XLike

Deliverable D1.3.1

Early prototype of data infrastructure

Editor:	Esteban García-Cuesta, iSOCO
Author(s):	Blaz Fortuna, JSI; Mitja Trampus, JSI; Blaz Novak, JSI; Esteban García-Cuesta, iSOCO; José Manuel Gómez, iSOCO;Xavier Carreras, UPC; Marko Tadić, UZG; Peter Penko, STA; Pat Moore, BLP; Achim Rettinger, KIT; Juanzi Li, THU; Pushpak Bhattacharyya ITTBombay; Evan Sandhaus, NYT;
Deliverable Nature:	Prototype (P)
Dissemination Level: (Confidentiality)	Public (PU)
Contractual Delivery Date:	M3
Actual Delivery Date:	M3
Suggested Readers:	Developers creating software components
Version:	1.0
Keywords:	datasets; linguistic corpora; news stream; news indexing

Disclaimer

Full Project Title:	Cross-lingual Knowledge Extraction
Short Project Title:	XLike
Number and Title of Work package:	WP1 – Definition and Data Provision
Document Title:	D1.3.1 – Early prototype of data infrastructure
Editor (Name, Affiliation)	Esteban García-Cuesta, iSOCO
Work package Leader (Name, affiliation)	Blaz Fortuna, JSI
Estimation of PM spent on the deliverable:	12

Executive Summary

This document presents the early description of the XLike data infrastructure which will be the main source of data to be used by the different components of the project.

The document is based mostly on the collected information from all the partners, and the functional/technical requirements of the project which are part of the tasks T1.1 and T1.2. The main goal of this deliverable is to describe this collections and a reusable and extendible data infrastructure covering the needs of publishers and related industry being represented through the case-studies.

Regarding the main outcome of this task T1.3, jointly with this deliverable, it is a prototype which is available at a public URL[1] and provides the initial data infrastructure for the project. This data infrastructure is already being used for the development of an example of Sandbox platform at T6.1 and T6.2.

This report also includes the list of identified sources and data models provided by all the partners of the project.

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

Table of Contents

Executive Summary 3
Table of Contents 4
List of Figures 5
List of Tables 6
Abbreviations 7
1 Introduction 8
2 Data Sources 9
3 News Stream 16
3.1 News Collection 16
3.1.1 System Architecture 16
3.1.2 Sources 17
3.1.3 Language distribution 18
3.1.4 Data volume 18
3.1.5 Responsiveness 18
3.1.6 Public Data and API 18
3.2 NewsMiner 19
3.2.1 Indexed fields 20
3.2.2 Query API 20
4 Conclusions 22
References 23

List of Figures

Figure 1 News aggregator system architecture..... 17
Figure 2. The daily number of downloaded articles..... 18

List of Tables

Table 1 Data sources.. 10

Abbreviations

NLP	Natural Language Processing
API	Application Programming Interface
SKOS	Simple Knowledge Organization System
CLEF	Cross-Language Evaluation Forum
SPARQL	SPARQL Protocol and RDF Query Language
RDF	Resource Description Framework
MSD	Morpho-Syntactic Description
CGI	Common Gateway Interface
RSS	Really Simple Syndication

1 Introduction

This deliverable provides a list of existing available data sources within the project, and details the existing news stream infrastructure which will be used and further be extended within the project. This set of initial sources covers a broad range of different types of media from digital newspapers to blogs, or different needed corpuses for NLP analysis.

The deliverable is split into two main parts. The first part provides a list of static corpora, which can be used to develop, train and evaluate language technologies. The second part provides a preliminary API to access a real time feed of news and social media.

2 Data Sources

As part of deliverable **D1.1.1 – Report and library on the existing technology and data** a comprehensive list of datasets available within the consortium was collected. We want to highlight that this initial set is not an intention to provide an exhaustive list or specific sources which will be used, but a rough set of sources from all the participants of the project and also to provide a common point for being updated in the future. The complete list is shown in the **Table 1** describing the following characteristics:

- **Data Entity**: name or identification of the data resource i.e. JSI news crawler
- **Data Responsible**: name of the institution or company responsible for the data source described, i.e. JSI
- **Data sources**: the type of data which is gathered, i.e. main stream news/blogs/twitter/Facebook/…
- **How can be accessed**: the method to get access to the data, i.e. API/WS/files/databases/…
- **Type of data**: the type of data which is stored, i.e. raw_text/categories/ontology/enriched_text/…
- **Amount of data**: the size of data which needs to be stored for being processed in the pipeline. This information can be expressed in /M/T/P/Bytes or as a data flow per time, i.e. 1TB or 150.000x15kB streams news per day.
- **License**: identifies if the data is only available for the project purposes (PR) or if it is also public for any other purposes (PU). The identification of the type of license would be desirable.
- **Web site URL**: where is the dataset available

