Módulo 15: NLP

v.1

Práctica de NLP

Javier Serrano, Jorge Cubero • May 17, 2015

Ejercicio - Módulo 15

1. Introducción

El objetivo de esta práctica es doble, por un lado responder al segundo ejercicio propuesto para el módulo 15 del máster de data science en la UTAD y por otro el servir de prólogo introductorio al proyecto final de máster, ya que en ambos casos se trabajará sobre el análisis de sentimiento en twitter, aunque enfocado en distintos objetivos.

Por un lado el presente trabajo que da respuesta a la actividad de evaluación se centrará en el análisis de sentimiento respecto a un partido político, en este caso se ha escogido el PSOE y como términos de búsqueda se ha utilizado: “PSOE”. Se ha escogido este por simplicidad, ya que no tiene tanta necesidad de desambiguación como otros partidos políticos que podrían necesitar añadir términos para desambiguar algunas de sus siglas (por ejemplo PP o Podemos), pero este no es el caso del término PSOE.

2. Información disponible

Desde que el objeto del análisis son tweets, hay que destacar que no es información 100% desestructurada, sino que con cada tweet nos descargamos unos metadatos, relativos al lenguaje, fecha de publicación, autor, etc. Que resulta también muy útiles a la hora de elaborar una metodología de análisis.

Además un factor propio de las redes sociales es el grado de transmisión de un mensaje, o cuántas veces un mensaje es repetido por un conjunto de personas. En este caso, como veremos más adelante, nos descargamos todos los tweets relativos a un tema, sin importar si estos están repetidos o no, y debemos ser conscientes de cómo queremos manejar dicha multiplicidad. En este caso hemos escogido trabajar sobre tweets individuales y ponderar los resultados por su multiplicidad.

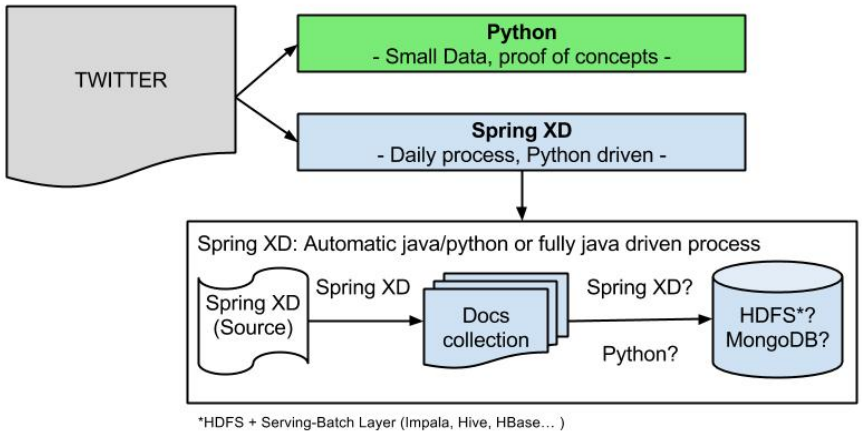
En este caso y utilizando la arquitectura descrita en el punto 3, hemos capturado tweets entre las 11:14:35 y las 11:58:41 del día 2015-05-17 buscando el término “PSOE”. En concreto se han capturado 13913 tweets de los cuáles sólo 1240 son diferentes, el resto son retweets. Esto equivale a una frecuencia de publicación de tweets (nuevos o no) de unos 5 tweets por segundo de media lo que equivaldría a más de 400k tweets diarios. Teniendo en cuenta que el archivo generado es de unos 59Mb, tendríamos que al cabo del día se generarían unos 2Gb de datos, siempre suponiendo que la tasa de publicación para tweets que contengan el término PSOE se mantiene constante, lo cuál en la práctica no es asumible.

