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

XLike

Deliverable D8.1.4

Impact and Continuation Plans for XLike Services and Software

Editor:	Esteban García-Cuesta, iSOCO
Author(s):	Esteban García-Cuesta, iSOCO; Alejandro Caparros, iSOCO; Matea Srebačić, UZG; Vanja Štefanec, UZG; Marko Tadić, UZG; Lluis Padró, UPC; Lei Zhang, KIT; Zhixing Li, THU; Blaz Fortuna, JSI;
Deliverable Nature:	Report
Dissemination Level: (Confidentiality)[1]	Public (PU)
Contractual Delivery Date:	M18
Actual Delivery Date:	M18
Suggested Readers:	Developers creating software components to be integrated, developers creating case study prototypes, software developers.
Version:	1.0
Keywords:	

[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:	WP8 – Dissemination, exploitation, and community building
Document Title:	D8.1.4 – Impact and Continuation Plans for XLike Services and Software
Editor (Name, Affiliation)	Esteban García-Cuesta, iSOCO
Work package Leader (Name, affiliation)	Marko Tadić, UZG
Estimation of PM spent on the deliverable:	8

Copyright notice

Executive Summary

This document presents the impact and continuation plans for XLike services and software: it describes the actual plans in order to ensure that the services and corpora developed within XLike are available within the project and they also remain available after the lifetime of the project. This document also explores the possibility for further development of a community around the created open source software.

Table of Contents

List of Figures

List of Tables

Abbreviations

D	Deliverable
T	Task
WP	Work Package

Definitions

Pipeline Refers to the flux of different processes which are applied to a set of raw data in order to analyse 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

The plans for ensuring that services and corpora developed within the XLike project will remain available after the lifetime of the project are based on three different levels.

The first level corresponds to the **partner level**. This level describes how the different partners are going to maintain the components which have been used or upgraded during the project but they had been developed previously and therefore their licensing is constrained to that previous work. The second level is the **XLike project** itself and it defines how the main functionalities and resources produced during the execution of the project will be maintained for internal use within the project but also for preserving it longer in the future. This level takes care of both, the source code produced by the different partners, and also of the correctness of its implementation towards reusing the services in other context or platforms.

This last remark will be the baseline for the third level. The **community level** provides a broader context where the different services and resources of the project can be reused and maintained beyond the scope of the project. This level is mostly based on community activities to enforce the use of the technological developments, on transferring of the developed functionalities to the end-users as early adopters of the technological advances of the project (as it was done during the first year for STA [1] and Bloomberg [2], and will be done during the second year for the new identified use cases [3]), and also on using the META-SHARE platform, which is well-known in the computational linguistic community, for dissemination and easy accessibility of resources. In our case we have used it for sharing XLike's linguistic (pre)processing pipelines.

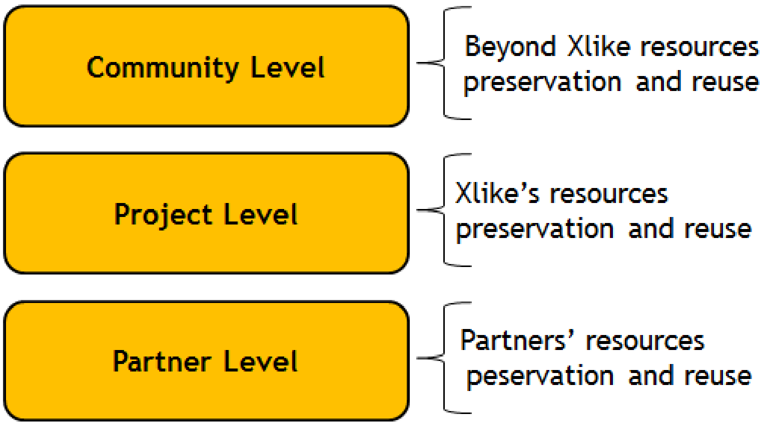
Community Level
Beyond Xlike resources preservation and reuse
Project Level
Xlike's resources preservation and reuse
Partner Level
Partners' resources peservation and reuse

Figure 1 Three levels approach for resources maintenance at XLike project.

The Figure 1 shows the above introduced layers and their main purposes ranging from a partner level (which focuses on partner's already available resources which mostly are constrained by previous work and legacy licensing issues), to a more generic community level which focuses on the preservation of XLike resources and cross-lingual functionalities beyond the project lifetime, going through the project level which is being adopted during the execution of the project. The project level is strongly related with the initially introduced development process in D6.1.1 "Early Toolkit Architecture Specification" [4] which provided guidelines for code sharing and components deployment as part of the SCRUM development methodology and the co-evolutionary implementation of the project guided by the use cases.

