



RENDER
FP7-ICT-2009-5
Contract no.: 257790
www.render-project.eu

RENDER

Deliverable D3.3.2

Final version of the of diversity-aware ranking

Editor:	Maurice Grinberg, Ontotext
Author(s):	Maurice Grinberg, Ontotext; Alex Simov, Ontotext; Simo Simov, Ontotext; Andreas Thalhammer, UIBK; Ioan Toma, UIBK ; Simon Hangl, UIBK.
Deliverable Nature:	Prototype (P)
Dissemination Level: (Confidentiality)	Public (PU)
Contractual Delivery Date:	March 2013
Actual Delivery Date:	March 2013
Suggested Readers:	Researchers and practitioners in the Linked Data and NLP fields
Version:	1.0
Keywords:	Diversity-aware ranking; clustering; spreading activation

Disclaimer

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

In case of Public (PU):

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

In case of Restricted to Programme (PP):

All RENDER consortium parties have agreed to make this document available on request to other framework programme participants.

In case of Restricted to Group (RE):

The information contained in this document is the proprietary confidential information of the RENDER consortium and may not be disclosed except in accordance with the consortium agreement. However, all RENDER consortium parties have agreed to make this document available to <group> / <purpose>.

In case of Consortium confidential (CO):

The information contained in this document is the proprietary confidential information of the RENDER consortium and may not be disclosed except in accordance with the consortium agreement.

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

Neither the RENDER consortium as a whole, nor a certain party of the RENDER 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:	RENDER – Reflecting Knowledge Diversity
Short Project Title:	RENDER
Number and Title of Work package:	WP3 Diversity representation and processing
Document Title:	D3.3.2 - Final version of the of diversity-aware ranking
Editor (Name, Affiliation)	Maurice Grinberg, Ontotext
Work package Leader (Name, affiliation)	Andreas Thalhammer, UIBK

Copyright notice

© 2010-2013 Participants in project RENDER

Executive Summary

This deliverable builds on the results and the ideas of D3.3.1 Prototype of diversity-aware ranking. It presents the Diversity-aware ranking service developed by UIBK and CLAS OWLIM plug-in developed by Ontotext.

The Diversity-aware ranking service is a middleware application using OWLIM and Sesame triple stores as a data layer and providing ranked data as JSON output based on effective and efficient ranking and clustering algorithms. The application allows for compact representation and visualization of large amounts of posts, news articles or tweets. The deliverable presents in brief the methods used (presented in detail in D3.3.1) and provides details about the current implementation and its use in RENDER and its evaluation.

The CLAS plug-in has been designed to provide OWLIM with machine learning capabilities that complement its formal reasoning capabilities. It can be accessed via a SPARQL endpoint and aims at combining the power of structured reasoning and statistical methods like spreading activation and clustering on graphs. The main principles of CLAS are extendibility and modularity. CLAS tools can be managed by a common SPARQL interface which allows the generation of sophisticated datasets extracted from large RDF triple stores. Such datasets can be further processed using the modules of CLAS and the results used for further data querying. In this deliverable, we demonstrate and evaluate the potential of CLAS, on a corpus of Google news articles, using the clustering library CLUTO and some of the approximate spreading activation methods presented in D3.3.1.

In this deliverable, the CLAS tool is presented in detail with examples demonstrating some of its capabilities and its potential to be used by a front end application. The deliverable presents further steps of improvement of CLAS tools performance for the Google news use case based on richer semantic annotation of news articles.

Table of Contents

Executive Summary	3
Table of Contents	4
List of Figures.....	5
List of Tables.....	5
Abbreviations.....	7
1 Introduction	8
2 CLAS Tools for Graph Based Ranking	9
2.1 Basic features of the CLAS plug-in	9
2.2 Using the CLAS via SPARQL	14
CLAS is accessible via:.....	14
2.2.1 Dataset generation	14
2.2.2 Clustering.....	14
2.2.3 Spreading Activation.....	16
2.2.4 CLAS plug-in configuration.....	17
2.3 CLAS Plug-in: Google News Articles and DBpedia Entities.....	18
2.3.1 Dataset Generation.....	19
2.3.2 Clustering.....	20
2.3.3 Spreading activation and within cluster similarity	23
2.4 Discussion and Future Work	27
3 The Diversity-Aware Ranking Service.....	29
3.1 Recapitulation of the similarity measures	30
3.2 Clustering algorithms in use.....	31
3.3 Description of the input parameters	33
3.4 Description of the output parameters.....	33
3.5 Evaluation of the Diversity-Aware Ranking Service	34
3.6 Presentation of the survey results.....	37
3.7 Discussion of the results	41
4 Conclusion and Discussion	43
References.....	44
Annex A CLAS Plug-in Example: Google News Corpus Analysis and Datasets Generation	45
A.1 Statistics about dmoz labels (sioc:tag).....	45
A.2 Statistics about dmoz topics (sioc:topic)	46
A.3 News with DBpedia URIs attached.....	47
Annex B CLAS Plug-in: Diversity-aware Ranking of Google News Articles Based on dmoz Labels (sioc:tag)	50
B.1 Dataset generation	50
Annex C Node Selection Based Spreading Activation (NSbSA)	54
C.1 Standard SA.....	54
C.2 SA as Non-Zero Elements Search	54
C.3 NSbSA Algorithm.....	55

List of Figures

Figure 1: CLAS in the Render technical architecture.	9
Figure 2: Schematic interaction of the CLAS plug-in with OWLIM via a SPARQL end point.....	10
Figure 3: CLAS tools workflow.	11
Figure 4: Workflow of the Clustering CLAS tool.	12
Figure 5: Workflow of the SA CLAS tool.	13
Figure 6: RDF representation for Google news documents [7].....	19
Figure 7: HTML 5 Web interface of the Diversity-Aware Ranking Service	29
Figure 8: The FOLDING algorithm (Source: [1])	31
Figure 9: The MAXIMUM algorithm (Source: [1])	32
Figure 10: Screenshot of the Web interface of that visualizes the documents and their clusters.	35
Figure 11: Frequency of the number DBpedia entity per news article.	48
Figure 12: Frequency of the number news articles per DBpedia entity.....	49

List of Tables

Table 1: Parameters controlling the behaviour of the CLAS plug-in.	18
Table 2: Modified breadth-first-search algorithm implementing NSbSA according to Eq. (2).	56

Abbreviations

CLAS	CLustering and Activation Spreading plug-in to OWLIM
CLUTO	A Clustering Toolkit, http://www.cs.umn.edu/~karypis/cluto/
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
KDO	Knowledge Diversity Ontology
NER	Named Entity Recognition
REST	Representational State Transfer
RDF	Resource Description Framework
RKS	Reference Knowledge Stack
SA	Spreading Activation
SIOC	Semantically-Interlinked Online Communities
SPARQL`	SPARQL Protocol and RDF Query Language
URL	Uniform Resource Locator
URI	Uniform Resource Indienticator
XML	Extensible Markup Language

1 Introduction

This deliverable presents the final version of the tools developed in the project for diversity aware ranking. It builds on the approaches and results presented in D3.3.1 “Prototype of diversity-aware ranking” [9]. The tools described here are available for use in Render use-cases. The deliverable describes two applications: the CLAS plug-in to OWLIM (sometimes referred to as CLAS) developed by Ontotext and the Diversity-aware ranking service developed by UIBK.

CLAS is a set of tools accessible as OWLIM plug-in which provide clustering and spreading activation capabilities. It can be accessed via a SPARQL endpoint, together with any other functionality of OWLIM. SPARQL¹ queries allow generating sophisticated datasets based on large RDF triple stores that augments the processing capabilities of CLAS. CLAS is extendible and modular by design and can combine various clustering algorithms, external clustering, and spreading activation. In this deliverable, we demonstrate and evaluate the potential of the CLAS on a corpus of Google news articles, using the clustering library CLUTO [6] and some of the approximate spreading activation methods presented in D3.3.1 [9]. The deliverable presents also a plan for further development of CLAS based diversity-aware ranking, related to the Google news use-case in Render which will be carried out in the remaining months of the project and will be reported in the respective use-case deliverables.

The Diversity-aware ranking service is a middleware application introduced in D3.3.1 [9]. It can use OWLIM and Sesame triple stores as a data layer and provides ranked data as JSON output based on effective and efficient ranking and clustering algorithms. The applications allows for compact representation and visualization of larger amounts of posts, news articles or tweets. The deliverable recapitulates briefly the methods used (described in detail in D3.3.1) and provides additional information about the current implementation and its use in RENDER. The results from a user evaluation are also presented and analyzed.

The deliverable has the following structure:

In Section 2, the CLAS plug-in implementation is described in detail with several detailed examples which illustrate the typical use of the proposed tools. Ideas of improvement of the functionality of the CLAS plug-in based on improved entity extraction from text are also presented.

In Section 3, the final version of the Diversity-Aware Ranking Service is described. Based on background information about the main approach and algorithms used, the last developments with examples and evaluation are presented.

In Section 4, the overall status of the diversity-aware ranking services is described and options for their integrated use in Render are discussed.

In Annex A and Annex B, additional examples of dataset extraction and clustering and spreading activation are given. In Annex C, information about the SA algorithm used is given.

¹ <http://www.w3.org/TR/2013/REC-sparql11-query-20130321/>

2 CLAS Tools for Graph Based Ranking

This section describes the implementation of a diversity-aware ranking approach based on spreading activation (SA) and clustering techniques [9]. The developments presented here build on work done in LarkC EC project [16][39] and present the implementation of a working OWLIM plug-in, called CLAS (standing for CLustering and Activation Spreading).

2.1 Basic features of the CLAS plug-in

CLAS is aimed to be part of the functionality provided by OWLIM for the project and a set of tools available for higher level components to use in specific scenarios. With respect to the Render technical architecture CLAS is situated as shown in Figure 1.

