Deliverable D6.2.2

Demonstrator Prototype

Editor:	Esteban Garcia-Cuesta, iSOCO
Author(s):	Esteban García-Cuesta (iSOCO), Andrej Muhic and Jan Rupnik (JSI), Mitja Trampus (JSI), Zhixing Li (THU), Xavier Carreras (UPC)
Deliverable Nature:	Prototype (P)
Dissemination Level: (Confidentiality)	Public (PU)
Contractual Delivery Date:	M24
Actual Delivery Date:	M24
Suggested Readers:	All partners of the XLike project consortium and end-users
Version:	1.0
Keywords:	Demo, prototype, end-users, cross-lingual, dissemination, validation, entity tracking, article tracking.

Disclaimer

Full Project Title:	Cross-lingual Knowledge Extraction
Short Project Title:	XLike
Number and Title of Work package:	WP6 – Integration and Toolkit
Document Title:	D6.2.2 – Demonstrator Prototype
Editor (Name, Affiliation)	Esteban Garcia-Cuesta iSOCO
Work package Leader (Name, affiliation)	Esteban García-Cuesta, iSOCO
Estimation of PM spent on the deliverable:	5 PM

Executive Summary

This document presents the demonstrator prototype which is the updated implementation of the year one early prototype. The demonstrator has adapted the early prototype to the new architecture specifications for the integration of the different modules provided by WP1, WP2, WP3, WP4 and WP5. The work done at each one of these modules is described in the corresponding technical deliverables: D2.2.2, D2.4.1, D3.1.2, D3.2.1, D3.3.1, D4.2.1, D4.3.1, and D5.2.1 which are also part of the second year work of the XLike project.

The main goal of this document, which is also the main goal of the WP6 work package, is to show the overall integration of all the components in order to obtain the demonstrator prototype of the Xlike project accomplishing with the requirements of the use cases which have been defined in D1.2.2 “Requirements for demonstrator” (for both end users, STA and Bloomberg).

This document upgrades the previous version D.6.2.1 “Early Prototype” and it is also partly based in the deliverable D6.1.2 “Final Toolkit architecture specification” which contains the final architecture of the project and the established development strategy for the second and third year.

Also, the document D1.2.2 is much related with this one and it defines the requirements needed for accomplishing with the STA and Bloomberg use cases which are the following up of year one: i) related or relevant articles (Bloomberg), and ii) article tracking (STA), and the new ones: iii) Content Advertising (Bloomberg), and iv) event identification (STA).

This report covers a description of the prototype developed until the end of year two and is organized as following: the description of the prototype is at Section 1 followed by the overall description of the architecture in Section2; a summary of the new use cases scenarios are introduced at Section 3 whereas the XLike toolkit is explained at Section 4; finally the conclusions are at Section 5 and the definition of the APIs and the deployed services are enclosed as annexes (Annex A and Annex B), and the data transformation including inputs and outputs at different stages of the XLike pipeline can be also found at Annex C.

This document is the second of three (D6.1.1 Y1, D6.2.1 Y2, and D6.2.3 Y3) which are associated with the prototypes development at the different stages of the project. These different stages of the project collects incrementally the ongoing work and the improvements obtained after solving the problems detected at each previous stage.

Table of Contents

Executive Summary 3
Table of Contents 4
List of Figures 5
List of Tables 6
Abbreviations 7
Definitions 8
1 Introduction 9
1.1 Integration overview and current status 9
1.2 Methodology 10
1.3 Authorship 10
1.4 Relation with Other Work Packages 11
2 Overall XLike Architecture 12
2.1 Technological Demo 14
2.2 Monitorization and performance 15
2.3 Demonstrator prototype Y2 17
3 Use case scenarios 19
3.1 Article tracking and event identification (STA) 19
3.2 Content Advertising (Bloomberg) 20
4 Toolkit components 22
4.1 Sandbox (WP6) 22
4.2 JSI NewsFeed (WP1) 23
4.3 Shallow linguistic processing (WP2) 23
4.4 Early informal language structure extraction prototype (WP2) 23
4.5 Final deep linguistic processing (WP2) 24
4.6 Final text annotation prototype (WP3) 24
4.7 Early ontological word-sense disambiguation prototype (WP3) 25
4.8 Statistical cross-lingual document linking (WP4) 25
4.9 Final Information Visualization (WP5) 27
4.10 API specification and prototype (WP1, WP6) 28
5 Conclusions 29
References 30
Annex A API Definition 31
Annex B Demos and prototypes 38
Annex C Data Format 39

List of Figures

Figure 1 Pipelines implemented in the XLike project 9
Figure 2 Integration of the different work packages into the XLike pipeline 11
Figure 3 XLike REST, HTTP + XML point to point communications 13
Figure 4 Stories visualization panel (part of the visualization component). 14
Figure 5 Technological Demo 15
Figure 6 Monitoring services status using pingdom tool. 16
Figure 7 Monitoring XLike functionalities using leftronic tool. 17
Figure 8 Xlike demonstrator 18
Figure 9 Article tracking scenario. 19
Figure 10 Article details panel. 20
Figure 11 Content advertisement using hootsuite interface 21
Figure 12 XLike Components' interactions. 22
Figure 13 Cross-lingual document linking standalone Y2 web application 26
Figure 14 Demonstrator visualization component. 27
Figure 15 QMiner architecture 28

List of Tables

Table 1 Uptime and average response time for the different services of the demonstrator. 16
Table 2 WP1 Definition and Data provision API 31
Table 3 WP2 Shallow/Deep Linguistic Processing API 31
Table 4 WP3 Final Annotation Text Prototype 33
Table 5 Example of use of the final text annotation prototype 34
Table 6 WP4 Cross-lingual Document Linking Prototype 36
Table 7 Final information visualization 36
Table 8 Demos and Prototypes 38
Table 9 Example of the XML format obtained by WP2 + annotations (WP3 and WP4) 39
Table 10 Complete XML data format of the XLike prototype 40

Abbreviations

API	**A**pplication **P**rogramming **I**nterface
D	**D**eliverable
NLP	**N**atural **L**anguage **P**rocessing
SOA	**S**ervice **O**riented **A**rchitecture
T	**T**ask
HCI	**H**uman **C**omputer **I**nteraction
URL	**U**niform **R**esource **L**ocator
REST	**Re**presentational **S**tate **T**ransfer
XML	e**X**tensible **M**arkup **L**anguage
XLike	**Cross**-**li**ngual **K**nowledge **E**xtraction
WP	**W**ork **P**ackage
WS	**W**eb **S**ervice
YX	Year X

Definitions

Pipeline	Refers to the flux of different processes which are applied to a set of raw data in order to analyze it and interpret it. In XLike project It covers the following phases: gathering data, pre-processing data, application of Natural Language Processing Tools, semantic interpretation, visualization, and finally domain interpretation
Hackathon	Is an event in which computer programmers and others in the field of software development, like graphic designers, interface designers, project managers and computational philologists, collaborate intensively on software projects[1].

[1] http://en.wikipedia.org/wiki/Hackathon

1 Introduction

1.1 Integration overview and current status

The XLike demonstrator prototype aims to provide the first stable and complete pipeline of the project which is able to process on real time the whole amount of articles collected by the JSI Newsfeed component. During the first year the main goal was to provide an initial version of the prototype which could allow a fast and easy integration framework for all the different functionalities developed (WP1-WP5) but not the most robust or reliable version of it.

During the second year we have been focused not that much on providing an integration framework (which had been previously established during the Y1) but on the improvement of the functionalities implemented during that first year (debugging, updating, and enhancing the infrastructure) and also in the inclusion of the new ones which have been implemented in order to satisfy the requirements of the uses cases defined by the end-users at D1.2.2 “Requirements for demonstrator”.

For achieving these goals we have continued using two different pipelines in parallel as we did during the Y1. The first one (Figure 1: Technological Demo Pipeline) has been used to continue the development of the functionalities associated to WP2 (e.g. final deep linguistic processing) and WP3 (e.g. final text annotation prototype), and its early testing. This pipeline is allocated at Xlike Sandbox[2] and is publically accessible. The second pipeline is the demonstrator[3] (Figure 1: Prototyping Pipeline) which contains the functionalities available at the end of the first year [7] plus the new ones implemented towards achieving the use cases from STA and Bloomberg Y2 (see [1] for a complete reference of functionalities and requirements specification).

Also, in order to provide a more robust platform and to gain overall control on the performance it a monitoring system for checking that the different core services are available has been deployed (see Annex A of this document for a complete description of the Xlike demonstrator services). Furthermore, automatic restarting of these services has been implemented for recovering from failures and do not disturb the correct functioning of the overall system.

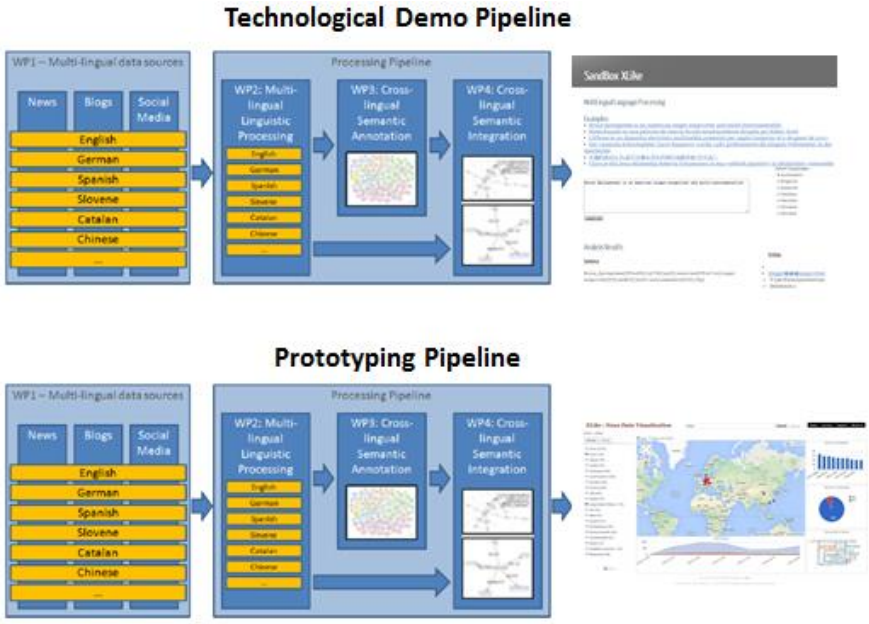

Figure 1 Pipelines implemented in the XLike project

[2] http://sandbox-xlike.isoco.com/demo/index
[3] http://sandbox-xlike.isoco.com/portal/index.html

As was done during the first year for implementing the early prototype, the demonstrator has been created following the methodology described in D6.1.1 "Early toolkit architecture specification" [5] and allowing the co-design of the different parts of the project between all the partners.

We have also continued with a non-central approach for services deployment following a data-centric approach where the data flows throughout the different services of the pipeline at the same time that it gets richer from an information point of view (see Annex C for a better understanding of the data enrichment process throughout the pipeline). Despite of this, the data format and schema have been updated in order to include the new functionalities and newly generated information. This approach still allows having a rapid prototyping (which is still appealing at this stage of the project) and it does not penalize the time response performance too much due to the transmitted data between components is text and it is not very large regarding the number of bytes that it contains.

For the last year of the project we expect that most of the components will be stable and therefore we are planning to switch to a central-approach for the deployment of the services making the best-effort in order to provide the best results in computer and time performance. The main reason for delaying this to the third year is that most of the needed functionalities for accomplishing with the different use cases scenarios and the industrial showcase will be already done by the beginning of the third year, and afterwards we do not expect to have many changes in the components and the development effort will be decreasing towards having a final stable fully functional prototype.

1.2 Methodology

The followed methodology was described at deliverable D6.1.1[5] and a co-evolutionary development model has been continuously applied in order to update the implementation iteratively and according to the use cases of the second year. The demonstrator prototype (M24) is the second prototyping milestone of WP6 where the implemented functionalities and developed components until the end of the second year are shown. This demonstrator includes the revision of the initial architecture specification (D6.1.1 [5], M3) which was updated at the final toolkit architecture specification (D6.1.2 [6], M15).

