xLiMe – crossLingual crossMedia knowledge extraction

# HANDBOOK

## 1. DATA PROCESSING INFRASTRUCTURE

The xLiMe architecture is build using several technologies. In the following sections, some of the best practices are listed to make xLiMe pipeline work trouble-free for data processing.

### 1.1 KAFKA

The Apache Kafka is the message broker used in xLiMe data processing infrastructure. It is useful for publishing/subscribing to the messages sent by various content providers. For xLiMe, Table 1 shows the recommend system configurations for installing Kafka.

| CPU | RAM | Hard Disk Storage | Ethernet Bandwidth | OS |
|---|---|---|---|---|
| 4 x Intel(R) Xeon(TM) 3.00GHz | 4 GB | 1.2 T | 1Gbit | Debian/Ubuntu Linux |

**Table 1 Kafka System Settings.**

Now to install, configuration settings suggested in the Quick start [1] with single partition is used. Once the Kakfa is setup, topics are created for publishing/subscribing messages. More information about creation of topics and publishing metadata in the Kafka can be found in in the section 4.1 [2] of Deliverable D1.2, while subscription information is provided in the Section 4.2.

### 1.2 VIRTUOSO AS STORAGE

In the earlier section, exchanging data with publish/subscribe mechanism is presented. Most of the applications require storage to use the streaming data. Virtuoso [3] is used as storage. For xLiMe, Table 2 shows the recommend system configurations for installing Virtuoso.

| CPU | RAM | Hard Disk Storage | OS |
|---|---|---|---|
| 8 x 2.0 GHz AMD Opteron | 40 GB | 3TB | Debian/Ubuntu Linux |

**Table 2 Virtuoso System Settings.**

To install Virtuoso, settings suggested can be used [4].

### 1.3 MONGODB AS STORAGE

For storing data generated in the Kafka publish/subscribe mechanism, MongoDB [5] open source version is used as a document store. Messages published in the Kafka by producers are converted into JSON/JSON-LD format by consumers to be further stored into MongoDB document store to support various applications.

For xLiMe, Table 3 shows the recommended system configurations for installing MongoDB.

| CPU | RAM | Hard Disk Storage | OS |
|---|---|---|---|
| 8 x 2.0 GHz AMD Opteron | 10GB | 1TB | Debian/Ubuntu Linux |

**Table 3 MongoDB System Settings.**

Now to install, settings listed for Ubuntu/Debian [6] for MongoDB can be used.

## 2. DATA MANAGEMENT

The data generated from various content providers are first encapsulated into the xLiMe Meta data model. More information about the data model can be found in the Deliverable 1.1 [7]. Once the data is encapsulated into data model, it is pushed into Kakfa for consumption and storage to facilitate various applications.

There are two important modes of storage which is used in xLiMe.

## 2.1 KAFKA TO MONGODB

Messages are sent to the Kafka topics by content providers which are then consumed by their subscribers/clients. Once the messages are collected, they are further pushed into the storage. MongoDB is one such mode where the storage is achieved by converting Kafka messages in xLiMe data model to JSON/JSON-LD format. In the following sections, two different ways of data model conversion is presented.

### 2.1.1 XLIME DATA MODEL TO JSON-LD

There are several existing tools like Jena [8] that converts RDF data into other formats. To convert xLiMe data-model into JSON-LD format the indigenous technology developed in the xLiMe project [9] is used.

### 2.1.2 XLIME DATA MODEL TO JSON

If the message requires further processing and needs to be stored in the JSON format for different applications. Then processing techniques described in the Section 2.1.2 of Deliverable 4.1 [10] can be used and the code for deployment can be found in xLiMe Semantic Integrator [11].

## 2.2 KAFKA TO VIRTUOSO

The Kafka messages in the xLiMe data model format can be stored directly in the Virtuoso without any intermediate processing. A Tomcat [12] based consumer services are built in xLiMe to capture the messages from Kafka to be further stored in Virtuoso. Currently, there are four such services in xLiMe and more information about the implementation of these services can be found at Kafka wiki [13].

## 2.2 BACKUP

In xLiMe, the life time of the data stored in Kafka is 14 days (i.e. around 2 weeks). To have a back-up of data older than 14 days, content is archived by taking a data dump. The backup for data dump is achieved with Kakfa-tools [14].

# 3. DATA INGESTION

Aforementioned sections presented the infrastructure used to host or exchange data. Also, the sections listed the ways to consume data in different formats. In the following sections, producer part of the data exchange is investigated.

## 3.1 ZATTOO DATA PRODUCER

The Zattoo sandbox is one the key providers of data in several formats.

Data in the format of video, audio, subtitles and EPG information from Zattoo is pre-processed and converted into xLiMe data format before it is pushed into the Kafka topic stream. Documentation about using Zattoo sandbox API's for collecting data can be found at the Zattoo developer wiki [15].

## 3.2 JSI DATA PRODUCER

News collections from different languages are one of the important data sources that have different applications in xLiMe. The newsfeed [16] provided by JSI is leveraged to collect news collections for the xLiMe supported languages. More information about the statistics of documents collected every day can be found in Section 2.1.4 of Deliverable 1.3 [17]. Once the news articles are collected, they are further converted into xLiMe data model before pushing into the Kafka topic stream.