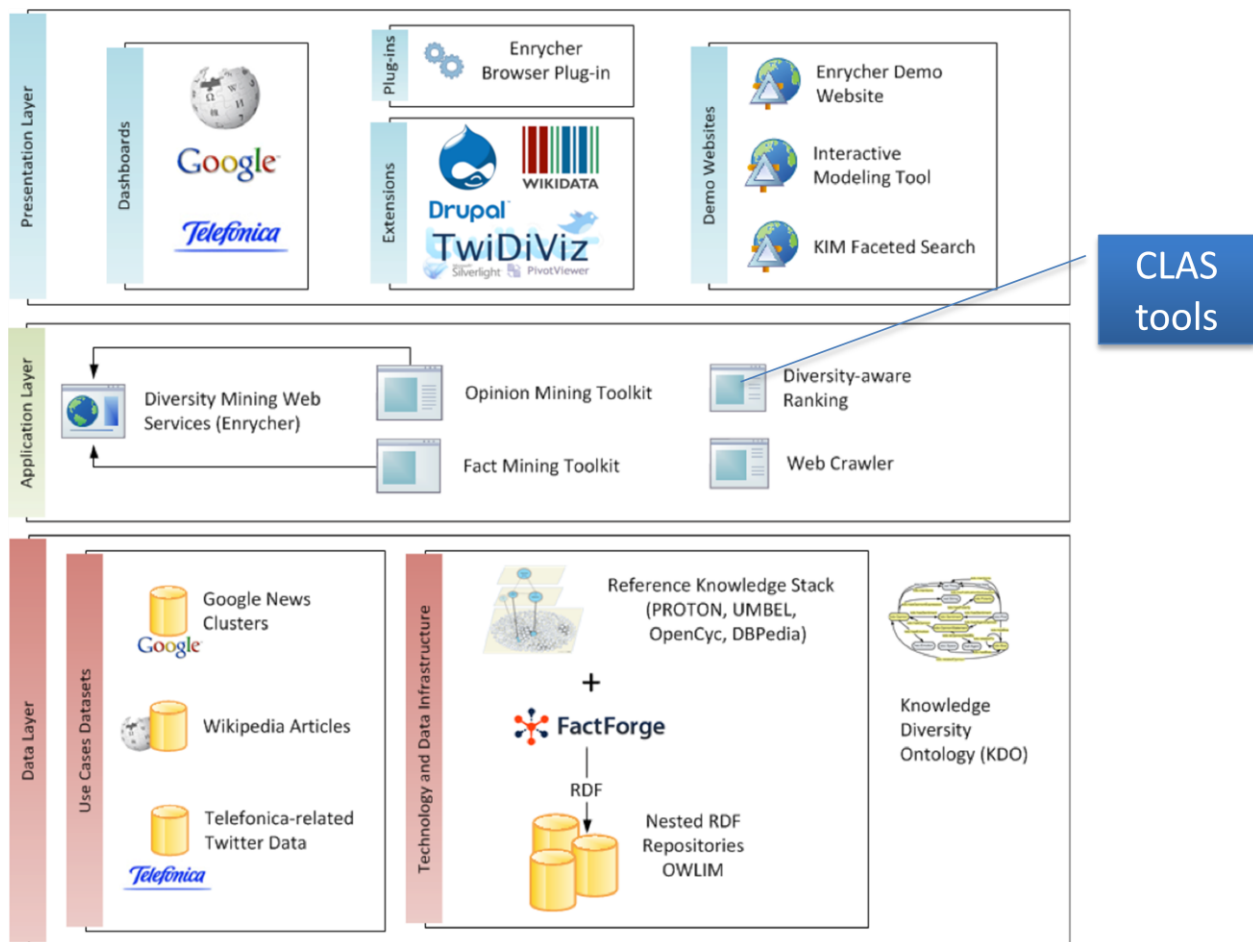


Figure 1: CLAS in the Render technical architecture.

The CLAS components and their connection to OWLIM are schematically represented in Figure 2. The CLAS plug-in integrates the functionality of OWLIM, with access to data via a SPARQL end-point, with the tools of CLAS. The latter use numerical information exported from OWLIM and can be distributed over several physical machines. CLAS can be extended with additional clustering algorithms or spreading activation (SA) mechanisms.

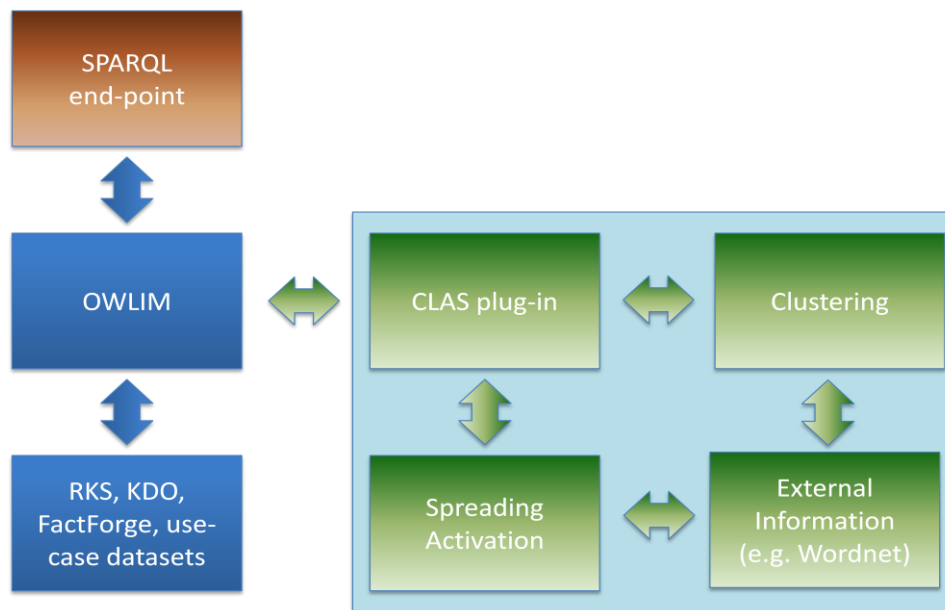


Figure 2: Schematic interaction of the CLAS plug-in with OWLIM via a SPARQL end point.

The CLAS plug-in is characterized by the following design principles and features:

- **Single SPARQL endpoint for all data and tools**
 - Allows to use the power of SPARQL for sophisticated data extraction for the CLAS plug-in;
 - Predefined syntactic structures for components interaction;
 - System predicates for meta-data retrieval;
 - Example: a matrix with the news articles as rows and topics and entities as columns can be generated and then used for clustering the news articles and extract their typicality and rank them accordingly with a few SPARQL queries which can be combined in a R or Java program (see Section 2.2 for examples).
- **Flexibility and extensibility**
 - Loosely coupled components which can interact via the CLAS plug-in;
 - Highly configurable environment which allows some of the components to work on distributed architectures;
 - Combination of off-line and on-line tools;
 - Example: CLAS uses CLUTO [6] as a clustering engine and can have access to all of its algorithms and options. Similarly, CLAS has a fast approximate SA algorithm (see Annex C and [4]) which can use a matrix of connections or a matrix of similarities obtained by the clustering module. CLUTO is installed on a separate computer while SA is installed on the same computer as OWLIM.
- **Computational efficiency**
 - Best numerical format data representation (e.g. for large sparse matrices);
 - Best numerical algorithms (e.g. specialized libraries for large data clustering);
 - Appropriate hardware (e.g. GPU devices);
 - Example: CLAS can use an implementation of SA for CUDA device [4] using sparse matrices stored in CSR (compressed sparse row format) which allows using matrices with millions of rows.

These design principles are illustrated by the workflow diagram for CLAS shown in Figure 3. In Figure 3, the data to be processed and additional datasets are positioned on the top of the diagram. Then, processing can go by a SPARQL query, following the right-hand side sequence boxes in Figure 3. On the other hand, it is possible to extract and organize data following the left hand side of Figure 3. Extraction of data can be done by a SPARQL query which is the default option. In some cases however, the data extracted from a dataset by a SPARQL query may need further processing (e.g. feature selection) or the dataset is to be extracted from a text corpus. In such cases, for providing the data needed by CLAS external tools can be run independently of CLAS. How CLAS is used via SPARQL queries will be explained in detail in the next sections.

For efficiency reasons, CLAS uses numerical data (graphs, matrices, etc.) in suitable formats. Any data extracted from the datasets via a SPARQL query or otherwise is exported in numerical format. Once the off-line pre-processing carried out, its results can be used in standard SPARQL queries. The two main tools, implemented so far are clustering and spreading activation. Typically, they can cluster or SA given a source of activation (called ‘seed’) using a sub-graph extracted from the original data or/and datasets. They can also interact by using the results from clustering in SA (e.g. by using the similarity or typicality matrices as weight matrices).

The typical use workflow is as follows. Let’s assume that all preliminary processing has been done off-line and a SPARQL query has been submitted to SPARQL end-point. The results of the query can be augmented by additional information available in CLAS, e.g. cluster ids, typicality (closeness to the centroid of the cluster), similarity to a prespecified entity, etc. By using this additional information the results can be presented in the best way to the user.