The use of this methodology which encourages working on small teams, which often are composed by members of different partners, and pursuing specific short-term goals has naturally provided a good context for preserving the developed software and the created resources. It also has simplified their reuse by allowing the easy transferring of source code and its deployment at any platform during the implementation of the project for testing purposes. In the next sections we will briefly introduce the main topics related to this deliverable for better understanding.

1.1 META-SHARE

The very diverse and heterogeneous landscape of huge amounts of digital and digitized resources (publications, datasets, multimedia files, processing tools, services and applications) has drastically transformed the requirements for their publication, archiving, discovery and long-term maintenance. Digital repositories provide the infrastructure for describing and documenting, storing, preserving, and making this information publicly available in an open, user-friendly and trusted way. Repositories represent an evolution of the digital libraries paradigm from open access towards permanent access principle, introducing also advanced search capabilities and large-scale distributed architectures.

META-SHARE aims at providing such an open, distributed, secure, and interoperable infrastructure for the Language Technology domain. *Open*, since the infrastructure is conceived as an ever-evolving, scalable resource base including free and for-a-fee resources and services; *distributed* because it consists of networked repositories/data centres accessible through common interfaces; *interoperable*, because the resource base will be standards-compliant, trying to overcome format, terminological and semantic differences; *secure*, since it will guarantee legally sound governance, legal compliance and secure access to licensable resources.

All of this makes it very suitable for sharing some of the main tools, services and resources developed within the XLike project.

1.2 Licensing

The licensing covers the software and data produced by XLike partners. The different components of the project are specified at D6.1.2 “Final Toolkit Architecture Specification” [5] and some of them have legacy licensing constraints which are also included in this deliverable. In Section 3 we describe the chosen by default licensing for the XLike project and the complete list of the licenses for all the components can be consulted at Annex A “Components Licensing”.

1.3 xLime as continuation of XLike

Recently the STREP project entitled xLime “crossLingual crossMedia knowledge extraction” has been approved in the ICT Call 10 FP7-ICT-2013-10 (objective ICT-2013.4.1 Content analytics and language technologies; target a) Cross-media content analytics) which is coordinated by one of the partners of the XLike consortium (KIT) including JSI and iSOCO also as partners of the consortium. The main goal of this new project is to extend the current XLike functionalities (which are mainly focused on cross-lingual knowledge extractions) to any type of media content (video, audio, and text). This new project will be a good opportunity for enlarging the current XLike cross-lingual functionalities to other contents but also an environment where some of the components/services developed would be reused and updated allowing the continuation of the work done within XLike so far.

1.4 Relation with Other Work Packages

This document includes services, corpora, and software components (e.g. those used for prototyping), that are developed within the XLike project. The complete set of components have been described in the D6.1.2 “Final Toolkit Architecture Specification” and the prototypes of the year one are described in D6.2.1 “Early Prototype” as part of the WP6 work.

2 META-SHARE

2.1 Introduction

The main goal of using META-SHARE within XLike project is to provide access to the cross-lingual technologies by using a well-known computational linguistic community array of repositories which will also provide dissemination capabilities beyond the project. The resources that have been identified so far can be split into: i) tools and services, and ii) corpora.

2.2 Tools and Services

By the term "tools and services" we refer to the set of different components which have been implemented in XLike and are going to be accessible not only at the project level, but also at the community level by sharing them through META-SHARE [6]. Concretely, these tools and services are going to be published through a META-SHARE node which is hosted by one of the partners of the project (UZG) at [7]. The tools and services which are being shared are the complete linguistic (pre)processing pipelines associated to the different XLike languages. These pipelines perform the multilingual analysis, cross-lingual analysis, and include also the previous language identification service needed for selecting the specific multilingual pipeline given an input text. So far these "services and tools" are available for: en, es, de, zh, ca, and sl, and we are developing the Croatian pipeline which will be also uploaded to META-SHARE throughout the year two of the XLike project.

Therefore the following tools and services have been added at META-SHARE as part of the impact and continuation plans of the project:

- Language identification: http://sandbox-xlike.isoco.com/services/language code/ident
- Language analysis English: http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-en
- Language analysis Spanish: http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-es
- Language analysis Chinese: http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-zh
- Language analysis German: http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-de
- Language analysis Catalan: http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-ca
- Language analysis Slovene: http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-sl

The complete list of the tools and services added to META-SHARE can be consulted at Annex B "XLike META-SHARE Descriptions" which also includes the META-SHARE URLs and identifiers.

