XLike
FP7-ICT-2011.4.2
Contract no.: 288342
www.xlike.org

XLike

Deliverable D6.1.2

Final toolkit architecture specification

Editor:	Esteban García-Cuesta, iSOCO
Author(s):	Esteban García-Cuesta, iSOCO; Alejandro Caparros, iSOCO; Blaž Fortuna, JSI; Xavier Carreras, UPC; Lei Zhang, KIT; Zhixing Li, THU; Achim Rettinger, KIT;
Deliverable Nature:	Report
Dissemination Level: (Confidentiality)[1]	Public (PU)
Contractual Delivery Date:	M15
Actual Delivery Date:	2.4.2013
Suggested Readers:	Developers creating software components to be integrated, developers creating case study prototypes, software developers.
Version:	1.0
Keywords:	toolkit; development; components; architecture;

[1] Please indicate the dissemination level using one of the following codes:
• **PU** = Public • **PP** = Restricted to other programme participants (including the Commission Services) • **RE** = Restricted to a group specified by the consortium (including the Commission Services) • **CO** = Confidential, only for members of the consortium (including the Commission Services) • **Restreint UE** = Classified with the classification level "Restreint UE" according to Commission Decision 2001/844 and amendments • **Confidentiel UE** = Classified with the mention of the classification level "Confidentiel UE" according to Commission Decision 2001/844 and amendments • **Secret UE** = Classified with the mention of the classification level "Secret UE" according to Commission Decision 2001/844 and amendments

Disclaimer

This document contains material, which is the copyright of certain XLike consortium parties, and may not be reproduced or copied without permission.

All XLike consortium parties have agreed to full publication of this document.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the XLike consortium as a whole, nor a certain party of the XLike consortium warrant that the information contained in this document is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using this information.

Full Project Title:	Cross-lingual Knowledge Extraction
Short Project Title:	XLike
Number and Title of Work package:	WP6 – Integration and Toolkit
Document Title:	D6.1.2 – Final toolkit architecture specification
Editor (Name, Affiliation)	Esteban García-Cuesta, iSOCO
Work package Leader (Name, affiliation)	Esteban García-Cuesta, iSOCO
Estimation of PM spent on the deliverable:	9PM

Copyright notice

Executive Summary

This document presents the description of the final XLike toolkit architecture which will be the main tangible outcome of the project.

The document is based on the functional and technical specifications, and the requirements of the project which are collected at D1.2.2 “Requirements for demonstrator” and are based on the use cases defined by Bloomberg and STA for the year two of the project. The main goal is to update the previously defined toolkit architecture D6.1.1 “Early Toolkit Architecture” which was the early definition of the toolkit architecture covering the needs of publishers and related industry for the first year and to fulfill the reviewers’ recommendations regarding the inclusion of references of how the use cases requirements are accomplished in the project.

The outcome of the task is a final report with an updated specification of the toolkit infrastructure, a summary of components including their association with the requirements that they accomplish with and its internal design description, and the current implementation of the sandbox for the X-LIKE project which allocates both the technological demo and the STA year one use case. This report will be used as reference for the rest of the project both from a development and technical point of view.

Table of Contents

Executive Summary 3
Table of Contents 4
List of Figures 6
List of Tables 7
Abbreviations 8
Definitions 9
1 Introduction 10
1.1 Sandbox Role 11
1.1.1 Sandbox Configuration Summary 13
2 Use Cases 14
2.1 Topic and Entity tracking (STA) (Identifier UC4) 14
2.2 Related/Relevant articles (Bloomberg) (Identifier UC1) 16
2.3 Content advertising (Bloomberg) (Identifier UC2) 16
2.4 Article tracking (STA) (Identifier UC3) 17
2.5 Event identification (STA) (Identifier UC5) 17
2.6 Technological Demo 18
3 XLike Toolkit Components 19
3.1 Acquisition 19
3.1.1 NewsFeed 19
3.1.2 iSOCO real time storage and access 20
3.2 Analysis and Interpretation 20
3.2.1 Shallow and Deep Linguistic Processing 20
3.2.1.1 Language Identification 20
3.2.1.2 Multilingual Analysis 21
3.2.1.3 Deep Linguistic Analysis 22
3.2.1.4 Linguistic Relation Extraction 23
3.2.1.5 Informal Languages Analysis 24
3.2.2 Text and Semantic Annotation 24
3.2.2.1 Name Entity Annotation 24
3.2.2.2 Wikipedia Miner Wikifier Annotation 25
3.2.2.3 Early Ontological Word-sense-disambiguation 26
3.2.2.4 Crowd Sourcing Word-sense-disambiguation 26
3.2.2.5 Final Ontological Word-sense-disambiguation 27
3.2.2.6 Final Text Annotation Service 27
3.2.2.7 Early Machine Translation based Semantic Annotation 28
3.2.2.8 Final Machine Translation based Semantic Annotation 29
3.2.2.9 Cross-lingual USP 29
3.2.3 Cross-lingual Document Linking 30
3.2.3.1 KIT Cross-lingual Similarity 30
3.2.3.2 JSI Cross-lingual Similarity 30
3.2.3.3 Cross-lingual Analysis 31
3.2.3.4 Cross-lingual Document Linking 32
3.2.4 Semantic Graphs Constructions and Event Extraction 33
3.2.4.1 Semantic Graph Extraction 33
3.2.4.2 Event Extraction 33
3.2.4.3 Detection of news reporting bias 34
3.2.4.4 Trend and complex event detection 35
3.3 Human Computer Interaction (HCI) 36
3.3.1 News Data Visualization 36
4 APIs Overview 38

4.1 Acquisition ... 38

4.1.1 NewsFeed ... 38

4.1.2 iSOCO real time storage and access ... 40

4.2 Analysis and Interpretation ... 40

4.2.1 Language Identification ... 40

4.2.2 Multilingual Analysis ... 41

4.2.3 Deep Linguistic Analysis ... 42

4.2.4 Linguistic Relation Extraction ... 42

4.2.5 Name Entity Annotation ... 43

4.2.6 KIT Cross-lingual Similarity ... 43

4.2.7 JSI Cross-lingual Similarity ... 44

4.2.8 Cross-lingual Analysis ... 45

5 Conclusions ... 46

6 References ... 47

Annex A Requirements-Components Relation ... 49

List of Figures

Figure 1: XLike Toolkit 10
Figure 2: XLike Toolkit deployment diagram 11
Figure 3 Annotation performance for selected languages 12
Figure 4: Cassandra data model for XLike multilingual processing 13
Figure 5 XLike Toolkit deployed dependencies 14
Figure 6 Entity tracking pipeline 15
Figure 7 Entity tracking 15
Figure 8 Relevant articles pipeline 16
Figure 9 Recommender article feature on Bloomberg.com 16
Figure 10 Content advertising pipeline 17
Figure 11 Article tracking pipeline 17
Figure 12 Event identification pipeline 18
Figure 13 Technological Demo functionalities 18

List of Tables

Table 1: Data Format 39
Table 2. Relation Between Requirements and Services 49

Abbreviations

API	Application Programming Interface
D	Deliverable
SOA	Service Oriented Architecture
T	Task
URL	Uniform Resource Locator
WP	Work Package

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

1 Introduction

One of the main tangible outcomes of the XLike project is the technological infrastructure for the efficient retrieval of information from multiple languages and to provide cross-language capabilities to analyse that data for building new applications on top of them (e.g. recommendation of similar articles in different languages). In order to produce this outcome we define a software architecture for the design and implementation of a cross lingual infrastructure, and we ground it by a reference of its implementation which is so called "XLike Toolkit". The architecture is described by different layers at Figure 1 grouping each one a common set of components which share a common abstract functionality. This infrastructure architecture was introduced at D6.1.1.[1] defining three main functional layers: acquisition, analysis and interpretation, and human-computer interface, and has been filled with a set of functional components RQX which provide the different functionalities at each layer compiling, jointly with the clients implementation, the toolkit of the project (see Annex A for the relation with the toolkit components). This process of generating the toolkit has been done following an agile development approach allowing the co-evolution of the architecture, the infrastructure, the toolkit, and the Xlike prototypes.

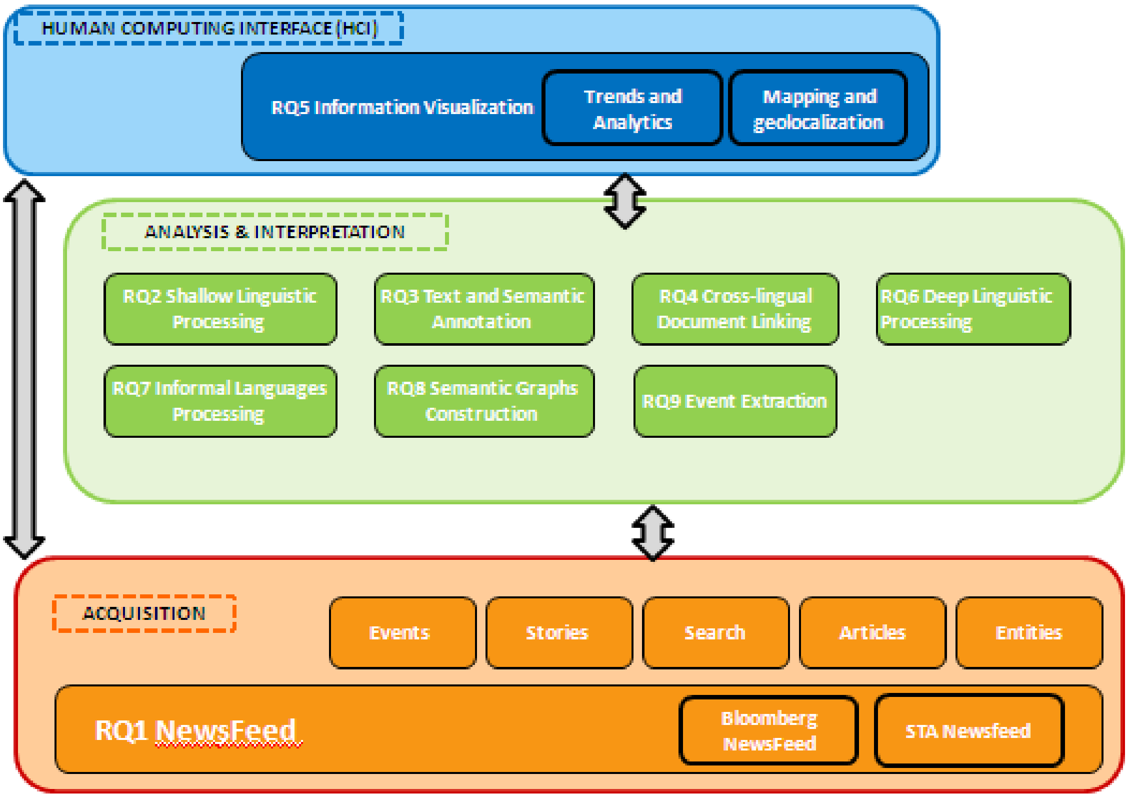

HUMAN COMPUTING INTERFACE (HCI)
RQ5 Information Visualization
Trends and Analytics
Mapping and geolocalization
ANALYSIS & INTERPRETATION
RQ2 Shallow Linguistic Processing
RQ3 Text and Semantic Annotation
RQ4 Cross-lingual Document Linking
RQ6 Deep Linguistic Processing
RQ7 Informal Languages Processing
RQ8 Semantic Graphs Construction
RQ9 Event Extraction
ACQUISITION
Events
Stories
Search
Articles
Entities
RQ1 NewsFeed
Bloomberg NewsFeed
STA Newsfeed

Figure 1: XLike Toolkit

Figure 1 represents the current XLike Toolkit of services[2] (both, implemented during the Y1 of the project and the expected ones detected at M15 to accomplish with the new use cases defined at D1.2.2 "Requirements for demonstrator" [2] and the industrial showcase defined at D8.2.1 "XLike Showcase specification" [3]) which covers the functionalities of data acquisition, storage and management, analysis and interpretation of data, and external access to these functionalities (exposed mostly via RESTful APIs along with client applications which provide access to the data and the indicated functionalities. This

[2] Note: the representation has been done by grouping the components by its associated functional requirement. This relationship can be consulted at Annex A.

document also includes the updates of the used components identified in the first version of the document and the new ones implemented and needed for accomplishing with the current identified new use cases.

1.1 Sandbox Role

The XLike Sandbox[3] is the platform where the implemented services have been deployed for testing and validation purposes, and for the deployment of the early prototype at the end of the first year. This Sandbox has been implemented during the first year of the project and will be maintained for the rest of the project for the same initial defined purposes of testing and validation. Currently not all the services are being deployed in the Sandbox as can be seen in the Figure 2. and some are being host at XLike partners' institutions (notice that the indicated new components still have not been implemented/deployed and are placed in the institution that is responsible for them).