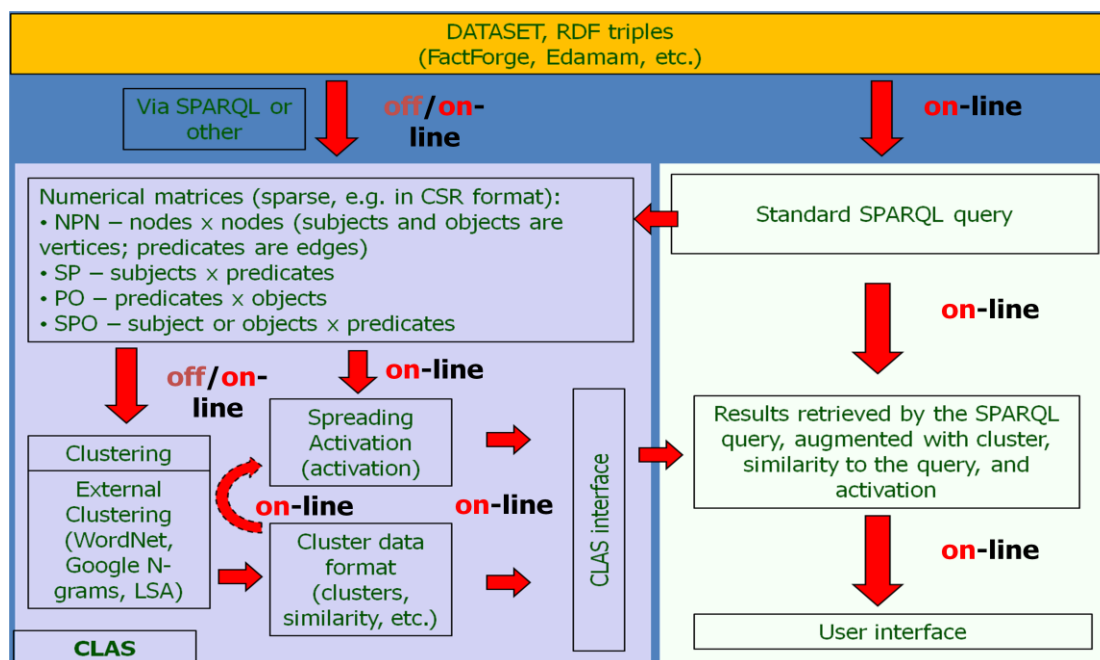


Figure 3: CLAS tools workflow.

It is important to stress how CLAS can provide information related to diversity which is central to the Render project. Concerning clustering, diversity can be related to distance from to the centroid for a set of items. Another related measure would be dissimilarity with respect to an item. Spreading activation can be used to assess similarity based on co-activation of feature or to map entities over a larger dataset by using graph vertices as connections and thus amplify existing differences. The level of activation of related entities can be used as features’ weights and allow for the expression of fine grained diversity and feature selection.

To summarize, the three main functional groups in the CLAS plug-in are:

- **Dataset extraction.** In order to be useful, clustering and SA should be based on relevant data structures. In the present implementation both operate on a matrix which is extracted from the original use-case dataset, i.e. the corpus of Google news articles and resources like e.g. DBpedia or

RKS, and which contains the connection of a set of items to a set of features. The connection weights (taken to be equal to 1 for the examples) can be used for SA or clustering. The dataset is usually generated by a SPARQL query based on the available datasets and could in principle combine them in any meaningful way (see Section 2.3.1). It uses the capabilities of OWLIM plug-ins and specific predicates and parameters used via SPARQL. If needed, datasets can be generated by external tools and stored in appropriate formats and made available to CLAS.

- Clustering module(s).** In the present implementation, the clustering is carried out by using the CLUTO (Clustering tool for high-dimensional datasets) library [6][7] and additional feature dimensionality reduction based on feature clustering. In the deliverable, as the clustering itself is not an issue, the default options of CLUTO 'vcluster' procedure are used which are described in detail in [6][7]. In general, any clustering library can be added and adapted to CLAS API. It is implemented in Java. The workflow process is shown in Figure 4.

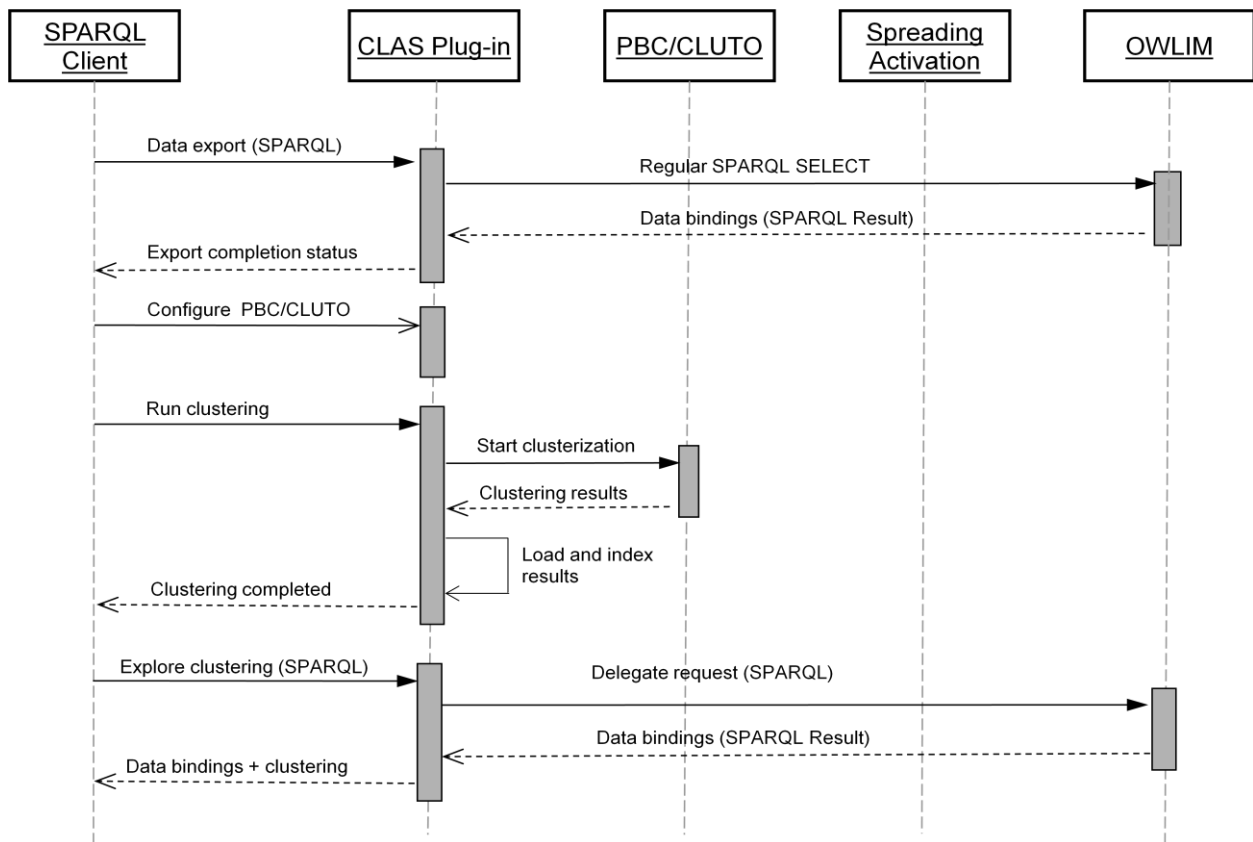


Figure 4: Workflow of the Clustering CLAS tool.

- Spreading activation module(s).** The present SA module implements a highly parallelizable graph based approximate SA (see Annex C and [9]). It is integrated with the clustering module and can use the results coming clustering like distances to the centroids, similarity matrix (if any due to storage limitations) as weight matrices. Multicore and CUDA (GPU) support is also provided. It is implemented in C++ wrapped with Java JNI. The workflow process is shown in Figure 5.

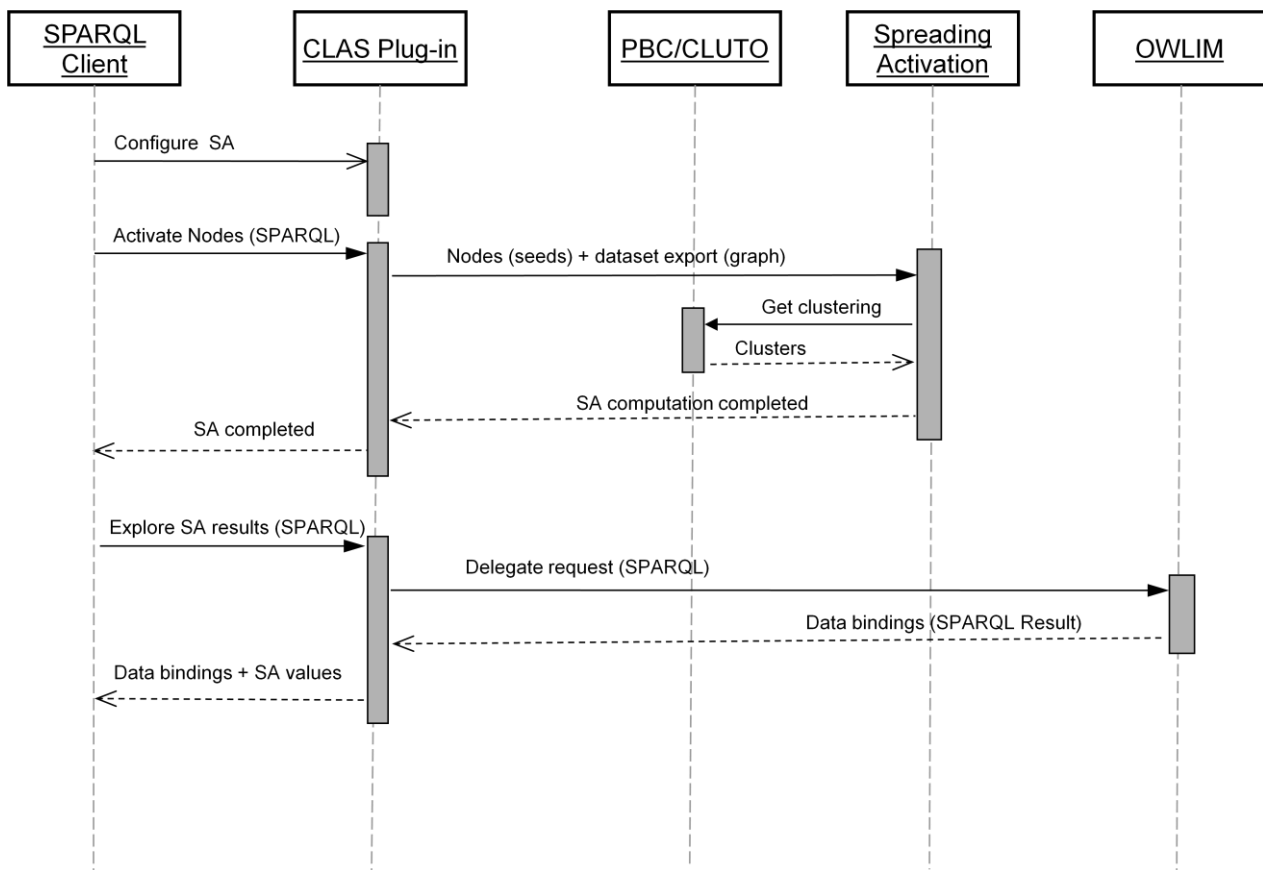


Figure 5: Workflow of the SA CLAS tool.

