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

XLike

Deliverable D1.1.1

Report and library on the existing technology and data

Editor:	Blaž Fortuna, JSI
Author(s):	Blaz Fortuna, JSI; Mitja Trampus, JSI; Blaz Novak, JSI; Esteban García-Cuesta, 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:	R
Dissemination Level: (Confidentiality)	Public (PU)
Contractual Delivery Date:	M3
Actual Delivery Date:	M3
Suggested Readers:	XLike project partners
Version:	1.0
Keywords:	Existing tools, existing datasets

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:	D1.1.1 – Report and library on the existing technology and data
Editor (Name, Affiliation)	Blaž Fortuna, JSI
Work package Leader (Name, affiliation)	Blaž Fortuna, JSI
Estimation of PM spent on the deliverable:	11

Executive Summary

This document presents a review of the different components and technology which is available to the XLike project. Therefore the main outcome of the deliverable is a collection of these components including its description, accessibility, availability, etc. This report is mainly split into two parts: the list of identified sources and data models, and the main components to analyze it and to provide the functionality needed by the project.

This document will be also used for the definition and design of other parts of the project. Among others the following tasks make use of this outcome: T1.3 and T6.1.

Table of Contents

List of Tables

Abbreviations

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
LOD	Linked Open Data
NLP	Natural Language Processing
ESA	Explicit Semantic Analysis
PoS	Part-of-Speech

1 Introduction

The purpose of this document is to gather, from all partners, comprehensive list of existing technology and data sources. The list will serve as an input for the requirements and specifications which are needed for the establishment of the XLike data infrastructure, Toolkit architecture and the definition of the functional interfaces between the different technologies.

The XLike project starts with a significant amount of pre-existing operational technology providing a good baseline for extending them during the project evolution. This document is going to focus on retrieving those existing elements regarding mainly with the next two issues:

- Existing data sources and data requirements
- Existing components and technical requirements

Based on these two points, an initial set of requirements and functionalities will be created in order to consolidate an overall architecture and functionality for the XLike project.

The overall template is structured into two main parts: data sources and components as described in the next sections.

This document will be used as source of information mainly for the M3 deliverables (D1.3.1, D6.1.1) though it also would be accessible and could be used by any partner for other purposes inside the XLike project.

1.1 Legend and Usage

Descriptive text is placed in square brackets [ ] in this template. The descriptive texts should be replaced in the final document.

Each section contains a table for describing the data sources and components. The goal is to add items to each table collecting as much information as possible from every partner in order to get a “big picture” of the available resources. All that information will be used for the architecture description and the early design of the XLike toolkit.

Each section also contains a legend section which explains each column purpose.

2 Data Sources

This section lists in Table1 data sources identified at the start of the project, which are available within the consortium. The table was collaboratively filled by consortium members.

Meaning of individual columns in Table1:

- **Data Entity**: name or identification of the data resource i.e. JSI news crawler
- **Data Responsible**: name of the institution or company responsible of 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 and covered languages:** 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. Languages which are represented in the data (when applicable). If possible, list languages.
- **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:** address of the web site which includes the documentation and information regarding the data source

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/tagged Corpues	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

Table 1 Data sources

3 Existing Components

This section includes the different components/modules which, at present, are available for the XLike project and can be provided by any of the partners of the consortium. This identification of components must also include the services which are going to be served by XLike platform or in which it is going to be based on.

The section consists of two lists. Table 2 provides a quick list of all the components. The components are then further described in Section 3.1.

Meaning of individual columns in Table 2:

- **Component**: name or identification of the component i.e. Enrycher
- **Responsible**: name of the institution or company responsible of the data source described, i.e. JSI
- **Sources availability:** type of availability of the source code yes/no/partly
- **How it is provided:** the type of accessibility i.e. library, web services, etc.
- **Programming Language:** type of language used by the components i.e. Java, C++, etc.
- **License:** type of license i.e. GNU/GPL, proprietary, etc.
- **Web site URL:** address of the web site which includes the documentation and information regarding the component

Table 2. List of software components.