Regarding the development process each WP leader has organized internally with the other work package members in order to accomplish with the specified tasks. At project integration level we have continued during the second year having bi-weekly Skype calls to provide updates on the current status of the different parts of the project and also weekly calls have been used in order to accomplish with specific deadlines following a SCRUM methodology.

Furthermore, two hackathons have been also scheduled jointly with the regular meetings[4] for helping the final integration of the Y2 demonstrator. The main goal of the hackathons were to develop software collaboratively in order to fill the software gaps between components and to provide a general overview of the project to different people who is involve at the different work packages..

For inter-task work a more informal communication protocol has been used via direct mailing (using the development XLike list), or Skype XLike group chat/calls/standups.

1.3 Authorship

This document results from the work of the Xlike developers/researchers at the different work packages of the project. iSOCO has leaded the process and has helped to define and standardize the different services, and it has provided a common data format at the different parts of the project. Relating the creation of the web services each partner (JSI, KIT, UZG, THU, and UPC) has been leading its own development and has

[4] June 19th at Dubrovnik, and October 2nd at Barcelona

followed the established standards in order to be compliant with the overall project strategy and to fulfil the use-cases requirements.

1.4 Relation with Other Work Packages

The demonstrator prototype includes the visualization for human-computer interaction (HCI) jointly with the different service calls needed to obtain the relevant data from the JSI Newsfeed according to the use cases defined for Y2.

Following the specifications of D6.1.2 [6] the different services are decouple each other and the only dependencies between components are by functionality. This functionalities are provided by the different partners and are split into the following layers according to the data processing and enrichment pipeline: i) source data and infrastructure, ii) analysis and interpretation, iii) application and interface as shown in the Figure 2.

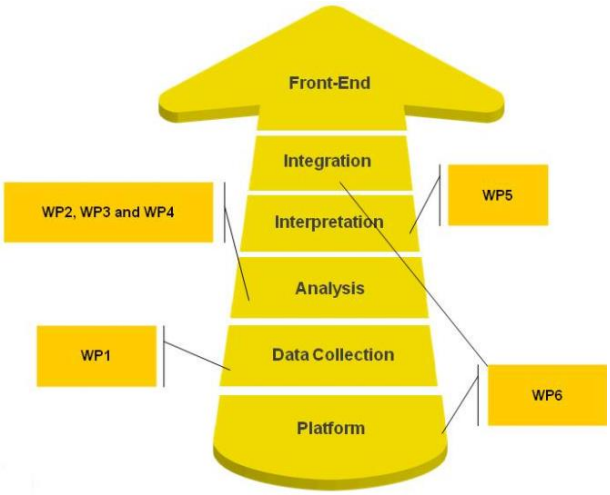

Figure 2 Integration of the different work packages into the XLike pipeline

The complete set of functional dependences between all the implemented components and its description can be consulted at D6.1.2 “Final Toolkit Specification”.

In the following the overall architecture of the Xlike prototype is described at Section 2. Afterwards the new use case scenarios are introduced at Section 3 describing their functionalities and how the demonstrator accomplishes with them. Then the Toolkit components are shown at Section 4, and Section 5 presents the conclusions and plans for the next year.

2 Overall XLike Architecture

In this section we summarize the overall architecture of the demonstrator and the interaction between the different components. As described in D6.1.2 [6] the overall architecture is divided into four main groups: i) **physical layer** which basically is composed by an infrastructure for accessing to the data and provide computing capabilities (this is the main role of the Sandbox[5]), ii) **acquisition** which provides data harvesting capabilities in order to collect data from the Web, repositories, or data providers, iii) **analysis and interpretation** which provides the linguistic processing and also the semantic enrichment analysis for extracting richer multilingual and linked information (WP2 e.g. relations), cross-lingual knowledge (WP3-WP4) and applications such as cross-lingual event registries (WP5), and iv) **user interface or Human-Computer interaction layer** which provides the interaction between the end-users and XLike functionalities (WP5).

One of the main characteristics that has been accomplished is a loose coupling between the different components. All the components have been implemented following a REST Web Service approach and their APIs are public[6] in order to facilitate the easy accessing and integration [5,6]. This approach also has accomplished with other three desired integration characteristics which are the interoperability between different languages, the reusability of the components being able to use the services by its functionality independently of the demonstrator, and the flexibility which allows moving services location from one place to another without having to pay a penalty for it. The last characteristic is especially useful in the project in order to scale the platform as needed mainly due to the collection of new data sources and higher computational requirements needed by the new and more complex linguistic and semantic implemented functionalities.

This demonstrator prototype includes the functionalities and components needed for accomplishing with the use cases defined at D1.2.2[1]. These use cases cover the four general applications of cross-lingual summarization, cross-lingual contextualization, cross-lingual plagiarism, and cross-lingual personalization and specifically the demonstrator accomplishes with the following Y2 STA use cases:

- **Article Tracking**
- **Event Identification**

There is another use case (**content advertising**) related with Bloomberg end-user which is not being accomplished by the demonstrator but by a dedicated application (implemented in Hootsuite[7]) which makes use of a recommendation service implemented within the overall pipeline of the project. All this use cases are explained at Section 3.

Regarding the demonstrator platform, the current pipelines are distributed at different locations as indicated in Figure 3. This allows a faster interaction within the development phase but it forces to be stick to a stable data description. During the year two the data format of the multilingual pipeline has been modified after the first hackathon and since then it has been preserved in order to avoid inter-components failures. This new data format being used for the implementation of this demonstrator can be consulted at Annex C of this document.

One of the drawbacks of a distributed architecture is that it also implies more points of failure. In order to avoid uncontrolled services a monitoring tool using third party software (Leftronic[8]) has been configured for detecting any failure on the services. This tool sends automatically an alert to the maintainer in order to check and recover the service and also alerts of possible bugs on the code. A description of this monitoring tool is introduced in section 2.2.

[5] http://sandbox-xlike.isoco.com
[6] The set of developed services for the demonstrator can be consulted at Annex A.
[7] https://hootsuite.com/
[8] https://www.leftronic.com/

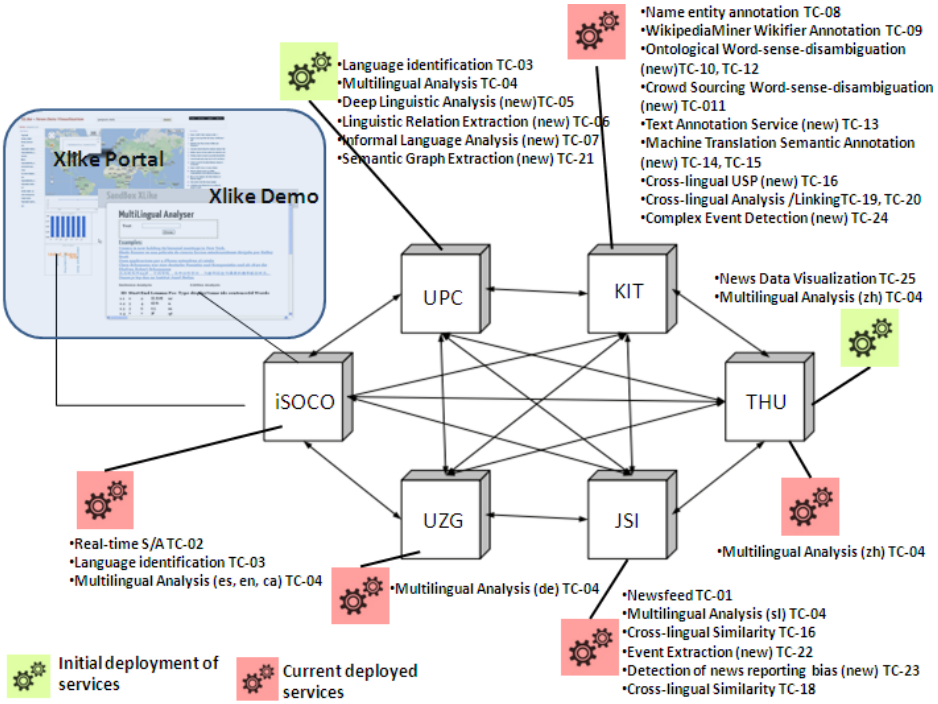

Figure 3 XLike REST, HTTP + XML point to point communications

Currently the demonstrator pipeline set of calls is the following[9]:

1. Service for collecting, indexing, and accessing to the pieces of news: it is provided by JSI newsfeed http://newsfeed.ijs.si/xlike/ and specific APIs as indicated at D5.2.2 "Final information visualization prototype"
2. Service for language identification: http://sandbox-xlike.isoco.com/services/language_code/ident and for shallow/deep language processing for the different language: http://sandbox-xlike.isoco.com/services/analysis_XX/analyze (being XX the language identifier coded following the ISO 639-2 specification)
3. Service for annotating the document with cross-lingual links and wiki links: http://km.aifb.kit.edu/services/annotation-XX (being XX the language identifier coded following the ISO 639-2 specification)
4. Service for retrieving the events/stories associated to a specific query: http://eventregistry.org/#/?query= (the query parameters can be consulted at D4.3.1 "Early event extraction prototype")
5. Visualization of the retrieved information using the final visualization prototype http://sandbox-isoco.com/portal/index.html,

and the data enrichment is the following according to those four calls:

1. It collects the different pieces of news from the different sources available at JSI Newsfeed (see D1.3.2 "Final Prototype of Data Infrastructure") index and stores them into a local repository. During the Y2 Twitter, Bloomberg and STA sources have been added. This step includes the raw data related with an article which includes publisher, text, date of retrieval, location, country, etc.

[9] Note: although the logical sequence is the presented, actually due to implementation reasons the calls are being executed following this order 1→3→2→4 allowing to the last service of the pipeline to gaining control over the others

2. Provides the language identifier for the given article and the multilingual analysis of the text. The language identification information is used for calling the corresponding NLP shallow/deep processing services. The analysis of the text provides an article XML file defining the original language of the document analysed and the linguistic information (see Annex C) which is added to the original data collected by the JSI Newsfeed.
3. The service includes the annotations to the document which relates the document with other cross-lingual ones (e.g. other directly related Wikipedia articles or similar ones). These relations can be based on topic similarity or on semantic relatedness between documents. These annotations include a list of descriptions with the URLs associated to the document and an additional parameter for language identification.
4. The event registry provides information that describes a set of articles which share some common information. This event registry includes the information regarding the identification of the event, the set or articles that it contains (possibly in different languages), its categories, its related concepts, and the associated sources.

This information is presented by the visualization component and it is worth to highlight that the event registry part is one of the major updates provided during this second year (see Figure 4).

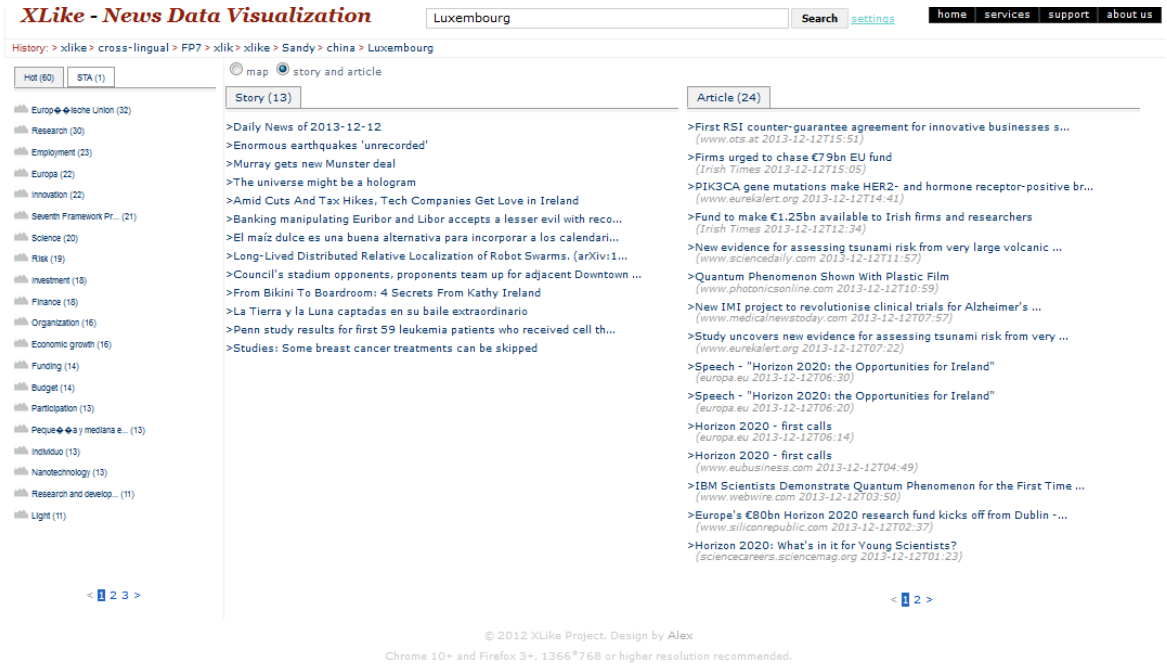

Figure 4 Stories visualization panel (part of the visualization component).