The general presentation of CLAS, presented in this section, aimed at presenting its design principles and implementation. To clarify how CLAS can be used in practice and give a more specific background for the extensive examples presented in the next section, a simple usage example can be given.

In the Google news use case, a corpus of about 300 000 news articles is available at <http://rendernews.ontotext.com>. The typical SPARQL queries which can be used to analyze the data and extract selected parts of it would ask for articles having specific topics or sentiment by using for instance predicates from KDO. On the other hand, we may be interested in cluster of news articles using as features the DBpedia² entities extracted for each news article. CLAS allows to cluster the news articles using any set of features available in the corpus enriched by information available in the dataset like DBpedia and moreover to perform several clusterings with different combinations of features allowing the grouping of news articles from various perspectives. If one is interested in specific news articles, the SA module allows to activate related news articles, once again based on generated ‘weight’ matrices, which in the present implementation of CLAS are the matrices used for clustering. Once again, each such matrix gives a specific perspective on the data, and their combination given a usage context, can exhibit non-trivial associations in the news datasets. For instance, using a news article as a source of activation (‘seed’), one can activate its selected features (e.g. topics or DBpedia entities) and in a second iteration activate all the other news articles having some of these features with a level of activation proportional to the number of features they share with the seed article. In some cases, e.g. when very large datasets have to be processed or a processing not available in CLAS must be carried out, additional tools can be used. Such tools can be any library or application for data analysis, clustering, etc. which used off-line can provide data in the formats usable by CLAS. For instance, one may want to apply a vector space model using the news articles as documents and apply LSA like procedure to generate a similarity matrix for them.

² <http://dbpedia.org>

2.2 Using the CLAS via SPARQL

CLAS is accessible via:

- SPARQL endpoint: [http://rendernews.ontotext.com/sparql.json?query=;](http://rendernews.ontotext.com/sparql.json?query=)
- Forest UI: <http://rendernews.ontotext.com/sparql>.

In the next sections, the presentation of CLAS and the related examples, we will assume the usage of the prefixes given in SPARQL 1 when appropriate.

SPARQL 1: Definition of prefixes for the CLAS plug-in.

```
PREFIX clas: <http://www.ontotext.com/owlim/plugin/CLAS#>
PREFIX kdo: <http://kdo.render-project.eu/kdo#>
PREFIX pkm: <http://www.ontotext.com/proton/protonkm#>
PREFIX sioc: <http://rdfs.org/sioc/ns#>
PREFIX dbp-prop: <http://dbpedia.org/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

Additional examples and demonstration of the capabilities of CLAS are given in Annex A (Google news corpus analysis and dataset generation) and Annex B (diversity-aware ranking based on `dmoz` labels).

2.2.1 Dataset generation

CLAS can use any dataset generated with a SPARQL query. This allows for the selection of relevant information from the data store for clustering and SA calculation.

The SPARQL query can be an arbitrarily complex `SELECT` query projecting two variables – `?s` and `?p` defining an edge or item-feature relation. Note that the names of the variables must be exactly as indicated. Any projection variables with different names are ignored. In the `SELECT` query, the following dataset specification contains information about the present installation of CLAS should be used:

```
FROM <http://www.ontotext.com/owlim/plugin/CLAS#dump>.
```

A single binding set is returned, containing extraction completion status (success and summary, or error and message). It is important to select JSON or XML download from the SPARQL end-point interface as the generated dataset is downloaded as a side effect of this selection. The resulting file is stored in a pre-specified directory and in an appropriate format for further processing by a clustering algorithm or SA.

The following example extracts all the pairs of news-article-dmoz-label (`sioc:tag`):

SPARQL 2: CLAS dataset generation based on `dmoz` labels.

```
SELECT ?s ?p
FROM <http://www.ontotext.com/owlim/plugin/CLAS#dump>
WHERE { ?s a kdo:NewsArticle ;
        sioc:tag ?p }
```

2.2.2 Clustering

After a dataset has been generated, clustering is performed by using an `ASK` query:

SPARQL 3: CLAS clustering command.

```
ASK { [] clas:doClustering "[command]" }
```

Where "command" is an optional parameter, interpreted by the clustering module (the value to be chosen in the present implementation is "direct"). When an empty string is given, the system default will be used. In the present version of CLAS, only one dataset can be used for both clustering and SA. This limitation will be removed in future versions.

Example: Performing clustering with the option "direct".

SPARQL 4: CLAS "direct" clustering command.

```
ASK { [] clas:doClustering "direct" }
```

Querying for clustering information (e.g. the cluster number), is done by using the 'magic' predicate:
clas:inCluster.

Example: Retrieving news articles and their respective clusters (one news article can belong to only one cluster).

SPARQL 5: CLAS cluster information retrieval: news and cluster.

```
SELECT ?news ?cluster
WHERE { ?news a kdo:NewsArticle .
        clas:inCluster ?cluster }
```

It is possible to get information about the typicality of a cluster element with respect to all elements by using the CLUTO's so-called z-score (see [6], [7] for details). For each cluster element, its z-score is the ratio of average similarity of this element to the other cluster elements, divided by the average of the same quantity for all cluster elements, and then z-transformed [6]. This gives a distribution with mean 0 and standard deviation 1. The higher the z-score of an element is, the more typical or close to the cluster centroid it is.

The z-scores given by CLUTO after clustering can be retrieved by the 'magic' predicate:

```
clas:hasClusterDistance.
```

Example: Retrieval of news, their respective cluster, and their typicality with respect to its elements.

SPARQL 6: CLAS cluster information retrieval: news, cluster and typicality.

```
SELECT ?news ?cluster ?typicality
WHERE { ?news a kdo:NewsArticle ;
        clas:inCluster ?cluster ;
        clas:hasClusterDistance ?typicality }
```

Additionally, for a selected cluster element, the similarity ('cosine' metric) between it and the cluster elements of its cluster can be calculated, using an ASK query with the 'magic' predicate:

```
clas:setClusterFocus
```

And the URI of the selected cluster member as follows:

SPARQL 7: CLAS cluster information retrieval: similarity of a selected cluster element to the other element in the cluster ('cosine' metric).

```
ASK { [] clas:setClusterFocus URI }
```

Example: Calculate the similarities between the news article <urn:document-news-0026c81c-e6b9-4bd0-bb60-59bf77650511> and the other members of its cluster.

SPARQL 8: CLAS cluster information retrieval: similarity of a selected news article to the other news articles in its the cluster ('cosine' metric).

```
ASK { [] clas:setClusterFocus
      <urn:document-news-0026c81c-e6b9-4bd0-bb60-59bf77650511> }
```

The results can be retrieved by using the 'magic' predicate: `clas:hasClusterFocusDistance`.

Example: Retrieval of news articles from a cluster containing a URI of interest and their respective similarity to it.

SPARQL 9: CLAS cluster information retrieval: similarity of a selected news article to the other news articles in its the cluster ('cosine' metric).

```
SELECT ?news ?similarity
WHERE { ?news a kdo:NewsArticle ;
         clas:hasClusterFocusDistance ?similarity
        FILTER ( ?similarity > "0.5"^^xsd:double )
      }
order by DESC(?similarity)
```

2.2.3 Spreading Activation

The SA starts by using selected URIs called seeds of activation. Some details about the approximate SA algorithm used are given in [4], [5] and in Annex C. In the present implementation of the CLAS tools the SA module uses the dataset generated for clustering. The seed nodes are selected and spreading activation takes place with a `SELECT` or `ASK` query containing the 'magic' predicate:

SPARQL 10: CLAS SA: defining seed for and running SA.

```
ASK { [] clas:activateNode node_1 .
      [] clas:activateNode node_2 .
      ...
      [] clas:activateNode node_n . },
```

In SPARQL 10, `node_1`,...,`node_n` must not be variables (no evaluation take place in the present implementation of CLAS) but should be strings corresponding to valid URIs (this feature will be made more flexible, allowing the usage and evaluation of variables).

The resulting activation can be retrieved by the 'magic' predicate: `clas:hasActivity`.

The following specifics have to be taken into account:

- Statement pattern objects **should** be concrete URI values. No real evaluation is performed here;
- If the query is composed of useful request and nodes activation specification, the latter should be wrapped in `OPTIONAL` clause to avoid empty result set joins;
- Seeds collection and activation calculation is done prior to actual query evaluation, so the activation values can be used in the same query.

Example: Selecting seeds and running the activation process ("false" response).

SPARQL 11: CLAS SA: specifying news articles as seeds and running SA.

```
ASK {
  [] clas:activateNode <urn:document-news-0026c81c-e6b9-4bd0-bb60-59bf77650511> .
  [] clas:activateNode <urn:document-news-4add75d3-7bd1-47fb-bc32-8cfd5a52d177> .
  [] clas:activateNode <urn:document-news-3df972a5-3ccd-4311-a320-208d5eacacfb>
}
```

Example: Retrieval of news articles and their activation values.

SPARQL 12: CLAS SA: extracting the activities of news articles based on the SA seeds from SPARQL 11.