2.3 Corpora

We currently have collected data from different media sources accumulating more than 200,000 RSS feeds, and Twitter (see also D1.3.1 "Early prototype of data infrastructure" [8] and D1.3.2 "Final prototype of data infrastructure [9] for more information). This data, jointly with the additional information generated by the XLike functionalities, is available through JSI Newsfeed service at http://newsfeed.ijs.si/stream/[2]and it provides a complete multi/cross-lingual corpus for the languages en, es, zh, de, ca and sl. The data format of this corpus can be consulted at D6.2.1 "Early prototype" [10] Annexes B and C.

We want to point out that due to copyright issues the service is currently only available for internal purposes but we expect to have a clear separation among the copyrighted and legally shareable collected

[2] The URL accepts optional parameters that can be consulted at http://newsfeed.ijs.si/

data in order to build a new corpus by the end of the year two. By then we will publish the legally shareable corpus at META-SHARE for gaining community impact and for dissemination.

Furthermore we are currently implementing a new real time infrastructure based on Apache Cassandra for providing scalability and easy transfer capabilities for the current infrastructure. There are already some of the functionalities transferred[3] (it provides accessibility to entities, relations, text, and tokens for the different XLike languages and for a few RSS's sources) though the number of analysed sources are limited and the data format will have to be updated to the new requirements of the year two demonstrator. We plan to update this infrastructure during the second part of the year two providing also accessibility to the generated analysed corpus.

[3] http://sandbox-xlike.isoco.com/xlike/isoco-news/search (it accepts the date parameter to specify the initial date of retrieval following the format yyyy-mm-dd hh:mm:ss CEST , e.g. date=2013-06-02%2010:00:00%20CEST).

3 Other Linguistic Infrastructures

We have been considering making XLike tools/services/resources accessible through other language/text-based research infrastructures, like CLARIN ERIC. CLARIN ERIC, as already established European Research Infrastructure in accordance with EU regulations on ERIC, would certainly be a good candidate for making XLike products accessible and visible, while on the other hand CLARIN RI users could benefit from a series of mature language processing chains, as XLike pipelines are. Since at this moment CLARIN national consortia have not been established in all countries of XLike languages (excl. English: for Spanish and Catalan – Spain, for Slovene – Slovenia, while for Chinese the participation of China in CLARIN is not yet expected), we were not being able to detect the respective national CLARIN participating institution for all XLike languages, that would be willing to make our pipelines accessible through CLARIN infrastructure. However, this doesn't mean that we will not try to be present also in the CLARIN ERIC RI, but by the end of the project we might be able to find such a node in CLARIN ERIC network and make XLike language tools/services/resources available through it.

4 Maintenance of source code beyond XLike

4.1 Introduction

The source code of the project also follows the above mentioned three levels of organization. At community and project levels the different components implemented during the execution of the project have been published at Github in a public repository, whereas the source code of those components that due to legacy licensing restrictions were not able to be publically published have been and will be maintained at partner level. For development purposes during the project this components have been shared privately[4].

We want to point out that at the community level the complete language pipelines have been published (including both services and clients) providing the complete enriched data which includes the different functionalities provided by the different work packages (WP1-WP5)[5]. The current enriched data format was defined during the Y1 of the project and can be consulted at Annex C of "D6.2.1 – Early Prototype" [10].

4.2 Published source code

All the components developed within the XLike project have been published publically in Github[6] (project level) or are available under the specified license by their respective creators (partners' level). The source code available at Github is organized by the WP that it belongs to allowing an easy collaboration between the different partners.

Furthermore, the fact that the development of the project has been done using the SCRUM methodology (which was proposed in [8]), involving sometimes people from different institutions, it has naturally lead towards a collaborative environment which is very similar to the one needed for continuing upgrading the current services and tools beyond the XLike project (moving from project level to community level).

4.3 Licensing

The project has adopted the GNU/GPL[7] licensing as default. However each one of the partners has its own types of licensing according to internal restrictions or legacy issues. The complete list of components implemented in the project and its licensing can be consulted at Annex A. This list also includes the maintenance plans for those components which are going to be maintained at partners' level.

[4] https://github.com/JozefStefanInstitute/xlike

[5] https://github.com/xlike-project/wp8/tree/master/pipeline_clients

[6] https://github.com/organizations/xlike-project

[7] http://www.gnu.org/licenses/gpl.html

5 Conclusions

In this document we have presented the current continuation plans for XLike services and software which has been organized on three different levels: partner level, project level, and community level.

At partner level we have included all those components which are being used within the project but due to legacy licensing restrictions or any other constraint they have to be maintained by the corresponding partner.