Component	Responsible	Sources availability	How it is provided	Programming Language	License	Web site URL
Enrycher	JSI	Partly	Web Service	Java, C++	Proprietary	http://enrycher.ijs.si/
NewsMiner	JSI	No	Web Service	C++	Proprietary	http://newsfeed.ijs.si/
FreeLing	UPC	Yes	Library	C++	GNU GPLv3	http://nlp.lsi.upc.edu/freeling
Treeler	UPC	Yes	Library	C++	GNU GPLv3	http://treeler.lsi.upc.edu
Research-ESA	KIT	Yes	Library and Web Service	Java	GNU Lesser GPL	http://code.google.com/p/research-esa/
ETALIS	KIT	Yes	Library and Web Service	Prolog	GNU Lesser GPL	http://code.google.com/p/etalis/
DUSP	KIT	Yes	Library	Java	Modified BSD License.)	https://code.google.com/p/dusp/
Ontology-based annotation tool	KIT	Yes	Library	Java	NA	https://code.google.com/p/dusp/
Chinese Character Segmentation and Pos Tagging	THU	No	Library	Java	Proprietary	NA
Entity Linking to Knowledgebase	THU	No	Library	Java	Proprietary	NA
Keywords Extraction	THU	No	Library	Java	Proprietary	NA
Chinese News Classification	THU	No	Library	Java	Propritetary	NA
Chinese News Clustering	THU	No	Library	Java	Propritetary	NA
MSD tagger and lemmatizer	UZG	no	binaries	C++	proprietary	NA

			web service	C++/CLI		
NE recognizer and classifier	UZG	No	binaries web service	C++ C++/CLI	proprietary	NA
LanguageIdentifier	UZG	Yes	full source	Perl	Apache 2.0	NA
CollTerm	UZG	Yes	full source	Python	Apache 2.0	NA
Dependency parser	UZG	No	binaries web service	Java	proprietary	NA
ccExtractor	UZG	Yes	full source		Apache 2.0	NA
TwitterGrab	UZG	Yes	would be accessible and could be used by any partner for other purposes inside the XLike project		Apache 2.0	NA
Tools for Hindi wordnet and ontology based search	IIT Bombay	NA	NA	NA	NA	NA
Hindi wordnet linked with SUMO Ontology	ITT Bombay	NA	NA	NA	NA	NA
UNL dictionary for English and Hindi	IIT Bombay	NA	NA	NA	NA	NA
UNL corpora in agricultural domain	IIT Bombay	NA	NA	NA	NA	NA
Resources/Tools for sentiment analysis	IIT Bombay	NA	NA	NA	NA	NA
Suggested Upper Merged Ontology	IIT Bombay	NA	NA	NA	NA	NA
Universal Networking Language (UNL)	ITT Bombay	NA	NA	NA	NA	NA

3.1 Component Descriptions

3.1.1 Enrycher

<table>
<tr><td rowspan="8">Service</td><td>Name*</td><td>Enrycher</td></tr>
<tr><td>Description*</td><td>Enrycher is a service-oriented system, providing shallow as well as deep text processing functionality at the text document level. The system consists of two major components. First is the architecture, which is design to easily scale with respect to number of articles that can be processed in parallel. Second is a set of components, which perform particular tasks in the processing pipeline (e.g. part-of-speech tagging, named entity extraction). The output can be either in Enrycher-defined XML schema or in RDF.</td></tr>
<tr><td>Related component(s)</td><td>Any NLP components (e.g. Freeling), can be used to provide support for additional languages.</td></tr>
<tr><td>Constraints</td><td>The system imposes only a small overhead on the processing pipeline. The performance mostly depends on the processing components.</td></tr>
<tr><td>Dependencies</td><td>No external dependencies.</td></tr>
<tr><td>Bottlenecks</td><td>The system is developed as a combination of C++ and Java. It supports services in either of the two languages, and can combine them into a processing pipeline.</td></tr>
<tr><td>Security</td><td>No</td></tr>
<tr><td colspan="2">Operations</td><td>
<table>
<tr><td>Name*</td><td>Enrich</td></tr>
<tr><td>Description*</td><td>Passes plain text document through the processing pipeline.</td></tr>
<tr><td>Communication type</td><td>request/reply</td></tr>
<tr><td>Input message*</td><td>Plain text</td></tr>
<tr><td>Output message*</td><td>XML in Enrycher XML schema</td></tr>
</table>
<table>
<tr><td>Name*</td><td>Enrich RDF</td></tr>
<tr><td>Description*</td><td>Same as “Enrich”, but with RDF output</td></tr>
<tr><td>Communication type</td><td>request/reply</td></tr>
<tr><td>Input message*</td><td>Plain text</td></tr>
<tr><td>Output message*</td><td>RDF using standard LOD vocabulary.</td></tr>
</table>
</td></tr>
</table>