2.1 Technological Demo

The Sandbox has been also used during the second year as a place for early testing of the new components and also for the updated ones of the Y1. All these components have been used later for the accomplishment with the above mentioned user scenarios either using the demonstrator or by the specific implementation of prototypes (e.g. Bloomberg content advertising Hootsuite application).

Figure 5 shows the updated Y2 technological demo visualization which provides the functionality related to the first and second year of the: WP2 (e.g. lemmatization, tokenization, POS tagging, entity recognition, relation extraction, etc.), WP3 (e.g. cross-lingual annotation), and it has been adapted to the new Y2 data schema.

Figure 5 Technological Demo

This technological demo provides the following functionalities adapted to the Y2 data schema:

- Text language identification
- Shallow/deep processing (including sentence splitting, tokenization, lemmatization, POS tagging, and entity recognition)
- Annotation of a text based on similarities functions.

2.2 Monitorization and performance

The different implemented components are distributed as a set of independent services as has been detailed above. In order to avoid on cascade failures due to functionality dependencies and for making the overall platform more stable and robust, a set of monitoring tools[10,11] have been configured for tracking the core services. These monitoring tools provide a quick alert whenever a service is down in order to check them out. Furthermore internal controls have been implemented for each deployed service in order to automatically restart them.

Regarding the performance, the multilingual pipeline supports multiple forks/threads allowing parallelization at operative system level. Moreover, the presented architecture also supports parallelization at computer level by duplicating the overall system and applying for instance a round robin queue scheduling approach. Due to currently we are able to analyse two third parts of the collected data, which has been considered enough for validation of the use cases, this change is planned to be done during the third year of the project once the development process is completed. This allows to keep going with the prototyping and fast development cycles before establishing the final platform.

Figure 6 shows the uptime results obtained for a period of six days during the last week of the year 2013 (24th – 29th of December) showing green symbols whenever the services were up all the time and the yellow exclamation meaning that there was a short period of outage. Furthermore Figure 7 shows the number of analysed articles for each one of the languages considering the different services of the multilingual pipeline which are shown in different colours.

[10] http://stats.pingdom.com/33sj6sc6keuh
[11] https://www.leftronic.com/share/9tcE6X/#9tcE6X

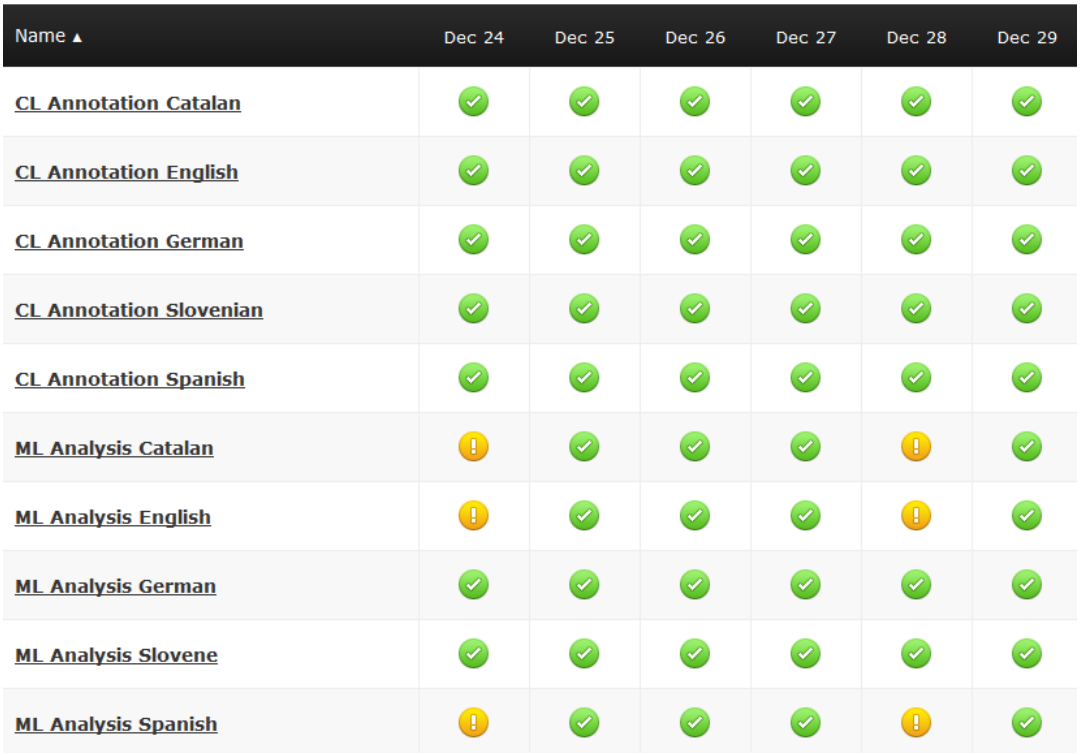

Figure 6 Monitoring services status using pingdom tool.

Table 1 also shows a summary of the uptime and average response for the above mentioned period of time. We can observe that the services are up most of the time which shows the robustness of their deployment and the code of the algorithms. Regarding the response time we have to take into account that we are using a third party tool for checking the services and it averages the time response using different locations such as North America and several European countries.

Table 1 Uptime and average response time for the different services of the demonstrator.

Service Name	Uptime last 7 days	Avg. res. Time ms.	Last updated
English Service	93.6%	268	30/12/2013
Spanish Service	93.6%	439	30/12/2013
Catalan Service	97%	497	30/12/2013
German Service	93.6%	347	30/12/2013
Slovenian Service	99%	338	30/12/2013
English annotation	99.8%	527	30/12/2013
Spanish annotation	99.8%	460	30/12/2013
German annotation	100%	456	30/12/2013
Catalan annotation	99.8%	420	30/12/2013
Slovenian annotation	100%	353	30/12/2013

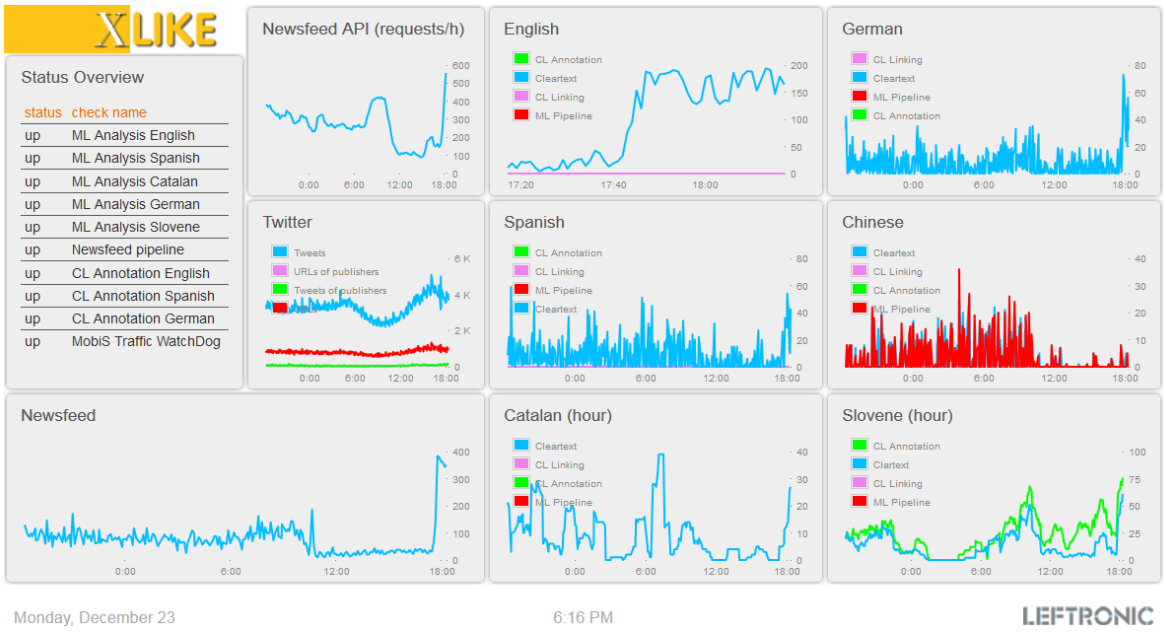

Figure 7 Monitoring XLike functionalities using leftronic tool.

Overall we can conclude that the platform is working well though still some minor improvements are needed to obtain full uptime of the services and process the full collected data.

2.3 Demonstrator prototype Y2[12]

The demonstrator prototype is focused on implementing the access to the functionalities developed within the project and also solving the use cases related with topic entity/article tracking, related relevant articles, and event detection. The demonstrator is allocated at the sandbox and it performs the needed calls to the services allocated at JSI in order to retrieve the proper information (see Figure 3).

During the second year it has been done some work for improving the performance of the demonstrator specially at algorithmic level of the different services and also a first effort has been put into establishing a unique general framework (including platform and software) in order to allow the validation of the defined use cases and to obtain the final prototype during the third year.

[12] http://sandbox-xlike.isoco.com/portal/index.html

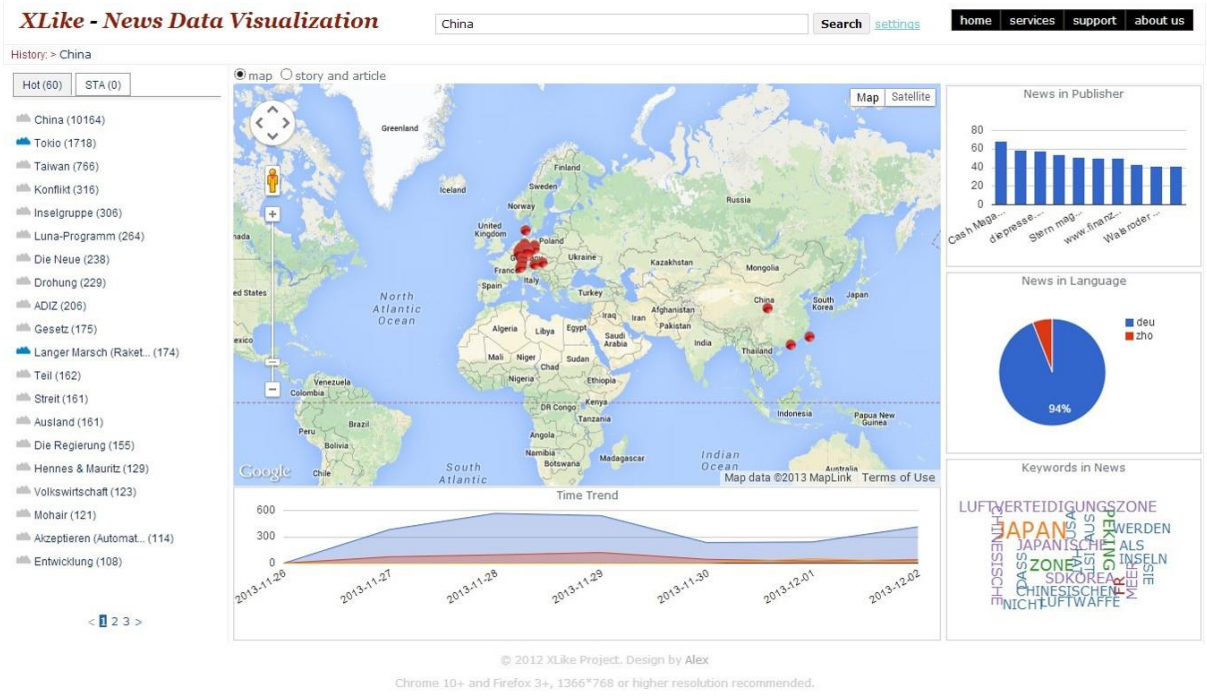

Figure 8 Xlike demonstrator

Figure 8 shows the visualization (WP5) of the prototype which is using the different services from WP1-WP5 and provides the interface for displaying the news articles, and the enriched and structured data (including the article itself, stories, and entities) in in a clear way. This demonstrator contains the following functionalities covering the Y1 and Y2 work (see D.5.2.2 “Final Visualization Prototype” for a deeper explanation of each one of the parts of the visualization interface and also for a description of the data format):

- Searching for keywords or specific entities: it allows searching for articles and stories which contains a specific keyword or entity. This functionality can be customized by time, language, location, and maximum size of results.
- Entity tracking: it returns articles related with pre-defined entities of interest.
- Articles map geolocalization and analytics: it provides trending information, map localization, and analytics related with the publishers, languages, and most important words (cloud of words).
- Article tracking and story building: it collects and builds up stories based on the similarities between the articles. This allows to find similar articles based on those similarities and also to build a story by grouping a set of articles which share semantic information.
- Topic tracking: it allows obtaining a quick overview of what is being said for a specific category.