Table 1 Data sources

Data Entity	Responsible	Data Sources	How can be accessed?	Type of data	Amount of data and covered languages	License	Web site URL
JSI news crawler	JSI	Main stream news (including New York Times and Bloomberg)	Web service API for access to raw feed and full-text search.	Each item contains: title and content, source, URL, publish date. Articles from selected sources are sent through Enrycher.	150.000 articles per day	PR	NA
Wikipedia	JSI	Wikipedia [1]	Available as text corpora.	Processed Wikipedia dump for top 200 languages (based on article counts). Each article is cleaned by removing wiki-text tags.	18,7 million articles	Public	http://www.wikipedia.org/
Data.NYTimes.com	NYT	New York Times Topics [2]	Can be download as a set of SKOS files	Linked Open Data	10,000 entities	Public	http://data.nytimes.com/
STA	STA	News stream	Web Service	Articles published by STA	Slovene, English	Private	http://www.sta.si/

Penn Treebank + Propbank	UPC	Newswire	Files	Sentences with syntactic and shallow semantic annotations	English (1M words)	Private	http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC99T42 http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2004T14
OntoNotes 4.0	UPC	Newswire, broadcast news, webtext	Files	Sentences with syntactic and shallow semantic and named entity annotations.	English (1.3M words), Chinese (800K words)	Private	http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2011T03
Google Web Treebank	UPC	Newswire, webtext	Files	Sentences with syntactic annotations	English (4K sentences)	Private	https://sites.google.com/site/sancl2012/home/shared-task
AnCora Corpus	UPC	Newswire, webtext	Files	Sentences with syntactic, shallow semantics, and named entity annotaitons.	Spanish (500K words) and Catalan (500K words)	Public	http://http://clic.ub.edu/corpus/ancora
CoNLL-2009 Datasets	UPC	Newswire	Files	Sentences with syntactic and shallow semantic annotaitons	Catalan (390K words), Chinese (609K w.), English (958K w.), German (648K w), Spanish (427K w.)	Private	http://ufal.mff.cuni.cz/conll2009-st/
CLEF	KIT	Test Suites	Files	Cross-Language Evaluation Forum (CLEF) Test Suites		Private	http://www.clef-initiative.eu/

MULTEXT	KIT	Questions and Answers	Files	Raw, tagged and aligned data from the Written Questions and Answers of the Official Journal of the European Community	English, French, German, Italian and Spanish	Private	http://aune.lpl.univ-ix.fr/projects/multext/
yahooanswers	KIT	Questions and Answers	Files	Questions and Answers from Yahoo! Answers		Private	http://answers.yahoo.com/
JRC-Acquis	KIT	Collection of legislative text	Files	Legislative documents of the European Union	Bulgarian, Czech, Danish, German, Greek, English, Spanish, Estonian, Finnish, French, Hungarian, Italian, Lithuanian, Latvian, Maltese, Dutch, Polish, Portuguese, Romanian, Slovak, Slovene and Swedish	Public	http://langtech.jrc.it/JRC-Acquis.html#Statistics

STA	STA	News stream	Web Service NewsML available by HTTP, FTP and e-mail	Articles published by STA	Slovene (currently 1,5 million articles, around 300 daily), English (currently around 160k articles, around 40 daily)	Private	http://www.sta.si/ access information http://www.sta.si/td.php and http://www.sta.si/en/td.php
Sina Events News	THU	News stream	Full-text search	Articles from Special News in Sina news portal	3405 events, 310247 news articles,7980 latent topics; Chinese	Private	http://www.newsminer.net/
HuDong Knowledge Base	THU	Hudong Encyclopedia [3]	SPARQL Endpoint	RDF Triples	19542 Concepts, 2381 Properties, 802593 Instances & 5237520 RDF Triples Chinese	Private	http://keg.cs.tsinghua.edu.cn/project/ChineseKB/
SETimes	UZG	News stream	not yet available for download	tagged textual structure, URL	~2 Mw per language, 10 languages: al, bg, bo, el, en, hr, mk, ro, sr, tr	Public	http://www.setimes.com/