```
SELECT ?news ?activity
WHERE { ?news a kdo:NewsArticle ;
          clas:hasActivity ?activity }
```

2.2.4 CLAS plug-in configuration

The plug-in maintains a global configuration table (see Table 1), containing specific configuration settings for the modules and integration parameters like: working directories, files, run-time libraries, etc. It is an open set of properties which can be manipulated by SPARQL queries.

The plug-in itself doesn't validate the parameters being set. It only provides means for **setting** and **reading** the values remotely. It is the responsibility of each sub-module to find its relevant properties.

Parameter names are specified as URIs having the namespace:

'http://www.ontotext.com/owlim/plugin/CLAS#'.

Parameters are set by using an ASK query with the 'magic' predicate: `clas:ParamValue`:

SPARQL 13: CLAS: setting the values of CLAS parameters.

```
ASK { clas:myparam clas:hasParamValue "value of the parameter" }
```

Parameter values can be retrieved by a `SELECT` query with the 'magic' predicate: `clas:hasParamValue`:

SPARQL 14: CLAS: getting the values of CLAS parameters.

```
SELECT ?val
WHERE { clas:myparam clas:hasParamValue ?val }
```

To list the whole configuration the following query must be used:

SPARQL 15: CLAS: getting the values of all CLAS parameters.

```
SELECT ?param ?val
WHERE { ?param clas:hasParamValue ?val }.
```

The parameters available in the present version of CLAS are described in Table 1.

Table 1: Parameters controlling the behaviour of the CLAS plug-in.

Parameter	Component	Default Value	Description
d	Clustering	OWLIM plug-ins default storage location + "/dump"	Server side file system directory where the clustering module stores its outputs
g	Clustering	<plug-in storage> + "dump/sp.edges"	Input file for clustering (as sparse numeric matrix), produced by the dataset extraction module
N	Clustering	1000	Number of result clusters
L	Clustering	e.g. "-zscore -showtree"	CLUTO command line parameters [6]
c	Clustering	All	Command, to be sent to the Clustering module, related to clustering strategy
i	Clustering	graph	Clustering input format
B	Clustering	-	Location of CLUTO binary files
saUseClustering	SA	false	Sets if the SA tool should ('true') or should not ('false') use results from clustering
saIterations	SA	2	Number of SA iterations
saMethod	SA	simple	Use approximate ('simple') or other SA method (not implemented)
saUseSymmetric Links	SA	False	Only symmetric connections are used in the present implementation!

2.3 CLAS Plug-in: Google News Articles and DBpedia Entities

In this deliverable, the basic potential of CLAS is demonstrated on a corpus of Google news items, provided by Google as part of the Google use case in Render. The usage scenario for this use-case is to start from a selected news and extract information about what this news is related to, what is mentioned in it, which other news from the corpus are similar to it and what is the typicality of the selected news article with respect to them. In the following subsections, the parameters and functionality of CLAS will be presented paying special attention to three main mechanisms – dataset extraction, clustering, spreading activation, and combination of the results.

The Dataset Extraction Tool extracts data for the Clustering and SA components. This enables flexible and powerful sub-dataset selection for further processing by the CLAS tools. In the present version, the data can be clustered by using the CLUTO library providing typicality and hierarchical cluster information [6], given a seed from the dataset approximate spreading activation can be performed, both based on the graph information and on hierarchical cluster similarity matrix. Additionally, distances within a pre-specified cluster can be extracted and a feature number reduction tool is provided (not presented in the deliverable).

An important feature of CLAS tools is their accessibility via SPARQL endpoint. This allows selecting of relevant information from the data store for clustering and SA calculation. In this subsection several examples of data analysis will be shown on the corpus of Google news located at <http://rendernews.ontotext.com>.

The data model behind the data is shown in Figure 6 as taken from [7] with addition of partial links to Wordnet³. The selection of a dataset consists of entities and their features which can be other entities, predicates or literals.

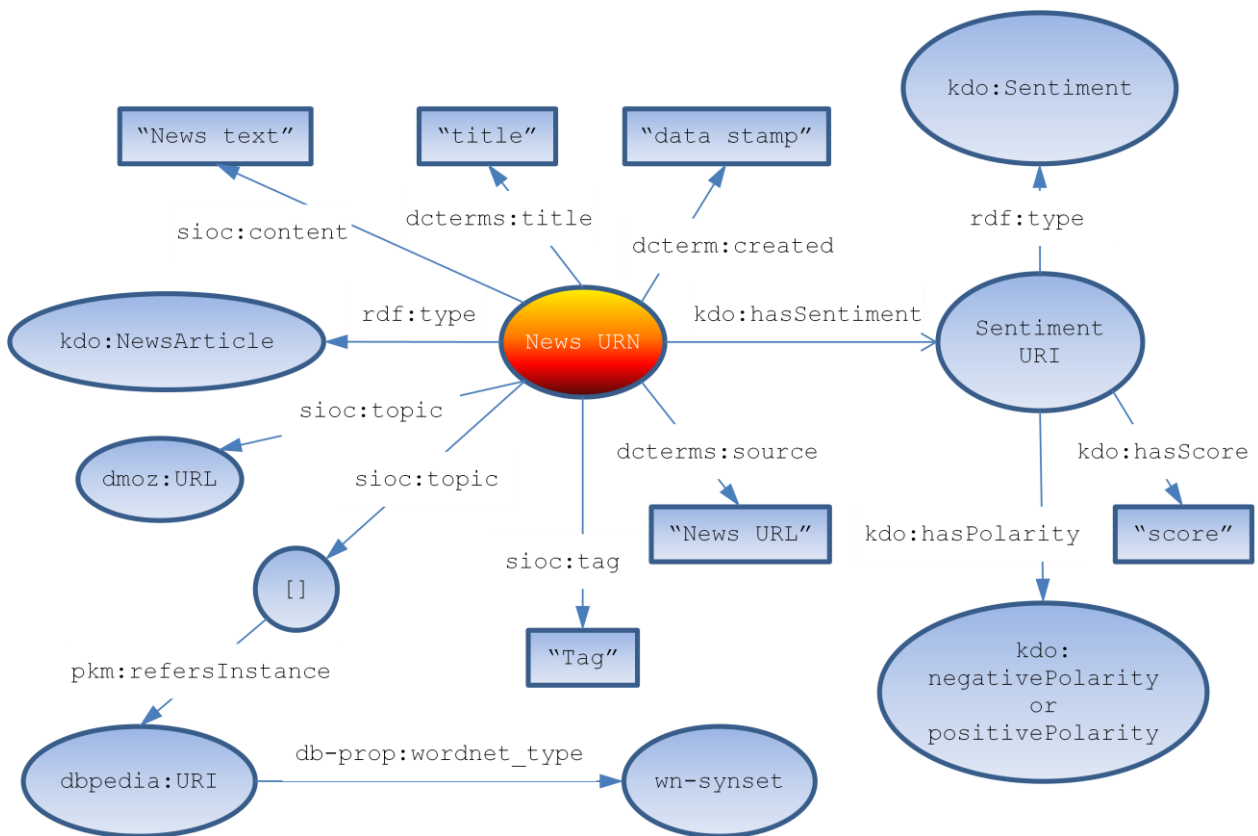


Figure 6: RDF representation for Google news documents [7].

There are four types of information available in the data which can be used by CLAS:

- dmoz⁴ labels made available via the the `sioc:tag` predicate (`<http://rdfs.org/sioc/ns#tag>`); available for all of the news articles (about 300 000);
- dmoz topics made available via the `sioc:topic` predicate (`<http://rdfs.org/sioc/ns#topic>`); available for all of the news articles (about 300 000);
- DBpedia URIs made available via blank nodes and `sioc:topic` and `pkm:refersToInstance` predicates (`<http://rdfs.org/sioc/ns#topic>` and `<http://www.ontotext.com/proton/protonkm#refersToInstance>`); concern about 35 000 news articles;
- Wordnet synssets for some of the DBpedia URIs made available via `db-prop:wordnet_type` predicate (`<http://dbpedia.org/property/wordnet_type>`); available for about 20 000 news articles.

2.3.1 Dataset Generation

SPARQL 16: News articles per DBpedia entities extracted.

```
SELECT ?s ?p
```

³ <http://wordnet.princeton.edu>

⁴ <http://www.dmoz.org>

```
FROM <http://www.ontotext.com/owlim/plugin/CLAS#dump>
WHERE { ?s sioc:topic/pkm:refersToInstance ?p }
```

Results for [PREFIX pkm:...](#) (100 of 316971)

s	p
urn:document-news-3d6cb913-755e-4ac5-8597-6e45a9900db3	dbr:United States
urn:document-news-4ce69a28-2714-4c2c-b11c-668685df8ae1	dbr:United States
urn:document-news-a8ebc03b-1872-4470-9c07-14e990940aad	dbr:United States
urn:document-news-23f24d6e-9775-4a99-8431-38dfb8188d9d	dbr:United States
urn:document-news-478756de-0081-4488-b166-7ac176c40d2d	dbr:United States

2.3.2 Clustering

The queries bellow (SPARQL 17-SPARQL 20), show the clustering of the corpus of Google news articles in 1000 clusters using the DBpedia entities extracted. Information about the clusters in which a news article belongs is retrieved. Moreover, two news articles from a single cluster are analyzed in terms of topics and tags.

SPARQL 17: Clustering with CLUTO in 1000 clusters and displaying the clusters and the number of news articles in each of them.

```
ASK { clas:L clas:hasParamValue "-zscore -showtree" }
```

```
ASK { clas:N clas:hasParamValue "1000" }
```

```
SELECT ?param ?val
WHERE { ?param clas:hasParamValue ?val }
```

```
ASK { [] clas:doClustering "direct" }
```

```
SELECT ?cluster (count(distinct ?s) as ?c_memb)
                (avg(?clusterDistance) as ?c_dist)
WHERE { ?s a kdo:NewsArticle ;
          clas:inCluster ?cluster ;
          clas:hasClusterDistance ?clusterDistance
        }
group by ?cluster having(?c_memb > 0)
order by desc(?c_memb)
```

Results for PREFIX_ (100 of 1000) View as [Exhibit](#)

cluster	c_memb	c_dist
807	1977	1.0957511380847656E-5
24	1874	-4.261472786403063E-6
26	1767	-5.87645727207941E-5
866	1676	1.2757756563308828E-5
395	1416	9.949152542389887E-6
569	1402	-9.272467903403312F-7

SPARQL 18: Clustering with CLUTO in 1000 clusters and displaying news articles and the cluster they belong to.