3 Use case scenarios

This section summarizes the two news use case scenarios on which the demonstrator prototype has been focused on and also includes the overview of how they have been achieved successfully by integrating the different deployed services in the XLike project. The use cases are the following:

3.1 Article tracking and event identification (STA)

This scenario pursues two main goals, i) allow the detection of republished articles for detecting and control of plagiarism, and ii) providing to editors a richer context for a specific article, keyword, or entity in order to help them in their daily work of creating news articles and news stories. Regarding the plagiarism use case, it doesn't necessarily occur in the same language but it can happen at different languages. Also, whenever a news editor wants to write a new article he may found interesting information at other languages sources which can enrich the writing or can help him to verify its authenticity. For those reasons there is a need of finding similarities independently of the language.

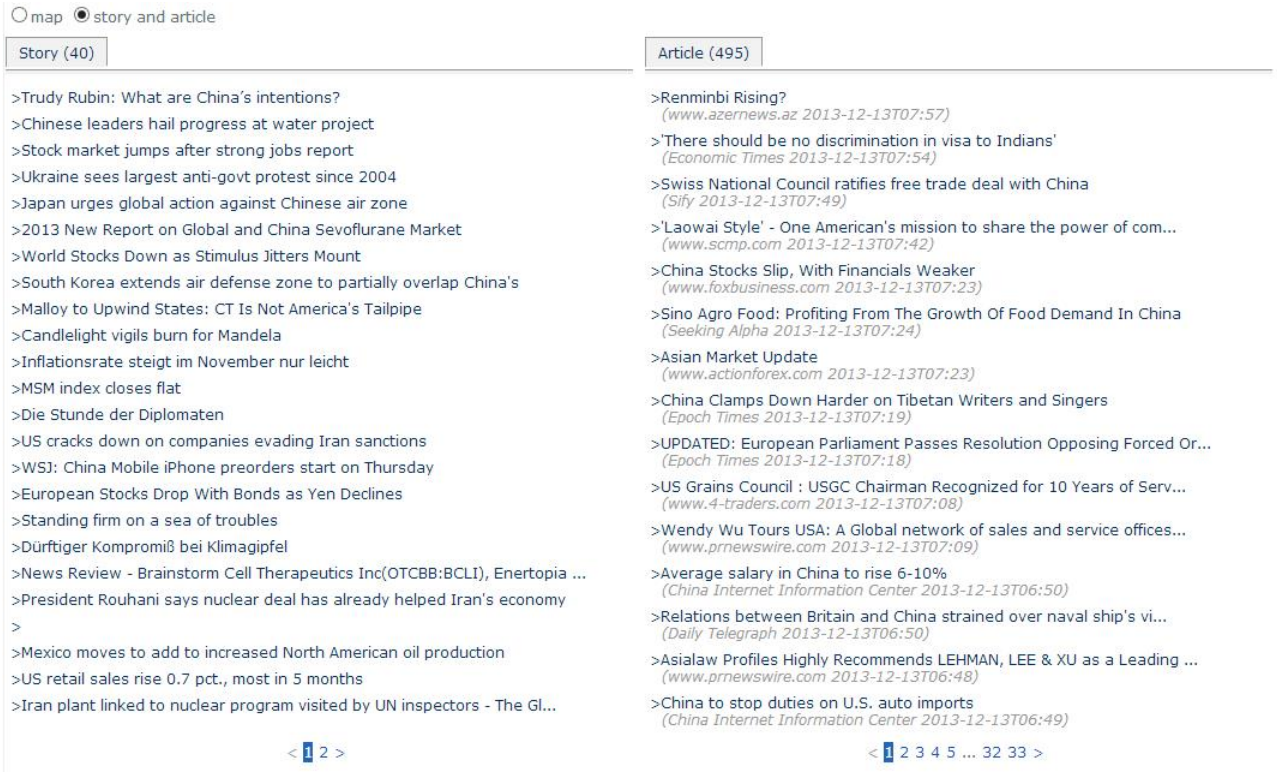

Figure 9 Article tracking scenario.

The story and article option of the demonstrator allows accessing to the story (a group of related articles which share semantic information) and see the articles included within that stories (see Figure 9). The different stories can be shown on the left part and on the right part the articles contained can be consulted by clicking of the story of interest. Furthermore the complete article description and information (such as entities that it contain, published time, URL, etc.) can be obtained by clicking on the article link (see Figure 10).

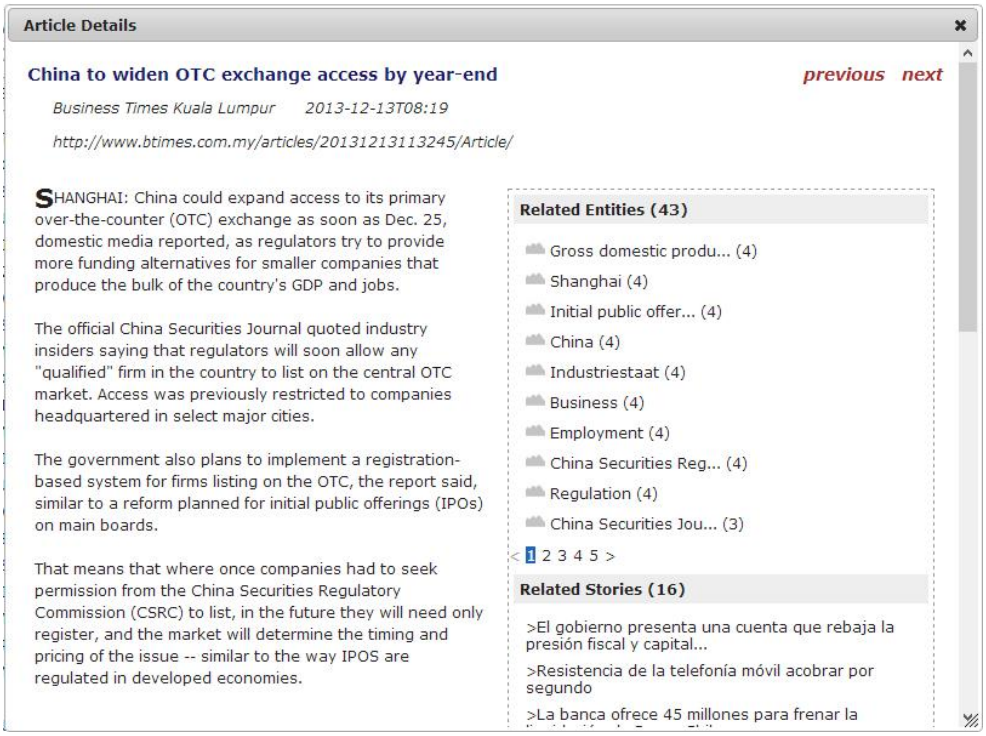

Figure 10 Article details panel.

The event detection scenario uses some of the technical functionalities which have been also covered by the industrial showcase [8] but they have been included into the demonstrator which is designed specifically for editors.

3.2 Content Advertising (Bloomberg)

This scenario pursues the advertising of contents for specific regions/locations based on local interest of the audience. The local interest has been gathered via monitorization of local stream channels and those channels are integrated into the JSI NewsFeed platform (WP1) which is already integrated with the multilingual pipelines (WP2), and the cross-lingual functionalities (WP3-WP4) providing a API[13] for retrieving the specific articles which are more related with the different regions as indicated (see Annex A for the description of the APIs). Due to best practices and easy use within Bloomberg company, the visualization has been developed using Hootsuite[14] platform which was already being used by Bloomberg editors. This application uses JQuery and secure connectivity for transferring information from the recommendation service to the Hootsuite framework allocated at Bloomberg.

[13] https://aidemo.ijs.si/xlike/hs/topstories
[14] https://hootsuite.com/

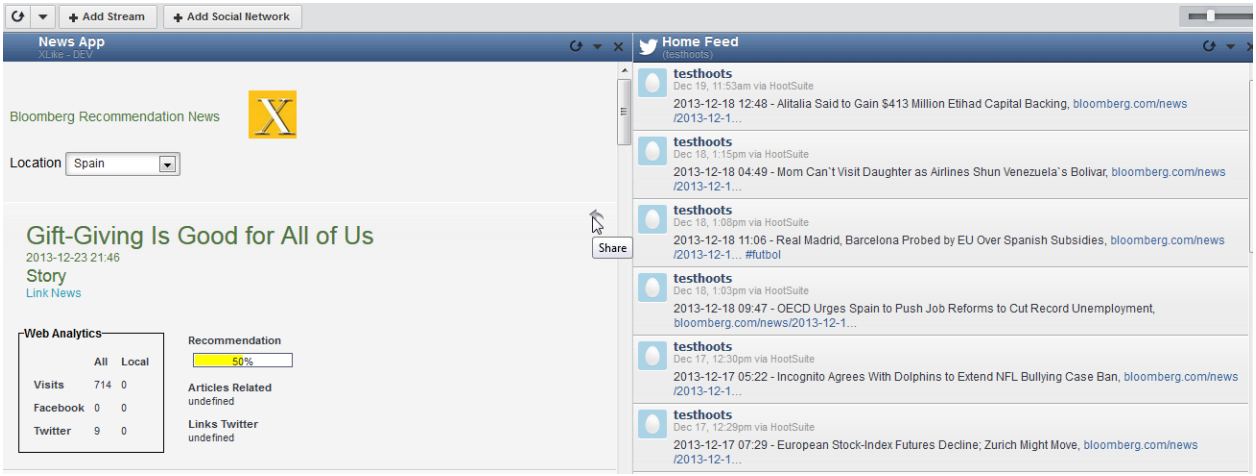

Figure 11 Content advertisement using hootsuite interface

Figure 11 shows the current interface where there are two main parts. The left part allows choosing the region of interest for the editor and displays those Bloomberg articles which are more related with that local interests. The panel includes the title, published time, URL link, recommendation score (ranging from 0 to 100), and some analytics to measure how many visitors came from the different social media where the article is being posted. This panel also allows the editor to share the article via Twitter whenever he thinks that the recommendation is relevant for that specific audience. On the right part the Twittered articles can be seen providing a quick feedback of the previously posted ones.

In the next a brief description of the different integrated and used components of the project for the development of the demonstrator prototype are introduced.

4 Toolkit components

The next subsections describe the components of the XLike toolkit. The toolkit is composed by 25 different components and their complete description can be found at D6.1.2 [6]. Also the global services API's that have been used for the demonstrator (this does not include internal components which accomplish with specific functionalities not needed to be called directly by the demonstrator) can be found at Annex A. We also include Figure 12 to have an overview of the components interaction between all the components of the project.