At project level all the components implemented within XLike project have been considered. This also includes the multi-cross/lingual pipelines which incorporate the most important technological functionalities implemented so far. The use of public repositories for sharing the source code of these components have been naturally adopted during the development of the project (due to the use of SCRUM methodology) which has turned out into a suitable way for preserving, reusing, and sharing it not only internally but also externally.

At community level the main goal is to increase the impact and community awareness about the services and the implemented functionalities. For this purpose one of the nodes of the platform META-SHARE (UZG) has been used for publishing XLike services towards preserving them beyond its lifetime. We are also considering the use of other platforms as CLARIN ERIC.

During the rest of the project we plan to continue with these three levels of organization for impact and continuation plans. At the same time we plan to continue updating and implementing the new components following this organization. We also plan to promote the use of the services published in META-SHARE by organizing hands-on sessions focused on building new applications and making use of this cross-lingual technological functionalities provided by the XLike project.

Furthermore, despite the fact that there is not any other deliverable for reporting these updates they will be included as part of the D6.2.2 Demonstrator prototype (M24) and D6.2.3 Fully functional prototype (M32) deliverables due to its closeness to the present document.

6 References

[1] XLike deliverable “D7.2.1 – Early Prototype and Validation Report”

[2] XLike deliverable “D7.1.1 – Early Prototype and Validation Report”

[3] XLike deliverable “D1.2.2 – Requirements for demonstrator”

[4] XLike deliverable “D6.1.1 – Early Toolkit Architecture Specification”.

[5] XLike deliverable “D6.1.2 – "Final toolkit architecture specification”.

[6] Meta-Share Platform available at http://www.meta-share.eu/ (last time accessed 26-06-2013).

[7] Meta-Share University of Zagreb node available at http://meta-share.ffzg.hr/ (last time accessed 26-06-2013).

[8] XLike deliverable “D1.3.1 – Early prototype of data infrastructure”

[9] XLike deliverable “D1.3.2 – Final prototype of data infrastructure”

[10] XLike deliverable “D6.2.1 – Early Prototype”.

7 Annex A Components Licensing

This Annex describes the licensing for each one of the components described in D6.1.2 [1].

Table 1 Components' licenses

Service Name	Identifier	Description	License (GPL, BSD3, MIT, Apache 2.0, LGPL3.0, others)	Maintenance Plan
Newsfeed	TC-01	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.	Source code available under BSD3[8]	The service is maintained and developed by JSI for several years since 2007, and is used to gather data for research purposes (See http://newsfeed.ijs.si/). Any commercial installation can be setup using the open-source code. Proprietary connectors are not part of the open source code, for example adapters for Bloomberg and STA private feeds.
iSOCO real time storage and access	TC-02	This component is based on Apache Cassandra and provides access to the data obtained from JSI NewsFeed and XLike pipeline services allowing rapid accessibility to the enriched raw data.	GPLv2	This component is maintained at project level.
Language Identification	TC-03	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.	The results of the service are licensed as CC-BY.	This component is part of FreeLing, an open-source library maintained at UPC. See http://nlp.lsi.upc.edu/freeling for details.

[8] https://github.com/JozefStefanInstitute/newsfeed

Multilingual Analysis (en, es, zh, ca, sl)	TC-04	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. This component is mostly based on Freeling.	The results of the service are licensed as CC-BY. This service makes use of Freeling which is licensed under GPL.	This component is part of FreeLing, an open-source library maintained at UPC. See http://nlp.lsi.upc.edu/freeling for details.
Multilingual Analysis (de)	TC-04	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.	The results of the service are licensed as CC-BY. The German pipeline uses several existing modules which have Apachev2.0 and GPLv2 or later licenses.	The service is maintained at project level and the different modules are maintained by the respectively owners.
Multilingual Analysis (sl)	TC-04	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:	Components for Slovene are close-source and are provided as free	The components for Slovene are part of main language infrastructure and are financed by The Ministry for Science.