hrWac	UZG	whole Croatian .hr domain crawled in 2011-06	not yet available for download	lemmatised, MSD-tagged (MulTextEast hr tagset)	~1,2 Bw, hr	Public	NA
siWaC	UZG	whole Slovenian .si domain crawled in 2011-06	not yet available for download	lemmatised, MSD-tagged (MulTextEast si tagset)	~380 Mw, si	Public	NA
hrenWaC	UZG	Parallel English-Croaitan sentences extracted from hr web	not yet available for download	sentence aligned	89,204 TUs	Public	NA
hr-en parallel corpus	UZG	newspaper published from 1998 to 2000	available for download through META-SHARE platform	TMX, sentence aligned	1.6 Mw hr, 1.9 Mw en, 62,534 TUs	Public	http://www.meta-net.eu/meta-share
Annotated and tagged Corpora	NYT	Tagged articles and categories	Web service API	Pieces of news from NYTimes newspaper	1.5 million manually tagged articles + 275.000 automatically	Private	http://developer.nytimes.com/docs

Semantic API	NYT	External mapping of NYT vocabulary	Web Service API	Semantic linking data	10,000 people, places, organizations and descriptors used to classify New York Times articles metadata	Private	http://developer.nytimes.com/docs/read/The_Semantic_API
Geo API	NYT	Combination of NYT vocabularies with GeoNames	Web Service API	Location information	Over 2000 places used to classify New York Times articles metadata	Private	http://developer.nytimes.com/docs/read/The_Semantic_API

3 News Stream

This section describes briefly the news collecting, indexing and querying services, developed at JSI which is available publically at http://newsfeed.ijs.si. The architecture where this component is deployed is also described in the next sub-section.

3.1 News Collection

The news aggregator is a piece of software developed at JSI which provides a real-time aggregated stream of textual news items provided by RSS-enabled news providers across the world. The pipeline performs the following main steps:

1) Periodically crawl a list of RSS feeds and a subset of Google News and obtain links to news articles
2) Downloads the articles, taking care not to overload any of the hosting servers
3) Parse each article to obtain
    a. Potential new RSS sources mentioned in the HTML, to be used in step (1)
    b. Cleartext version of the article body
4) Process articles with Enrycher [4] (at the moment for English articles only)
5) Expose two streams of news articles (cleartext and Enrycher-processed) to end users as a series of XML files.

3.1.1 System Architecture

The Figure 1**.** shows the architecture of the infrastructure of the JSI news feed component and how it is deployed to make it available externally via a web service API.

The first part of the aggregator is based around a PostgreSQL database running on a Linux server. The database contains a list of RSS feeds, which are periodically downloaded by the RSS monitoring component. RSS feeds contain a list of news article URLs and some associated metadata, such as tags, publication date, etc. Articles that are not already present in the database are added to a list of article URLs, and marked for download. Tags and publication date are also stored alongside, if found in the RSS.

A separate component periodically retrieves the list of new articles and fetches them from the web. The complete HTML is stored in the database, and simultaneously sent to a set of cleaning processes over a *0mq* message queue.

The cleaning process converts the HTML into UTF-8 encoding and updates the database with it. Then, it determines which part of the HTML contains the useful text, and discards the remainder and all of the tags. Finally, a naïve Bayes classifier is used to determine the primary language.

The cleaned version of the text is stored back in the database, and sent over a message queue to consumers.

Documents that are determined to be in English language are sent to the Enrycher web service, where named entities are extracted and resolved, and the entire document is categorized into a DMoz [5] topic hierarchy.