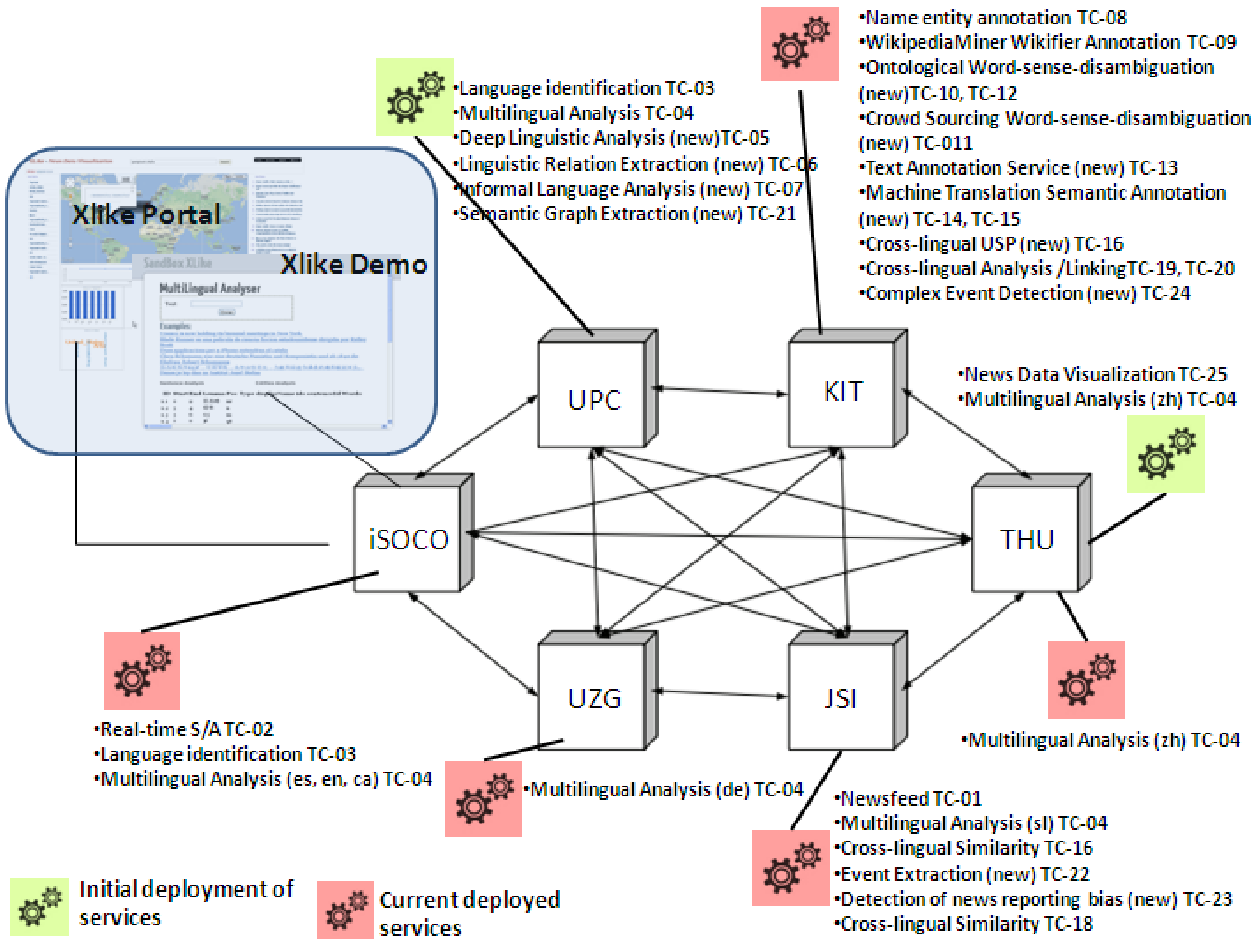

•Name entity annotation TC-08
•WikipediaMiner Wikifier Annotation TC-09
•Ontological Word-sense-disambiguation (new)TC-10, TC-12
•Crowd Sourcing Word-sense-disambiguation (new) TC-011
•Text Annotation Service (new) TC-13
•Machine Translation Semantic Annotation (new) TC-14, TC-15
•Cross-lingual USP (new) TC-16
•Cross-lingual Analysis /LinkingTC-19, TC-20
•Complex Event Detection (new) TC-24
•Language identification TC-03
•Multilingual Analysis TC-04
•Deep Linguistic Analysis (new)TC-05
•Linguistic Relation Extraction (new) TC-06
•Informal Language Analysis (new) TC-07
•Semantic Graph Extraction (new) TC-21
Xlike Portal
Xlike Demo
UPC
KIT
•News Data Visualization TC-25
•Multilingual Analysis (zh) TC-04
iSOCO
THU
UZG
JSI
•Multilingual Analysis (zh) TC-04
•Real-time S/A TC-02
•Language identification TC-03
•Multilingual Analysis (es, en, ca) TC-04
•Multilingual Analysis (de) TC-04
•Newsfeed TC-01
•Multilingual Analysis (sl) TC-04
•Cross-lingual Similarity TC-16
•Event Extraction (new) TC-22
•Detection of news reporting bias (new) TC-23
•Cross-lingual Similarity TC-18
🟩 Initial deployment of services
🟥 Current deployed services

Figure 2: XLike Toolkit deployment diagram

During the year two the industrial showcase defined at (D8.2.1 "Showcase Specification" [3]) has to be accomplished (M24) and the Sandbox will be upgraded for that purpose. Among other needed changes the scalability problem needs to be addressed in order to be able to accomplish with the real time needs of the showcase. For this purpose the Sandbox and the overall pipeline has been designed and is being implemented to allow the use of this Sandbox for real-time purposes and making use of the whole pipeline as implemented in the current XLike Toolkit. This challenge is possible to achieve during the year two

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

mainly because a big effort have been done during the first year towards standardizing to **RESTful Web Services** the different components which were already available to the project (e.g. converting Freeling from a stand-alone application to a set of web services following the SOA approach TC-04), and following the same SOA approach for the creation of the new ones (e.g. cross-lingual similarity service TC-16).

Regarding the scalability of the current deployed platform some initial test have been perform for the STA entity tracking use case (which includes the language identification, multi-language analysis, text annotation, and cross-lingual document linking) with the current state of the platform which was introduced at D6.1.1 "Early toolkit architecture specification" [4]. This initial performance test has shown that the different components used to solve that use case scales following a linear logarithmic relation (marked with a black line in the Figure 3) meaning that all the components scales well from an algorithms coding point of view [6]. Despite this, there is still a lack of scalability of the platform for storage and accessibility aspects.

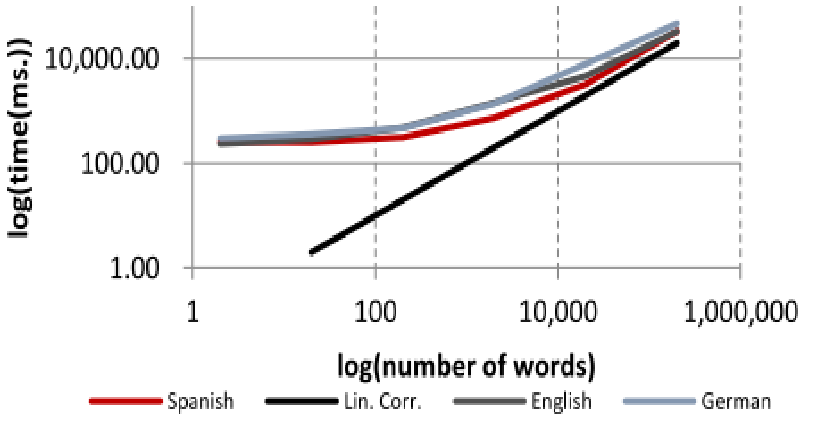


Figure 3 Annotation performance for selected languages

To address this requirement a new component (TC-02) is being developed to provide large scale and real-time capabilities to the XLike project by using the open source distributed database management system Apache Cassandra[4]. Cassandra is classified as no-SQL Database initially developed by Facebook and currently is used by a lot of companies like EBay, HP, Twitter or Spotify.

Currently we are adapting Cassandra to X-Like architecture as storage service for allowing an easy way to store and access the data. One of the most important capabilities of Cassandra is the possibility to create a cluster of nodes. At present, we have only implemented one node but it can be scaled to any number of nodes as needed. Also, two RESTful services have been implemented for providing entities and related articles functionalities over data stored already in Cassandra. This set of services will be enlarged and enhanced as needed to provide the functionalities to accomplish with the industrial showcase (e.g. providing access to event) and the new use cases of the project.

We also have made a rough study based on the current newsfeed data volume having a total amount of 228MB generated per day which could be used as a first approximation to the real industrial environment.
This turns out into a total of 81GB per year before analyzing and enriching the text with the results obtained after applying the XLike pipeline. We also did a quick study of how much information was generated by XLike project enrichment pipeline (including also the source metadata) and we obtained an expected amount of 1809MB per day turning out into 644GB per year. Such amount of data is difficult to manage and to query on real time mainly due to the read/write rapid needs. For optimization purposes it is very important to design a good data model and the implemented for XLike is shown in the Figure 4 which follows the description of the data format described in D6.2.1 "Early Prototype" (Annex C) [5].

[4] http://cassandra.apache.org/

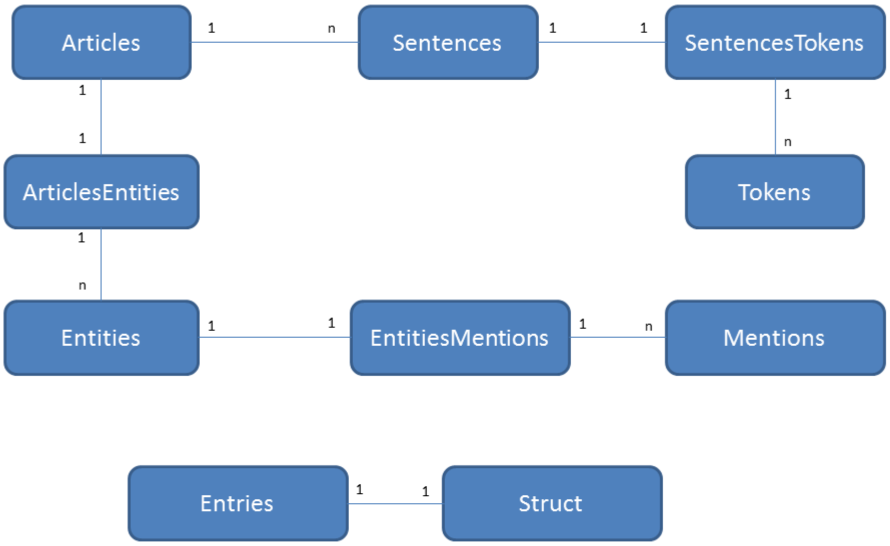


Figure 4: Cassandra data model for XLike multilingual processing

This model is already implemented in Cassandra although new items will be needed and the model will be updated during the second year of the project.

1.1.1 Sandbox Configuration Summary

This subsection is an update of the configuration summary described at D6.1.1 (M3) [4] showing the current specifications of the Sandbox running environment:

- Debian “squeeze” 2.6.32-5-amd64
- Java 1.6
- MySQL 5.1.49
- Git 1.7.2.5
- Apache tomcat 6.0.35 (Apache wicket infrastructure)
- 4store 1.1.1-1
- Apache Cassandra 1.2.1.

The remainder of this document is organized as follows. Section 2 shows the different use cases identified at D1.2.2 "Requirements for demonstrator" [2] which have been the motivators for the development of the different components, and shows also the interactions between them. Section 3 introduces the toolkit components and highlights those implemented at the end of the year one and the ones expected to be done for the year two of the project. Next, section 4 includes the APIs specifications for each one of the toolkit components and finally we present some conclusions at section 5. The Annex A shows the relation between the technical and functional requirements described at D1.2.2 "Requirements for demonstrator" [2] and the implemented components to accomplish with each one of them.

2 Use Cases

To illustrate the usage of the XLike toolkit, and the interactions between their components, we describe the early prototype D6.1.1. "Early toolkit architecture specification” [1] used for the STA use case during the validation of the Y1 of the project. The early prototype accomplished with a set of smaller use case scenarios which make use of the XLike toolkit following the overall pipeline. It is worth highlighting that different tools are used for the different use cases showing its capabilities of independency (e.g. entity extraction) and its interoperability. The Bloomberg prototypes are also described showing the tools used. These scenarios are depicted in the deployment diagram in Figure 5.

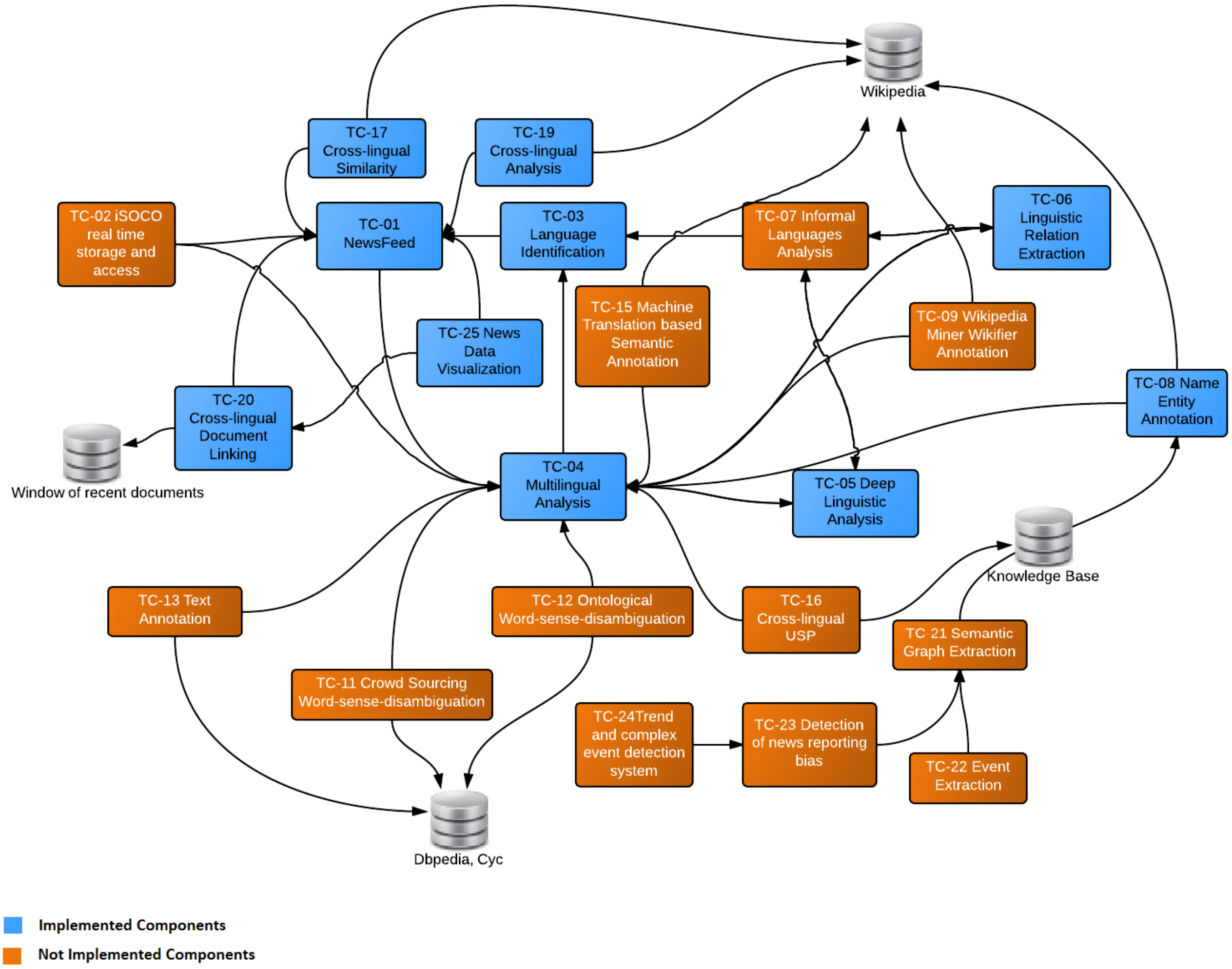

Wikipedia
TC-17 Cross-lingual Similarity
TC-19 Cross-lingual Analysis
TC-02 iSOCO real time storage and access
TC-01 NewsFeed
TC-03 Language Identification
TC-07 Informal Languages Analysis
TC-06 Linguistic Relation Extraction
TC-25 News Data Visualization
TC-15 Machine Translation based Semantic Annotation
TC-09 Wikipedia Miner Wikifier Annotation
TC-08 Name Entity Annotation
TC-20 Cross-lingual Document Linking
Window of recent documents
TC-04 Multilingual Analysis
TC-05 Deep Linguistic Analysis
Knowledge Base
TC-13 Text Annotation
TC-12 Ontological Word-sense-disambiguation
TC-16 Cross-lingual USP
TC 21 Semantic Graph Extraction
TC-11 Crowd Sourcing Word-sense-disambiguation
TC-24Trend and complex event detection system
TC-23 Detection of news reporting bias
TC-22 Event Extraction
Dbpedia, Cyc
🟦 Implemented Components
🟧 Not Implemented Components

Figure 5 XLike Toolkit deployed dependencies

2.1 Topic and Entity tracking (STA) (Identifier UC4)

The main purpose of this use case is to be able to track articles, across media and languages, which are of interest for the press agency (e.g. a topic, an entity, or even a definition). The Figure 6 illustrates the set of functional requirements which are needed following the overall XLike pipeline metaphor. The NewsFeed (RQ1) component retrieves the text from the sources under analysis parsing them and storing them in a PostgreSQL database. The stored data contains the information related to its source, timestamp, publisher, etc. Afterwards the text is analyzed by using the multilingual shallow linguistic processing components (RQ2) and the name entity annotation components (RQ3) to obtain the lemmas and the entities as detailed at Annex C of D6.2.1. "Early Prototype” [5] which are also stored in the database.