3. Flujo de trabajo

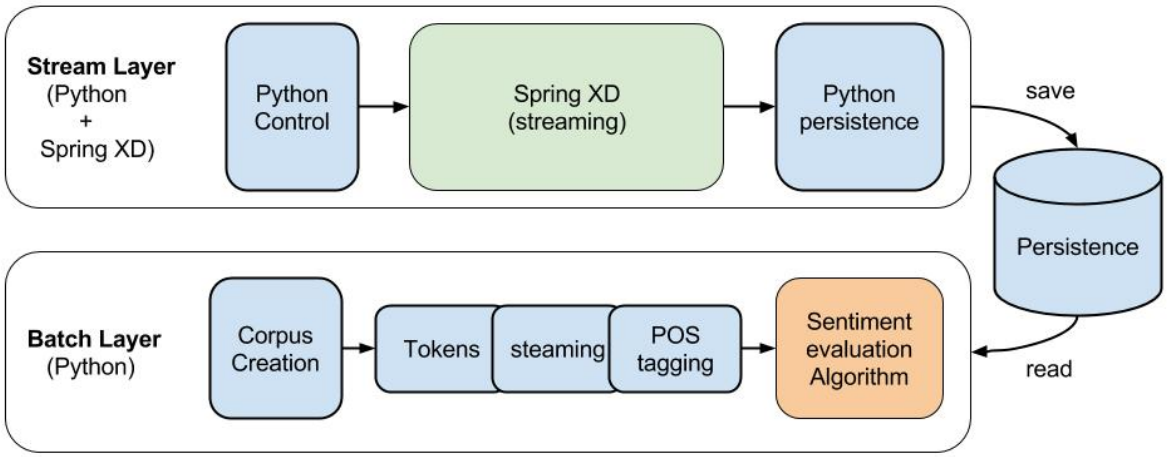
Debido a esta demanda de datos, sabiendo que para utilizar hadoop una partición de archivos en bloques de 128Mb es la más común, si estimamos unos 2Gb por día, tenemos que aproximadamente ocuparían 16 bloques de memoria en una arquitectura de ficheros HDFS. Por lo tanto juzgamos adecuado su uso para un problema real de análisis de sentimiento en twitter.

Sin embargo, en la práctica se trabaja usualmente con un lenguaje de desarrollo rápido (scripting) como python, para desarrollar la metodología que luego se aplicará en el proceso productivo en NRT (near real-time).

A la derecha vemos un esquema del uso de ambos proceso, y profundizamos en la captación de la información en NRT.



Veamos cómo sería un entorno mixto de desarrollo/producción. En concreto tendremos 2 procesos ejecutándose en paralelo en un sistema de este tipo, un proceso NRT que denominamos proceso stream y un proceso batch:



En este caso se ha escogido utilizar Spring XD como framework principal para montar la capa de stream, creando desde python el stream de datos y utilizando python para dirigir los mecanismos de persistencia, aunque Spring XD ofrece la posibilidad de realizar persistencia directamente en DBs distribuidas como MongoDB o bien en HDFS. También ofrece la posibilidad de desarrollar módulo (paquetes java) con clases que nos permitan preprocesar la información antes de almacenarla, por si se desearan por ejemplo aplicar filtros a los tweets en real time.

Dentro de la capa batch utilizaríamos algunas herramientas también desarrolladas en java y publicadas por el grupo de NLP de la universidad de Stanford, en concreto, el framework de POS tagging disponible en: http://nlp.stanford.edu/software/index.shtml Aunque como se pueden apreciar existen más herramientas en esta misma web para análisis más complejos. En este caso hemos optado por esta herramienta ya que dispone de taggers para idioma español y son fácilmente usables y configurables desde python.