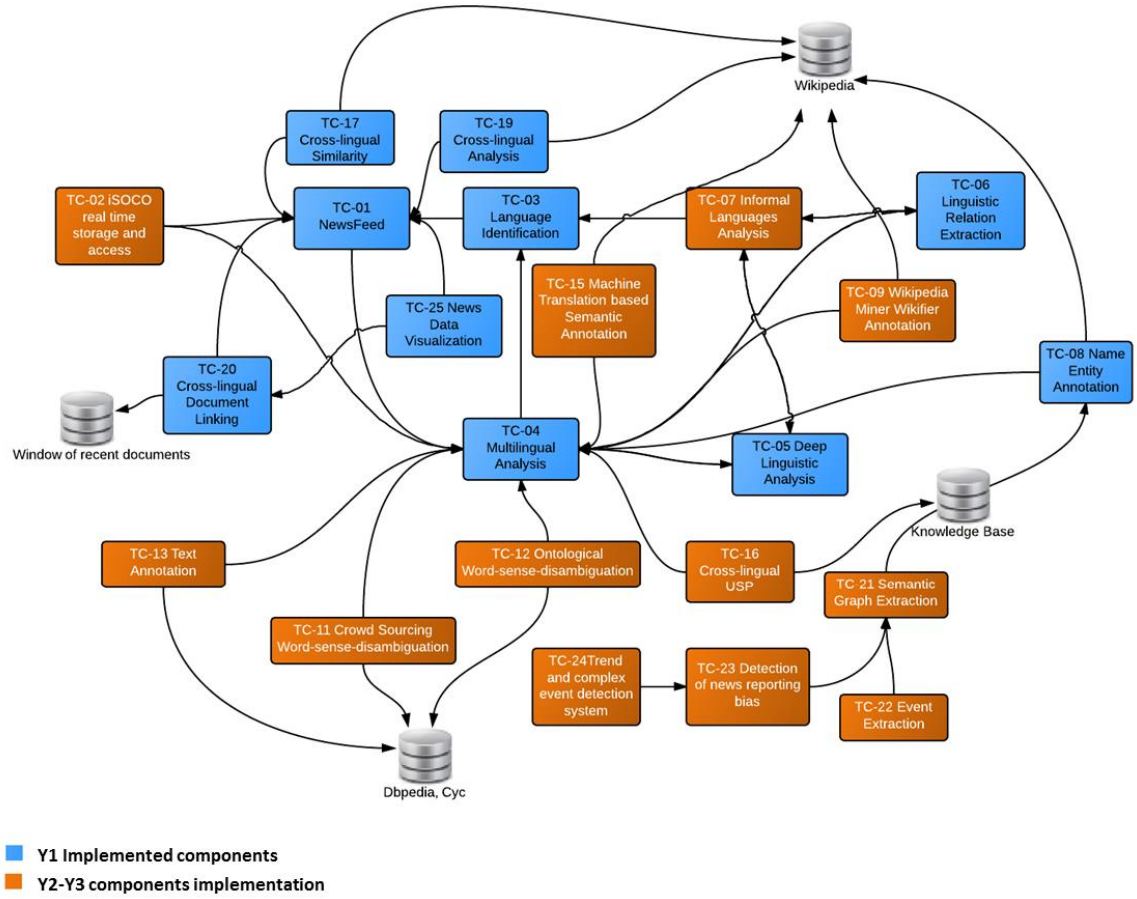

Figure 12 XLike Components' interactions.

In the next we describe the main components resulting from the different work packages.

4.1 Sandbox[15] (WP6)

The Sandbox is a kind of playground and has been used for early testing during the first and second year of the project. For instance during the second year the Bloomberg news content advertising application has been firstly deployed in the Sandbox for verifying its correctness before being deployed at Bloomberg systems. Also some of the functionalities are currently being hosted at the Sandbox (e.g. es, en, ca multilingual pipelines) due to we are still finishing and testing some functionalities. So far, we have used this Sandbox for both, i) showing and early testing of technical functionalities, and ii) for hosting demos and prototypes.

At this stage the Sandbox is reaching its mature state and during the first months of the third year it will be considered final. Afterwards, it also will be deprecated for being used as a testing platform and all the

[15] http://sandbox-xlike.isoco.com

functionalities that are deployed or currently under testing phase will be finished and move into a more scalable and stable platform.

4.2 JSI NewsFeed (WP1)

The JSI Newsfeed provides the inputs to the multilingual pipelines and the demonstrator, and also collects the outputs of the different analysis embedding them verbatim into the final XML formatted indexed articles (see Annex C). This component allows the accessibility to the complete raw data via a web service API[16], and also to indexed specific searches (entities, stories, articles, etc.) via web services APIs as described in D5.2.2 "Final Visualization Prototype". Recall that this component is a central point of the architecture being the provider and the consumer of the different linguistic and semantic functionalities provided by the WP2-WP5.

During the second year several improvements have been applied. The inclusion of 250k new blogs is one of the major advances resulting in a collection of more than 200k new articles per day. This improvement in data harvesting required significant changes in the crawler infrastructure to accommodate it for the increase in traffic. Scalability improvements allowed reducing the average delay between when an article is published and when it is crawled, and in order to reduce the average delay the monitoring of tweeter feeds of top news publishers was also started allowing higher frequency than using regular RSS feeds. Moreover, two new languages (Italian and French) have been also included.

The description of its internal architecture can be found at the updated and final version of the data infrastructure (D1.3.2 "Final prototype of data infrastructure") including how it is deployed to make it available externally via a web service API[17] (see D1.3.2 "Final prototype data infrastructure" and D5.2.2 "Final visualization prototype"). Currently the acquisition process is being done by the partner JSI and hosted in their institution.

4.3 Shallow linguistic processing (WP2)

The shallow processing functionalities were implemented during the first year of the project and they covered language detection, sentence splitting, tokenization, lemmatisation, POS/MSD-tagging, and named entity recognition. During the second year this functionalities have been tested and improved towards making them more robust from an execution point of view. Also the output of this component has been modified in order to accomplish with the new multi-lingual schema which covers a richer representation including frames which contains the deep linguistic processing and n-relations. As was done during the first year an important work on standardization of the different languages outputs has been performed. This effort has been grounded in D2.2.2 "Final deep linguistic processing" which has collected the modifications and final version of the WP2 output schema (it also can be consulted at[18)] for both shallow (due to there was not any deliverable regarding the shallow processing during the Y2) and deep processing.

4.4 Early informal language structure extraction prototype (WP2)

Since the NLP tools developed for standard language analysis performs much worse on informal languages due to the domain adaption, this component is focused and specialized in providing the same functionalities that the shallow and deep processing components but for informal languages. The component is already working for English, Chinese, and Spanish languages and it is expected to be extended to the other ones during the last year of the project.

[16] http://newsfeed.ijs.si/stream/
[17] http://newsfeed.ijs.si/xlike/
[18] https://github.com/xlike-project/wp6/blob/master/schemas/document2013.xsd

The different languages have different peculiarities and therefore there are different modules for each one of them making them also independent. For Chinese two different modules were implemented, one for the discovery of vocabulary words via learning from unlabelled data and another tag via clustering the newly discovered words. For English, a new implemented tool called “TweetsNLP” has been used for shallow processing and then the main entities are extracted through a learning process. Finally, for Spanish also a normalization method has been developed as a pre-processing module.

The input of this component is the pieces of news themselves collected by JSI newsfeed and identified by its source information (e.g. Twitter). This component is still ongoing work and is not fully incorporated to the project pipelines but will be fully finalize and integrated during the third year. Regarding the results obtained so far they show very slight improvements on performance but during the next year these components will be upgraded, extended to all the XLike languages, and included in the multilingual pipelines.

4.5 Final deep linguistic processing (WP2)

The final deep linguistic processing component has extended the previous early component which only collected the syntactic dependencies in the form of a labelled tree to a more complicated predicate-argument structure which augments the semantic interpretability. This has also led a refinement of the data schema provided during the first year to include these new structures providing the WP2 final schema (see Annex C).

This enhancement allows having different frames which represents a predicate and allows their representation not only as SVO triples, but as frames encoding several predicate arguments (e.g. subject, direct object, indirect object, location, time, etc.). It also allows representing predicates where some arguments are in turn other predicates (e.g. "The Prime Minister declared that the Government will ban abortion"), and predicates expressed not only via verbs, but also with nominalizations (e.g. "Microsoft acquisition of Skype caused the stocks to rise.").

This argument extraction is performed via Semantic Role Labelling (SRL) for languages where training corpus was available, and via hand-written rules for the rest. In the latter case, semantic resources integrating WordNet, FrameNet, and VerbNet are used to bridge the gap between syntactic function and semantic roles. Finally, the deep processing includes also Word Sense Disambiguation (WSD) based on WordNet for those languages that have this resource available. This step provides the necessary cross-linguistic linking that makes it possible a language-independent semantic representation of the text content.

At this stage of the project all languages are compliant with the new schema. During the last year is not expected any modification of the functionalities and outputs of this component but only updates related with the performance or full integration within the overall platform.

4.6 Final text annotation prototype (WP3)

While the early annotation prototype aims to annotate articles with information regarding its entities and their description by linking them to Wikipedia articles, the final annotation prototype provides a set of services which semantically enrich the articles with the resources from the data sources of Linked Open Data (LOD).

First, we construct cross-lingual linked data lexica, also called xLiD-Lexica, by extracting surface forms in all XLike languages of DBpedia resources from Wikipedia. Basically, we make use the following sources in Wikipedia: 1) titles of the pages which provide the most common name for resource, 2) redirect pages which indicate synonyms, abbreviations or other variations of resource, 3) disambiguation pages which are useful in extracting abbreviations or other aliases of resource and 4) anchor texts which are very useful source of synonyms and other variations of resource. In order to derive cross-lingual grounds, we use cross-

language links in Wikipedia, which connect “equivalent” resources across languages. Besides the extracted surface forms, we also exploit statistics of the cross-lingual groundings. Multilingual information access can be facilitated by the availability of such cross-lingual lexicon, for example allowing for an easy mapping of natural language expressions in different languages to English ontology. We build the SPARQL endpoint over the xLiD-Lexica, which provides cross-lingual lexical information about DBpedia resources. The endpoint is provided using OpenLink Virtuoso as the back-end database engine. This RDF data set used for this endpoint is extracted from Wikipedia dumps of July 2013 in English, German, Spanish, Catalan, Slovenian and Chinese. It contains 295 million triples of lexical information about DBpedia resources.

Based on the xLiD-Lexica, we build the final text annotation prototype. In order to recognize mentions, disambiguate their meaning, generate the annotations of the multilingual text, we take into account two components: local mention-to-entity compatibility and global entity-to-entity coherence. The first component captures the most likely entity behind the mention based on cross-lingual groundings and the entity that best fits the context based on cross-lingual document linking describe in D4.1.1 “Cross-lingual document linking prototype”. The second component collectively captures the entities as annotations of one document that are related based on the structure of the KB since entities that appear together in one document tend to be related to each other. We provide the different services for annotating both raw text and web pages. In addition, the services according to the XLike service schema of the pipeline have been provided, which take the output of multi-linguistic processing in WP2 as input and adds the annotations with knowledge resources.

This component add to the previously collected articles a set of annotations for the different languages which are related with the recognized entities. It also includes its description and the confidence score. These descriptions provide a summary of the document by the Wikipedia links for the different languages under study that it contains. This added information is compliant with the defined data schema.

4.7 Early ontological word-sense disambiguation prototype (WP3)

The purpose of the early ontological word-sense-disambiguation prototype is to identify the sense of words and phrases (i.e. meaning) in multilingual text using resources in the knowledge bases, such as DBpedia and OpenCyc. Our system is based on the named entities detected by the NERC tools described in D2.1.1 [2] for all XLike languages. On top of that an approach for finding the corresponding resources in the knowledge base in the target language is deployed. In particular, we make use of the consistency of the classes of the resources in the ontology and types of the named entities to filter out the inconsistent candidate resources from the knowledge base. The evaluation results show that the consistency between the types of detected named entities and the classes of resources in the ontology can help increase the precision but decrease the number of the links.