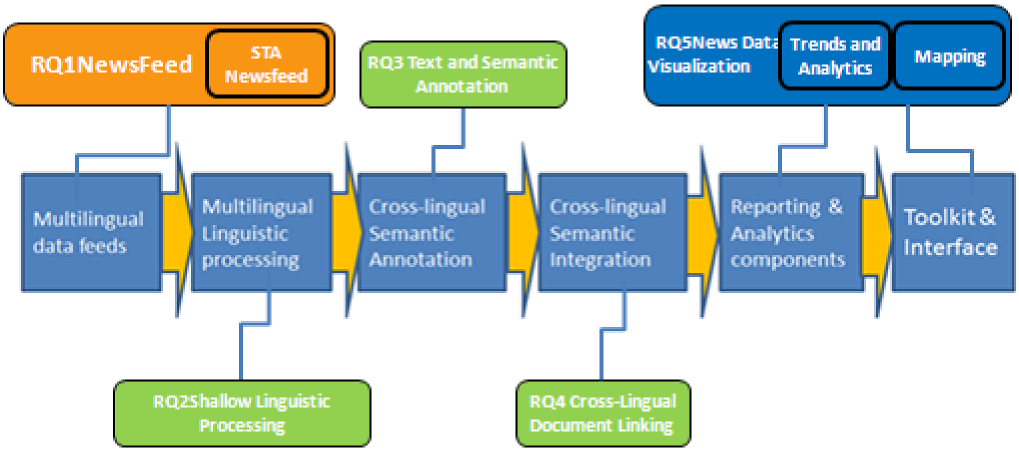
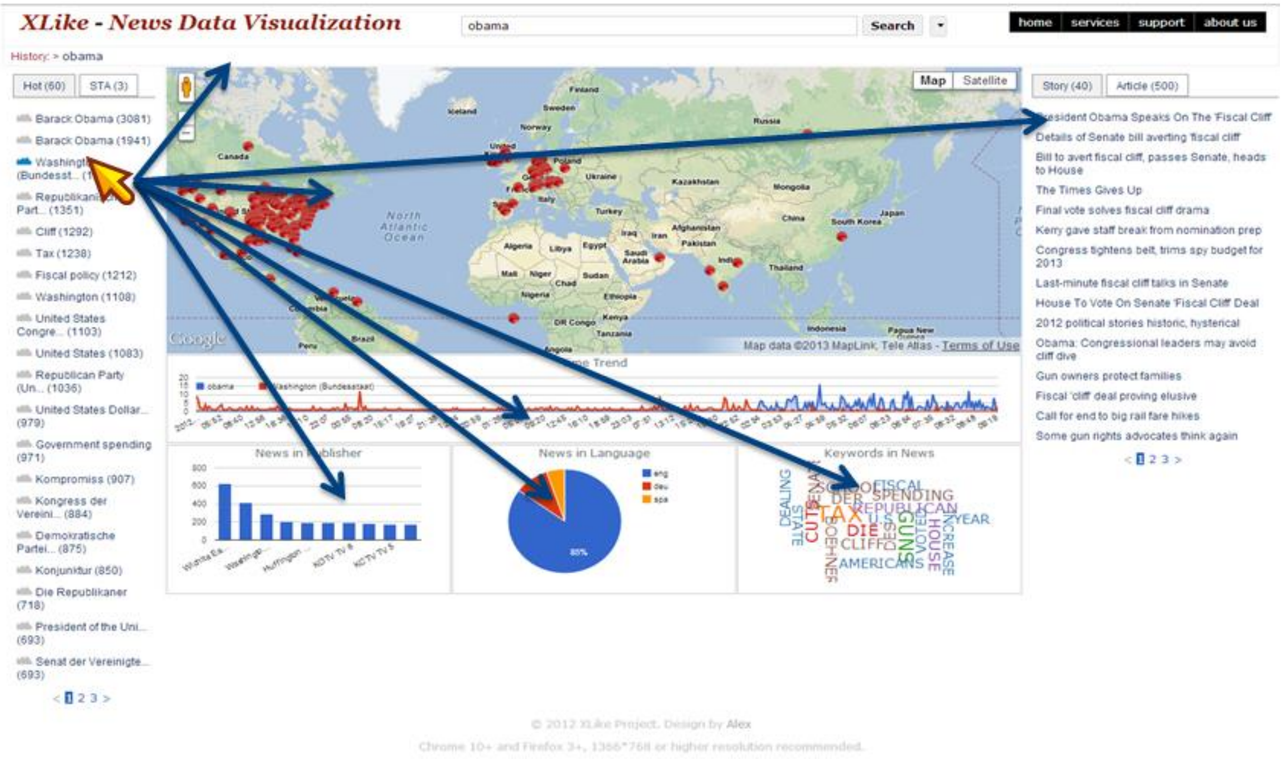

RQ1NewsFeed
STA Newsfeed
RQ3 Text and Semantic Annotation
RQ5News Data Visualization
Trends and Analytics
Mapping
Multilingual data feeds
Multilingual Linguistic processing
Cross-lingual Semantic Annotation
Cross-lingual Semantic Integration
Reporting & Analytics components
Toolkit & Interface
RQ2Shallow Linguistic Processing
RQ4 Cross-Lingual Document Linking

Figure 6 Entity tracking pipeline

The cross-lingual document linking component (RQ3) uses a 24h window of the retrieved articles by Newsfeed to compare among them and extract the similar ones written in any language (en, es, de and zh so far).

The Figure 7 shows how the entity tracked is presented on the left part of the picture and once it is clicked the similar articles are also shown in the right part of the web application. It is also shown how the cross-lingual component works by finding articles at different languages as shown in the pie. For the previously specified STA entities to be tracked there is a tab labelled with STA that allows accessing to them and by clicking allows finding articles related.



Barack Obama (3081)
Barack Obama (1941)
Washingt... (Bundesst... (1
Republikani... Part... (1351)
Cliff (1292)
Tax (1238)
Fiscal policy (1212)
Washington (1108)
United States Congre... (1103)
United States (1083)
Republican Party (Un... (1036)
United States Dollar... (979)
Government spending (971)
Kompromiss (907)
Kongress der Vereini... (884)
Demokratische Partei... (875)
Konjunktur (850)
Die Republikaner (718)
President of the Uni... (693)
Senat der Vereinigte... (693)
< 1 2 3 >

President Obama Speaks On The 'Fiscal Cliff'
Details of Senate bill averting 'fiscal cliff'
Bill to avert fiscal cliff, passes Senate, heads to House
The Times Gives Up
Final vote solves fiscal cliff drama
Kerry gave staff break from nomination prep
Congress tightens belt, trims spy budget for 2013
Last-minute fiscal cliff talks in Senate
House To Vote On Senate 'Fiscal Cliff' Deal
2012 political stories historic, hysterical
Obama: Congressional leaders may avoid cliff dive
Gun owners protect families
Fiscal 'cliff' deal proving elusive
Call for end to big rail fare hikes
Some gun rights advocates think again
< 1 2 3 >

© 2012 XLike Project. Design by Alex
Chrome 10+ and Firefox 3+, 1366*768 or higher resolution recommended.

Figure 7 Entity tracking

2.2 Related/Relevant articles (Bloomberg) (Identifier UC1)

The main purpose of this scenario is to recommend similar articles but not only from the original language but also from other languages. The Figure 8 illustrates the set of components which are used following the overall pipeline metaphor. The NewsFeed (RQ1) component retrieves the articles from the Bloomberg site and stores them in a PostgreSQL database according to the XLike data format after parsing and analyzing it (RQ2). Then, similar articles are searched by using the implemented cross-lingual similarity tool (RQ4) providing a set of relevant articles which are shown in the web site of Bloomberg as can bee seen in the Figure 9.

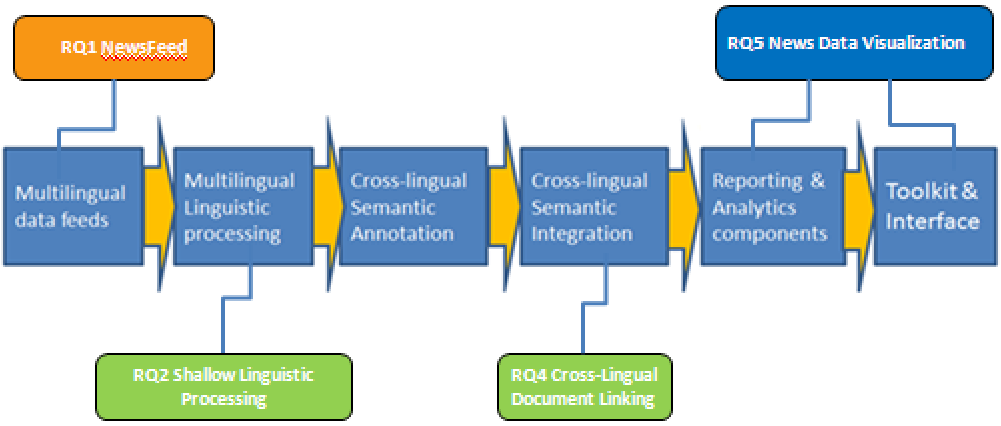
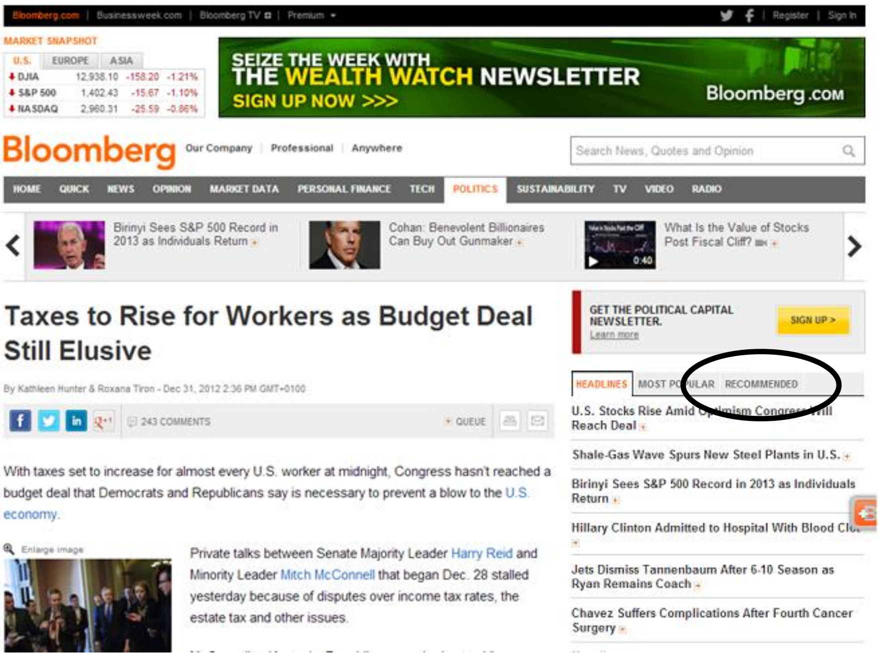

RQ1 NewsFeed
RQ5 News Data Visualization
Multilingual data feeds
Multilingual Linguistic processing
Cross-lingual Semantic Annotation
Cross-lingual Semantic Integration
Reporting & Analytics components
Toolkit & Interface
RQ2 Shallow Linguistic Processing
RQ4 Cross-Lingual Document Linking

Figure 8 Relevant articles pipeline



DJIA	12,938.10	-158.20	-1.21%
S&P 500	1,402.43	-15.67	-1.10%
NASDAQ	2,960.31	-25.59	-0.86%

Birinyi Sees S&P 500 Record in 2013 as Individuals Return

Cohan: Benevolent Billionaires Can Buy Out Gunmaker

What Is the Value of Stocks Post Fiscal Cliff?

Taxes to Rise for Workers as Budget Deal Still Elusive

By Kathleen Hunter & Roxana Tiron - Dec 31, 2012 2:36 PM GMT+0100

With taxes set to increase for almost every U.S. worker at midnight, Congress hasn't reached a budget deal that Democrats and Republicans say is necessary to prevent a blow to the U.S. economy.

Private talks between Senate Majority Leader Harry Reid and Minority Leader Mitch McConnell that began Dec. 28 stalled yesterday because of disputes over income tax rates, the estate tax and other issues.

U.S. Stocks Rise Amid Optimism Congress Will Reach Deal

Shale-Gas Wave Spurs New Steel Plants in U.S.

Birinyi Sees S&P 500 Record in 2013 as Individuals Return

Hillary Clinton Admitted to Hospital With Blood Clot

Jets Dismiss Tannenbaum After 6-10 Season as Ryan Remains Coach

Chavez Suffers Complications After Fourth Cancer Surgery

Figure 9 Recommender article feature on Bloomberg.com

2.3 Content advertising (Bloomberg) (Identifier UC2)

This is a new use case requirement for Y2 and its main goal is to enlarge the Bloomberg audience by advertising their content in other regions and media (e.g. by a Facebook fan page). Although the use case is still ongoing we summarize the pipeline that will be implemented and the needed components. The Newsfeed gather data from the sources of interest (RQ1), then identifying relevant/hot topics (by using the informal language processing (RQ7), shallow language processing (RQ2), and deep language processing (RQ6) from that local source provides the input for finding the Bloomberg similar articles independently of the language of the sources (RQ4).