3.1.2 NewsMiner

<table>
<tr><td rowspan="7">Service</td><td>Name*</td><td>NewsMiner</td></tr>
<tr><td>Description*</td><td>System for processing and indexing news articles. The system can connect to article pipe from JSI news crawler or Spinn3r. Each article is sent through Enrycher service, to extract and resolve named entities and categories. Finally, the article is indexed for content, entities and sources. The system is exposes API providing search functionality over the articles.</td></tr>
<tr><td>Related component(s)</td><td>JSI News Crawler, Enrycher</td></tr>
<tr><td>Constraints</td><td>In current state can handle stream of mainstream articles from Spinner feed (crawls 10,000 major mainstream news sources) on a single workstation.</td></tr>
<tr><td>Dependencies</td><td>JSI News Crawler, Enrycher</td></tr>
<tr><td>Bottlenecks</td><td>Indexing is currently done in a single process. However, at the current state, it can handle the sources, which are planned to be used within XLike.</td></tr>
<tr><td>Security</td><td>No</td></tr>
<tr><td colspan="2">Operations</td><td>
<table>
<tr><td>Name*</td><td>Search</td></tr>
<tr><td>Description*</td><td>Search over news article corpus.</td></tr>
<tr><td>Communication type</td><td>request/reply</td></tr>
<tr><td>Input message*</td><td>Search query, specified in NewsMiner specific format.</td></tr>
<tr><td>Output message*</td><td>XML or JSon document listing matching articles.</td></tr>
</table>
<table>
<tr><td>Name*</td><td>Subscribe</td></tr>
<tr><td>Description*</td><td>Push service, for all or subset of matching articles. Article is sent forward as soon as it passes Enrycher stage in the pipeline.</td></tr>
<tr><td>Communication type</td><td>publish/subscribe</td></tr>
<tr><td>Input message*</td><td>Optional filter (same format as search query)</td></tr>
<tr><td>Output message*</td><td>XML or JSon document containing full article.</td></tr>
</table>
</td></tr>
</table>

3.1.3 FreeLing

Service	Name*	FreeLing
	Description*	Open-source C++ Library of Language Analyzers for building end-to-end NLP pipelines. A configuration for a usual NLP pipeline can be quite complex, and here we will hide most of the details. In essence, we will consider that an NLP pipeline consists of four main modules that are run in sequence: 1 – Tokenization; 2 – Tagging; 3 – Parsing; 4 – Extraction. For each language in XLike we will have one instantiation of such operation. Hence, for 6 languages we will have 24 types of Freeling modules.
	Related component(s)	Treeler – FreeLing uses Treeler to build statistical taggers and parsers.
	Constraints	Each module may need from 1 to 8 Gb of memory. Each module may need up to 2Gb of disk in order to store internal data. About running times, tokenization and tagging run at 500 words per second or faster; parsing modules run at 50 words per second or faster. These speeds may be improved depending on particular language, domain, and desired accuracy level.
	Dependencies	Input texts shall respect the data formats specified in XLike toolkit.
	Bottlenecks	Speed may be an issue in order to process large-scale collections. It is recommended to implement a system of parallelization: N identical instances shall be created; to process M documents, each instance should process M/N documents.
	Security	none

Operations

Name*	Tokenize
Description*	Segments and tokenizes a document into paragraphs, sentences and tokens. Performs basic spell checking.
Languages	English, Spanish, Catalan (for the rest of languages similar modules can be developed using FreeLing)
Communication type	
Input message*	A text in an appropriate format
Output message*	A data structure representing the document segmented into tokens.