		- 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.	binaries[9]	
Deep Linguistic Analysis	TC-05	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.	The results of the service are licensed as CC-BY. This service makes use of Treeler which is licensed under GPL.	This component is part of Treeler, an open-source library maintained at UPC. See http://treeler.lsi.upc.edu for details.
Linguistic Relation Extraction	TC-06	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.	The results of the services are licensed as CC-BY	This component is maintained at project level.
Informal Language Analysis	TC-07	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.	The results of the service are licensed as CC-BY. This service makes use of Freeling which is licensed under GPL.	This component is part of FreeLing, an open-source library maintained at UPC. See http://nlp.lsi.upc.edu/freeling for details.
Name Entity	TC-08	This component discovers the Wikipedia annotations associated to a	GPLv2	

[9] http://razclenjevalnik.slovenscina.eu/Programska_oprema.aspx

Annotation		document by matching the names of the detected entities against the Wikipedia titles.		
Wikipedia Miner Wikifier Annotation	TC-09	This component adds the Wikipedia annotations to a document using the Wikipedia Miner wikifer based approach.	GPLv2	This component is maintained at project level.
Early Ontological Word-sense-disambiguation	TC-10	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.	Under development and to be published as GPLv2	This component is maintained at project level.
Crowd Sourcing Word-sense-disambiguation	TC-11	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.	GPLv2	This component is maintained at project level.
Final Ontological Word-sense-disambiguation	TC-12	These components provide the final version of ontology based word-sense disambiguation supporting all ontological knowledge resources handled by the final disambiguation tools.	Source code will be available under BSD3	WSD components are important part of JSI research agenda will be developed and reused in the work on transforming text to structured and reasonable form.
Final Text Annotation Service	TC-13	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.	GPLv2	This component is maintained at project level.
Early Machine Translation based Semantic Annotation	TC-14	This component provides first version of semantic annotation prototype based on machine translation trained on the initial data.	GPLv2	This component is maintained at project level.
Final Machine Translation	TC-15	This component provides the final version of semantic annotation prototype based trained on the extended dataset.	GPLv2	This component is maintained at project level.

based Semantic Annotation				
Cross-lingual USP	TC-16	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.	GPLv2	This component is maintained at project level.
KIT Cross-lingual Similarity	TC-17	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.	GPLv2	This component is maintained at project level.
JSI Cross-lingual Similarity	TC-18	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.	Close-sourced; considering to open source it under BSD3 license	The service is used as a component of Newsfeed service, and is maintained as part of the same system.
Cross-lingual Analysis	TC-19	This component retrieves related Wikipedia articles in a specified language given an input document.	GPLv2	This component is maintained at project level.
Cross-lingual Document Linking	TC-20	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.	Close-sourced; considering to open source it under BSD3 license	The service is used as a component of Newsfeed service, and is maintained as part of the same system.
Semantic Graph Extraction	TC-21	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.	The results of this service are licensed as CC-BY	This component is maintained at project level.
Event Extraction	TC-22	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.	Code under development, license to be decided	The service is planned to be used as a component of Newsfeed service, and will be maintained as part of the same system.
Detection of news	TC-23	Detection of news reporting bias is a service whose main target is to detect news with differences in reporting about the same events across	Code under development, will be	This component will depend on availability of linguistically processed data (news corpora). JSI will continue run the service as part of demonstration

reporting bias		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.	licensed as BSD3.	services running on top of Newsfeed in a maintainable manner (e.g. using it only on a subset of relevant news sources).
Trend and complex event detection system	TC-24	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.	Code under development, will be licensed as GPLv2	This component is maintained at project level.
News Data Visualization	TC-25	This component uses some of the existing text visualization and network visualization techniques to show real-time information spreading across	BSD3 for front end	This component is maintained at project level.

		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		

8 Annex B XLike Meta-Share Descriptions

In Table 2 the complete language pipelines of the XLike project are linked to the Meta-Share identifiers and the URLs where the description of these services can be obtained. The complete list of META-SHARE metadata description for the different components of the project can be found in[10].

Table 2 Meta-share pipelines' descriptions

Component Identifier	Description	Meta-share identifier	Meta-Share URL	License	Maintenance plan
TC-03	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.	XLike_001_LangId	http://meta-share.ffzg.hr/repository/browse/xlike-language-identifier/dc08a87ad8ef11e28a985ef2e4e6c59e988a23f60c904483834a5e9fbfe45c45/	The results of this service are licensed as CC-BY.	This service is maintained at community level.
TC-04, TC-06, TC-08, TC-09	English pipeline.	XLike_En_pipeline	http://meta-share.ffzg.hr/repository/browse/xlike-english-pipeline/b367e514e33c11e28a985ef2e4e6c59e8e0a4d15b1d547f6afdf3d78a437f31c/	The results of this service are licensed as CC-BY.	This service is maintained at community level.