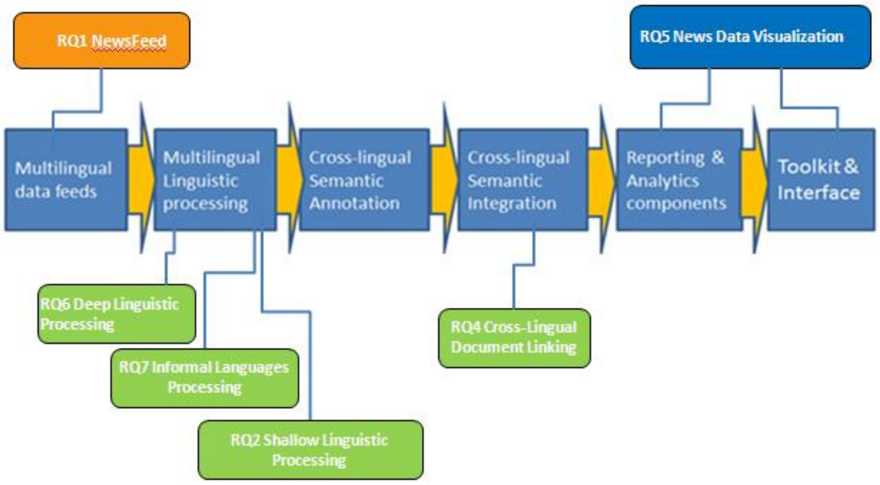

RQ1 NewsFeed
RQ5 News Data Visualization
Multilingual data feeds
Multilingual Linguistic processing
Cross-lingual Semantic Annotation
Cross-lingual Semantic Integration
Reporting & Analytics components
Toolkit & Interface
RQ6 Deep Linguistic Processing
RQ7 Informal Languages Processing
RQ2 Shallow Linguistic Processing
RQ4 Cross-Lingual Document Linking

Figure 10 Content advertising pipeline

2.4 Article tracking (STA) (Identifier UC3)

This scenario has its main goal on detecting republished articles in order to detect and control plagiarism. The Newsfeed gather data from STA and other main streams (RQ1), then the analyzed(RA2) and annotated text (RQ3) is cross-linked using the cross-lingual tool (RQ4) obtaining a set of candidates for possible plagiarism of the STA articles under study. This pipeline is shown in the Figure 11 including the visualization (RQ5).

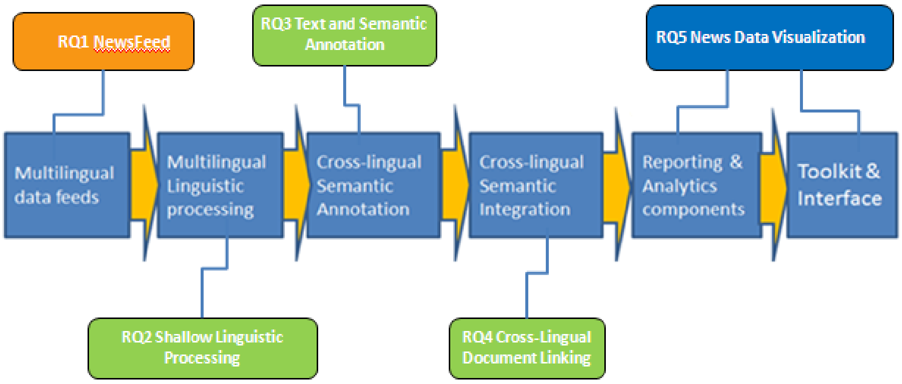

RQ1 NewsFeed
RQ3 Text and Semantic Annotation
RQ5 News Data Visualization
Multilingual data feeds
Multilingual Linguistic processing
Cross-lingual Semantic Annotation
Cross-lingual Semantic Integration
Reporting & Analytics components
Toolkit & Interface
RQ2 Shallow Linguistic Processing
RQ4 Cross-Lingual Document Linking

Figure 11 Article tracking pipeline

2.5 Event identification (STA) (Identifier UC5)

The main purpose of this use case is to extract events from articles. The newsfeed gathers data from the different sources (formal or informal texts) (RQ1), this data is analyzed by the shallow and deep linguistic processing components (RQ2, RQ6, RQ7) and then they are annotated semantically (RQ3) allowing the construction of the semantic graphs (RQ8). For allowing the detection of events at different languages the articles are linked using the cross-lingual document linking (RQ4). Finally the events are extracted using this information (RQ9).

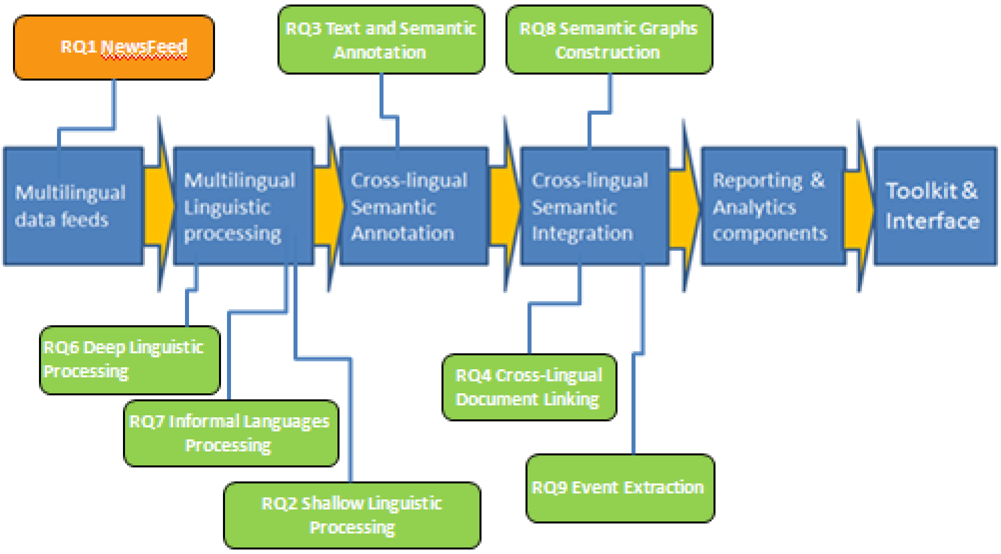
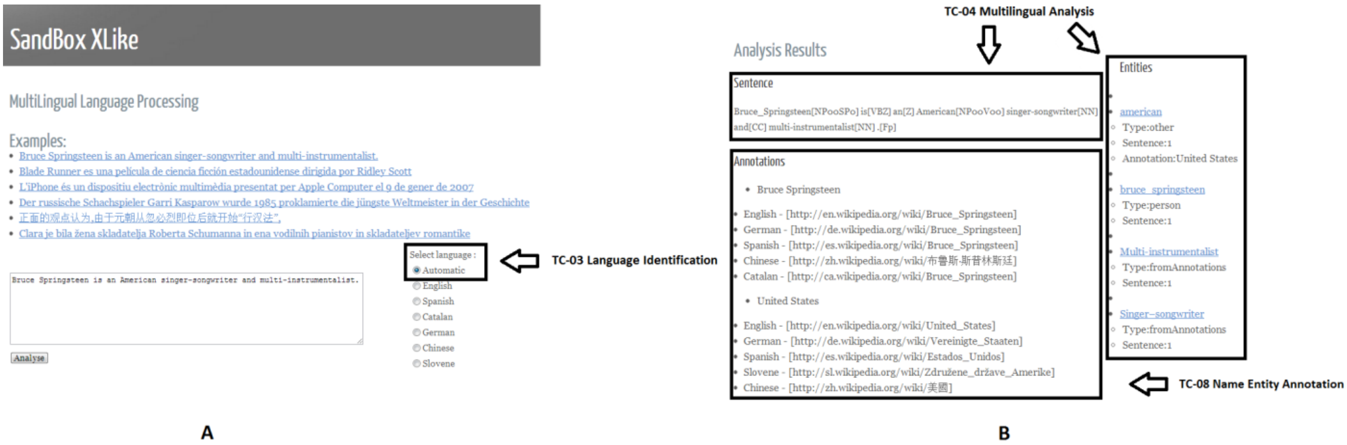

RQ1 NewsFeed
RQ3 Text and Semantic Annotation
RQ8 Semantic Graphs Construction
Multilingual data feeds
Multilingual Linguistic processing
Cross-lingual Semantic Annotation
Cross-lingual Semantic Integration
Reporting & Analytics components
Toolkit & Interface
RQ6 Deep Linguistic Processing
RQ7 Informal Languages Processing
RQ2 Shallow Linguistic Processing
RQ4 Cross-Lingual Document Linking
RQ9 Event Extraction

Figure 12 Event identification pipeline

2.6 Technological Demo

Here we briefly describe the technological demo as a use case which was proposed for testing and dissemination purposes of some implemented components of the project. The Figure 13 shows an example where the picture A of the figure shows the input text holder and some examples including the possibility to select the sentence language or to do it automatically. Once the text is selected then the analysis can be perform by clicking the "Analyze" button which executes (TC-03) (language identification), and performs the multilingual analysis (TC-04). Then the extracted entities obtained by calling the Name Entity Annotation components (TC-06, TC-07) are retrieved and shown in the picture B: it shows the sentence that has been analysed, the entities' area shows the entities extracted from the sentence (TC-07), and finally, the annotations' parts show the results of the name entity annotation service (TC-06).


SandBox XLike
TC-03 Language Identification
A
TC-04 Multilingual Analysis
TC-08 Name Entity Annotation
B

MultiLingual Language Processing

Examples:
- Bruce Springsteen is an American singer-songwriter and multi-instrumentalist.
- Blade Runner es una película de ciencia ficción estadounidense dirigida por Ridley Scott
- L'iPhone és un dispositiu electrònic multimèdia presentat per Apple Computer el 9 de gener de 2007
- Der russische Schachspieler Garri Kasparow wurde 1985 proklamierte die jüngste Weltmeister in der Geschichte
- 正面的观点认为,由于元朝从忽必烈即位后就开始"行汉法",
- Clara je bila žena skladatelja Roberta Schumanna in ena vodilnih pianistov in skladateljev romantike

```
Bruce Springsteen is an American singer-songwriter and multi-instrumentalist.
```

Analyse

Select language :
● Automatic
○ English
○ Spanish
○ Catalan
○ German
○ Chinese
○ Slovene

Analysis Results

Sentence

Bruce_Springsteen[NPooSPo] is[VBZ] an[Z] American[NPooVoo] singer-songwriter[NN] and[CC] multi-instrumentalist[NN] .[Fp]

Annotations

- Bruce Springsteen
- English - [http://en.wikipedia.org/wiki/Bruce_Springsteen]
- German - [http://de.wikipedia.org/wiki/Bruce_Springsteen]
- Spanish - [http://es.wikipedia.org/wiki/Bruce_Springsteen]
- Chinese - [http://zh.wikipedia.org/wiki/布魯斯·斯普林斯廷]
- Catalan - [http://ca.wikipedia.org/wiki/Bruce_Springsteen]
- United States
- English - [http://en.wikipedia.org/wiki/United_States]
- German - [http://de.wikipedia.org/wiki/Vereinigte_Staaten]
- Spanish - [http://es.wikipedia.org/wiki/Estados_Unidos]
- Slovene - [http://sl.wikipedia.org/wiki/Združene_države_Amerike]
- Chinese - [http://zh.wikipedia.org/wiki/美國]

Entities

- american
  - Type:other
  - Sentence:1
  - Annotation:United States
- bruce_springsteen
  - Type:person
  - Sentence:1
- Multi-instrumentalist
  - Type:fromAnnotations
  - Sentence:1
- Singer–songwriter
  - Type:fromAnnotations
  - Sentence:1

Figure 13 Technological Demo functionalities

In the next section, the different components of the toolkit are presented.

3 XLike Toolkit Components

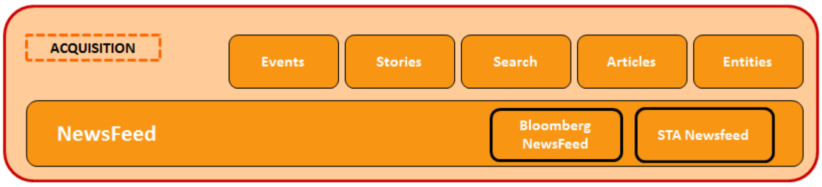
This section describes the set of components which are part of the XLike toolkit. This includes both the already implemented components during the year one and the new ones which have been identified at month 15 as part of the year two ongoing work. These components are distinguished by indicating if they are already implemented or they are still pending. This is also represented in the global components diagram (see Figure 5, brown coloured components are the new ones).

This XLike toolkit consists of 24 components distributed as follows by functionality: 2 in the acquisition layer, 21 in the analysis and interpretation layer, and 1 in the access and human-computer interaction layer. In the following, we include a brief summary of each one of the components including: a brief description, the services URIs, the source code repository, the components dependencies, and some additional information and notes. A more detailed description of the implemented services is done in Section 4.

3.1 Acquisition

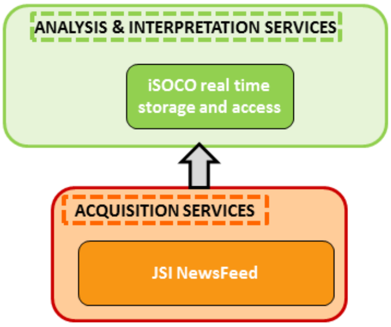
This section describes the components associated to the acquisition layer (see Figure 1).

