Análisis de sentimientos con Hadoop

Sistemas Distribuidos de Procesamiento de Datos

Descripción

El análisis de sentimientos es un área de aplicación que está cobrando bastante importancia en los últimos años. La idea es utilizar un programa que automáticamente extraiga emociones de un texto, para poder obtener actitudes y sentimientos existentes en la red, ya sea en blogs, comentarios, foros, Twitter etc… Así se pueden rastrear opiniones sobre productos, marcas, personas para determinar si están bien o mal vistos en la red o hasta establecer la sensación o actitud de un segmento de la red.

Para poder obtener un análisis realista se suele trabajar sobre un gran volumen de información de forma que se cuente con un número elevado de datos sobre los que realizar el análisis. En este sentido suele ser bastante interesante utilizar técnicas de paralelismo escalables, cómo MapReduce, para su ejecución. En concreto, en esta práctica se abordará el problema de implementar un programa de análisis de sentimientos que utilice datos de Twitter y trabaje con ellos utilizando el algoritmo MapReduce.

Objetivos parciales

1. El primer paso es ser capaz de obtener un gran conjunto de datos de Twitter a analizar. Para ello, se puede acceder al stream de los últimos *tweets* públicos realizados. Aunque sólo Twitter tiene una serie de restricciones del acceso a los mismos (https://dev.twitter.com/rest/public/rate-limiting), si se deja ejecutando durante cierto tiempo se puede conseguir un tamaño elevado de *tweets* a analizar.

   Para poder acceder a la transmisión por streaming desde Python se necesita:

   a. Instalar la biblioteca oauth2 de Python para poderse autenticar correctamente (se recomienda utilización de Python 2.7)

      conda install -c chuongdo oauth2

   b. Crearse una cuenta de Twitter, sino se cuenta con ella.

   c. Ir a https://dev.twitter.com/apps e iniciar sesión de Twitter.

   d. Hacer clic en "Create New app", y:

      i. Rellenar el formulario y aceptar las condiciones. Poner en una página web ficticia si se prefiere.

      ii. En la página siguiente, desplazarse hacia abajo y hacer clic en "Create my access token". Con el token se puede acceder al stream de twitter

e. En el fichero proporcionado “twitterstream.py”, establecer las variables correspondientes a consumer key, consumer secret, access token, y access secret.

```
access_token_key = "<Enter your access token key>”
access_token_secret = "<Enter access token secret>”
…
consumer_key = "<Enter consumer key>"
consumer_secret = "<Enter consumer secret>"
```

f. Ejecutar “python twitterstream.py” para ver que todo funciona correctamente (es decir se obtienen datos del stream de Twitter en formato json). Se puede detener su funcionamiento con Ctrl-C.

g. Redirigir el comando anterior (python twitterstream.py > file.json) a un archivo dejando bastante tiempo para almacenar en el mismo una muestra grande de datos de Twitter antes de terminar el programa. Este archivo debe ser situado en un bucket de S3 o en HDFS (en el cluster de Cloudera o EMR) para poder trabajar posteriormente con el mediante mapReduce.

2. Para realizar un análisis de sentimientos sencillo, es bastante habitual basarse en estudios previos que asignan una valoración de sentimiento (por ejemplo positivo, negativo en diferente grado) a cada palabra. En este sentido, se pretende deducir el sentimiento de cada *tweet*, basándose en las puntuaciones obtenidas por cada una de las palabras del mismo. El sentimiento de un *tweet* será equivalente a la suma de las puntuaciones de sentimiento para cada término encontrado en el *tweet*. Para la realización de la práctica se pueden utilizar 2 estudios de partida diferentes.

   a. Uno de los estudios de sentimientos más utilizados se conoce como AFINN. En concreto, el archivo proporcionado AFINN-111.txt contiene una lista de las puntuaciones de sentimiento pre-calculados para términos en inglés. Cada línea del archivo contiene una palabra o frase seguida de un valor de sentimiento que va desde -5 a 5 (más negativo a más positivo). Se puede consultar el archivo AFINN-README.txt para más información

   b. Para el idioma español, se puede utilizar la traducción del estudio ANEW, que en su versión española se denomina SPANEW. Para ello, se ha proporcionado un archivo denominado Redondo_words.txt en el cual se encuentran las palabras en español analizadas estableciendo un valor decimal entre 0 y 10 (0 más negativo, 10 más positivo).

