eXtream: a System for Real-time Monitoring of Dynamic Web Sources

Marcos Fernández-Pichel
Rodrigo Martínez-Castaño
David E. Losada
Juan C. Pichel
marcosfernandez.pichel@usc.es
rodrigo.martinez@usc.es
david.losada@usc.es
juancarlos.pichel@usc.es
Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela
15782 Santiago de Compostela, Spain

ABSTRACT

In this work, we introduce eXtream, a Big Data platform whose main goal is to deploy modular and customisable processing topologies for massive analysis of web data in real time. The system offers a reduced group of pre-installed modules that can be easily combined in a visual way. Additionally, an advanced user can upload new modules and extend an existing topology. This tool facilitates the development of many Information Retrieval and Big Data applications, such as query-based real-time filtering or topic analysis services on Social Media data. To demonstrate it, we have also developed an initial web-based demonstrator.

CCS CONCEPTS

• **Information Retrieval** → **Real-time**; • **Text Mining** → Social Media Analytics.

KEYWORDS

Big Data, Real Time, Web Streams, Datasets

1 INTRODUCTION

Processing Social Media data is a challenge and doing it in real time is critical for many added-value applications. For example, according to Twitter[1], the number of daily posted tweets is higher than 500 million (around 5, 787 tweets per second).

eXtream is a Big Data platform for building topologies oriented to real-time processing of web or stream data. It has many potential use cases, such as doing a reputation analysis about a company, its products or its competitors from social media data. It can also be used by Information Retrieval (IR) experts to collect social media texts and create their own datasets.

This tool is built with the Python framework Catenae [1], which has several advantages over other existing technologies (e.g., inherent horizontal scalability and inter-module communication through RPC). However, eXtream goes a step further offering a set of text mining modules which can be easily interconnected using the GUI provided by our web-based demonstrator. As a consequence, users are able to construct data processing topologies avoiding the low level details such as manually deploying Docker containers. Moreover, eXtream has several important differences with respect to other existing frameworks (see Table 1 for details). For instance, our system can combine real-time processing with the capability of doing batch tasks. It also supports Python natively and allows defining cycles among the topology modules.

As said before, eXtream offers a reduced group of pre-installed modules. Among them, we can highlight a query-based filter or a topic analysis module. Note that, in addition, advanced users can easily extend the processing pipelines adding new user-defined modules.

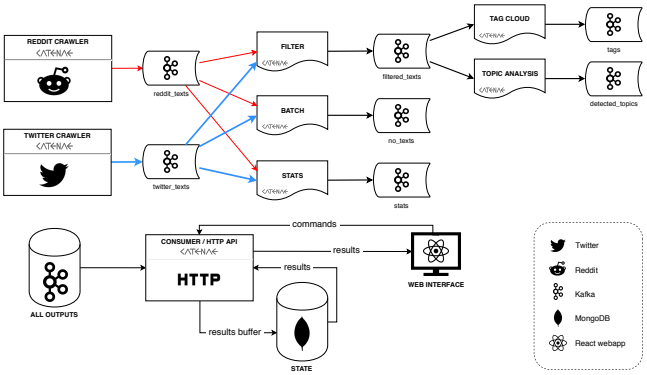

Figure 1: Example of a possible eXtream topology using the pre-installed modules (top) and visualization and persistence pipeline (bottom).

2 SYSTEM OVERVIEW AND IMPLEMENTATION

A topology example that interconnects all the current available pre-installed modules of eXtream is displayed in Figure 1 (top). This is just a simple example of the countless possible configurations. All the topology modules and sources are interconnected using Apache Kafka[2] topics, and the system state is kept in a MongoDB database. It is worth noting that all the implementation and deploying details

[1]https://business.twitter.com/en.html

[2]https://kafka.apache.org/

Technology	Streaming	Python-native	Execution cycles	Easy deployment management	Graphical definition of topologies	Docker oriented	Resource assignment
Hadoop MR	No	No	No	No	No	No	Yes
Spark	Yes	No	No	Yes	No	No	Yes
Storm	Yes	No	Yes	No	No	No	Yes
Stream Parse	Yes	No	Yes	Yes	No	No	Yes
Kafka	Yes	Yes	Yes	No	No	No	No
Kafka Streams	Yes	Yes	Yes	No	No	No	No
eXtream	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

Table 1: Main features of eXtream and other related technologies and frameworks.

are completely hidden to the users of our platform. In particular, the available modules are:

- A **Reddit crawler** [2] and a **Twitter crawler** [3] that inject text streams into the topologies built with eXtream.
- A **real-time filtering** module is essential since it acts as a first distilling step of the data that eXtream receives in real time. It is a query-based filter where the current implementation supports exact and inexact[3] matching. This module can be easily customised to implement any IR filter and it can help IR experts create their own collections.
- A **dynamic tag cloud generator** represents a primitive form of summary. It removes stopwords and normalises the words to build a tag cloud from the resulting bag of words.
- A **topic analysis module** attempts to discover the hidden topics in the texts. We have used Gensim [4] and LDA [5], which perform unsupervised learning over a corpus.
- A **stats module** that returns the number of distinct users and texts that are currently being analyzed by the platform.
- eXtream also supports **batch tasks**. As a first example, we provide a module that counts the number of recovered texts over a certain period.