3.1.1 NewsFeed

Service Name	**NewsFeed**
Identifier	TC-01
Description	Newsfeed is a service for real-time crawling and cleaning of news articles. It focuses on mainstream news sources with public RSS feeds. Each new article is crawled, processed to extract clean text by remove boilerplate, and annotated with source (e.g. type, location) and article (e.g. publish time, images, language) level meta-data. Besides public sources, Newsfeed also aggregates: • XLike specific sources, such as Bloomberg and STA. • Twitter public streaming feed-processed to end users.
URI	The website of the service is available at http://newsfeed.ijs.si/
Source Code Repository	Processing pipeline is available at: https://github.com/JozefStefanInstitute/newsfeed Code of crawler is not public, and is being hosted internally by JSI.
APIs Implemented	There are several APIs, specific to source: • Public: http://newsfeed.ijs.si/stream/ • Bloomberg: http://newsfeed.ijs.si/stream/bloomberg • STA: http://newsfeed.ijs.si/stream/sta • Agency: http://newsfeed.ijs.si/stream/sta_friends • Twitter: http://newsfeed.ijs.si/stream/twitter APIs are password protected using standard HTTP authentication. Output XML format is described on Newsfeed homepage.
Services Used	 ACQUISITION Events Stories Search Articles Entities NewsFeed Bloomberg NewsFeed STA Newsfeed

	Newsfeed uses APIs provided by Bloomberg and STA for accessing private content.
Additional Information	The tutorial is available in the web http://newsfeed.ijs.si/ This component is also deeply described at D1.3.1.
Notes	Webpage: http://newsfeed.ijs.si/

3.1.2 iSOCO real time storage and access

Service Name	**iSOCO real time storage and access**
Identifier	TC-02
Description	This component is a service based on Apache Cassandra which aims to store data from JSI NewsFeed and XLike pipeline functionality services allowing rapid accessibility to the enriched raw data.
URI	To be implemented at http://sandbox-xlike.isoco.com/services/
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	- Trending Entities: to be implemented at http://sandbox-xlike.isoco.com/services/trends returning the top most trending entities in XML or JSON format. - Related Articles: http://sandbox-xlike.isoco.com/services/tweets returning the related articles, in XML or JSON format, of a given entity as parameter.
Services Used	 ANALYSIS & INTERPRETATION SERVICES iSOCO real time storage and access ACQUISITION SERVICES JSI NewsFeed
Additional Information	None.
Notes	None.

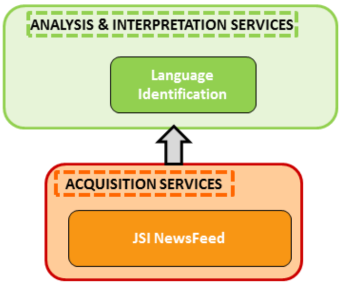
3.2 Analysis and Interpretation

This section describes the components associated to the analysis and interpretation layer (see Figure 1).

3.2.1 Shallow and Deep Linguistic Processing

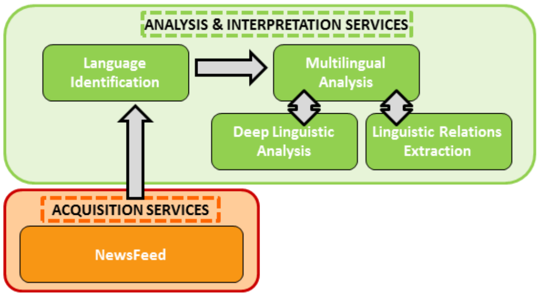
3.2.1.1 Language Identification

Service Name	**Language Identification**
Identifier	TC-03

Description	The language identification service takes as the input parameter the free text in utf8 encoding and outputs the language code following the ISO 639-2 specification. This service is based on the well-known method that uses character-based n-gram language models for each target language.
URI	http://sandbox-xlike.isoco.com/services/
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	The API is implemented at http://sandbox-xlike.isoco.com/services/language_code/ident receiving a XML POST body with the text as input.
Services Used	 ANALYSIS & INTERPRETATION SERVICES Language Identification ACQUISITION SERVICES JSI NewsFeed The language identification service takes as the input parameter the free text in utf8 encoding and outputs the language code.
Additional Information	None.
Notes	None.

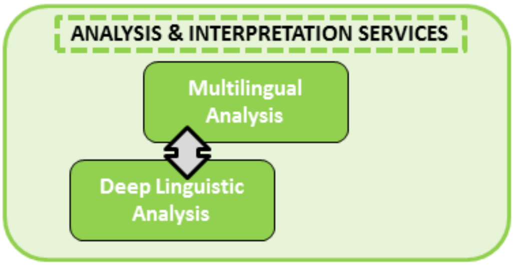
3.2.1.2 Multilingual Analysis

Service Name	**Multilingual Analysis**
Identifier	TC-04
Description	Language analysis services consisting of six analysis pipelines, one for each target language. All pipelines implement the same API, thus providing a uniform service for multi-lingual analysis. Each pipeline implements the following linguistic analysis processes: • sentence splitting and tokenization (shallow) • lemmatisation, part-of-speech tagging and morpho-syntactic annotation (shallow) • named entity recognition and classification (shallow) • syntactic parsing (deep) • semantic role labelling (deep) • extraction of tokens, lemmas, entities and relations The services of this component implement the shallow processes. For the deep processes, they use the UPC Deep Analysis component described below. For extraction of relations, they use the UPC Extraction component, also described below.
URI	• English, Spanish and Catalan: http://sandbox-xlike.isoco.com/services/ • German: http://lt.ffzg.hr:9090/xlike/

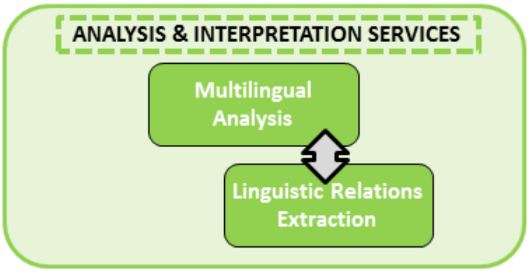
	• Chinese: http://keg.cs.tsinghua.edu.cn:8080/ • Slovenian: http://km.aifb.kit.edu/services
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	The are several APIs one for each language: • Spanish, English, and Catalan: http://sandbox-xlike.isoco.com/services/analysis_xx/analyze (where xx is the language code) • German: http://lt.ffzg.hr:9090/xlike/analysis_de/analyze • Chinese: http://keg.cs.tsinghua.edu.cn:8080/analysis_zh/analyze • Slovenian: http://km.aifb.kit.edu/services/ner-annotation-sl/ The input is the text to be analyzed in XML POST body format and the output is an XML document with the results of the analysis.
Services Used	 ANALYSIS & INTERPRETATION SERVICES Language Identification Multilingual Analysis Deep Linguistic Analysis Linguistic Relations Extraction ACQUISITION SERVICES NewsFeed The Multilingual Analysis service analyzes documents and extracts entities that appear in the documents, together with their relations. Internally, it uses the services provided by the deep processing and the extraction components.
Additional Information	None.
Notes	None.

3.2.1.3 Deep Linguistic Analysis

Service Name	**Deep Linguistic Analysis**
Identifier	TC-05
Description	This component provides specific NLP services related to deep linguistic analysis. Specifically, they provide syntactic parsing and semantic role labeling methods for all languages. These services are used by the Multilingual Analysis component described above.
URI	To be implemented: http://sandbox-xlike.isoco.com/services/
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	Syntactic Parsing: • to be implemented at http://sandbox-xlike.isoco.com/services/ xlike_upc_services /srl receiving the text to analyze and the type of parser as inputs and returning syntactic role labelling in CoNLL format.

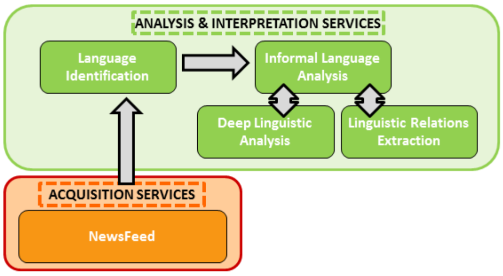
	Semantic Role Labeling: • to be implemented at http://sandbox-xlike.isoco.com/services/ xlike_upc_services/ parse receiving the text to analyze and the type of parser as inputs in XML POST body format and returning the semantic role labelling in CoNLL format. The output is a text file in CoNLL format with syntactic and semantic role labeling structures added.
Services Used	 ANALYSIS & INTERPRETATION SERVICES Multilingual Analysis Deep Linguistic Analysis Deep processing services are called from the multilingual analysis component.
Additional Information	None.
Notes	None.

3.2.1.4 Linguistic Relation Extraction

Service Name	**Linguistic Relation Extraction**
Identifier	TC-06
Description	This component provides specific NLP services related to the extraction of linguistic relations. The extraction methods rely on syntactic parse trees and semantic roles. The extraction methods provided here are used by the Multilingual Analysis component in order to extract relations from linguistic structure.
URI	To be implemented at http://sandbox-xlike.isoco.com/services/
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	Relation Extraction: • to be implemented http://sandbox-xlike.isoco.com/services / xlike_upc_services /extract receiving the text to analyze and the type of the extraction method as inputs in XML POST body format and returning the discovered relations in the text in a XML document.
Services Used	 ANALYSIS & INTERPRETATION SERVICES Multilingual Analysis Linguistic Relations Extraction The linguistic relations extraction services are called from the multilingual analysis component.
Additional	None.

Information	
Notes	None.

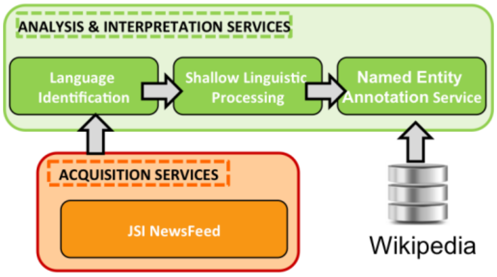
3.2.1.5 Informal Languages Analysis

Service Name	**Informal Language Analysis**
Identifier	TC-07
Description	This component analyzes documents written in informal language (for example, documents crawled from social media), and extracts entities and relations from them. This component is a clone of the multilingual analysis component, the main difference being that the methods and models are adapted to improve robustness of linguistic analysis and extraction of linguistic structure. As in the standard counterpart, the component subdivides into a pipeline for each language.
URI	To be implemented at http://sandbox-xlike.isoco.com/services/
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	Informal analysis: • to be implemented at http://sandbox-xlike.isoco.com/services /informal_xx/analyze (where xx is the language code) receiving the text to be analyzed as input in XML POST body format and returning a XML with the results of the analysis.
Services Used	 ANALYSIS & INTERPRETATION SERVICES Language Identification Informal Language Analysis Deep Linguistic Analysis Linguistic Relations Extraction ACQUISITION SERVICES NewsFeed
Additional Information	None.
Notes	This component will be eventually integrated with the multilingual analysis component. The goal is to build a single component for multilingual analysis that can be used to analyze language in a uniform fashion, irrespective of the language of the document and its level of formality.

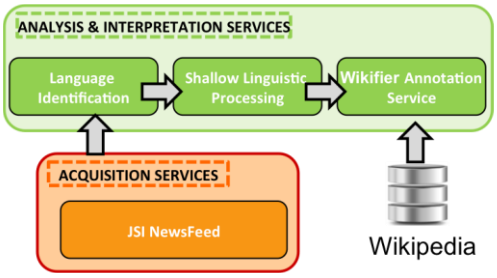
3.2.2 Text and Semantic Annotation

3.2.2.1 Name Entity Annotation

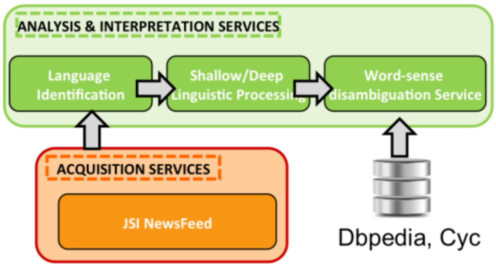
Service Name	**Name Entity Annotation**
Identifier	TC-08
Description	This component discovers the Wikipedia annotations associated to a document by matching the names of the detected entities against the Wikipedia titles.
URI	http://km.aifb.kit.edu/services/ (where xx is the language)

Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	- Text/Semantic annotation: http://km.aifb.kit.edu/services/ner-annotation-xx/ receiving as input the output of the multilingual analysis (TC-04) and returning the set of annotations which match the detected entities in XML format.
Services Used	 ANALYSIS & INTERPRETATION SERVICES Language Identification Shallow Linguistic Processing Named Entity Annotation Service ACQUISITION SERVICES JSI NewsFeed Wikipedia The name entity annotation service annotates the entities provided by the multilingual processing analysis with the Wikipedia articles of the same title for the different languages.
Additional Information	None.
Notes	None.

3.2.2.2 Wikipedia Miner Wikifier Annotation

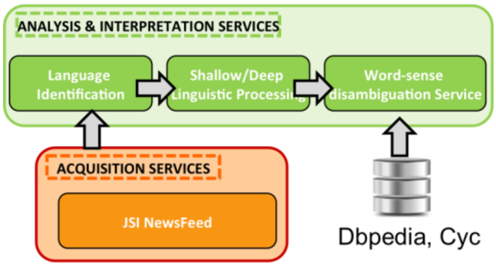
Service Name	**Wikipedia Miner Wikifier Annotation**
Identifier	TC-09
Description	This component adds the Wikipedia annotations to a document using the Wikipedia Miner wikifer based approach.
URI	http://km.aifb.kit.edu/services/
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	- Wikifier annotator: http://km.aifb.kit.edu/services/annotation-xx/ (where xx is the language code) receiving as input the output of the multilingual analysis (TC-04) and returning the set of annotations obtained by the Wikifier.
Services Used	 ANALYSIS & INTERPRETATION SERVICES Language Identification Shallow Linguistic Processing Wikifier Annotation Service ACQUISITION SERVICES JSI NewsFeed Wikipedia The Wikipedia Miner Wikifier annotation service annotates the entities provided by the multilingual processing analysis with the Wikipedia articles of the same title for the different languages.
Additional Information	None.
Notes	None.

3.2.2.3 Early Ontological Word-sense-disambiguation

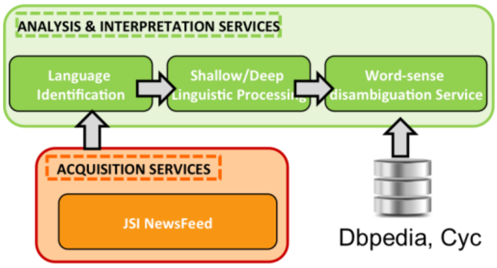
Service Name	**Early Ontological Word-sense-disambiguation**
Identifier	TC-10
Description	This component takes the output of multi-linguistic processing in WP2 as input and adds the annotations with knowledge resources, such as DBpedia, Cyc etc. by matching the names of the detected entities against the labels of the knowledge resources.
URI	http://km.aifb.kit.edu/services/
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	• Word-sense-disambiguation: http://km.aifb.kit.edu/services/ner-disambiguation-xx/ (where xx is the language code) receiving as input the output of the multilingual analysis (TC-04) and returning the set of annotations obtained by the knowledge bases.
Services Used	 ANALYSIS & INTERPRETATION SERVICES Language Identification Shallow/Deep Linguistic Processi Word-sense disambiguation Service ACQUISITION SERVICES JSI NewsFeed Dbpedia, Cyc The word-sense disambiguation service annotates the entities provided by the multilingual processing analysis with the knowledge resources for the different languages.
Additional Information	None.
Notes	None.