## 3.3 VICO DATA PRODUCER

Social media data in xLiMe is collected from VICO. More information about the data that is stored or collected at VICO can be found in the Section 2.1.2 of Deliverable 1.3.

# 4. ANNOTATIONS

Some of the components in the xLiMe pipeline depend on the semantic annotations applied on the textual information identified with entities and their attributes present in the external knowledge bases. There are two important services which provide these annotations in the xLiMe pipeline. All textual resources such as ASR, subtitles, news and social media are the users of these annotation services.

## 4.1 KIT ANNOTATOR

For employing annotations, xLiMe leverages xLiSa [18] for monolingual and cross-lingual annotations. More information about the functionality can be found in the Deliverables 3.3.1 [19] and D3.3.2 [20].

## 4.2 JSI ENRYCHER

Another annotator which is used for annotating the News stream is JSI Enrycher [21]. More information about its usage in xLiMe can be seen in the Deliverables 3.3.1 and 3.3.2.

## 4.3 VIDEO OCR

Extracting textual information from a Video is achieved with an optical character recognition (OCR). This can be seen as a pseudo annotation achieved on visual information describing the content of a video. More information about extracting OCR and related functionality can be seen in the Deliverables 2.2.1 [22] and 2.2.2 [23].

## 4.4 OBJECT DETECTION

In xLiMe, object detection in a video plays a prominent role for cross-modal recommendations and search. There are three important types of objects that are explored to satisfy the use-cases.

### 4.4.1 LOGO DETECTION

Main goal of the logo detection is to identify the deutsche telecom logos that are found in the TV videos. More information about the technology to rebuild the detection system can be found in the Deliverables 3.2.1 [24] and 3.2.2 [25].

### 4.4.2 FLAG DETECTION

The goal of flag detection is to identify the British flags in videos in the context of "Brexit" use-case. More information can be seen in the "Brexit" dataset that will be released.

### 4.4.3 SHOE DETECTION

Aim of the shoe detection in a video is to track the shoes that are shown in the social media streams. More information about it is provided in the Deliverable 7.4.2 [26].

## 5. RECOMMENDATIONS

The goal of the recommendations is to compare the content across the modalities and languages. Modalities such as video are converted into text using speech transcriptions and subtitles, while news and social media content is cleaned to extract only relevant information. In xLiMe, recommendations are provided for two important use-cases as described in deliverables 7.4.2 [26] and 7.2.3 [27].

### 5.1 RECOMMENDATIONS FOR ZATTOO USE-CASES

Most of the recommendation logic is developed in deliverables 4.1 [28] and 4.2 [29] for supporting use-cases of Zattoo.

- The recommendations for the first use-case caters the need of finding the similar social media and News content for a given TV show snippet (40 sec video chunk). More information about the technology developed for this use-case can be found in Deliverable 4.1.
- The recommendations for the second use-case cater the need of finding similar TV show snippets (i.e. 40 sec chunks) for a given News article. More information about the technology developed for this use-case can be found in Deliverable 4.2.

- The recommendations for the third use-case cater the need of finding similar TV programs for a given TV program. More information about the technology developed for this use-case can be found in the Deliverable 7.2.3.

Also, the workflow to collect, integrate and recommend is provided in the xLiMe Semantic Integrator framework [30].

## 5.2 RECOMMENDATIONS FOR ECONDA USE-CASES

Most of the recommendation logic is developed in the Deliverable 5.4 [31] to support the use-case of Econda. One of the goals of this use-case is to find similar categories or brands mentioned in social media which are also present in Deichmann web shop. The outcome and evaluation results for textual recommendation are provided in the Deliverable 7.4.2.

# 6. RECOMMENDATIONS

This appendix provides the workflow followed in the xLiMe pipeline along with their components. Each component in the workflow is well documented and is supported with a deliverable for further reading or for usage.

# 7. REFERENCES

[1] https://kafka.apache.org/quickstart
[2] Deliverable 1.2
[3] http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/
[4] http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VOSDebianNotes
[5] https://www.mongodb.com/scale/database-software-open-source
[6] https://docs.mongodb.com/v3.0/administration/install-on-linux/
[7] Deliverable 1.1
[8] https://jena.apache.org/index.html
[9] https://github.com/rdenaux/xlime-showcase-dataloader
[10] Deliverable 4.1
[11] https://github.com/adityamogadala/xLiMeSemanticIntegrator/tree/master/xlimedataparser
[12] http://tomcat.apache.org/

[13] https://cwiki.apache.org/confluence/display/KAFKA/Consumer+Group+Example
[14] https://cwiki.apache.org/confluence/display/KAFKA/System+Tools
[15] https://developer.zattoo.com/
[16] http://newsfeed.ijs.si/
[17] Deliverable 1.3
[18] http://km.aifb.kit.edu/sites/xlisa/
[19] Deliverable 3.3.1
[20] Deliverable 3.3.2
[21] https://github.com/JozefStefanInstitute/EnrycherAPI
[22] Deliverable 2.2.1
[23] Deliverable 2.2.2
[24] Deliverable 3.2.1
[25] Deliverable 3.2.2
[26] Deliverable 7.4.2
[27] Deliverable 7.2.3
[28] Deliverable 4.1
[29] Deliverable 4.2
[30] http://adityamogadala.github.io/xLiMeSemanticIntegrator/
[31] Deliverable 5.4