```

SELECT ?s ?cluster
WHERE {
  ?s a kdo:NewsArticle ;
    clas:inCluster ?cluster }
    
```

Results for SELECT ?s... (100 of 297946)

s	cluster
urn:document-news-1dff73f0-bccf-43b8-af56-a09ce9656fbc	945
urn:document-news-9e54e539-9ca2-4c44-9a56-4d0c81e0e2e4	63
urn:document-news-bcc92aba-2dd8-46f5-8600-5d4e9361e201	63
urn:document-news-6d97dca7-d230-4813-9485-469116c04d44	709
urn:document-news-a091c341-cfa1-4249-a255-2affa4269297	154
urn:document-news-f1aaeddd-abef-48d4-9410-4509d6db0347	494
urn:document-news-1899ff94-8c2c-42fe-9566-dce6b3381d3f	63

SPARQL 19: Example of news article with high typicality (z-score in CLUTO [6]) from cluster No. 63.

<urn:document-news-1899ff94-8c2c-42fe-9566-dce6b3381d3f>

Tags: Pakistan, Asia, Regional, Society_and_Culture, Warfare_and_Conflict

Topics:

- <http://www.dmoz.org/Top/Regional/Asia/India/Government>
- http://www.dmoz.org/Top/Regional/Asia/India/Society_and_Culture
- <http://www.dmoz.org/Top/Regional/Asia/Pakistan>
- http://www.dmoz.org/Top/Regional/Asia/Pakistan/Business_and_Economy
- http://www.dmoz.org/Top/Regional/Asia/Pakistan/Society_and_Culture
- http://www.dmoz.org/Top/Society/Issues/Warfare_and_Conflict
- http://www.dmoz.org/Top/Society/Issues/Warfare_and_Conflict/Specific_Conflicts
- http://www.dmoz.org/Top/Society/Issues/Warfare_and_Conflict/Specific_Conflicts/War_on_T...
- http://www.dmoz.org/Top/Society/Issues/Warfare_and_Conflict/Specific_Conflicts/War_on_T...
- http://www.dmoz.org/Top/Society/Issues/Warfare_and_Conflict/Specific_Conflicts/War_on_T...

document-news-b61c3ece-9194-4cfb-ad63-bae8476b74f8 RDF Search and Explore

Source: <urn:document-news-b61c3ece-9194-4cfb-ad63-bae8476b74f8>

Subject (21) Predicate Object All Download in: [JSON](#) | [RDF](#) | [N3/Turtle](#) | [N-Triples](#)

Statements in which the resource exists as a subject. Named Graph Language Inference

Predicate	Object
rdf:type	kdo:NewsArticle
dcterms:title	India and Pakistan swap nuclear data
dcterms:source	http://www.ft.com/cms/s/0/24c71820-d86f-11dd-bcc0-000077b07658.html
dcterms:created	2009-01-03T01:07:52Z
kdo:hasSentiment	urn:sentiment-of-news-14411808
sioc:content	India and Pakistan swap nuclear data India and Pakistan exchanged sensitive nuclear information on Wednesday in a rare conciliatory gesture after the sharp deterioration in relations following the Mumbai terror attacks. The step followed the personal intervention of George W. Bush, US president, who made new year's eve phone calls to Asif Ali Zardari, Pakistan's president, and Manmohan Singh, India's premier. A senior Pakistani official said Mr Bush "urged both India and Pakistan to de-escalate the situation by taking measures such as deploying air force units back to peacetime locations". The sharing of details of atomic sites between the nuclear-armed neighbours took place under a 17-year-old agreement that binds the two countries from attacking each other's facilities. In a related goodwill gesture, up to 40 Pakistani nationals arrived in Karachi after being released from jail in India where they had been arrested for visa infringements. The moves followed a month-long escalation in tensions between the two countries which saw Pakistan shift troops from its border with Afghanistan to that with India and cancel leave for its soldiers. India blames Lashkar-e-Taiba, a Pakistani militant group, for the November attack on some of Mumbai's best-known landmarks and has pushed Islamabad to hand over leaders of the group for trial,

SPARQL 20: Example of news article with low typicality (z-score in CLUTO [6]) from cluster No. 63.

```
<urn:document-news-6bb3019a-a140-4bbd-84e7-b538de62200d>
```

Tags: Pakistan, Asia, Regional, Society_and_Culture, Business_and_Economy

Topics:

- <http://www.dmoz.org/Top/Computers/Hardware/Peripherals/Printers>
- <http://www.dmoz.org/Top/Computers/Hardware/Peripherals/Printers/Supplies>
- <http://www.dmoz.org/Top/Regional/Asia/India/Government>
- <http://www.dmoz.org/Top/Regional/Asia/Pakistan>
- http://www.dmoz.org/Top/Regional/Asia/Pakistan/Business_and_Economy
- <http://www.dmoz.org/Top/Regional/Asia/Pakistan/Provinces>
- http://www.dmoz.org/Top/Regional/Asia/Pakistan/Society_and_Culture
- http://www.dmoz.org/Top/Society/Issues/Terrorism/Terrorist_Organizations
- http://www.dmoz.org/Top/Society/Issues/Warfare_and_Conflict
- http://www.dmoz.org/Top/Society/Issues/Warfare_and_Conflict/Specific_Conflicts

document-news-6bb3019a-a140-4bbd-84e7-b538de62200d not rank RDF Search and Explore

Source: <urn:document-news-6bb3019a-a140-4bbd-84e7-b538de62200d>

Subject (21) Predicate Object All Download in: [JSON](#) | [RDF](#) | [N3/Turtle](#) | [N-Triples](#)

Statements in which the resource exists as a subject. Named Graph Language Inference

Predicate	Object
rdf:type	kdo:NewsArticle
dcterms:title	Bounty for Hafiz Saeed to get information leading to arrest or conviction: US
dcterms:source	http://www.nation.com.pk/pakistan-news-newspaper-daily-english-online/national/05-Apr-2...
dcterms:created	2012-04-05T22:35:39Z
kdo:hasSentiment	urn:sentiment-of-news-43663147
sioc:content	As a defiant Lashkar-e-Taiba chief Hafiz Mohammed Saeed challenged the United States to prove charges against him, Washington clarified its offer of \$10 million bounty was not for finding him, but for information leading to his arrest and conviction. "Just to clarify, the \$10 million is for information not about his location, but information that leads to an arrest or conviction," state department spokesman Mark Toner told reporters on Wednesday. "And this is information that could withstand judicial scrutiny, so I think what's important here is we're not seeking this guy's location. We all know where he is," he said when asked about the rationale of the reward. Asked if he was suggesting that there is no information available to prosecute Saeed, Toner said: "There is information, there is intelligence that is not necessarily usable in a court of law." The spokesman suggested that the timing of the reward offer had "nothing to do with the ongoing parliamentary review" in Pakistan about relations with the US following the US military raid that killed Osama bin Laden last May and the mistaken killing of 24 Pakistani troops

2.3.3 Spreading activation and within cluster similarity

Once clustering has been performed successfully, one can use clusters to find similarities between their elements. For instance, if a news article is selected, the elements of the cluster can be ranked with respect to their similarity to the selected news article. The selected news article can be used as a seed of activation and the news articles belonging to the same cluster can be analyzed in terms of similarity and activation. On the other hand, clustering allows evaluating typicality and thus we have three measures that can be used for ranking depending on the needs. While similarity depends on the features selected and the two news article whose similarity is evaluated, typicality depends additionally on all cluster members as it involves all inter cluster similarities. Thus, the similarity and especially typicality can be considered as measures of diversity and used for diversity-aware ranking. Typically, lower similarity corresponds to lower typicality but the typicality is sensitive and relative to the other cluster elements (see SPARQL 21).

SPARQL 21: Example of similarity of a news article from cluster No. 63 to the rest of the news articles in the cluster together with typicality (z-score in CLUTO [6]) and activation.

```
ASK { [] clas:setClusterFocus
      <urn:document-news-1899ff94-8c2c-42fe-9566-dce6b3381d3f> }

ASK { [] clas:activateNode
      <urn:document-news-1899ff94-8c2c-42fe-9566-dce6b3381d3f> }

SELECT ?news ?activity ?similarity ?typicality ?cluster
WHERE { ?news a kdo:NewsArticle ;
          clas:hasActivity ?activity ;
          clas:inCluster ?cluster ;
```

```

        clas:hasClusterFocusDistance ?similarity ;
        clas:hasClusterDistance ?typicality
    FILTER ( ?activity > "0"^^xsd:double )
}
order by DESC(?activity)

```

Results for PREFIX... (100 of 111)

View as [Exhibit](#)

news	activity	similarity	typicality
urn:document-news-189...	9.0	1.0	-0.314205
urn:document-news-7c1...	7.0	0.6864064729836442	-0.963705
urn:document-news-ec3...	7.0	0.875	-0.11554
urn:document-news-a71...	6.0	0.5669467095138409	-0.861355
urn:document-news-358...	6.0	0.75	0.006868
urn:document-news-f14...	6.0	0.6396021490668313	-0.007857
urn:document-news-eed...	5.0	0.7905694150420948	1.489946
urn:document-news-d5d...	5.0	0.6681531047810609	0.0373
urn:document-news-6a6...	5.0	0.4902903378454601	-0.741008
urn:document-news-496...	5.0	0.4902903378454601	-0.474879
urn:document-news-69d...	5.0	0.5590169943749475	0.17411

In SPARQL 22, the normalized activation is used to evaluate the similarity between news articles. In the chosen data model, each news article is connected to the DBpedia entities extracted from its content, and as shown above (see SPARQL 16) the SA connectivity matrix defines a bi-partite graph. So, as seen in SPARQL 21, the activation of the news articles for 2 SA iterations gives the following final activations: for the seed the initial activation (1) plus the number of DBpedia entities; for the other news articles the activation is equal to the number of shared DBpedia entities with the seed article. To normalize the activation, one can divide by the number of DBpedia entities, thus weighting the overlap between the seed news article and another article. Normalized activation of 1 will correspond to activation equal to the number of DBpedia entities. In SPARQL 22, the normalized activation for the seed is equal to 1.125 as the activation for the seed is obtained by the initial activation plus the number of connections divided by the number of connections.

SPARQL 22: Normalized activation as cross-cluster similarity between news articles.

Seed: <urn:document-news-1899ff94-8c2c-42fe-9566-dce6b3381d3f>