3.2.2.4 Crowd Sourcing Word-sense-disambiguation

Service Name	**Crowd Sourcing Word-sense-disambiguation**
Identifier	TC-11
Description	This component provides bootstrapping additional annotations and ontological structure given the training documents and the knowledge base. Based on the results, the previous word-sense disambiguation service will be improved.
URI	To be implemented at http://km.aifb.kit.edu/services/
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	None.

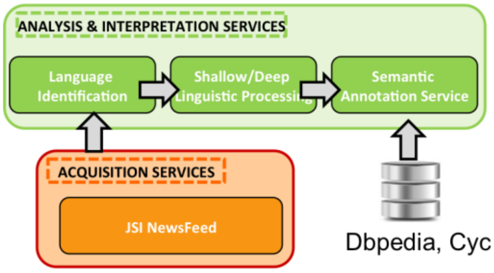
Services Used	 ANALYSIS & INTERPRETATION SERVICES Language Identification Shallow/Deep Linguistic Processing Word-sense disambiguation Service ACQUISITION SERVICES JSI NewsFeed Dbpedia, Cyc The word-sense disambiguation service annotates the entity mentions with the knowledge resources for the different languages.
Additional Information	None.
Notes	None.

3.2.2.5 Final Ontological Word-sense-disambiguation

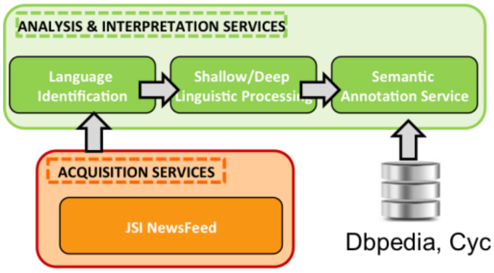
Service Name	**Final Ontological Word-sense-disambiguation**
Identifier	TC-12
Description	This components provides the final version of ontology based word-sense disambiguation supporting all ontological knowledge resources handled by the final disambiguation tools.
URI	To be implemented at http://km.aifb.kit.edu/services/
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	None
Services Used	 ANALYSIS & INTERPRETATION SERVICES Language Identification Shallow/Deep Linguistic Processing Word-sense disambiguation Service ACQUISITION SERVICES JSI NewsFeed Dbpedia, Cyc The word-sense disambiguation service annotates the entity mentions with the knowledge resources for the different languages.
Additional Information	None.
Notes	None.

3.2.2.6 Final Text Annotation Service

Service Name	**Final Text Annotation Service**
Identifier	TC-13
Description	This component provides the final version of text annotation tool, which annotates documents with all

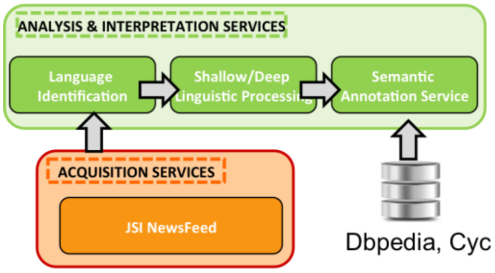
	knowledge resources handled by the final annotation tools, including entities, relations and triples.
URI	To be implemented at http://km.aifb.kit.edu/services/
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	• Final text annotation: http://km.aifb.kit.edu/services/ner-annotation-xx/ (where xx is the language code) receiving as input the output of the multilingual analysis (TC-04) and returning the set of annotations which match the detected entities in XML format.
Services Used	 ANALYSIS & INTERPRETATION SERVICES Language Identification Shallow/Deep Linguistic Processing Semantic Annotation Service ACQUISITION SERVICES JSI NewsFeed Dbpedia, Cyc The semantic annotation service annotates the entity mentions and relation patterns with the knowledge resources for the different languages.
Additional Information	None.
Notes	None.

3.2.2.7 Early Machine Translation based Semantic Annotation

Service Name	**Early Machine Translation based Semantic Annotation**
Identifier	TC-14
Description	This component provides first version of semantic annotation prototype based on machine translation trained on the initial data.
URI	To be implemented.
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	None.
Services Used	 ANALYSIS & INTERPRETATION SERVICES Language Identification Shallow/Deep Linguistic Processing Semantic Annotation Service ACQUISITION SERVICES JSI NewsFeed Dbpedia, Cyc The semantic annotation service annotates the entity mentions and relation patterns with the knowledge resources for the different languages.
Additional Information	None.

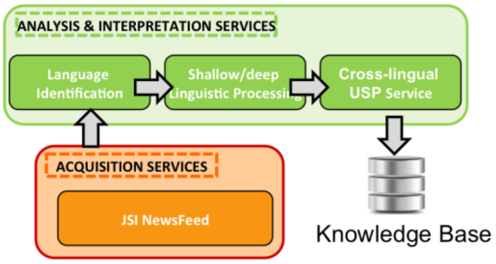
Notes	None.

3.2.2.8 Final Machine Translation based Semantic Annotation

Service Name	**Final Machine Translation based Semantic Annotation**
Identifier	TC-15
Description	This component provides the final version of semantic annotation prototype based trained on the extended dataset.
URI	To be implemented.
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	None
Services Used	 ANALYSIS & INTERPRETATION SERVICES Language Identification Shallow/Deep Linguistic Processing Semantic Annotation Service ACQUISITION SERVICES JSI NewsFeed Dbpedia, Cyc The semantic annotation service annotates the entity mentions and relation patterns with the knowledge resources for the different languages.
Additional Information	None.
Notes	None.

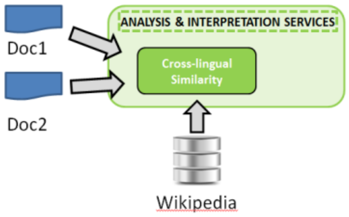
3.2.2.9 Cross-lingual USP

Service Name	**Cross-lingual USP**
Identifier	TC-16
Description	This component provides the extension of USP techniques in a cross-lingual setting, which tries to build clusters of syntactic variations across languages but of the same meaning.
URI	To be implemented.
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	None.

Services Used	 ANALYSIS & INTERPRETATION SERVICES Language Identification Shallow/deep Linguistic Processing Cross-lingual USP Service ACQUISITION SERVICES JSI NewsFeed Knowledge Base It does have dependencies of specific knowledge based and of previous multilingual analysis results (shallow or deep).
Additional Information	None.
Notes	None.

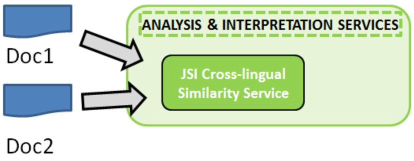
3.2.3 Cross-lingual Document Linking

3.2.3.1 KIT Cross-lingual Similarity

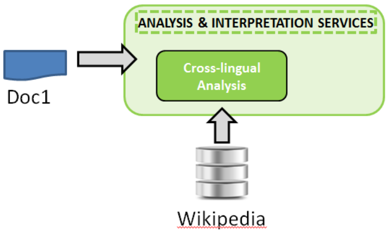
Service Name	**Cross-lingual Similarity**
Identifier	TC-17
Description	Cross-lingual Similarity component determines the cross-lingual similarity between two Documents. This web service is based on Cross-lingual extension of Explicit Semantic Analysis (ESA) and uses Wikipedia dumps as knowledge source.
URI	http://km.aifb.kit.edu/services/
Source Code Repository	Not available.
APIs Implemented	• Cross-lingual Similarity: http://km.aifb.kit.edu/services/clesa/similarity receiving as input two text documents (doc1, doc2) and their languages (lang1, lang2) as Rest parameters and returns their similarity score.
Services Used	 Doc1 Doc2 ANALYSIS & INTERPRETATION SERVICES Cross-lingual Similarity Wikipedia This web service is based on Cross-lingual extension of Explicit Semantic Analysis (ESA) and uses Wikipedia dumps from Mai 2012 as knowledge source.
Additional Information	None.
Notes	None.

3.2.3.2 JSI Cross-lingual Similarity

Service Name	**Cross-lingual Similarity**

Identifier	TC-18
Description	The JSI Cross-lingual Similarity component consists of two main services. First one (CLSI) is used to compute similarity between two documents doc1 and doc2 in two languages, lang1 and lang2; and the second one (REVEAL) enables the insight in how the similarity is computed returning the words in language lang1 and lang2 that add the most to the similarity.
URI	http://xling.ijs.si:1111/
Source Code Repository	Not available
APIs Implemented	• JSI Cross-lingual Similarity : http://xling.ijs.si:1111/clsi receiving two text documents (doc1,doc2) and their languages (lang1, lang2) as input parameters of the Rest service and returning the similarity score between them as output. • JSI Cross-lingual reveal: http://xling.ijs.si:1111/reveal receiving two text documents (doc1, doc2) and their languages (lang1, lang2) as input parameters of the Rest service and returning a set of words as outputs.
Services Used	 Doc1 Doc2 ANALYSIS & INTERPRETATION SERVICES JSI Cross-lingual Similarity Service This component does not have dependencies on other components of the toolkit.
Additional Information	None.
Notes	None.

3.2.3.3 Cross-lingual Analysis

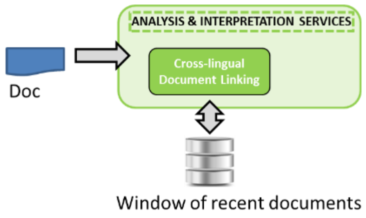
Service Name	**Cross-lingual Analysis**
Identifier	TC-19
Description	This component retrieves related Wikipedia articles in a specified language given an input document.
URI	http://km.aifb.kit.edu/services/
Source Code Repository	Not available
APIs Implemented	• Cross-lingual Analysis: implemented at http://km.aifb.kit.edu/services/clesa/analyzer receiving a text document (doc1), its language(lang1)and another specified language (lang2) as Rest parameters and returning a set of related Wikipedia articles.
Services Used	 ANALYSIS & INTERPRETATION SERVICES Cross-lingual Analysis Doc1 Wikipedia

	It uses Wikipedia dumps from Mai 2012 and supports English, German, Spanish, French, Catalan and Slovenian.
Additional Information	None
Notes	None

3.2.3.4 Cross-lingual Document Linking

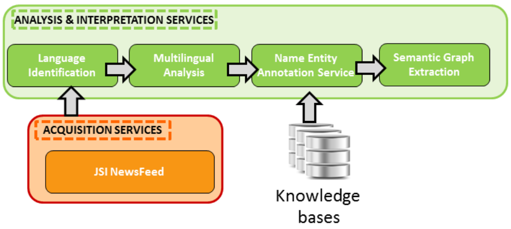
Service Name	Cross-lingual Document Linking
Identifier	TC-20
Description	Cross-lingual document linking component links articles across languages based on content similarity. Service keeps a window of recent articles, against which it matches new articles, and returns list of most similar ones. The window of recent articles is updated with new articles, as they are sent to the service.
URI	http://xling.ijs.si:10001/
Source Code Repository	To be published at https://github.com/xlike-project/

APIs Implemented	**Similar articles:** implemented at http://xling.ijs.si:10001/bloombergostxml receiving as input newsfeed XML format and returning as output the set of similar articles in JSON format. Output:
	{
	    "id":66701562,
	    "similar_articles_spa": […],
	    "similar_articles_deu": […],
	    "similar_articles_fra": […],
	    "similar_articles_ita": […]
	  }
	Bloombergness: implemented at http://xling.ijs.si:10001/bloombergostxml receiving as input newsfeed XML format and returning as output a JSON object providing list of similar Bloomberg articles and article bloombergness score. Output:
	  {
	    "id":51522271,
	    "similar_articles_bloomberg": [{"id":66701562, "sim":0.1364}],
	    "bloombergness":0.0230
	  }

Services Used	 ANALYSIS & INTERPRETATION SERVICES Cross-lingual Document Linking Doc Window of recent documents
Additional Information	This service is internal to JSI network, as it is being called only directly from Newsfeed.
Notes	None.

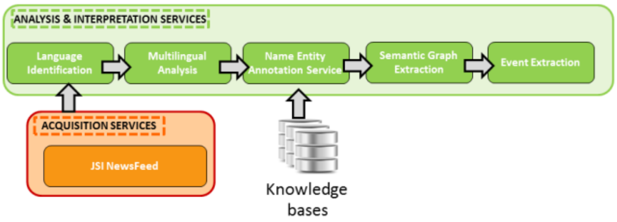
3.2.4 Semantic Graphs Constructions and Event Extraction

3.2.4.1 Semantic Graph Extraction

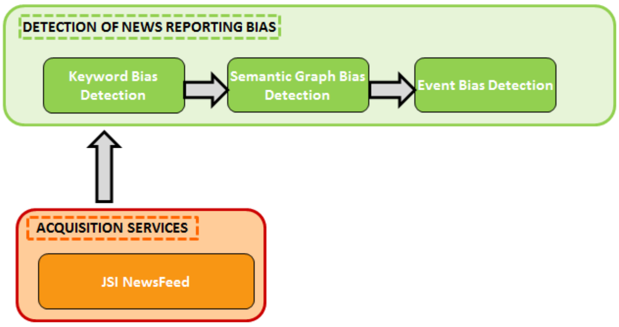
Service Name	**Semantic Graph Extraction**
Identifier	TC-21
Description	This component takes as input a text document, processed by services from WP2 and WP3, and returns a set of assertions (e.g. triples) which are identified in the document.
URI	To be implemented.
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	Service will provide three APIs, corresponding to three different approaches tackled in T4.2. All three APIs will get same input, but will differ in the output format, depending on the formalism most suitable to encode the results (will be identified as part of T4.2).
Services Used	 ANALYSIS & INTERPRETATION SERVICES Language Identification Multilingual Analysis Name Entity Annotation Service Semantic Graph Extraction ACQUISITION SERVICES JSI NewsFeed Knowledge bases
Additional Information	None.
Notes	None.

3.2.4.2 Event Extraction

Service Name	**Event Extraction**
Identifier	TC-22
Description	This service takes as input a set of text documents, already processed by Semantic Graph Extraction service, and uses them to fill up one of the predefined event extraction templates. The templates are defined in an interactive offline processed by a separate user-facing tool.
URI	To be implemented.
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	Service will provide an API, which will on the input receive a set of processed documents, and returns event description in a form of a formalism best suited for the task.

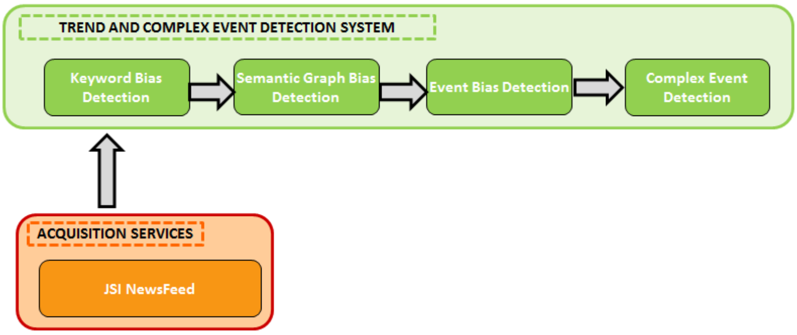
Services Used	 ANALYSIS & INTERPRETATION SERVICES Language Identification Multilingual Analysis Name Entity Annotation Service Semantic Graph Extraction Event Extraction ACQUISITION SERVICES JSI NewsFeed Knowledge bases This component depends on the knowledge bases and the previous components of the pipeline for the linguistic analysis, the recognition of entities, and the construction of the semantic graphs.
Additional Information	None.
Notes	None.