[10] https://github.com/xlike-project/wp8/blob/master/META-SHARE/Meta-Share%20X-Like.docx

TC-04, TC-06, TC-08, TC-09	Spanish pipeline	XLike_Es_pipeline	http://meta-share.ffzg.hr/repository/browse/xlike-spanish-pipeline/c90a5ef8e34411e28a985ef2e4e6c59e8da8ab9a1a7c4c4abbcf2ad998254c19/	The results of this service are licensed as CC-BY.	This service is maintained at community level.
TC-04, TC-06, TC-08, TC-09	Chinese pipeline	XLike_Zh_pipeline	http://meta-share.ffzg.hr/repository/browse/xlike-chinese-pipeline/77f0f07ee34211e28a985ef2e4e6c59eae8ec40cbeed48c6808219c2de8d7a64/	The results of this service are licensed as CC-BY.	This service is maintained at community level.
TC-04, TC-06, TC-08, TC-09	German pipeline	XLike_De_pipeline	http://meta-share.ffzg.hr/repository/browse/xlike-german-pipeline/07d7723ce34111e28a985ef2e4e6c59ed192c0d92d9e441cb46addd75415be96/	The results of this service are licensed as CC-BY.	This service is maintained at community level.
TC-04, TC-06, TC-08, TC-09	Catalan pipeline	XLike_Ca_pipeline	http://meta-share.ffzg.hr/repository/browse/xlike-catalan-pipeline/6eeb5f22e33e11e28a985ef2e4e6c59e159d640f495e4246801a00b6d66ccdef/	The results of this service are licensed as CC-BY.	This service is maintained at community level.
TC-04, TC-06, TC-08, TC-09	Slovene pipeline	XLike_Sl_pipeline	http://meta-share.ffzg.hr/repository/browse/xlike-slovene-pipeline/cb0882dae34311e28a985ef2e4e6c59edfe2580002a548aa92b89bf113883ac9/	The results of this service are licensed as CC-BY.	This service is maintained at community level.

In the next we provided the description of use for the available information for each one the above mentioned Meta-Share resources.

8.1 Language Identification

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.

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

Parameters:

- **text**: text to be analyzed. Input: the input follows the next format.

Input format: XML

```
<?xml version="1.0"?>
<!DOCTYPE analyze [
<!ELEMENT analyze (text)>
<!ELEMENT text (#PCDATA)>
]>
```

Response format: XML

```
<?xml version="1.0"?>
<!DOCTYPE language [
<!ELEMENT language (#PCDATA)>
]>
```

- **Response example:** <language>en</language>

Example request:

POST	http://sandbox-xlike.isoco.com/services/language_code/ident
POST Data	text=This is a Hello World Example
Content-type	application/x-www-form-urlencoded

Example of a call using curl: curl -X POST --data "text=This is a hello world example" http://sandbox-xlike.isoco.com/services/language_code/ident

A Java client can be downloaded at:
https://github.com/xlike-project/wp8/blob/master/pipeline_clients/XLikeLangIden.java

8.2 English Analysis

Description: this service executes the English analysis pipeline of the XLike project. This service provides the following functionality:

- sentence splitting and tokenization (shallow)
- lemmatization, part-of-speech tagging and morpho-syntactic annotation (shallow)
- named entity recognition and classification (shallow)
- syntactic parsing (deep)
- semantic role labeling (deep)
- extraction of tokens, lemmas, entities and relations

URL: http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-en

Parameters (a complete description of the parameters can be consulted at http://www.xlike.org/wp-content/uploads/2012/03/D2.1.1-Shallow-linguistic-processing-prototype.pdf):

- text: text to be analyzed.
- target: one of the following values, specifying the desired output elements.
  - tokens: plain tokens.
  - lemmas: tokens with lemma, and msd attributes.
  - entities: tokens with lemma, msd and ne attributes.
- conll: whether to return the analysis in CoNLL format.

Input format: XML

```
<?xml version="1.0"?>
<!DOCTYPE analyze [
<!ELEMENT analyze (text, target, conll)>
<!ELEMENT text (#PCDATA)>
<!ELEMENT target (#PCDATA)>
<!ELEMENT conll (#PCDATA)>
]>
<analyze>
        <text>Article</text>
        <target>relations</target>
        <conll>true</conll>
</analyze>
```

Response format: XML (the schema is available at https://github.com/xlike-project/wp2/blob/master/xlike_pipeline_annotations.xsd)

Example Request:

POST	http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-en
POST Data	text=This is a Hello World Example&target=entities&conll=false
Content-type	application/x-www-form-urlencoded

Example of a call using curl: curl -X POST --data "text=This is a hello world example&target=entities&conll=true" http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-en

A Java client can be downloaded at: https://github.com/xlike-project/wp8/blob/master/pipeline_clients/English.java

8.3 Spanish Analysis

Description: this service executes the Spanish analysis pipeline of the XLike project. This service provides the following functionality:

- sentence splitting and tokenization (shallow)
- lemmatization, part-of-speech tagging and morpho-syntactic annotation (shallow)
- named entity recognition and classification (shallow)
- syntactic parsing (deep)
- semantic role labeling (deep)
- extraction of tokens, lemmas, entities and relations

URL: http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-es

Parameters (a complete description of the parameters can be consulted at http://www.xlike.org/wp-content/uploads/2012/03/D2.1.1-Shallow-linguistic-processing-prototype.pdf):

- text: text to be analyzed.
- target: one of the following values, specifying the desired output elements.
  - tokens: plain tokens.
  - lemmas: tokens with lemma, and msd attributes.
  - entities: tokens with lemma, msd and ne attributes.
- conll: whether to return the analysis in CoNLL format.

Input format: XML

```
<?xml version="1.0"?>
<!DOCTYPE analyze [
<!ELEMENT analyze (text, target, conll)>
<!ELEMENT text (#PCDATA)>
<!ELEMENT target (#PCDATA)>
<!ELEMENT conll (#PCDATA)>
]>
<analyze>
        <text>Article</text>
        <target>relations</target>
        <conll>true</conll>
</analyze>
```

Response format: XML (the schema is available at https://github.com/xlike-project/wp2/blob/master/xlike_pipeline_annotations.xsd)

Example Request:

POST	http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-es
POST Data	text=Este es el ejemplo de Hola Mundo&target=entities&conll=false
Content-type	application/x-www-form-urlencoded

Example of a call using curl: curl -X POST --data "text=Este es el ejemplo de Hola Mundo&target=entities&conll=true" http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-es

A Java client can be downloaded at:
https://github.com/xlike-project/wp8/blob/master/pipeline_clients/Spanish.java

8.4 German Analysis

Description: this service executes the German analysis pipeline of the XLike project. This service provides the following functionality:

- sentence splitting and tokenization (shallow)
- lemmatization, part-of-speech tagging and morpho-syntactic annotation (shallow)
- named entity recognition and classification (shallow)
- syntactic parsing (deep)
- semantic role labeling (deep)
- extraction of tokens, lemmas, entities and relations

URL: http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-de

Parameters (a complete description of the parameters can be consulted at http://www.xlike.org/wp-content/uploads/2012/03/D2.1.1-Shallow-linguistic-processing-prototype.pdf):

- text: text to be analyzed.
- target: one of the following values, specifying the desired output elements.
  - tokens: plain tokens.
  - lemmas: tokens with lemma, and msd attributes.
  - entities: tokens with lemma, msd and ne attributes.
- conll: whether to return the analysis in CoNLL format.

Input format: XML

```
<?xml version="1.0"?>
<!DOCTYPE analyze [
<!ELEMENT analyze (text, target, conll)>
<!ELEMENT text (#PCDATA)>
<!ELEMENT target (#PCDATA)>
<!ELEMENT conll (#PCDATA)>
]>
<analyze>
        <text>Article</text>
        <target>relations</target>
        <conll>true</conll>
</analyze>
```

Response format: XML (the schema is available at https://github.com/xlike-project/wp2/blob/master/xlike_pipeline_annotations.xsd)

Example Request:

POST	http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-de
POST Data	text= Dies ist die Hallo Welt Beispiel &target=entities&conll=false
Content-type	application/x-www-form-urlencoded

Example of a call using curl: curl -X POST --data "text= Dies ist die Hallo Welt Beispiel &target=entities&conll=true" http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-de

A Java client can be downloaded at:
https://github.com/xlike-project/wp8/blob/master/pipeline_clients/German.java

8.5 Chinese Analysis

Description: this service executes the Chinese analysis pipeline of the XLike project. This service provides the following functionality:

- sentence splitting and tokenization (shallow)
- lemmatization, part-of-speech tagging and morpho-syntactic annotation (shallow)
- named entity recognition and classification (shallow)
- syntactic parsing (deep)
- semantic role labeling (deep)
- extraction of tokens, lemmas, entities and relations

URL: http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-zh

Parameters (a complete description of the parameters can be consulted at http://www.xlike.org/wp-content/uploads/2012/03/D2.1.1-Shallow-linguistic-processing-prototype.pdf):

- text: text to be analyzed.
- target: one of the following values, specifying the desired output elements.
  - tokens: plain tokens.
  - lemmas: tokens with lemma, and msd attributes.
  - entities: tokens with lemma, msd and ne attributes.
- conll: whether to return the analysis in CoNLL format.