   Cada palabra o frase que se encuentra en un *tweet*, pero no en los ficheros indicados se debe dar una puntuación de 0. Esto provoca que si por ejemplo, el *tweet* no está en el idioma analizado, su puntuación obtenida sea 0. Para utilizar los datos en el archivo:

   - Será necesario abrir un fichero adicional en el mapper. Para poder acceder al mismo será necesario incluirlo en la opción –files en hadoop jar. Par ver el fichero situado en el directorio actual puede ser recomendable hacer:

```
import sys
sys.path.append('.')
```

- Para gestionar los ficheros puede que resulte útil construir un diccionario. Tened en cuenta que el formato de los archivos está delimitado por tabuladores, lo que significa que el término y la puntuación son separados "\t" .El siguiente fragmento puede ser útil:

```
file = open("file.txt")
scores = {} # initialize an empty dictionary
for line in file:
        term, score  = line.split("\t")  # The file is tab-delimited.
        scores[term] = int(score)     # Convert the score to an integer
                                      # In case of Redondo_words.txt,
                                      #a float should be used
        print scores.items()          # Print every (term, score) pair in
                                      # the dictionary
```

3. Con la información de sentimiento de cada *tweet*, se pretende obtener la felicidad agregada de diferentes grupos de personas. En concreto, se pretende obtener la felicidad de cada división provincial o autonómica en España o división estatal en Estados Unidos (en función del idioma utilizado en el apartado anterior) utilizando un algoritmo mapReduce escrito en Python.

   La función json.loads analiza una cadena de JSON. Podéis consultar la documentación de Twitter con el fin de determinar los campos a analizar.

   https://dev.twitter.com/overview/api

   Adicionalmente, para saber la localización, se puede utilizar el objeto Places del *tweet* indicado.

   https://dev.twitter.com/overview/api/places

   En el mismo hay diferentes opciones que se pueden utilizar para determinar su origen.

   1. Pais (country_code), nombre y estado o provincial (full_name, …)
   2. Las coordenadas del objeto

   Podéis desarrollar vuestra propia estrategia para determinar donde se ha realizado cada *tweet* pero se recomienda limitar los *tweets* a analizar a los de los países de interés (España o Estados Unidos). De igual forma, la utilización de servicios de terceros para resolver la localización puede ser complicada por lo que se recomienda no basar la localización en las coordenadas.

   Adicionalmente, hay que entender que trabajar con datos reales presenta bastantes dificultades. Por ejemplo, no todos los *tweets* tendrán una clave de texto o place (los datos reales son así). En este sentido, estad preparados para depurar, y no dudéis en quitar los *tweets* que no podáis manejar. Por ejemplo, mensajes de Twitter de habla no analizada.

4. De forma optativa, adicionalmente al trabajo desarrollado, se pueden plantear soluciones de análisis de *tweets* que profundicen un poco más en la ejecución de algoritmos mapReduce. Por ejemplo se pueden plantear ejemplos como:

    a. Plantear un escenario en el cual se pueda obtener la división provincial o estatal con la gente más feliz. Como resultado, el programa debería imprimir el nombre o abreviatura de dicha división.

    b. Plantear el cálculo de las 10 palabras *trending topic*, cuyo análisis tiene su origen en los *hashtag* (etiquetas precedidas por #) que sirven para organizar *tweets* sobre un tema concreto.

Nota: No se debe hacer un programa separado para cada objetivo, sino un único programa genérico que cumpla con todos los objetivos simultáneamente. La fase de ejecución debe quedar reflejada en un archivo .txt que se entregara junto al código de la práctica desarrollado)

Configuración de la cuenta AWS

El proyecto se puede desarrollar en un ordenador personal (realizando comprobación de su funcionamiento mediante pipes que unan los mappers y reducers) sin embargo, se deberá ejecutar finalmente para evaluar prestaciones en un servicio distribuido como Cloudera o Amazon Web Services (AWS) Elastic Map Reduce (EMR). Para esto último los alumnos pueden configurarse una cuenta AWS. Para su creación, se recomienda utilizar la cuenta de correo @alumnos.urjc.es para poder obtener en las ventajas del programa AWS Educate (100$ adicionales de gasto) al cual pertenece la Universidad Rey Juan Carlos. Esto implicará asociar su propia tarjeta de crédito a dicha cuenta pero AWS no cobra ningún coste por defecto, es decir no hay coste por mantener la cuenta abierta sin usar ningún servicio. Adicionalmente el primer año de creación se cuenta con una capa gratuita que permite la utilización de recursos sencillos pero suficientes para poder mostrar la ejecución en un entorno cloud real. Sin embargo, se recomienda tener especial cuidado con los recursos utilizados (apagándolos cuando no sean necesarios entre otras cosas y seleccionado aquellos recursos asociados a la capa gratuita) para no incurrir en gastos innecesarios.

Para inscribirse es necesario ir a la página http://aws.amazon.com y pinchar en “Inscríbase”. Será necesario introducir el email para crear una nueva cuenta, e introducir los datos personales y los datos de facturación.

Amazon puede realizar una verificación de identidad realizando una llamada telefónica. Será necesario introducir un número de teléfono al cual llamaran para verificar el PIN introducido en la página. Adicionalmente, puede requerir el envío de un fax para comprobar la dirección. Una vez realizado ya podréis acceder a la consola de AWS https://aws.amazon.com/es/console/

Una vez que se cuente con acceso a la consola se pueden obtener 100$ adicionales para su uso durante el primer año a los alumnos de la Universidad. Para ello es necesario ir a https://aws.amazon.com/education/awseducate/ y unirse a AWS Educate en el apartado de Estudiantes. Se requerirá todos los datos de contacto, incluyendo el nombre correcto de la

Universidad y el mail de alumno de la misma. En el tercer paso se solicita el Id de cuenta de AWS. Se debe seleccionar que se cuenta ya con una cuenta AWS y especificar indicar el identificador numérico que podéis encontrar en la consola de AWS en el apartado de "Mi cuenta" bajo el dato Account id.

Ejecución en AWS EMR

En la consola AWS, se puede crear un cluster y configurar sus pasos de ejecución seleccionando "Add step" por cada paso que sea necesario. Al finalizar se terminará el cluster

- Elegir Streaming como tipo de paso.
- En Mapper, seleccionar el código que tiene que estar en AWS S3
- En Reducer, seleccionar el código que tiene que estar en AWS S3
- En input seleccionar la ubicación de los datos de entrada en AWS S3
- En output, seleccionar la ubicación de salida en AWS S3. No debe existir el directorio
- Arguments: se pueden incluir parámetros adicionales como -cacheFile. Para poner ficheros adionales en una cache distribuida.

Igualmente se puede utilizar mrJob configurando las claves de acceso a AWS

Entrega de prácticas

La entrega de prácticas se hará a través de aulaVirtual en las fechas anunciadas en el mismo. Se debe entregar un fichero .zip incluyendo el código realizado (archivos .py necesarios debidamente comentado y un archivo .txt incluyendo los pasos necesarios para su correcta ejecución) y una memoria de la misma en formato pdf. La memoria debe incluir:

- Índice de contenidos
- Autor/es
- Descripción del código: incluyendo descripción de las principales funciones implementadas. No se debe incluir código fuente
- Diferentes formas de ejecución realizadas incluyendo los parámetros utilizados.
- Evaluación en la medida de los posible de escalabilidad y elasticidad de la solución propuesta.
- Comentarios personales: incluyendo problemas encontrados, críticas constructiva, propuesta de mejoras y evaluación del tiempo dedicado.

Evaluación de la práctica

La práctica se evaluará comprobando el correcto funcionamiento de los distintos objetivos, y valorando la simplicidad del código, los comentarios, la óptima gestión de recursos y la calidad de la memoria. Se podrá solicitar una defensa oral de la práctica si lo considerase necesario.

Autoría de la práctica

La práctica se realizara en grupos de 3 personas siempre y cuando incluyan a gente de diferentes perfiles (p.ej. perfil informático y matemático).