```

SELECT ?news ?normlzd_activity_per_news
      ?number_of_dbUris ?activity ?cluster
WHERE {
  {SELECT ?news ?activity ?cluster
      (count(?dbUri) as ?number_of_dbUris)
   WHERE {?news a kdo:NewsArticle ;
            clas:hasActivity ?activity ;
            clas:inCluster ?cluster ;
            clas:sioc:topic/pkm:refersToInstance ?dbUri .
        }
    group by ?news ?activity ?cluster
    order by asc(?cluster)
  }
  BIND((?activity/?number_of_dbUris) as ?normlzd_activity_per_news)
  FILTER(?normlzd_activity_per_news > "0.5"^^xsd:double)
}
order by desc(?normlzd_activity_per_news)

```


Results for PREFIX... (98) View as [Exhibit](#) Download

news	normlzd_activity_per_news	number_of_dbUris	activity	cluster
um:document-news-189...	1.125	8	9.0	63
um:document-news-ee...	1.0	5	5.0	63
um:document-news-ff5...	1.0	1	1.0	87
um:document-news-b68...	1.0	4	4.0	88
um:document-news-83a...	1.0	2	2.0	88
um:document-news-894...	1.0	3	3.0	88
um:document-news-023...	1.0	4	4.0	88
um:document-news-59e...	1.0	1	1.0	123

SPARQL 23 illustrates this point by explicitly quoting the DBpedia entities for the seed and the most similar news article.

SPARQL 23: Example of news articles ordered by decreasing activation with DBpedia entities and dmoz topics. Seed: <urn:document-news-1899ff94-8c2c-42fe-9566-dce6b3381d3f>.

```

SELECT ?news ?activity ?normlzd_activity_per_news ?dbUris
      (GROUP_CONCAT(?topic; separator=", ") as ?topics)
WHERE {
  {SELECT ?news ?activity (count(?dbUri) as ?number_of_dbUris)
        (GROUP_CONCAT(?dbUri; separator=", ") as ?dbUris)
   WHERE {?news a kdo:NewsArticle ;
           clas:hasActivity ?activity ;
           sioc:topic/pkm:refersToInstance ?dbUri .
        }
  group by ?news ?activity
  order by asc(?activity)
}
?news sioc:topic ?topic .
FILTER (!isBlank(?topic))
BIND((?activity/?number_of_dbUris) as ?normlzd_activity_per_news)
FILTER(?normlzd_activity_per_news > "0.5"^^xsd:double)
}
group by ?news ?activity ?normlzd_activity_per_news ?dbUris ?sentiment
order by desc(?activity)
    