3.2.4.3 Detection of news reporting bias

Service Name	Detection of news reporting bias
Identifier	TC-23
Description	Detection of news reporting bias is a service whose main target is to detect news with differences in reporting about the same events across sources, languages and time. To identify significant differences on the cross-lingual keyword based, semantic graph and event description levels, statistical and machine learning methods will be used. Reporting tools capable of efficient summarization of detected bias will be developed, so can be extended to many sources, languages and longer periods of time. This component should be composed by the next modules: • Keyword bias detection. • Semantic graph bias detection. • Event bias detection.
URI	To be implemented.
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	Not available.
Services Used	 DETECTION OF NEWS REPORTING BIAS Keyword Bias Detection Semantic Graph Bias Detection Event Bias Detection ACQUISITION SERVICES JSI NewsFeed The bias detection service uses the keyword extraction service, semantic graph building service and event description services and then detects the biases based on the differences between news on these three facets.
Additional	None.

Information	
Notes	None.

3.2.4.4 Trend and complex event detection

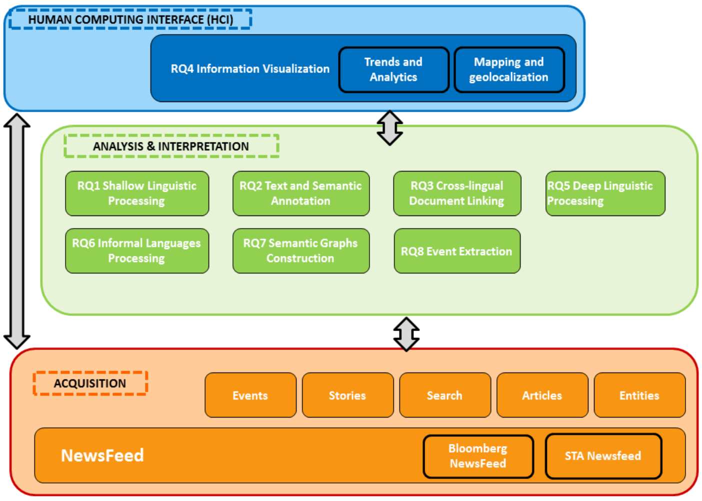
Service Name	**Trend and complex event detection system**
Identifier	TC-24
Description	The main objective of this component is to detect trends and identify complex events from event stream provided by event extraction from semantic graphs task. Trends are defined as significant distribution changes on the cross-lingual keyword based, semantic graph and event description level over short, medium and long period of time and techniques will be developed to deal with these diverse representations. Complex events are defined as set of atomic events, which occur over some periods of time and can only together compose one larger event. This task is of particular interest for the financial case study where one needs to detect trends and complex events as early and as fast as possible. The different between detection of news reporting bias and trend and complex event detection system is that trend and complex event detection system aims to find the trends over all news and detection of news reporting bias aims to detect the bias over news reported by different publishers. This component is composed by: • Keyword trends detection. • Semantic graph trends detection. • Event trends detection. • Complex event detection.
URI	To be implemented.
Source Code Repository	To be published at https://github.com/xlike-project/
APIs Implemented	Not available.
Services Used	 TREND AND COMPLEX EVENT DETECTION SYSTEM Keyword Bias Detection Semantic Graph Bias Detection Event Bias Detection Complex Event Detection ACQUISITION SERVICES JSI NewsFeed The trend and complex event detection system uses the keyword extraction service, semantic graph building service, event description service, complex event detection and then detects the trends and complex events.
Additional Information	None.
Notes	None.

3.3 Human Computer Interaction (HCI)

This section describes the components associated to the human-computer interaction layer (see Figure 1).

3.3.1 News Data Visualization

Service Name	News Data Visualization
Identifier	TC-25
Description	This component uses some of the existing text visualization and network visualization techniques to show real-time information spreading across the globe for the events and news. This service contains two modules: the backend data services and the frontend visualization component. Each of them contains several services Backend data services ○ Stories API ○ Entities API ○ Story API ○ Entity API ○ Article API ○ Search API Frontend Component ○ Geographical distribution visualization ○ Language, publisher, time distribution visualization ○ Tracking by entity, article, story, keyword
URI	http://sandbox-xlike.isoco.com http://newsfeed.ijs.si/xlike/
Source Code Repository	https://github.com/xlike-project/wp5
APIs Implemented	• entities AP: http://newsfeed.ijs.si/xlike/entities • stories API: http://newsfeed.ijs.si/xlike/stories • entity API: http://newsfeed.ijs.si/xlike/entity • story API: http://newsfeed.ijs.si/xlike/story • article API: http://newsfeed.ijs.si/xlike/article • search API: http://newsfeed.ijs.si/xlike/search • HCI Portal: http://sandbox-xlike.isoco.com/portal/

Services Used	 HUMAN COMPUTING INTERFACE (HCI) RQ4 Information Visualization Trends and Analytics Mapping and geolocalization ANALYSIS & INTERPRETATION RQ1 Shallow Linguistic Processing RQ2 Text and Semantic Annotation RQ3 Cross-lingual Document Linking RQ5 Deep Linguistic Processing RQ6 Informal Languages Processing RQ7 Semantic Graphs Construction RQ8 Event Extraction ACQUISITION Events Stories Search Articles Entities NewsFeed Bloomberg NewsFeed STA Newsfeed This component will use the language processing service provided by WP2, cross-lingual annotation services and cross-lingual similarity calculating services provided by WP4.
Additional Information	None.
Notes	None.

4 APIs Overview

This section defines the APIs provided by the different components, at M15 of the project, as a set of services which interoperate each other and also with the final end-users. These APIs are organized in three layers as described in the introduction of this report grouping them by functionality: acquisition, analysis and interpretation, and human-computer interaction services.

4.1 Acquisition

4.1.1 NewsFeed

API overview:

The NewsFeed is a stream of news that is used by the architecture to obtain articles and news from the tracked sources. The API is implemented as web services using JSON format .

API operations:

- **stories**

  Service Address: http://newsfeed.ijs.si/xlike/stories

  Inputs: this service doesn't need any input.

  Outputs: returns the most popular stories (maximum40). A story corresponds to a collection of articles which talk about the same topic or event

- **entities**

  Service Address: http://newsfeed.ijs.si/xlike/entities

  Inputs: this service doesn't need any input

  Outputs: return the most popular entities (maximum60). An entity is a special word or phrase representing a person, a location, an organization, or some abstract concept

- **entity**

  Service Address: http://newsfeed.ijs.si/xlike/entity

  Inputs: uri=entityUri.

  Outputs: returns an entity, a list of articles, a list of stories, a list of entities, a list of STA entities, a list of keywords and a list of timestamps

- **article**

  Service Address: http://newsfeed.ijs.si/xlike/article

  Inputs: id=articleID.

  Outputs: returns an article, a list of articles, a list of stories, a list of entities, a list of STA entities, a list of keywords and a list of timestamps

- **story**

  Service Address: http://newsfeed.ijs.si/xlike/story

  Inputs: id=storyID.

  Outputs: returns a story, a list of articles, a list of entities, a list of STA entities, a list of keywords and a list of timestamps

- **search**

  Service Address: http://newsfeed.ijs.si/xlike/search

Inputs: q=keyword.

Outputs: returns a list of articles, a list of stories, a list of entities, a list of STA entities, a list of keywords and a list of timestamps

In the next table the data format of the used structures is shown.

Table 1: Data Format

Data	Practical Format	Related APIs
article	<id, url, title, body, publisher, country, city, location, language,time>	article
article list	<count, {<id, url, title, publisher, country, city, location, language,time >}>	article, story, entity, search
entity	<uri, type,{<language, label>},{<key, value>}>	entity
entity list	{<uri,{language, label}, frequency>}	article, story, entity, search, entities
Story	<id, label, abstract, count(the number of articles in this story)>	Story
story list	{<id, label>}	article, entity, search, stories
keyword list	{<keyword, count>}	article, entity, search, stories
timestamp list	{<time, count>}	article, entity, search, stories
STA entity list	{<uri, label, frequency>}	article, entity, search, stories

API usage:

All these APIs are called by visualization component via the its URL address baseURL/APIname?[key parameters][supporting parameters]. An example of a call is the following:

http://newsfeed.ijs.si/xlike/search?q=obama&page=0&pagesize=100&ts=1d&lang=eng&lang=chi&group=general&country=Slovenia

This service supports a list of miscellaneous parameters. These supporting parameters are not mandatory but can be used to filter the set of desired articles/news:

- **pagesize:** how many articles will be returned.
- **page:** the page offset of returned articles.
- **lang:** specify the language of articles; default value is empty which means all languages are permit.
- **group:** specify the groups of STA entities, works only in private API. Default value is empty which means all groups are permit. In public API, the value of this parameter is fixed as “general”.
- **ts:** the time span of the returned articles. “h” for hour, “d” for day and “w” for week.
- **country:** specify the country or region of articles, default value is empty which means all countries are permit.

4.1.2 iSOCO real time storage and access

API overview:

The iSOCO real time storage and access is a set of services based on Cassandra which aims to store data from NewsFeed and language analysis services for providing fast real time access (read + write) to data.

API operations:

It has been developed a service to retrieve entities with the highest number of occurrences and its related articles.

- **Trending Entities**

  Service Address: to be published. Its API RESTful format is base_url/ trends.

  Inputs: this service doesn't need any input.

  Outputs: returns a list of sorted entities ordered by its frequency.

- **Related Articles**

  Service Address: to be published. Its API RESTful format is base_url/tweets?entity=parameter

  Inputs: the parameter is a text with the selected entity

  Outputs: returns a list of articles associated to an entity

API usage:

This service is used for obtaining the trending entities of the newsfeed data and its related articles. An example of call is:

- base_url/ trends
- base_url/tweets?entity=Ljubljana

4.2 Analysis and Interpretation

4.2.1 Language Identification

API overview:

The language identification service classifies a text in one of the XLike languages. The service takes as the input parameter the free text to be analyzed in utf8 encoding format and outputs the language code following the ISO 639-2 specification (en, es, ca, de, zh, sl).

API operations:

- **Language Identification Service**

  Service Address: http://sandbox-xlike.isoco.com/services/language_code/ident

  Inputs: it receives an XML POST body <analyze> <text>Article</text> </analyze> being the parameter Article a free text in utf8 encoded format.

  Outputs: returns the language code following the ISO 639-2 specification (en, es, ca, de, zh, sl)

API usage:

This service is used to identify the language of a specified text. The output of this service is used to choose the correct multilingual analysis pipeline to be executed. An example of the body XML POST call is the following:

<analyze><text>Bruce Springsteen is an American singer-songwriter and multi instrumentalist.</text></analyze>

4.2.2 Multilingual Analysis

API overview:

Multilingual analysis consists of six individual pipelines (one for each XLike language) which contain the functionality of: sentence splitting, tokenisation, lemmatisation, POS- or MSD-tagging, named entity recognition, and relations.

API operations:

- **Multilingual Analysis:**

  Service Address:

  English, Spanish, Catalan: http://sandbox-xlike.isoco.com/services/analysis_xx/analyze (being xx the language code following the ISO 639-2 specification (en, es, ca))

  German: http://lt.ffzg.hr:9090/xlike/analysis_de/analyze

  Chinese: http://keg.cs.tsinghua.edu.cn:8080/analysis_zh/analyze

  Slovenian: http://enrycher.ijs.si/xlike/analysis_sl/analyze

  Inputs: it receives an XML POST body

  <analyze><text>Article</text><target>relations</target><conll>true</conll></analyze>,

  being the parameter article a free text utf8 encoded, and conll whether to return the analysis in CoNLL format or not.

  Outputs: the output is a XML file containing the information about sentences, text, tokens, entities, and mentions (see Annex C at [5] to see an example of this data format).

  Sentence: each one of the sentences of the text

  Text: the original source text

  Tokens: the token as it occurs in the input text

  Pos: the POS/MSD tag (disambiguated)

  Lemma: the lemma of the word (disambiguated)

  id: the position of the token in the sentence (starting at 1), which serves as an identifier for the token

  start: the start character of the token in the input text

  end: the ending character of the token in the input text

  Entity: the recognized entity in the text

  Type: the categorization of the entity

  DisplayName: the title of the entity

  Mention: where the entity is mentioned in the text

  sentenceId: id referencing to the sentence where is mentioned

  id: identifier of the token mentioned

  words: the word that it refers to

API usage:

These services are called to obtain the shallow linguistic processing of a text and are used for the different uses cases identified. An example of a the body POST call is the following:

```
<analyze><text>Bruce Springsteen is an American singer-songwriter and multi-instrumentalist.</text><target>relations</target><conll>true</conll></analyze>
```

4.2.3 Deep Linguistic Analysis

API overview:

This service provides syntactic parsing and semantic role labeling methods for all languages. These services are used by the Multilingual Analysis component (TC-04).

API operations:

- **Parse:**

  Service address: http://sandbox-xlike.isoco.com/services/xlike_upc_services/parse

  Inputs: it receives an XML POST request whit the following parameters

  Lang: the free text language.

  Text: the free text in utf8 encoding.

  Parser: type of parser.

  Outputs: the output is a text file in CoNLL format with syntactic and semantic role labeling structures added.

- **Semantic Role Labeling:**

  Service address: http://sandbox-xlike.isoco.com/services/xlike_upc_services/srl

  Inputs: it receives an XML POST request whit the following parameters:

  Lang: the free text language.

  Text: the free text in utf8 encoding.

  Parser: type of parser (first-order or second-order).

  Outputs: the output is a text file in CoNLL format with syntactic and semantic role labeling structures added.

API usage:

An example of parsing and semantic role labelling body POST calls are the following:

```
<parse><lang>en</lang><text> Bruce Springsteen is an American singer-songwriter and multi-instrumentalist.</text><parser>first-order</parser></parse>
```

```
<srl><lang>en</lang><text> Bruce Springsteen is an American singer-songwriter and multi-instrumentalist.</text><parser>second-order</parser></srl>
```

4.2.4 Linguistic Relation Extraction

API overview:

The extraction service relies on syntactic parse trees and the semantic identified roles for extracting the relations between different parts of a sentence.