Furthermore, we also address the problem of determining the similarity between concepts defined in a knowledge base such as ontology. We propose a concept similarity algorithm based on geometric models for representing concepts and relationships, which can be applied to different types of ontologies. The key idea is the concept weighting scheme which allows for quantifying the degree of abstractness of concepts. The evaluation settings involving two ontologies with different characteristics, Wordnet and OpenCyc, validate and highlight the advantages of the proposed approach. Using our measure, which closely resembles the human judgment of similarity, we can reliably recreate predefined concept clusters, and generate more informative concept paths.

4.8 Statistical cross-lingual document linking (WP4)

The statistical cross-lingual document linking[3] researches how to compute similarities between documents written in different languages based on the Wikipedia multi-lingual comparable corpus. During the first year this component implemented three different methods for linking different languages

documents and mapping them each other on real time: 1) Latent Semantic Indexing (LSI), 2) its generalization Canonical Correlation Analysis (CCA), and 3) Explicit Semantic Analysis (ESA).

For 1) and 2) during the second year this component has been updated mainly for improving its performance and also for allowing a larger number of languages. Regarding the performance the implementation has been improved by: i) removing dependencies from MATLAB (used as initial development framework), which was difficult to maintain due to its commercial license, and now all the code is contained within the JSI code base glib, ii) the classes have been also extended for the vector space model representation of texts including a more efficient code for word and character n-gram counting as well as stemming using Snowball stemmers and tokenization; for Chinese tokenization, we included optional support for the high precision Chinese tokenizer - ICTCLAS.; iii) the algebraic algorithms and implemented models have been adapted for using high performance libraries such as Intel MKL[19] and OpenBLAS[20] that can be enabled by using pre-processor definitions and, iv) the methods and models have been adapted to be cross-platform (Windows and Linux).

Regarding the number of languages, at Y1 we dealt with English, German, Spanish, Chinese, Slovenian, and Catalan for 1) and 2) and English, German, Spanish, Slovenian and Catalan for 3). During the Y2 1) and 2) have been extended to one hundred languages from the top of Wikipedia languages (most used).

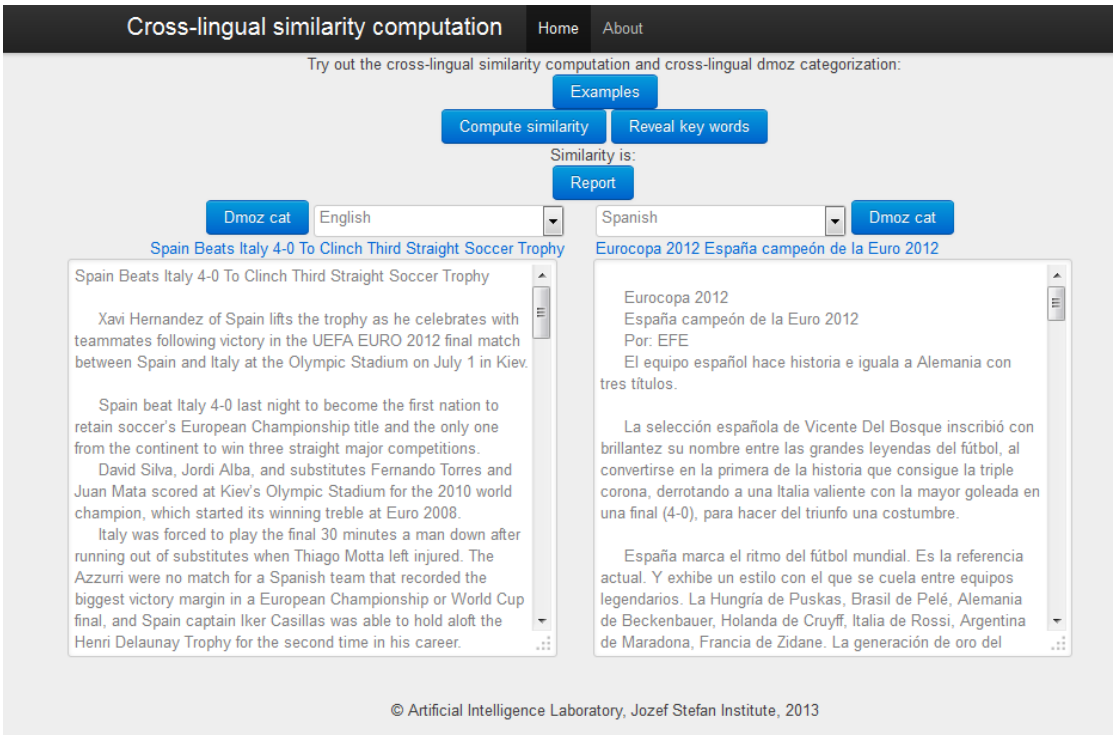

Figure 13 Cross-lingual document linking standalone Y2 web application

Figure 13 shows the updated web prototype for 1) and 2) which is available at[21]. The prototype allows comparing between documents written in any of the one hundred trained languages and reveals the information that added the most to the similarity score. Specifically the demo shows the similarity computation, interpretation of the scores (a user can reveal the keywords that contributed the most to

[19] http://software.intel.com/en-us/intel-mkl
[20] https://www.tacc.utexas.edu/tacc-software/gotoblas2
[21] http://pankretas.ijs.si:1221/wikipedia.html

similarity) and a new feature: cross-lingual document categorization obtained from the Open Directory Project[22].

The cross-lingual categorization for 100 languages is based on training pair-wise latent spaces between English and all the other languages and using centroid classifiers in the latent spaces. This means that 99 models are being used corresponding to 99 different latent bases in the English space.

The service for 3) which allows the use of the ESA approach to obtain a specified number of documents from Wikipedia similar to a given one has been also updated to improve its performance [23] (see D3.1.2 "Final text annotation prototype" for a description of the service).

4.9 Final Information Visualization (WP5)

The final information visualization component [4] displays the analyzed and structured data extracted from news agencies, web blogs, and social media sources (see Annex A WP1 Definition and Data Provision). This structured data can be split into source metadata (e.g. time of publication, geographical location, publisher, etc.) and semantic information provided by the different components of the XLike pipeline WP2-WP5 (e.g. entities, keywords, stories/events, etc.). The visualization has four main panels of information showing: i) current trending topics and STA entities of interest, ii) location of the pieces of news resulting from the search, iii) analytics related with the search which are directly interpretable, and iv) a list of related stories to the search. These component and its functionalities are designed following the use case requirements described at D1.2.2 "Requirements for demonstrator".

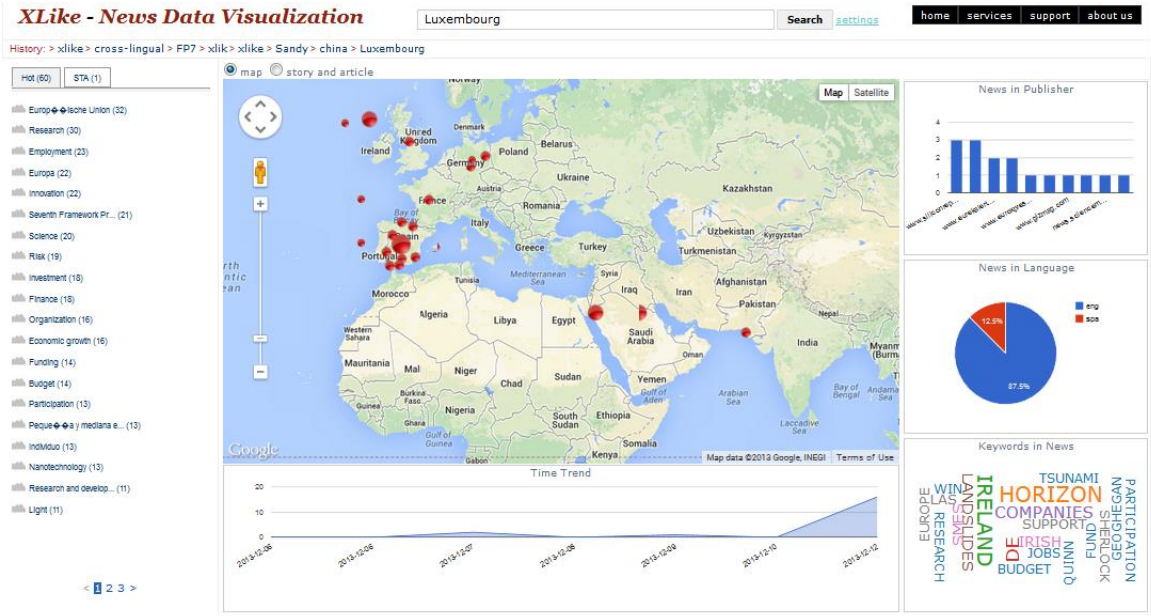

Figure 14 Demonstrator visualization component.

The early prototype visualization[24] is shown in Figure 14 and its complete description is available at D5.2.2. The visualization component has been developed by using JavaScript technologies included into HTML and making calls to the Google API for showing the geographical map where the articles of interest are pinned. This implementation specification accomplishes with the architecture desired characteristics of **loose coupling, autonomy, and reusability.**

[22] http://www.dmoz.es/
[23] http://km.aifb.kit.edu/services/webpage-annotation/
[24] http://sandbox-xlike.isoco.com/portal/index.html

4.10 API specification and prototype (WP1, WP6)

The data storage and analytics backend is based on API specification and prototype [9], which The API is based on QMiner open-source project. It is an analytics platform for large-scale data stores and real-time streams containing structured and unstructured data. It is designed to for scaling to millions of instances on high-end commodity hardware, providing efficient storage, retrieval and analytics mechanisms with real-time response.

QMiner provides and integration of NoSQL-like storage backend with machine learning algorithms. The integration allows for sharing of resources between analytics and storage layers, reducing the redundancy in data structure. For example, free-text index and vector-space model can share the pre-processing (tokenization, normalization, etc.) and vocabulary, result in lower memory footprint and lower latency when operating on streaming data.

QMiner architecture consists of several layers, as can be seen in Figure 15. The data is located at the bottom of the architecture diagram, and is stored either externally (e.g. in a database) or internally. Data Layer accesses the data through adapters, which must expose the data sources through a predefined interface. Data Layer provides efficient access to the data by indexing the records, and providing means to sample given a distribution over the records. Analytics layer provides support to define and construct feature vectors out of records and implements several machine learning algorithms, which can be applied to them. All implemented algorithms leverage the support provided by Data Layer. The system can be accessed via JavaScript API, located in the top layer.

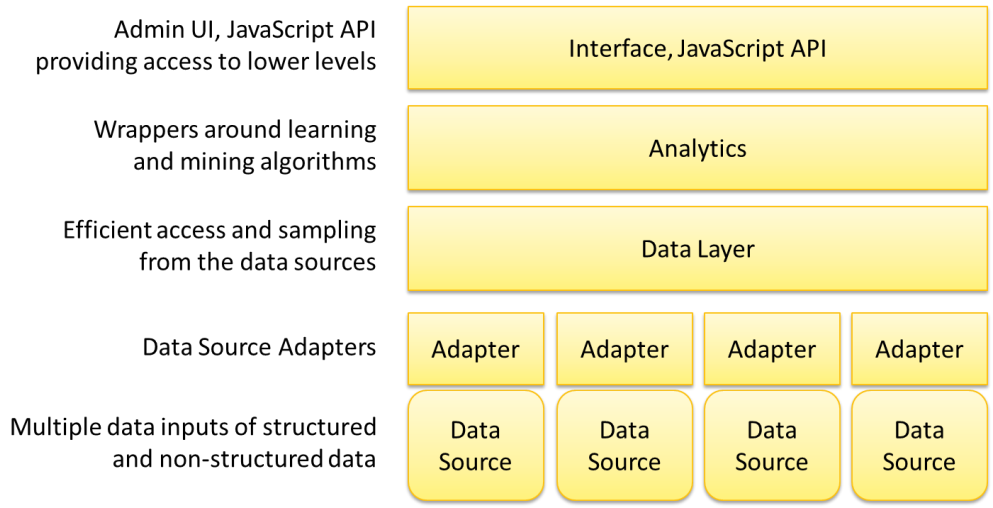

Figure 15 QMiner architecture

The system is fully developed in C++, and can run on Linux or Windows.