Input format: XML

```
<?xml version="1.0"?>
<!DOCTYPE analyze [
<!ELEMENT analyze (text, target, conll)>
<!ELEMENT text (#PCDATA)>
<!ELEMENT target (#PCDATA)>
<!ELEMENT conll (#PCDATA)>
]>
<analyze>
        <text>Article</text>
        <target>relations</target>
        <conll>true</conll>
</analyze>
```

Response format: XML(the schema is available at https://github.com/xlike-project/wp2/blob/master/xlike_pipeline_annotations.xsd)

Example Request:

POST	http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-zh
POST Data	text=这是一个例子，“世界，你好&target=entities&conll=false
Content-type	application/x-www-form-urlencoded

Example of a call using curl:
curl -X POST –data “text=这是一个例子，“世界，你好&target=entities&conll=true" http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-zh

A Java client can be downloaded at:
https://github.com/xlike-project/wp8/blob/master/pipeline_clients/Chinese.java

8.6 Catalan Analysis

Description: this service executes the Catalan analysis pipeline of the XLike project. This service provides the following functionality:

- sentence splitting and tokenization (shallow)
- lemmatization, part-of-speech tagging and morpho-syntactic annotation (shallow)
- named entity recognition and classification (shallow)
- syntactic parsing (deep)
- semantic role labeling (deep)
- extraction of tokens, lemmas, entities and relations

URL: http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-ca

Parameters (a complete description of the parameters can be consulted at http://www.xlike.org/wp-content/uploads/2012/03/D2.1.1-Shallow-linguistic-processing-prototype.pdf):

- text: text to be analyzed.
- target: one of the following values, specifying the desired output elements.
  - tokens: plain tokens.
  - lemmas: tokens with lemma, and msd attributes.
  - entities: tokens with lemma, msd and ne attributes.
- conll: whether to return the analysis in CoNLL format.

Input format: XML

```
<?xml version="1.0"?>
<!DOCTYPE analyze [
<!ELEMENT analyze (text, target, conll)>
<!ELEMENT text (#PCDATA)>
<!ELEMENT target (#PCDATA)>
<!ELEMENT conll (#PCDATA)>
]>
<analyze>
        <text>Article</text>
        <target>relations</target>
        <conll>true</conll>
</analyze>
```

Response format: XML (the schema is available at https://github.com/xlike-project/wp2/blob/master/xlike_pipeline_annotations.xsd)

Example Request:

POST	http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-ca
POST Data	text= Aquest és l'exemple de hola món&target=entities&conll=false
Content-type	application/x-www-form-urlencoded

Example of a call using curl: curl -X POST --data "text= Aquest és l'exemple de hola món&target=entities&conll=true" http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-ca

A Java client can be downloaded at: https://github.com/xlike-project/wp8/blob/master/pipeline_clients/Catalan.java

8.7 Slovenian Analysis

Description: this service executes the Slovenian analysis pipeline of the XLike project. This service provides the following functionality:

- sentence splitting and tokenization (shallow)
- lemmatization, part-of-speech tagging and morpho-syntactic annotation (shallow)
- named entity recognition and classification (shallow)
- syntactic parsing (deep)
- semantic role labeling (deep)
- extraction of tokens, lemmas, entities and relations

URL: http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-sl

Parameters (a complete description of the parameters can be consulted at http://www.xlike.org/wp-content/uploads/2012/03/D2.1.1-Shallow-linguistic-processing-prototype.pdf):

- text: text to be analyzed.
- target: one of the following values, specifying the desired output elements.
  - tokens: plain tokens.
  - lemmas: tokens with lemma, and msd attributes.
  - entities: tokens with lemma, msd and ne attributes.
- conll: whether to return the analysis in CoNLL format.

Input format: XML

```
<?xml version="1.0"?>
<!DOCTYPE analyze [
<!ELEMENT analyze (text, target, conll)>
<!ELEMENT text (#PCDATA)>
<!ELEMENT target (#PCDATA)>
<!ELEMENT conll (#PCDATA)>
]>
<analyze>
        <text>Article</text>
        <target>relations</target>
        <conll>true</conll>
</analyze>
```

Response format: XML (the schema is available at https://github.com/xlike-project/wp2/blob/master/xlike_pipeline_annotations.xsd)

Example Request:

POST	http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-sl
POST Data	text= To je primer zdravo svet&target=entities&conll=false
Content-type	application/x-www-form-urlencoded

Example of a call using curl: curl -X POST --data "text=To je primer zdravo svet &target=entities&conll=true" http://sandbox-xlike.isoco.com/xlike-pipeline/services/xlike-sl

A Java client can be downloaded at**:** https://github.com/xlike-project/wp8/blob/master/pipeline_clients/Slovene.java