Name*	Tag
Description*	Applies sequential tagging models to the sentences of a document, in order to predict PoS tags, named entities, and other annotations typically modelled in a sequential fashion.
Languages	English, Spanish, Catalan (for the rest of languages similar modules can be developed using FreeLing)
Communication type	
Input message*	A data structure representing a tokenized document.
Output message*	The same data structure augmented with tagging annotations and uncertainty values.

Name*	Parse
Description*	Applies parsing models to tagged sentences of a document, in order to predict syntactic parse tree, semantic roles, and other annotations typically modelled with probabilistic grammars.

	Languages	English, Spanish, Catalan (for the rest of languages similar modules can be developed using FreeLing)
	Communication type	
	Input message*	A data structure representing tagged document.
	Output message*	The same data structure augmented with parsing annotations and uncertainty values.

Name*	Extract
Description*	Applies a relation extraction module to a parsed document.
Languages	English, Spanish, Catalan (for the rest of languages similar modules can be developed using FreeLing)
Communication type	
Input message*	A data structure representing a document tagged and parsed.
Output message*	A set of relations, in the form of triples, each anchored to the passage in the document that realizes that relation.

3.1.4 Treeler

Service	Name*	Treeler
	Description*	An Open-source library of Structured Prediction Methods for NLP. It contains a number of statistical models and core algorithms that are used for tagging and parsing language. The configuration of a particular statistical method can be quite complex. Treeler will be basically used within the XLike Toolkit as part of FreeLing. Nonetheless, we want to note that FreeLing and Treeler are different software projects that complement each other.
	Related component(s)	FreeLing – many statistical modules in FreeLing use Treeler.
	Constraints	1 to 8 Gb of memory. Running time depends on model type, language, domain, and accuracy level that is required.
	Dependencies	None
	Bottlenecks	Time and memory consumption may be a bottleneck.
	Security	None
Operations		Operations of Treeler not specified in this document.

3.1.5 Research-ESA

[Research-ESA]Service		
	Name*	Research-ESA
	Description*	An implementation of Explicit Semantic Analysis for research
	Related component(s)	[Related components and the type of relation within the XLike project]
	Constraints	[Time constraints; computing constraints; etc.]
	Dependencies	TERRIER information retrieval framework
	Bottlenecks	[Description of possible problems regarding its integration within the XLike platform]
	Security	[Does the service need any kind of authentication?]

Operations

Name*	buildIndex
Description*	Build an inverted index of a Wikipedia Database provided in the original MediaWiki database schema
Languages	
Communication type	[request/reply or publish/subscribe]
Input message*	A Wikipedia Database provided in the original MediaWiki database schema
Output message*	Inverted index of the input Wikipedia Database

Name*	computeVector
Description*	Compute ESA vectors for any text
Languages	
Communication type	[request/reply or publish/subscribe]
Input message*	Any text
Output message*	ESA vectors of the input text

Name*	computeSimilarity
Description*	Compute Cosine Similarity of ESA vectors (which can be used as semantic similarity measure)
Languages	English, German and French
Communication type	[request/reply or publish/subscribe]
Input message*	ESA vectors
Output message*	Cosine similarity of the input ESA vectors

3.1.6 ETALIS

Service	Name*	ETAILS
	Description*	Event-driven Transaction Logic Inference System
	Related component(s)	[Related components and the type of relation within the XLike project]
	Constraints	[Time constraints; computing constraints; etc.]
	Dependencies	[Dependencies with other external components]
	Bottlenecks	[Description of possible problems regarding its integration within the XLike platform]
	Security	[Does the service need any kind of authentication?]
Operations		

Name*	detectComplexEvents
Description*	Detection of complex events and reasoning over states (with logic rules)
Languages	English
Communication type	[request/reply or publish/subscribe]
Input message*	Event stream
Output message*	Complex events and trends

Name*	eventProcessing
Description*	Support for Event Processing SPARQL (EP-SPARQL) language
Languages	
Communication type	[request/reply or publish/subscribe]
Input message*	EP-SPARQL query
Output message*	EP-SPARQL result