5 Conclusions

This document has presented the description and second milestone implementation of the XLike prototype resulting from the work done during the first and second years of the project at the different work packages (WP1-WP5) and the WP6 integration and implementation guide.

The second year prototype has been mostly focused on the accomplishment of the new defined uses cases i) article tracking and event identification (STA), and ii) content advertisement (Bloomberg). Also, two different HCI have been implemented for validating independently these scenarios and they can be consulted at D7.2.2. Despite of the use of different interfaces the core functionalities used for both are common and are the result of the implementation of different components done during the second year.

The continuation of using the technical demo for rapid development and early testing of the ongoing work has been proof to be very effective jointly with bi-weekly Skype calls. This has allowed a good partner communication and quick responses to unpredicted problems. Furthermore weekly calls and SCRUM methodology for accomplishing with specific deadlines has been proof to be very effective towards integration and catch up whenever a new person got involved in the project. Due to during the last year the developing effort tends to decrease these approach will tend to be less useful but we aim to keep using the bi-weekly calls for maintaining communication and empower dissemination activities.

The Sandbox has been enhanced with monitoring tools in order to allow failures detections on services which could also indicate an implementation problem This has allowed to boost the debugging of services in order to obtain a demonstrator prototype more robust which was one of the main goals. Currently the services are reaching almost 100% of reliability and we are processing two third parts of the collected data. It is also expected that the deployment of new functionalities during the Y3 of the project will be decreasing during its first months and then a final prototype will be available. At that stage we are expecting to move all the components (XLike Toolkit) to a scalable platform to provide the best-effort approach in order to gain some improvements on time performance and also on the volume of data able to be processed.

This document is the second of three (D6.1.1 Y1, D6.2.1, Y2, D6.2.3 Y3) which updates the previous one by incorporating the new functionalities implemented during the second year, the platform performance, and also incorporates the feedback obtained by the end-users STA and Bloomberg.

References

[1] XLike deliverable[25] *D1.2.2 – "Requirements for demonstrator prototype"*

[2] XLike deliverable *D2.1.1 – "Shallow linguistic processing prototype"*

[3] XLike deliverable *D4.1.1 – "Cross-lingual document linking prototype"*

[4] XLike deliverable *D5.2.2 – "Final information visualization prototype"*

[5] XLike deliverable *D6.1.1 – "Early toolkit architecture specification"*

[6] XLike deliverable *D6.1.2 – "Final toolkit architecture specification"*

[7] XLike deliverable *D6.2.1 – "Early Prototype"*

[8] XLike deliverable *D8.2.2 – "XLike showcase"*

[9] XLike deliverable "D6.3.1 – API specification and prototype"

[25] All the deliverables of the XLike project are accessible at: http://www.xlike.org/deliverables/

Annex A API Definition

This Annex collects the APIs for the web services of all the different components used for the development of the XLike Y2-Prototype

WP1 Definition and Data Provision

Table 2 WP1 Definition and Data provision API

Language	Description	URL SandBox	Parameters	Example of use
INDEPENDENT	Stories	http://newsfeed.ijs.si/xlike/stories	Id: identification of the story to be searched which contains a set of articles	http://newsfeed.ijs.si/xlike/stories?id=1
INDEPENDENT	Entities	http://newsfeed.ijs.si/xlike/entities	Id: identification of the entity to be searched which is contained in a set of article	http://newsfeed.ijs.si/xlike/entities?id=1
INDEPENDENT	Articles	http://newsfeed.ijs.si/xlike/article	Id: identification of the article to be searched	http://newsfeed.ijs.si/xlike/articles?id=1
INDEPENDENT	Search functionalities	http://newsfeed.ijs.si/xlike/search	q: the keyword to be searched which is contained in a set of articles	http://newsfeed.ijs.si/xlike/search?q=obama

WP2 Shallow/Deep linguistic processing

Table 3 WP2 Shallow/Deep Linguistic Processing API

Language	Description	URL SandBox	Parameters	Example of use

Language Identification Service	base_url/language_code/ident	http://sandbox-xlike.isoco.com/services/language_code/ident	<analyze> <text>Article</text> </analyze>	<analyze><text>Bruce Springsteen is an American singer-songwriter and multi-instrumentalist.</text></analyze>
English Service	base_url/ analysis_en /analyze	http://sandbox-xlike.isoco.com/services/analysis_en/analyze	<analyze> <text>Article</text> <target>relations</target> <conll>true</conll> </analyze>	<analyze><text>Bruce Springsteen is an American singer-songwriter and multi-instrumentalist.</text><target>relations</target><conll>true</conll></analyze>
Spanish Service	base_url/ analysis_es /analyze	http://sandbox-xlike.isoco.com/services/analysis_es/analyze	<analyze> <text>Article</text> <target>relations</target> <conll>true</conll> </analyze>	<analyze><text>Blade Runner es una película de ciencia ficción estadounidense dirigida por Ridley Scott.</text><target>relations</target><conll>true</conll></analyze>
Catalan Service	base_url/ analysis_ca /analyze	http://sandbox-xlike.isoco.com/services/analysis_ca/analyze	<analyze> <text>Article</text> <target>relations</target> <conll>true</conll> </analyze>	<analyze><text>L'iPhone és un dispositiu electrònic multimèdia presentat per Apple Computer el 9 de gener de 2007.</text><target>relations</target><conll>true</conll></analyze>
German Service	base_url/ analysis_de /analyze	http://lt.ffzg.hr:9090/xlike/analysis_de/analyze	<analyze> <text>Article</text> <conll>true</conll> </analyze>	<analyze><text>Clara Schumann war eine deutsche Pianistin und Komponistin und ab 1840 die Ehefrau Robert Schumanns.</text><conll>true</conll></analyze>
Chinese Service	base_url/ analysis_zh /analyze	http://keg.cs.tsinghua.edu.cn:8080/analysis_zh/analyze	<analyze> <text>Article</text> <conll>true</conll> </analyze>	<analyze><text>正面的观点认为,由于元朝从忽必烈即位后就开始“行汉法”,</text><conll>true</conll></analyze>
Slovenian Service	base_url/ analysis_sl /analyze?text=text to identify	http://aidemo.ijs.si/xlike/analysis_sl/analyze	<analyze> <text>Article</text> <conll>true</conll> </analyze>	<analyze><text>Clara je bila žena skladatelja Roberta Schumanna in ena vodilnih pianistov in skladateljev romantike.</text><conll>true</conll></analyze>

WP3 Final annotation text prototype

Table 4 WP3 Final Annotation Text Prototype

Language	Description	URL SandBox	Parameters	Example of use
English Service	Wififier: base_url/ annotation-en/ NER based: base_url/ner-annotation-en	http://km.aifb.kit.edu/services/annotation-en/ http://km.aifb.kit.edu/services/ner-annotation-en/	> <item> > <sentences> > <sentence id=""> > <text> </text> > <tokens> > <token pos=" " end="" lemma=" " id="" start=""> </token> > </tokens> > </sentence> > </sentences> > <entities> > <entity type=" " displayName=" " id=""> > <mentions> > <mention sentenceId="" id="" words=" "></mention> > </mentions> > </entity> > </entities> > </item>	See Table 5 for description
Spanish Service	Wikifier: base_url/ annotation-es/ NER based: base_url/ner-annotation-es	http://km.aifb.kit.edu/services/annotation-es/ http://km.aifb.kit.edu/services/ner-annotation-es/	> Same as English Service	See Table 5 for description (the schema is the same than for Spanish)
Catalan Service	NERbased: base_url/ner-annotation-ca	http://km.aifb.kit.edu/services/ner-annotation-ca/	> Same as English Service	See Table 5 for description (the schema is the same than for Spanish)

German Service	Wikifier: base_url/ annotation-de/ NER based: base_url/ner-annotation-de	http://km.aifb.kit.edu/services/annotation-de/ http://km.aifb.kit.edu/services/ner-annotation-de/	› Same as English Service	See Table 5 for description (the schema is the same than for Spanish)
Chinese Service	NER based: base_url/ner-annotation-zh	http://km.aifb.kit.edu/services/ner-annotation-zh/	› Same as English Service	See Table 5 for description (the schema is the same than for Spanish)
Slovenian Service	NER based: base_url/ner-annotation-sl	http://km.aifb.kit.edu/services/ner-annotation-sl/	› Same as English Service	See Table 5 for description (the schema is the same than for Spanish)

Table 5 Example of use of the final text annotation prototype

```
› <item>
› <services>
› <service name="UPC-analysis" date="2013-10-29"> </service>
› </services>
› <sentences>
› <sentence id="1">
› <text>Bruce Springsteen is an American singer-songwriter and multi-instrumentalist.</text>
› <tokens>
› <token pos="NP00SP0" end="17" lemma="bruce_springsteen" id="1.1" start="0">Bruce_Springsteen</token><token pos="VBZ" end="20" lemma="be" id="1.2" start="18">is</token>
› <token pos="Z" end="23" lemma="1" id="1.3" start="21">an</token>
› <token pos="NP00V00" end="32" lemma="american" id="1.4" start="24">American</token>
› <token pos="NN" end="50" lemma="singer-songwriter" id="1.5" start="33">singer-songwriter</token><token pos="CC" end="54" lemma="and" id="1.6" start="51">and</token>
› <token pos="NN" end="76" lemma="multi-instrumentalist" id="1.7" start="55">multi-instrumentalist</token>
› <token pos="Fp" end="77" lemma="." id="1.8" start="76">.</token>
› </tokens>
› </sentence>
› </sentences>
```

```
› <?xml version="1.0" encoding="UTF-8" standalone="no"?>
› <item>
› <services>
› <service date="2013-10-29" name="UPC-analysis"/>
› <service date="2013-12-23" name="KIT-annotation"/>
› </services>
› <sentences>
› <sentence id="1">
› <text>Bruce Springsteen is an American singer-songwriter and multi-instrumentalist.</text>
› <tokens>
› <token end="17" id="1.1" lemma="bruce_springsteen" pos="NP00SP0" start="0">Bruce_Springsteen</token>
› <token end="20" id="1.2" lemma="be" pos="VBZ" start="18">is</token>
› <token end="23" id="1.3" lemma="1" pos="Z" start="21">an</token>
› <token end="32" id="1.4" lemma="american" pos="NP00V00" start="24">American</token>
› <token end="50" id="1.5" lemma="singer-songwriter" pos="NN" start="33">singer-songwriter</token>
› <token end="54" id="1.6" lemma="and" pos="CC" start="51">and</token>
› <token end="76" id="1.7" lemma="multi-instrumentalist" pos="NN" start="55">multi-instrumentalist</token>
› <token end="77" id="1.8" lemma="." pos="Fp" start="76">.</token>
› </tokens>
› </sentence>
› </sentences>
› <nodes>
```

```
> <nodes>
> <node type="entity" class="other" displayName="american" id="E2">
> <mentions>
> <mention sentenceId="1" id="E2.1" words="American"><mention_token
id="1.4"> </mention_token>
> </mention>
> </mentions>
> </node>
> <node type="entity" class="person" displayName="bruce_springsteen"
id="E1">
> <mentions>
> <mention sentenceId="1" id="E1.1" words="Bruce Springsteen">
> <mention_token id="1.1"> </mention_token> </mention>
> </mentions>
> </node>
> </nodes>
> </item>
```

```
> <node class="other" displayName="american" id="E2" type="entity">
> <mentions>
> <mention id="E2.1" sentenceId="1" words="American">
> <mention_token id="1.4"> </mention_token> </mention> </mentions>
> <descriptions>
> <description URI="http://en.wikipedia.org/wiki/United_States" confidence="1.0"
displayName="United States" knowledgeBase="Wikipedia" lang="en"/>
> <description URI="http://de.wikipedia.org/wiki/Vereinigte_Staaten" confidence="1.0"
displayName="United States" knowledgeBase="Wikipedia" lang="de"/>
> <description URI="http://es.wikipedia.org/wiki/Estados_Unidos" confidence="1.0"
displayName="United States" knowledgeBase="Wikipedia" lang="es"/>
> <description URI="http://sl.wikipedia.org/wiki/ZdruÅ¾ene_drÅ¾ave_Amerike"
confidence="1.0" displayName="United States" knowledgeBase="Wikipedia" lang="sl"/>
> <description URI="http://zh.wikipedia.org/wiki/ç¾å" confidence="1.0" displayName="United
States" knowledgeBase="Wikipedia" lang="zh"/>
> <description URI="http://ca.wikipedia.org/wiki/Estats_Units_d'AmÃ¨rica" confidence="1.0"
displayName="United States" knowledgeBase="Wikipedia" lang="ca"/>
> </descriptions>
> </node>
> <node class="person" displayName="bruce_springsteen" id="E1" type="entity">
> <mentions>
> <mention id="E1.1" sentenceId="1" words="Bruce Springsteen">
> <mention_token id="1.1"> </mention_token> </mention>
> </mentions>
> <descriptions>
> <description URI="http://en.wikipedia.org/wiki/United_States" confidence="1.0"
displayName="United States" knowledgeBase="Wikipedia" lang="en"/>
> <description URI="http://de.wikipedia.org/wiki/Vereinigte_Staaten" confidence="1.0"
displayName="United States" knowledgeBase="Wikipedia" lang="de"/>
> <description URI="http://es.wikipedia.org/wiki/Estados_Unidos" confidence="1.0"
displayName="United States" knowledgeBase="Wikipedia" lang="es"/>
> <description URI="http://sl.wikipedia.org/wiki/Združene_države_Amerike" confidence="1.0"
displayName="United States" knowledgeBase="Wikipedia" lang="sl"/>
> <description URI="http://zh.wikipedia.org/wiki/美國" “ confidence="1.0"
displayName="United States" knowledgeBase="Wikipedia" lang="zh"/>
> <description URI="http://ca.wikipedia.org/wiki/ Estats_Units_d'Amèrica" confidence="1.0"
displayName="United States" knowledgeBase="Wikipedia" lang="ca"/>
> </descriptions>
> </node>
> </nodes>
> </item>
```