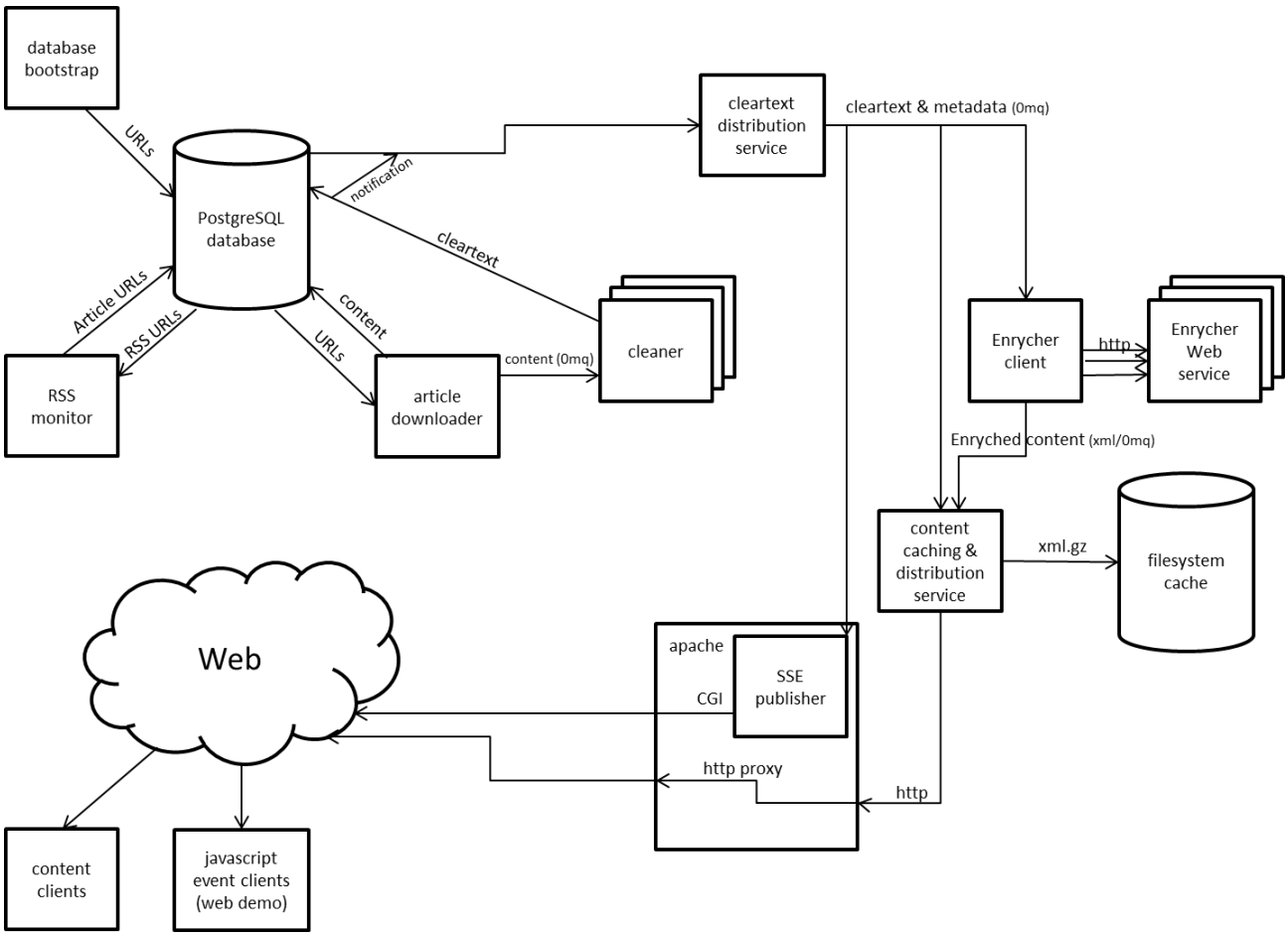

Figure 1 News aggregator system architecture

Both the cleartext and the enriched versions of documents are fed to a filesystem cache, which stores a sequence of compressed xml files, each containing a series of documents in the order they have arrived through the processing pipeline. The caching service exposes an HTTP interface to the world through an Apache transparent proxy, serving those compressed xml files on user request.

The Apache server also hosts a CGI process capable of generating HTML5 server-side events, which contains the article metadata and cleartext as payload. These events can be consumed using Javascripts EventSource object in a web browser.

3.1.2 Sources

We have accumulated about 200,000 RSS feeds, though not all of them are active, relevant or of sufficient quality. Currently, only about 3,400 are being actively monitored in order to keep the data quality high. The list of sources is constantly being changed – stale sources get removed automatically, new sources get added from crawled articles. In addition, we occasionally manually prune the list of sources using simple heuristics. The list was bootstrapped from publically available compilations, e.g. Kidon Media Link (www.kidon.com).

Besides the RSS feeds, we use Google News [6] as another source of articles. We periodically crawl the US English edition and a few other language editions, randomly chosen at each crawl. As news articles are later parsed for links to RSS feeds, this helps diversify our list of feeds while keeping the quality high.

The sources are not limited to any particular geography or language.

3.1.3 Language distribution

Automatic language detection of articles is still work in progress, so we can only give very rough preliminary estimates at this point, based on crude heuristics or manual inspection of samples of data.