Furthemore, a module placed at the end of every topology receives the output data (see Figure 1 at the bottom). It also implements a RESTful API in order to visualize results. Figure 2 displays the GUI main view and a possible topology example. It should be noticed that each kind of module has its own *dashboard* or view. For instance, Figure 3 shows the result of searching the exact query *Amazon* in Reddit. Finally, we provide a demonstration video showing eXtream in operation[4].

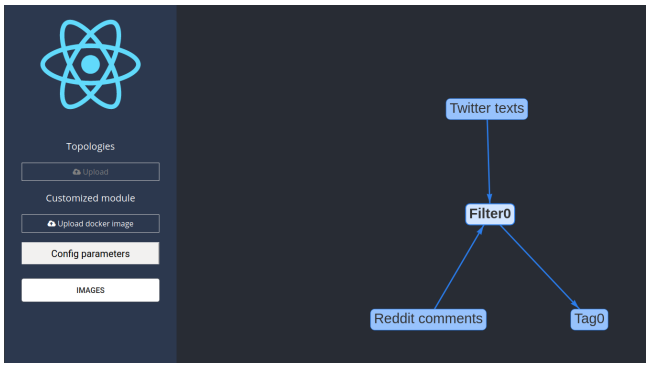

Figure 2: Main view of the eXtream GUI.

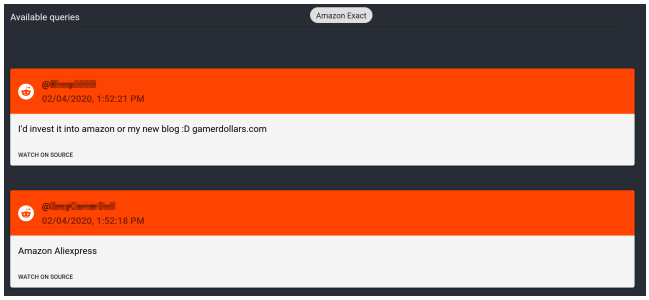

Figure 3: Filter view of the eXtream GUI.

3 CONCLUSIONS

Flexibility is a core strength of this system, which can be customisable for other IR or related purposes with little effort. This tool is available for the research community to expand it and employ it for numerous Information Access tasks[5].

ACKNOWLEDGMENTS

This work was funded by FEDER/Ministerio de Ciencia, Innovación y Universidades – Agencia Estatal de Investigación/ Project (RTI2018-093336-B-C21). This work has received financial support from the Consellería de Educación, Universidade e Formación Profesional (accreditation 2019-2022 ED431G-2019/04, ED431C 2018/29 , ED431C 2018/19) and the European Regional Development Fund (ERDF), which acknowledges the CiTIUS-Research Center in Intelligent Technologies of the University of Santiago de Compostela as a Research Center of the Galician University System.

[3] Stopwords are removed and the remaining words can appear in any order.
[4] https://youtu.be/5Aw4mAc9lTc
[5] https://github.com/MarcosFP97/eXtream

REFERENCES

[1] Rodrigo Martínez-Castaño, Juan C. Pichel, and David E. Losada. Building python-based topologies for massive processing of social media data in real time. In *Proceedings of the 5th Spanish Conference on Information Retrieval*, pages 1–8. ACM, 2018.
[2] Rodrigo Martínez-Castaño, Juan C. Pichel, David E. Losada, and Fabio Crestani. A micromodule approach for building real-time systems with python-based models: Application to early risk detection of depression on social media. In *Advances in Information Retrieval - 40th European Conference on IR Research, ECIR*, volume 10772 of *Lecture Notes in Computer Science*, pages 801–805. Springer, 2018.
[3] Rodrigo Martínez-Castaño, Juan C. Pichel, and Pablo Gamallo. Polypus: a big data self-deployable architecture for microblogging text extraction and real-time sentiment analysis. *CoRR*, abs/1801.03710, 2018.
[4] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50. ELRA, May 2010.
[5] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.