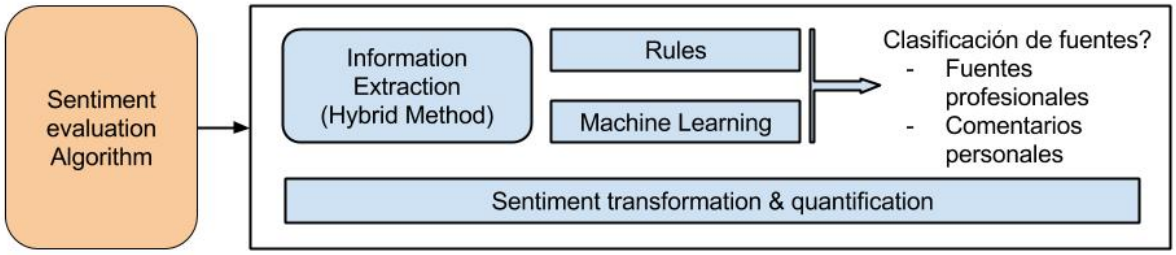
En cuanto al algoritmo de evaluación del sentimiento, nos referimos a algoritmos de procesos de extracción de la información. En concreto lo más factible sería dividir las fuentes de tweets según tipología, distinguiendo al menos los casos siguientes:

- Fuentes Oficiales: dentro de estas fuentes, englobamos: medios de comunicación, organismos oficiales, partidos políticos, etc.
- Opiniones personales: Todas las que no pertenecen a las entidades del punto anterior.

Hemos distinguido ambos tipos de fuentes ya que las primeras suelen presentar información mejor estructurada sintácticamente, con frases completas y buen vocabulario y con menor grado de figuras literarias que dificultan su análisis semántico (metáforas, símiles, etc.), en general utilizan un lenguaje más objetivo (prescindiendo por ejemplo de la ironía que pueden tener usuarios del segundo grupo) y más fácil de analizar por métodos de machine learning.

Mientras que los del segundo grupo probablemente requieren un enfoque más híbrido, definiendo reglas gramaticales para la interpretación de los mensajes, etc. mientras se aplican métodos de machine learning para clasificar los mensajes, incluso podríamos usar por ejemplo random forest para separar los mensajes según las reglas que apliquen.

Además en paralelo tendríamos que medir el sentimiento en función de las reglas que cumplan, es decir, el diseño del apartado “Sentiment Evaluation Recognition” sería más o menos el siguiente:

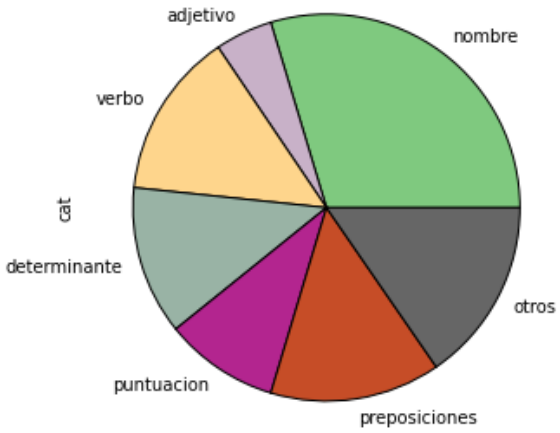


4. Pruebas de Herramientas y Resultados

Con esta memoria se adjunta también el archivo creado con Spring XD (tweetsearch.out) y un script que cubriría la mayor parte del trabajo en la capa batch, sin incluir los algoritmos de sentiment evaluation analysis, ni extracción de la información.

En este nivel ya se pueden aplicar algunos algoritmos sencillos utilizando el POS tagging, por ejemplo un primera clasificación de nombres con connotaciones positivas y negativas, etc. Sin embargo, por el uso de ironía y construcciones complejas, probablemente algoritmos de IE que se basen en relaciones entre entidades ofrezcan una mejora en los resultados sobre todo visto el alto contenido de nombre y verbos.

A la derecha vemos la distribución de palabras al aplicar POS tagging a los primeros 100 tweets (eliminando referencias a usuarios, hashtags y urls).



No obstante este método no está exento de errores, por ejemplo: las palabras “corrupto” y “culto” se clasifican aquí como nombres, aunque en el contexto están funcionando como adjetivos.

Una vez realizado este análisis, vemos que podríamos optar por analizar en más profundidad nombres y verbos, pero el análisis de adjetivos debido a su baja frecuencia no parece de alta utilidad.