WP4 Cross-lingual document linking prototype

Table 6 WP4 Cross-lingual Document Linking Prototype

Language	Description	URL SandBox	Parameters	Example of use
EN, ES, DE, SL, CA	Document similarity (to be use for comparison between definition of a STA use case and an article)	http://km.aifb.kit.edu/services/clesa/similarity	doc1 lang1 doc2 lang2	http://km.aifb.kit.edu/services/clesa/similarity?doc1=Bruce%20Springsteen%20is%20an%20American%20singer-songwriter%20and%20multi-instrumentalist.&lang1=en&doc2=Bruce%20Springsteen%20es%20un%20cantante%20y%20m%C3%BAsico%20americano&lang2=es
EN, ES, DE, SL, CA	Wikipedia analysis based on ESA	http://km.aifb.kit.edu/services/clesa/analyzer	Doc1 lang1 lang2 retrieve	http://km.aifb.kit.edu/services/clesa/analyzer?doc=Bruce%20Springsteen%20is%20an%20American%20singer-songwriter%20and%20multi-instrumentalist.&lang1=en&lang2=es&retrieve=2
EN, DE,ES, SL, CA, ZH	Document similarity between two given documents/articles	http://xling.ijs.si:1111/clsi	Doc1 Lang1 Doc2 Lang2	http://xling.ijs.si:1111/clsi?doc1=car&lang1=en&doc2=coche&lang2=es
EN, ES, DE, FR, IT	Bloomberg articles related with an specific language/region	https://aidemo.ijs.si/xlike/hs/topstories	Region	https://aidemo.ijs.si/xlike/hs/topstories?region=de

WP5 Final Information Visualization and Event Detection

Table 7 Final information visualization

Language	Description	URL SandBox	Parameters	Example of use
ALL	Xlike - News Data Visualization	http://sandbox-isoco.com/portal/index.html	None	Visual human-computer interaction
ALL	XLike Entity/Keyword search Service	http://mustang.ijs.si:8082/xlike/search	Q url page pagesize lang x6 group callback	http://mustang.ijs.si:8082/xlike/search?q=Keyword&url=EntityURI&page=0&pagesize=500&ts=30d&lang=eng&lang=deu&lang=spa&lang=zho&lang=slv&lang=cat&group=general&callback=

Annex B Demos and prototypes

This Annex contains the updated references to all the demos and prototypes implemented during the Y1 and Y2 of the XLike project.

Table 8 Demos and Prototypes

Language	Description	URL SandBox
INDEPENDENT	Technological demo which provides the WP2 and WP3 annotation functionalities of the project	http://sandbox-xlike.isoco.com/demo/
INDEPENDENT	Demonstrator prototype of the project (Public)	http://sandbox-xlike.isoco.com/portal/
INDEPENDENT	Early prototype of the XLike Project to be used for validation purposes of the STA use case (Private[26])	http://sandbox-xlike.isoco.com/portal/STA/
INDEPENDENT	Cross-lingual similarity demo	http://xling.ijs.si:1111/wikipedia.html
German (all other languages expected for Y2)	Cross-lingual similarity demo applied to Bloomberg use case	http://xling.ijs.si:1111/bloomberg.html
INDEPENDENT	Cross-lingual Recommendation Application (Bloomberg use case private[27])	http://sandbox-xlike.isoco.com/portal/newsApp/index.html
INDEPENDENT	Event Registry application	http://eventregistry.org/

[26] This prototype makes use of private STA data and therefore it has been agreed to forbid its access to the public.
[27] This demo makes use of private Bloomberg data and therefore it has been agreed to forbid its access to the public.

Annex C Data Format

This annex contains an example of the data and its format collected after going through the pipeline from WP1, WP2, WP3, and WP4 (see Table 9) and shows how this data enriches the raw article by adding it verbatim to the previous information obtained by Newsfeed (see Table 10).

Table 9 Example of the XML format obtained by WP2 + annotations (WP3 and WP4)

```
<item>
  <services>
    <service date="2013-10-29" name="UPC-analysis"/>
    <service date="2013-12-23" name="KIT-annotation"/>
  </services>
  <sentences>
    <sentence id="1">
      <text>Bruce Springsteen is an American singer-songwriter and multi-instrumentalist.</text>
      <tokens>
        <token end="17" id="1.1" lemma="bruce_springsteen" pos="NP00SP0" start="0">Bruce_Springsteen</token>
        <token end="20" id="1.2" lemma="be" pos="VBZ" start="18">is</token>
        <token end="23" id="1.3" lemma="1" pos="Z" start="21">an</token>
        <token end="32" id="1.4" lemma="american" pos="NP00V00" start="24">American</token>
        <token end="50" id="1.5" lemma="singer-songwriter" pos="NN" start="33">singer-songwriter</token>
        <token end="54" id="1.6" lemma="and" pos="CC" start="51">and</token>
        <token end="76" id="1.7" lemma="multi-instrumentalist" pos="NN" start="55">multi-instrumentalist</token>
        <token end="77" id="1.8" lemma="." pos="Fp" start="76">.</token>
      </tokens>
    </sentence>
  </sentences>
  <nodes>
    <node class="other" displayName="american" id="E2" type="entity">
      <mentions>
        <mention id="E2.1" sentenceId="1" words="American">
          <mention_token id="1.4"> </mention_token> </mention> </mentions>
      <descriptions>
        <description URI="http://en.wikipedia.org/wiki/United_States" confidence="1.0" displayName="United
States" knowledgeBase="Wikipedia" lang="en"/>
        <description URI="http://de.wikipedia.org/wiki/Vereinigte_Staaten" confidence="1.0" displayName="United
States" knowledgeBase="Wikipedia" lang="de"/>
        <description URI="http://es.wikipedia.org/wiki/Estados_Unidos" confidence="1.0" displayName="United
States" knowledgeBase="Wikipedia" lang="es"/>
        <description URI="http://sl.wikipedia.org/wiki/Združene_države_Amerike" confidence="1.0"
displayName="United States" knowledgeBase="Wikipedia" lang="sl"/>
        <description URI="http://zh.wikipedia.org/wiki/美國 " confidence="1.0" displayName="United States"
knowledgeBase="Wikipedia" lang="zh"/>
        <description URI="http://ca.wikipedia.org/wiki/ Estats_Units_d'Amèrica" confidence="1.0"
displayName="United States" knowledgeBase="Wikipedia" lang="ca"/>
      </descriptions>
    </node>
    <node class="person" displayName="bruce_springsteen" id="E1" type="entity">
      <mentions>
        <mention id="E1.1" sentenceId="1" words="Bruce Springsteen">
          <mention_token id="1.1"> </mention_token> </mention>
      </mentions> <descriptions>
        <description URI="http://en.wikipedia.org/wiki/Bruce_Springsteen" confidence="0.564" displayName="Bruce
Springsteen" knowledgeBase="Wikipedia" lang="en"/>
        <description URI="http://de.wikipedia.org/wiki/Bruce_Springsteen" confidence="0.564" displayName="Bruce
Springsteen" knowledgeBase="Wikipedia" lang="de"/>
        <description URI="http://es.wikipedia.org/wiki/Bruce_Springsteen" confidence="0.564" displayName="Bruce
Springsteen" knowledgeBase="Wikipedia" lang="es"/>
        <description URI="http://zh.wikipedia.org/wiki/布鲁斯·斯普林斯廷" confidence="0.564"
displayName="Bruce Springsteen" knowledgeBase="Wikipedia" lang="zh"/>
        <description URI="http://ca.wikipedia.org/wiki/Bruce_Springsteen" confidence="0.564" displayName="Bruce
Springsteen" knowledgeBase="Wikipedia" lang="ca"/>
      </descriptions>
    </node>
  </nodes>
</item>
```

The schema for this example is the following and can be found at[28], and the global structure of the xml file including the information of the article provided by the WP1 is the following[29],[30] [7]:

Table 10 Complete XML data format of the XLike prototype

```
<article id="internal article ID; consistent across streams">
   <source>
      <hostname> Publisher hostname </hostname>
      <title> Name of the publisher; failing that, title of the RSS feed </title>
      <location?>
         <longitude?> publisher longitude in degrees </longitude>
         <latitude?> publisher latitude in degrees </latitude>
         <city?> publisher city </city>
         <country?> publisher country </country>
      </location>
      <tags?>
         <tag+> a tag for the publisher; the vocabulary is not controlled </tag>
      </tags>
   </source>
   <feed+>
      <uri> URL from which the article was discovered; typically the RSS feed </uri>
   </feed>
   <uri> URL from which the article was downloaded </uri>
   <publish-date?> The publication time and date. </publish-date>
   <retrieve-date> The retrieval time and date. </retrieve-date>
   <lang> 3-letter ISO 639-2 language code </lang>
   <location? +>
      <longitude?> story content longitude in degrees </longitude>
      <latitude?> story content latitude in degrees </latitude>
      <city?> story city </city>
      <country?> story country </country>
   </location>
   <tags?>
      <tag+> a tag for the article; the vocabulary is not controlled </tag>
   </tags>
   <img?> The URL of a related image, usually a thumbnail. </img>
   <title> Title. Can be empty if we fail to identify it. </title>
   <body-cleartext>
      Clear text body of the article, formatted only with <p> tags
   </body-cleartext>
   <body-rych?; only English, Slovene>
      Enriched article body; an XML subtree as returned by Enrycher.
   </body-rych>
   <body-xlike?; only English, Spanish, Catalan>
      Enriched article body; an XML subtree as returned by iSOCO; experimental.
   </body-xlike>
</article>
```

Article information

Enrycher information

XLike information

[28] https://github.com/xlike-project/wp6/blob/master/schemas/document2013.xsd

[29] http://newsfeed.ijs.si/

[30] The enrycher provided information can be consulted at http://enrycher.ijs.si/