We expect to cover some 50 languages. English is the most frequent with an estimated 50% of articles being in that language. The remaining major European languages (German, Spanish, Italian, French, and Russian) are expected to be represented by 3 to 10 percent of the articles. We cannot yet give a reliable estimate on the number of articles in "minor" languages, e.g. Slovene or Catalan.

3.1.4 Data volume

The crawler currently downloads 50,000-100,000 articles per day from across 100,000 different websites. The current archive contains about 25 million articles and begins in May 2008.

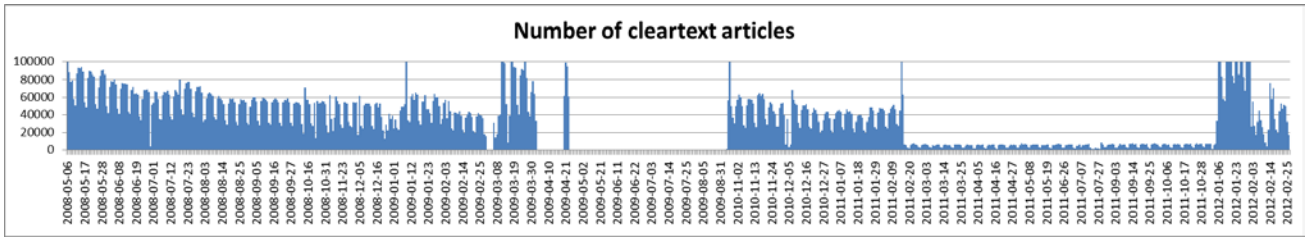

Figure 2. The daily number of downloaded articles.

A weekly pattern is nicely observable. Through most of 2011, only Google News was used as an article source, hence the significantly lower volume in that period.

The median and average article body length is 1,750 and 2,400 bytes, respectively.

3.1.5 Responsiveness

We poll the RSS feeds at varying time intervals from 5 minutes to 12 hours depending on the feed's past activity. Google News is crawled every two hours.

Based on articles where the RSS feed provides the original time of publication, we estimate 70% of articles are fully processed by our pipeline within 3 hours of being published, and 90% are processed within 12 hours.

3.1.6 Public Data and API

Two streams of news are made available:

1) The stream of all downloaded news articles
2) The stream of English articles, processed with Enrycher

3.1.6.1 Accessing the Data

Each stream of articles is serialized into XML and segmented by time into .gz files with several megabytes in size. The data is made available through a simple HTTP service that can be polled for new files periodically. Although this scheme introduces minor additional delays, it has the major benefit of keeping the API simple and easy to use.

To access the data, a request should be made to a URL of the form

`http://newsfeed.ijs.si/stream/`*`STREAM_NAME`*`?after=`*`TIMESTAMP`*

This returns the oldest available .gz file created after *TIMESTAMP* or HTTP error 404 if no recent enough file exists.

The `after` parameter is optional; if omitted, the oldest .gz file overall is returned. (We will attempt to maintain a month's worth of articles available through this API. The remainder of the archive is available to project partners on request.) *TIMESTAMP* must be given in the ISO 9601 format `yyyy-mm-ddThh:mm:ssZ` (Z and T are literals).

We also provide a python script which polls the server at one-minute intervals and copies new .gz files to the local disk as they are made available on the server. See http://newsfeed.ijs.si/.

Note: Due to the streaming nature of the pipeline, the articles in .gz files are only approximately chronologically sorted; they are sorted in the order in which they were processed rather than published.

3.1.6.2 Stream Serialization Format and Contents

Each .gz file contains a single XML tree. The root element, `<article-set>`, contains zero or more articles in the following format:

```
<article id="internal article ID; consistent across streams">
   <source-uri>URL from which the article was discovered; typically the RSS feed</source-uri>
   <source-title>Title for the source</source-title>
   <source-type>MAINSTREAM_NEWS</source-type>
   <article-lang>3-letter ISO 639-2 language code</article-lang>
   <article-uri>URL from which the article was downloaded</article-uri>
   <article-date approx="1 or 0">The publication time and date. If this was not made available
in the RSS feed, then the time and date when our crawler found the article. In the latter case, the attribute
approx will be "1". The value takes the yyyy-mm-ddThh:mm:ssZ format </article-date>
   <article-title>Title. Can be empty if we fail to identify it.</article-title>
   <article-body>See below.</article-body>
</article>
```

The `<article-body>` element contains:

- for the cleartext stream: cleartext body of the article, segmented into paragraphs with
- for the Enrycher stream: an XML subtree as returned by Enrycher. See documentation at http://enrycher.ijs.si/ for the exact format. Content-wise, the subtree contains document categorization into DMOZ, named entity detection and named entity resolution (i.e. entities are linked to DBpedia and YAGO).

