News Trends Processing Using Open Linked Data

Antonio Garrote
Universidad de Salamanca, Spain

María N. Moreno García
Universidad de Salamanca, Spain

EXECUTIVE SUMMARY

In this chapter we describe a news trends detection system built with the aim of detecting daily trends in a big collection of news articles extracted from the web and expose the computed trends data as open linked data that can be consumed by other components of the IT infrastructure. Due to the sheer amount of data being processed, the system relies on big data technologies to process raw news data and compute the trends that will be later exposed as open linked data. Thanks to the open linked data interface, data can be easily consumed by other components of the application, like a JavaScript front-end, or re-used by different IT systems. The case is a good example of how open linked data can be used to provide a convenient interface to big data systems.

ORGANIZATION BACKGROUND

The organization involved in the project is an Internet company providing data analysis services for different kinds of web data. Integration of different data sources capable of generating more useful insights on clients' data is an essential task in the company strategy.

From the organizational point of view, the software development process is accomplished by small and highly autonomous engineering teams responsible for different projects and relying on services provided by other teams. Use of big data technologies means that many times resources like computer clusters must be shared by different teams. This practice demands a high degree of cooperation between teams.

Within this context of small teams building a network of services that are used and combined by other teams, the use of open linked data makes it possible for the easy inter-operability between data resources as well as provides a shared vocabulary for the outcome data, processed by big data systems like Apache Hadoop.

Data management in big data systems is a hard problem. The organization undergoing the development project described in this case generates and processes tera bytes of data on a daily basis. Most of these data have been so far stored as plain tab-separated files in the Hadoop Distributed File System (HDFS) (Shvachko et al., 2010).

Re-using and cataloging the available data sources have been traditionally an important issue, due to the distributed nature of the development teams in the organization. As a consequence, problems like finding if the right information is already available in some part of the cluster file system or if some particular data generation process is still in use have been hard to solve, usually involving a lot of communication overhead between members of different teams.

This situation was slightly improved when a more structured data storage technology like Apache Hive started to be used instead of direct access to plain HDFS files. Hive provides a data abstraction layer in the form of data tables with a certain data schema and a relation SQL-like data retrieval language that can be used on top of the map-reduce platform offered by Hadoop. The use of a schema and an easy interface to query the stored data made it easier for non technical users to retrieve information from the cluster as well as provided a better definition of the available data. However, the problem of finding available data in the cluster remained a problem.

When making available structured information about news trends started to be considered as development project, linked data appeared as a possible alternative to provide a more open interface to the available data stored in the cluster, as well as a mechanism to interlink isolated data sets using well known web technologies like URIs and hyper-links.

CASE DESCRIPTION

The main goal of the project was to make available daily news trends as a structured data source that could be used as an additional input in any data analysis task being performed in the organization. Computation of the news trends was to be achieved in a series of steps involving: