mpi4py.scipy.org. (2017). Tutorial MPI for Python 2.0.0 documentation. Available at: https://mpi4py.scipy.org/docs/usrman/tutorial.html. 4 Oct. 2017.

[6] Honarkhah, M and Caers, J. (2017). K-means https://es.wikipedia.org/wiki/K-means 30 Oct. 2017.

[7] https://spark.apache.org/docs/2.0.2/api/java/org/apache/spark/SparkContext.html

[8] AnonymousUser. Introducción a Apache Spark. Capítulo 1. http://reader.digitalbooks.pro/book/preview/41061/capitulo$_1$.$xhtmlFeatureExtrac$ $RDD-basedAPI-Spark2.2.0Documentation, Spark.apache.org$, 2017.[$On$ $https : //spark.apache.org/docs/2.2.0/mllib - feature - extraction.html.[Accessed : 01 - Nov - 2017]$.

[9] Apache/spark, GitHub, 2017. [Online]. Available: https://github.com/apache/spark/blob/master/examples/src/main/python/mllib/tf$_i df_e xample.py.[Accessed : 01 - Nov - 2017].Apache/spark, GitHub, 2017.[Online].Available : https : //github.com/apache/spark/blob/master/examples/src/main/python/mllib/$ $19 - Nov - 2017]$.