3.2 NewsMiner

News Miner is a system for processing and indexing news corpora and is entirely based on the code base developed by JSI. At the moment, the system has adapters for two news feeds: News Collection service described in 3.1 or to Spinn3r feed [7]. Processing of a news feed contains the following steps, each being executed by an independent process:

1) Retrieve the articles from the news feed (e.g. using News Collection service's Python script),
2) Parse the article and prepare the fields for indexing (e.g. tokenizing text),
3) Add the article to the index.

Once the article is indexed, it can be accessed using query web service interface. In the remaining of this subsection provides descriptions of how the article is processed and what is the query API.

3.2.1 Indexed fields

Each article is indexed across several dimensions (facets) using inverted index. The system allows for retrieval of any Boolean combination of the facets, with only a limited subset being exposed through the API at the moment. The following list provides description of the facets:

- **Article Content** –words from the title or the body of the article
- **Article Title** – words from the title of the article
- **Article Body** – words from the body of the article
- **Article URI** – link to location from which the article was retrieved
- **Article Language** – language of the article, as identified by News Collection service (e.g. "*eng*")
- **Article Date** – date on which the article was crawled (e.g. "*2012-03-08*")
- **Article Year** – year on which the article was crawled (e.g. "*2012*")
- **Article Month** – year on which the article was crawled (e.g. "*2012*")
- **Article Day of Month** – month on which the article was crawled (e.g. "*March*")
- **Article Day of Week** – year on which the article was crawled (e.g. "*Thursday*")
- **Article Time of Day** – year on which the article was crawled (e.g. "*Afternoon*")
- **Article Hour** – year on which the article was crawled (e.g. "*17*")
- **Source URI** – link to location of the article source
- **Source Title** – title of the article source (e.g. "*New York Times*")
- **Source Type** – type of the article source (e.g. mainstream media)

The index will be further extended during the first year of the project to include entities, and other article meta-data, extracted by Enrycher and provided through News Collection service.

3.2.2 Query API

The index can be used to query collected news article through a web service. The service exposes a limited set of facets, provided using any combination of the following parameters:

- `q="slovenia hockey"` — retrieve all articles with keywords "*slovenia hockey*"
- `qt="slovenia hockey"` — retrieve all articles with keywords "*slovenia hockey*" in title
- `qb="slovenia hockey"` — retrieve all articles with keywords "*slovenia hockey*" in body
- `lang=eng` — retrieve all English articles
- `date=2012-03-16` — retrieve all articles published on 16th of March, 2012
- `offset=3` — used to retrieve more than first 100 articles; for example offset 3 returns articles from 301 to 400

Here is an example of a query, asking for all English articles containing keyword "*Slovenia*", crawled on 15th of March 2012: http://newsfeed.ijs.si/query/news-search?q=slovenia&lang=eng&date=2012-03-15.

The results are returned in XML format, similar to the format used by News Collection service. The following list summarizes the differences:

- The article IDs are independent from the IDs used by News Collection service
- Each article has a search rank attribute; rank is derived and assigned according to recentness, with the most recently crawled article from the result set having rank 0.

4 Conclusions

This deliverable has introduced the main source infrastructure to be used by the XLike project. This infrastructure will provide access to two main sources of data: corpuses needed for natural language processing tasks and streams of data to be analyzed (news and social media).

For that purpose, a first set of resources provided by all the partners is presented which are going to be used during the whole project. Regarding the access to streams of on-line data, a review of the JSI crawler service has been presented including its main characteristics, properties, and API.

Though it may be useful to have other crawlers or corpuses, the presented set is going to be the baseline and the main support for the completion of the project. If it would appear any other need during the development of the project which is not covered by the list presented here then a new component would be incorporated or developed.

References

[1] Wikipedia (http://www.wikipedia.org)

[2] New York Times Topics (http://www.nytimes.com/pages/topics/)

[3] Hudong Encyclopedia (http://www.hudong.com/)

[4] Enrycher (http://enrycher.ijs.si)

[5] DMoz (http://www.dmoz.org/)

[6] Google News (http://news.google.com/)

[7] Spinn3r (http://spinn3r.com/)