API operations:

- **Extract:**

Service address: *http*://sandbox-xlike.isoco.com/services/xlike_upc_services/extract

Inputs: it receives an XML POST request whit the following parameters:

Lang: the free text language.

Text: the free text in utf8 encoding.

Type: type of extraction method (syntactic or semantic)

Outputs: the output is an XML document listing the relations present in the document.

API usage:

These services are called by the Multilingual Analysis component. An example of the body XML POST call is the following:

```
<extract><lang>en</lang><text> Bruce Springsteen is an American singer-songwriter and
multi-instrumentalist.</text><type>syntactic</type></extract>
```

4.2.5 Name Entity Annotation

API overview:

This service annotates the sentence with the identified entities that contains. It takes as input the output of multi-linguistic processing and adds the Wikipedia annotations by matching the names of the detected entities against the Wikipedia titles.

API operations:

- **Name Entity Annotation:**

  Service address: http://km.aifb.kit.edu/services/ner-annotation-xx/

  (being xx the language code following the ISO 639-2 specification (en, es, ca, de, zh, sl))

  Inputs: it receives an XML POST request that is the output of multilingual analysis.

  Output: the output of name entity annotation service is a copy of the XML created in multilingual analysis (TC-04) adding annotations tags. Each annotation tag includes the Wikipedia annotations.

API usage:

These services are used together with multilingual analysis and return a XML with Wikipedia annotations for each entity. An example of the body XML POST call is the following:

```
<item> <sentences> <sentence id="1"><text>Unesco</text> <tokens> <token pos="NP00SP0" end="6"
lemma="unesco" id="1.1" start="0">Unesco</token> </tokens> </sentence> <entities> <entity
type="person" displayName="unesco" id="1"> <mentions> <mention sentenceId="1" id="1.1"
words="Unesco"></mention> </mentions> </entity> </entities> </item>
```

4.2.6 KIT Cross-lingual Similarity

API overview:

This service obtains the similarity score between two specific documents independently of the language. Is based on cross-lingual extension of Explicit Semantic Analysis (ESA) and uses Wikipedia dumps as knowledge source.

API operations:

- **Cross-lingual Similarity:**

  Service address: http://km.aifb.kit.edu/services/clesa/similarity

Inputs parameters:

Doc1: first document to be compared.

Lang1: language of first document.

Doc2: second document to be compared.

Lang2: language of second document.

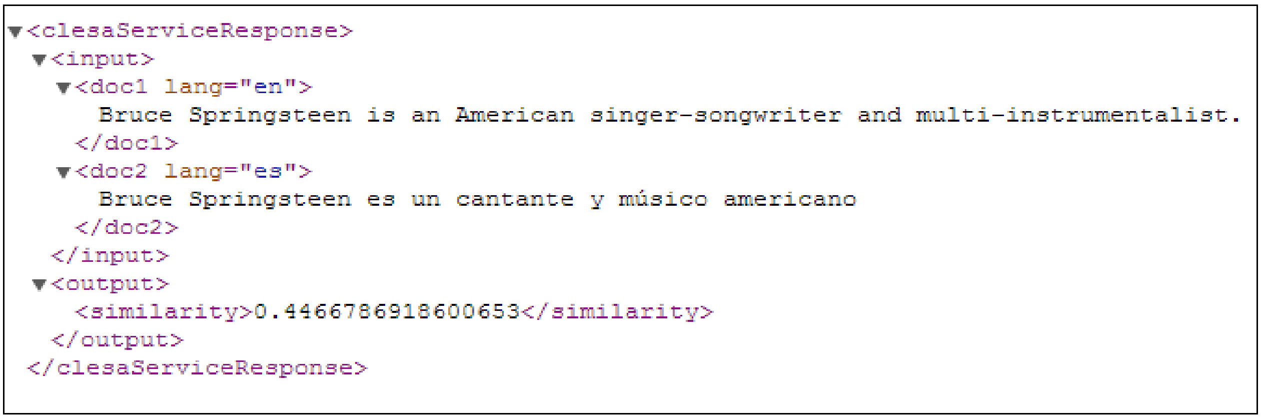
Output: the output consists of the xml elements input and output. The input element contains two doc elements with a language attribute. The output element contains the similarity score between the two input documents.

API usage:

An example of the Rest service call is the following:

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

And the response of the above example call is:



```
▼<clesaServiceResponse>
  ▼<input>
    ▼<doc1 lang="en">
       Bruce Springsteen is an American singer-songwriter and multi-instrumentalist.
     </doc1>
    ▼<doc2 lang="es">
       Bruce Springsteen es un cantante y músico americano
     </doc2>
   </input>
  ▼<output>
     <similarity>0.4466786918600653</similarity>
   </output>
 </clesaServiceResponse>
```

4.2.7 JSI Cross-lingual Similarity

API overview

This service computes the similarity between two documents written in any XLike two languages.

API operations:

- **JSI Cross-lingual Similarity:**

  Service address: http://xling.ijs.si:1111/clsi

  Input parameters:

  Doc1: first document to be compared.

  Lang1: language of first document.

  Doc2: second document to be compared.

  Lang2: language of second document.

Output: this service returns the similarity score between document 1 and document 2.

API usage:

An example of the Rest service call is the following:

http://xling.ijs.si:2222/clsi?doc1=Car%20is%20the%20most%20commonly%20used%20vehicle%20in%20Slovenia&lang1=en&doc2=Como%20es%20el%20veh%C3%ADculo%20m%C3%A1s%20utilizado%20en%20Eslovenia&lang2=es

And the response of the above example call is the similarity value: 0.90

4.2.8 Cross-lingual Analysis

API overview:

This service is based Explicit Semantic Analysis (ESA) and language links in Wikipedia. It uses Wikipedia dumps from Mai 2012 and supports English, German, Spanish, French, Catalan and Slovenian. The service can be called by using POST or GET service calls.

API operations:

- **Cross-lingual Analysis:**

  Service address: http://km.aifb.kit.edu/services/clesa/analyzer

  Input parameters:

  Doc1: document to be analysed.

  Lang1: language of document.

  Lang2: language of Wikipedia articles to retrieve.

  Retrieve: number of Wikipedia articles to retrieve

  Output: contains a copy of the input text and its language information as parameter, and a vector which contains a set of related concepts. The concept element has the following attributes:

  lang: the language of the related Wikipedia article.

  title: the title of the related Wikipedia article.

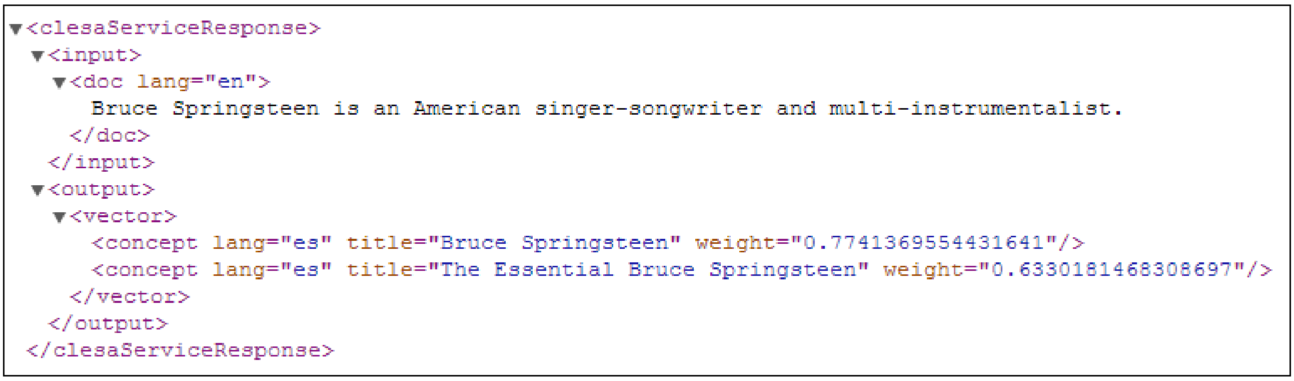
  weigth: measure of the similarity between input document and the Wikipedia article.

API usage

An example of the Rest service call is the following:

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

And the response of the above example call is:



```
▼<clesaServiceResponse>
  ▼<input>
    ▼<doc lang="en">
       Bruce Springsteen is an American singer-songwriter and multi-instrumentalist.
     </doc>
   </input>
  ▼<output>
    ▼<vector>
       <concept lang="es" title="Bruce Springsteen" weight="0.7741369554431641"/>
       <concept lang="es" title="The Essential Bruce Springsteen" weight="0.6330181468308697"/>
     </vector>
   </output>
 </clesaServiceResponse>
```

5 Conclusions

In this document we have presented the current state of the XLike toolkit at month 15 of the project lifetime. This toolkit is the result of the components implemented during the year one of the project towards the completion of the early prototype, and the new ones detected during the study of the new use cases to be accomplished during the second year of the project. A special focus has been put on identifying the components which are already implemented versus the new ones.

This infrastructure architecture, which was early defined at month three [1] has been updated with a new set of components and their related implemented services that provide the different functionalities of the XLike toolkit. This process of generating the toolkit has been done by allowing the co-evolution of the architecture, use cases, and XLike prototypes simultaneously. This heterogeneous environment has been glued by following an agile development approach and focusing on the defined use cases. It is also important highlighting that the introduced technological demo has been used as a glue towards converting to web services some parts of the already available, by the partners of the consortium, components (e.g. Freeling) obtaining successful results and a dissemination tool as well.

As can be seen in this document, we already have more than thirty services running which cover the whole pipeline of the project as it was shown in the early prototype deliverable [5]. The plan for the next period is to continue with the implementation of the identified new components as well with the development of the platform to provide real-time and scalability capabilities to the XLike project. These two major guidelines aim to support the new proposed use cases (Bloomberg and STA) [2] and the industrial showcase [3]. We also want to highlight the importance of publishing the toolkit and we would like to enhance its components towards supporting further functionalities for a independent language social Web.

This document reports the latest advances regarding the toolkit architecture which were previously defined at [1] and it represents the first complete definition of the APIs covered so far in the project. This toolkit architecture will be used for the development of the industrial showcase at month 24, the demonstrator prototype at month 24, the fully functional prototype at month 36, and the needed updates of this API specification will be included in these prototypes for covering the new functionalities that could appear from now until then.

6 References

[1] XLike deliverable “D6.1.1 – "Early toolkit architecture specification”

[2] XLike deliverable “D1.2.2 – "Requirements for demonstrator"

[3] XLike deliverable “D8.2.1 – "XLike Showcase specification”

[4] XLike deliverable “D6.1.1 – "Early toolkit architecture specification”

[5] XLike deliverable "D6.2.1 - "Early Prototype”

[6] Zhixing Li.; Zhang P.; García-Cuesta, E.; and Fortuna, B. “Demo: A Cross-lingual News Analytics Platform. Submitted to SIGIR’2013 demo panel”.

[7] XLike deliverable “D1.2.1 – “Requirements for early prototype”

Annex A Requirements-Components Relation

In this annex we show the relation between the requirements identified for year one and two [2], and the components implemented or to be developed for accomplishing with those requirements which have been defined and described in this document D6.1.2 "Final Toolkit architecture specification".

Table 2. Relation Between Requirements and Services

Requirement Identifier	Component identifier	Description
UC1	• TC-01 • TC-06 • TC-07 • TC-17 • TC-18 • TC-19 • TC-20 • TC-25	• NewsFeed • Linguistic Relation Extraction • Informal Language Analysis • KIT Cross-lingual Similarity • JSI Cross-lingual Similarity • Cross-lingual Analysis • Cross-lingual Document Linking • News Data Visualization Component
UC2	• TC-01 • TC-06 • TC-07 • TC-17 • TC-18 • TC-19 • TC-20 • TC-25	• NewsFeed • Linguistic Relation Extraction • Informal Language Analysis • KIT Cross-lingual Similarity • JSI Cross-lingual Similarity • Cross-lingual Analysis • Cross-lingual Document Linking • News Data Visualization Component
UC3	• TC-01 • TC-06 • TC-07 • TC-17 • TC-18 • TC-19 • TC-20 • TC-25	• NewsFeed • Linguistic Relation Extraction • Informal Language Analysis • KIT Cross-lingual Similarity • JSI Cross-lingual Similarity • Cross-lingual Analysis • Cross-lingual Document Linking • News Data Visualization Component
UC4	• TC-01 • TC-03 • TC-04 • TC-08 • TC-09 • TC-10 • TC-13 • TC-16 • TC-17 • TC-18 • TC-19 • TC-20 • TC-25	• NewsFeed • Language Identification • Multilingual Analysis • Name Entity Annotation • Wikipedia Miner Wikifier Annotation • Early Ontological Word-sense-disambiguation • Final Text Annotation Service • Cross-lingual USP • KIT Cross-lingual Similarity • JSI Cross-lingual Similarity • Cross-lingual Analysis • Cross-lingual Document Linking • News Data Visualization Component

UC5	• TC-01 • TC-03 • TC-04 • TC-05 • TC-08 • TC-09 • TC-17 • TC-18 • TC-19 • TC-20 • TC-21 • TC-22 • TC-23 • TC-24 • TC-25	• NewsFeed • Language Identification • Multilingual Analysis • Deep Linguistic Analysis • Name Entity Annotation • Wikipedia Miner Wikifier Annotation • KIT Cross-lingual Similarity • JSI Cross-lingual Similarity • Cross-lingual Analysis • Cross-lingual Document Linking • Semantic Graph Extraction • Event Extraction • Detection of news reporting bias • Trend and complex event detection • News Data Visualization Component
RQ1	• TC-01	• NewsFeed
RQ2	• TC-03 • TC-04	• Language Identification • Multilingual Analysis
RQ3	• TC-08 • TC-09 • TC-10 • TC-11 • TC-12 • TC-13	• Name Entity Annotation • Wikipedia Miner Wikifier Annotation • Early Ontological Word-sense-disambiguation • Crowd Sourcing Word-sense-disambiguation • Final Ontological Word-sense-disambiguation • Final Text Annotation Service
RQ4	• TC-17 • TC-18 • TC-19 • TC-20	• KIT Cross-lingual Similarity • JSI Cross-lingual Similarity • Cross-lingual Analysis • Cross-lingual Document Linking
RQ5	• TC-25	• News Data Visualization Component
RQ6	• TC-05	• Deep Linguistic Analysis
RQ7	• TC-06 • TC-07	• Linguistic Relation Extraction • Informal Language Analysis
RQ8	• TC-21	• Semantic Graph Extraction
RQ9	• TC-22 • TC-23 • TC-24	• Event Extraction • Detection of news reporting bias • Trend and complex event detection