3.1.7 DUSP

Service	Name*	DUSP
	Description*	Distributed unsupervised parsing
	Related component(s)	[Related components and the type of relation within the XLike project]
	Constraints	[Time constraints; computing constraints; etc.]
	Dependencies	USP
	Bottlenecks	[Description of possible problems regarding its integration within the XLike platform]
	Security	[Does the service need any kind of authentication?]
Operations		

Name*	clusterExpressions
Description*	Learn target relations and objects which can be viewed as clusters of syntactic or lexical variations of the same meaning
Languages	English
Communication type	[request/reply or publish/subscribe]
Input message*	syntactic parsing tree of the texts
Output message*	clusters of syntactic or lexical variations of the same meaning

3.1.8 Ontology-based annotation tool

Service	Name*	Ontology-based annotation tool
	Description*	Semi-automatic Annotation
	Related component(s)	[Related components and the type of relation within the XLike project]
	Constraints	[Time constraints; computing constraints; etc.]
	Dependencies	Gate
	Bottlenecks	[Description of possible problems regarding its integration within the XLike platform]
	Security	[Does the service need any kind of authentication?]
Operations		

Name*	findSynonymicExpressions
Description*	Finding synonymic expressions for classes and relations in the Nanotechnology domain: statistical collocation and syntactical variation analysis of words
Languages	English
Communication type	[request/reply or publish/subscribe]
Input message*	An ontology and textual documents
Output message*	Annotated documents

3.1.9 Chinese Character Segmentation and Pos Tagging

Service	Name*	Chinese Character Segmentation and Pos Tagging
	Description*	Library for segmenting Chinese character and tagging Pos for articles
	Related component(s)	
	Constraints	
	Dependencies	This service depends on dictionary
	Bottlenecks	The discovery of new characters to update dictionary
	Security	User authentication required
Operations		

Name*	SplitWords
Description*	Segment Chinese character and tag Pos for text
Languages	Chinese
Communication type	no
Input message*	A string or text of an article
Output message*	Chinese segmentation and POS tagging results String

3.1.10 Entity Linking to Knowledgebase

Service	Name*	Entity Linking to Knowledgebase
	Description*	Library for giving linking result from a news content
	Related component(s)	Chinese Character Segmentation and Pos Tagging
	Constraints	
	Dependencies	News content & knowledge base
	Bottlenecks	The source site is down or not accessible
	Security	User authentication required
Operations		

Name*	Disambiguation
Description*	Gets disambiguated result from news
Languages	Chinese
Communication type	request/reply
Input message*	Pos tag result
Output message*	text message

3.1.11 Keywords Extraction

Service	Name*	Keywords Extraction
	Description*	Library for extracting keywords from Chinese article
	Related component(s)	Chinese Character Segmentation and Pos Tagging
	Constraints	The article cannot be too short
	Dependencies	Chinese Character Segmentation and the length of an article
	Bottlenecks	The length of an article
	Security	User authentication required
Operations		

Name*	getKeywords
Description*	Extract keywords from a text
Languages	Chinese
Communication type	
Input message*	A string or text of an article
Output message*	Keywords extracted from an article

3.1.12 Chinese News Classification

Service	Name*	Chinese News Classification
	Description*	Library for assigning a label for a newly added news text
	Related component(s)	Chinese Character Segmentation and Pos Tagging
	Constraints	Need to access the original data model files in real-time
	Dependencies	Word splitting dependent files
	Bottlenecks	The new appeared class may be ignored
	Constraints	No constraints
	Security	User authentication required
Operations		

Name*	assignClassLabel
Description*	Receive a news text and assign a label for it
Languages	Chinese
Communication type	no
Input message*	News literal text
Output message*	A class label

3.1.13 Chinese News Clustering

Service	Name*	Chinese News Clustering
	Description*	Library for automatically clustering a set of news
	Related component(s)	Chinese Character Segmentation and Pos Tagging
	Constraints	Need to access the original data model files in real-time
	Dependencies	Word splitting dependent files
	Bottlenecks	Need to determine the cluster number artificially or it would be too time consuming to determine the cluster number
	Constraints	No constraints
	Security	User authentication required
Operations		

Name*	clusterNewsSet
Description*	Receive a news set and segment them to several clusters
Languages	Chinese
Communication type	no
Input message*	News literal text set
Output message*	Several clusters of news text

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/)