```



```

WHERE { ?news a kdo:NewsArticle ;
         sioc:topic/pkm:refersToInstance ?dbUri .
         ?dbUri clas:hasActivity ?activity .
}
group by ?dbUri ?activity
order by desc(?activity)

```

Results for PREFIX... (100 of 3803)

View as E

dbUri	activity	news_per_dbUri
dbr:India	971.0	971
dbr:Pakistan	538.0	538
dbr:Mumbai	412.0	412
dbr:United_States	230.0	2215
dbr:Islamabad	222.0	222
dbr:U.S.	217.0	4177
dbr:Afghanistan	213.0	705
dbr:China	201.0	863
dbr:Reuters	154.0	2237
dbr:Taliban	152.0	329
dbr:New_Delhi	144.0	194
dbr:Japan	126.0	916
dbr:Russia	119.0	602

2.4 Discussion and Future Work

In the previous sections, we presented extensive analyses of the corpus of Google news articles, available at <http://rendernews.ontotext.com>. The main focus of this exploration was to assess to what extent the information extracted from the news content (sioc:content) allowed for meaningful clustering of the data and more importantly ranking of the news articles with respect to diversity of their content.

The quality of the clustering was not discussed in detail, as the focus of the presentation of CLAS was to demonstrate the type of analyses and their added value and not a particular application. Another reason for that was the focus on DBpedia entities which are most relevant to graph based diversity-aware ranking in CLAS and the realization that they do not reflect well the news article topics and content as they cover only names, locations and organizations.

Several extensions of the existing tools and functionality of CLAS are under development and planned to be available before the end of the project. Such extensions are:

- Various clustering and SA datasets generation;
- Additional clustering algorithms and typicality measures;
- Rich data model concerning clustering and SA for RDF based storage and SPARQL access.

These additional functionalities will make possible the flexible usage of CLAS by combining results from various clustering algorithms on different datasets, uncover the structure of existing clusters related to diversity, combine activation and clustering, etc.

On the other hand, the CLAS analysis of the Google news corpus on <http://rendernews.ontotext.com> showed that the available information in terms of gmoz labels and topics, and DBpedia entities related to

names, locations, and organizations do not reflect to a sufficient extent the specificity and thus the potential diversity of content of the news articles so that the CLAS tools could demonstrate their full potential.

So, together with the development of CLAS, Ontotext will focus in extracting additional information from the Google news content which is not available so far. This information will be related to:

- **Linked Open Data connectivity of the content set.** Ontotext will try to achieve a richer and topic related semantic annotation which will allow connection to other concepts and LOD⁵ sources. Relevant links between the content of the Google news articles and the linked data cloud can be utilized as a good basis for the clustering algorithms and enhance the spreading activation utility in semantic stores. It will facilitate semantic and content search for providing access to additional information for an end user;
- **Recommendation of related and historical content.** The intuition behind this idea is that an event, about which a news story is worth to write, does not happen in isolation of other events. To capture the diversity and more importantly the sentiment and objectivity of one's point of view, similar news and especially historical/legacy articles can be added to the story context;
- **Summarization and visualization.** The usability of the richer information and related CLAS output needs high quality presentation to the end user with interactive capabilities of search and context choice with respect to content and historical contexts.


To achieve the above goals, Ontotext will use the recently published Google Wiki-links⁶ corpus to improve the enrichment of Google news articles with related Linked Open Data entities. In total, it features 40 million disambiguated mentions of Wikipedia concepts, which are found within 10 million web pages. This data set helps to disambiguate and cross-reference between any mention of things in a document to mention descriptions in Wikipedia.

⁵ <http://linkeddata.org>

⁶ <https://code.google.com/p/wiki-links/>

3 The Diversity-Aware Ranking Service

Diversity-Aware Ranking Service:



Main parameters:

SPARQL endpoint*:

SPARQL Restrictions[^] (on document/statement ?s):

SPARQL Order By[^] (e.g. ASC(?s)):

Rank focus[^]:

Special parameters:

Force algorithm[^]:

Epsilon[^]:

Gamma[^]:

Sentiment score normalization[^]: (uncheck this box only if **all** sentiment values are in the interval [0,1])

Testing and evaluation parameters:

Fix random element (maximum algorithm)[^]:

Debug mode[^]:

* mandatory
^ optional

More information about this ranking service can be found in RENDER deliverables D3.3.1 and D3.3.2 ([click here](#)). More information about the RENDER project can be found [here](#). The source code of this service is licensed under GPL v3 and can be found [here](#). If any questions remain, please contact [Andreas Thalhammer](#).

RENDER is funded by




Figure 7: HTML 5 Web interface of the Diversity-Aware Ranking Service

The RENDER Diversity-Aware Ranking Service is a middleware application using OWLIM and Sesame triple stores as a data layer and providing ranked data as JSON output. The ranking algorithms were selected with respect to performance and effectiveness (i.e. they come from the image retrieval domain). The ranking also includes clustering that enables compact representation and visualization of larger amounts of posts, news articles or tweets. Moreover, each cluster also includes a “representative”⁷ that is selected during the process. The representative is a document that represents all other articles of its respective cluster in the best way. In the following we list the set of new features that were introduced since the prototype version of the Diversity-Aware Ranking Service described in D3.3.1:

- Easy to understand HTML 5 Web interface (see Figure 7)
- Sentiment score normalization
- SPARQL “ORDER BY”
- Maximum algorithm
- Automatic algorithm selection
- Debug mode
- Status and performance parameters included in the JSON output

The RENDER Diversity-Aware Ranking Service is licensed under the GNU General Public License (GPL v3). The deployed service in version 2 can be found at:

- <http://ranking.render-project.eu>

⁷ In this case, a representative is a document or statement that serves as a label for the respective cluster.

The source code is available at:

- <https://github.com/athalhammer/RENDER-ranking-service>

The service is implemented in Java 6 with JAX-RS to be deployed on a GlassFish v3 server.

As already described in D3.3.1, the service makes use of the two dimensions of “topic” and “sentiment” published in a KDO⁸-compliant way (i.e. using the vocabulary terms: `kdo:Sentiment`, `kdo:hasSentiment`, `kdo:hasScore`, `kdo:hasPolarity`, `kdo:Polarity` and `sioc:topic`). As an example, such a KDO-compliant output is produced by the Enrycher⁹ RDF extractor. The data should be accessible through a SPARQL endpoint¹⁰

The remainder of this chapter is structured as follows: In Section 3.1 we recapitulate the similarity measures in use and also explain the sentiment normalization formula that was added in version 2. Section 3.2 revisits the FOLDING algorithm of D3.3.1 and explains the newly added MAXIMUM algorithm that doesn’t assume a pre-ranking. In Section 3.3 we discuss the input parameters of the service while explain the output in Section 3.4.

3.1 Recapitulation of the similarity measures

The similarity measures are based on two dimensions, namely “topic” and “sentiment”. For each dimension, we measure the similarity separately and form a linear combination afterwards. In this short recap we also introduce the normalization of sentiment scores as it is done by the service.

Topic Similarity:

For topic similarity, we make use of the Jaccard¹¹ similarity index. The function `topics(x)` retrieves the set of topics of the statement `x` (topics are extracted as described in D2.2.1):

$$Jacc(S_1, S_2) = \frac{|topics(S_1) \cap topics(S_2)|}{|topics(S_1) \cup topics(S_2)|}$$

Sentiment Similarity:

The sentiment score is extracted by the opinion mining toolkit (cf. D2.1.1). The function `score(x)` denotes the double sentiment score value of the sentiment of the statement `x`:

$$Sent(S_1, S_2) = 1 - |score(S_1) - score(S_2)|$$

Sentiment Normalization:

It has to be noted that the `Sent` measure only works out of the box for scores in the interval $[0, 1]$. For other sentiment scores, such as those extracted by Enrycher, we have to engage a normalization process beforehand. The constants `MIN` and `MAX` are the minimum respectively maximum sentiment scores contained in the triple store. The normalization is executed only if `MIN < MAX`.

$$normalized_score(x) = \frac{score(x) - MIN}{MAX - MIN}$$

In the other case, if `MIN == MAX`, the scores keep their value and `Sent(S1, S2)` will result in 1 in any case (maximum similarity).

Combining topic and sentiment similarity:

Both similarity scores, `Jacc` and `Sent`, range in the interval between 0 and 1 where 1 means full similarity and 0 means no similarity at all. Therefore, we can combine the two scores by simply averaging them:

$$SimAvg(S_1, S_2) = \frac{Jacc(S_1, S_2) + Sent(S_1, S_2)}{2}$$

⁸ The Knowledge Diversity Ontology (KDO): <http://kdo.render-project.eu/>

⁹ Enrycher: <http://enrycher.ijs.si/>

¹⁰ SPARQL endpoints are URL that expose SPARQL via HTTP binding (Source: <http://www.w3.org/wiki/SPARQL>)

¹¹ http://en.wikipedia.org/wiki/Jaccard_index, last checked on 14.02.2013

More in general, we can denote the similarity score by linearly combining them.

$$\text{SimLin}(S_1, S_2) = \gamma \cdot \text{Jacc}(S_1, S_2) + (1 - \gamma) \cdot \text{Sent}(S_1, S_2)$$

The restriction on this combination is: $0 \leq \gamma \leq 1$. Therefore, SimAvg can be represented by SimLin with $\gamma = 0.5$.

3.2 Clustering algorithms in use

We employ two different algorithms, namely FOLDING and MAXIMUM adapted from [1]. If an ORDER BY statement is given as a parameter, the system automatically assumes a pre-ranking and employs the FOLDING algorithm as it was specifically designed for pre-ranked lists. If no ORDER BY statement is given the MAXIMUM algorithm is used as it does not need a pre-ranking. Either way, there is also the option to enforce the application of one of the two algorithms. Of course, this function has to be regarded with care with respect to the FOLDING algorithm as it requires a relevance-based ranking in order to provide clusters that make sense. The FOLDING algorithm was already explained in D3.3.1, but for reasons of completeness, we also include its description at this point.

FOLDING algorithm:

The folding algorithm assumes a ranked list as input. There are two disjoint lists maintained, the representatives and the rest. At the start, the ranked input is the rest. The algorithm selects the first element of the rest (i.e. the ranked input list) as a representative. In the following, each element of the rest is compared to the representatives and added to the representatives list in case its similarity to all existing representatives is less than a certain reference point (epsilon). When all representatives are established, the each element in the rest is assigned to the cluster of which the representative is most similar to it. Figure 8 denotes the algorithm in pseudo code taken from [1].

Algorithm 1 Folding

Input: Ranked list L of I

Output: Clustering C

- 1: Let the image L_1 be the first representative R_1
 - 2: **for** Each image L_i **do**
 - 3: **if** $d(L_i, R_j) > \epsilon(*)$ for all representatives R_j **then**
 - 4: add L_i to the set of representatives R
 - 5: **for** Each image $L_i \notin R$ **do**
 - 6: Find representative R_j that is closest to L_i
 - 7: Assign L_i to the cluster of R_j
- (*) ϵ is defined as the mean distance all images have to the *average image* in I
-

Figure 8: The FOLDING algorithm (Source: [1])

MAXIMUM algorithm:

The MAXIMUM algorithm is similar to FOLDING but has some distinct features. The MAXIMUM algorithm belongs to the class of randomized algorithms. Again there are two disjoint lists, the representatives and the rest which is assigned to the input at the beginning. The first element of the representatives is selected randomly from the rest. Then, the algorithm adds the element which has minimum maximum similarity (or maximum minimum distance) to the representatives. If this minimum maximum similarity is at some point less than epsilon, all representatives are found and the remaining elements in the rest list are assigned to the clusters with closest representatives. Figure 9 shows the pseudo code of the MAXIMUM algorithm from [1]. The following example of the algorithm treats an array of (normalized) sentiment scores:

[0.2, 0.23, 0.17, 0.45, 0.67, 0.97, 0.95]

Let us assume the randomly selected first representative is 0.23 and the epsilon is fixed to 0.5. We proceed with the representative search: the element which is selected next is 0.97 as it has the lowest similarity score to 0.23. Now, the rest array is [0.2, 0.17, 0.45, 0.67, 0.95] and the representatives are [0.23, 0.97]. Now, we compare each element from the representatives to the rest with the similarity measure Sent:

Compare to 0.20: [0.97, 0.23]
 Compare to 0.17: [0.94, 0.20]
 Compare to 0.45: [0.78, 0.48]
 Compare to 0.67: [0.56, 0.70]
 Compare to 0.95: [0.28, 0.98]

Each of these arrays has a maximum. The element with the smallest maximum element is selected as representative. In this case, the element with sentiment score 0.45 is chosen as the next representative as it has 0.70 as the greatest minimum similarity. The algorithm terminates at this point as the maximum similarity within the representatives is higher than 0.5.

Algorithm 2 Maxmin

Input: Set S containing I

Output: Clustering C

```

1: Select the first representative  $R_1$  randomly
2: while All pairwise distances in  $R > \epsilon$  do
3:   for Each image  $L_i \notin R$  do
4:     Let  $d_i$  be  $\arg \min_{d(L_i, R_j), R_j \in R}$ 
5:   Add to  $R$  the image with  $\arg \max_{d_i}$ 
6: for Each image  $S_i \notin R$  do
7:   Find representative  $R_j$  that is closest to  $S_i$ 
8:   Assign  $S_i$  to the cluster of  $R_j$ 

```

Figure 9: The MAXIMUM algorithm (Source: [1])

Epsilon estimation:

For estimating the epsilon, we measure the average similarity of each item to each other item. A pseudo code example is given in the following Listing 1.

<p>Input: List L containing Statements</p> <p>Output: double value of Epsilon</p> <hr/> <pre> SumAll := 0; For each Statement s1 in L Sum := 0; For each Statement s2 in L If (s1 != s2) Sum := Sum + SimLin(s1, s2); Avg := Sum / (size(L) - 1); SumAll := SumAll + Avg; return SumAll / size(L); </pre>

Listing 1: Pseudo code for the estimation of epsilon

3.3 Description of the input parameters

In addition to the HTML5 Web interface, the ranking service can be called via a simple GET request at

- <http://ranking.render-project.eu/rank>

We distinguish between main and special input parameters.

Main parameters:

- **[endpoint]** The Sesame/OWLIM SPARQL endpoint where KDO-conform data is contained.
- **[restrictions]** Restrictions can be put on the document/statement ?s. For example if the topics of ?s should overlap with another document's topics:

```
urn:document-8efbadc0-8f38-47dc-b2c0-230a732ace52> sioc:topic ?t.
?s sioc:topic ?t.
```

- **[orderBy]** One can specify additional parameters in accordance to which should be ordered. If this parameter is set, the FOLDING algorithm is applied by default. E.g. one can order in accordance to sentiment in the following way:

```
restrictions: ?s kdo:hasSentiment ?sent. ?sent kdo:hasScore ?score.
```

```
orderBy: DESC (?score)
```

- **[rank]** This parameter defines an emphasis on a certain property (either kdo:hasSentiment or sioc:topic). The full URI for the property is needed e.g.

```
http://kdo.render-project.eu/kdo#hasSentiment
```

Special parameters:

- **[algorithm]** The ranking algorithm is selected automatically by default (depending on whether the orderBy parameter is set or not). With the algorithm parameter, either FOLDING or MAXIMUM can be enforced. Type, String.
- **[gamma]** This enables individual tuning of the gamma weighting in the similarity measure. This parameter overrides the [rank] parameter when set. Type, double.
- **[epsilon]** This parameter enables individually setting the epsilon parameter of the two algorithms. This parameter omits the automatic epsilon estimation. Type, double.
- **[normalization]** This parameter enables normalization of the sentiment scores. Type, boolean.

Testing and evaluation parameters:

- **[debug]** Provides additional information if an error appears. Type, boolean.
- **[random]** Fixes the random element for the maximum algorithm in order to provide different test users with the same set of clusters

The **endpoint** parameter is the only mandatory parameter. All other parameters are optional.

3.4 Description of the output parameters

The following status and performance parameters are contained in the JSON output:

- **[status]** Either "ok" or "error"
- **[Selected algorithm]** Either "folding" or "maximum"
- **[Number of documents]** The number of documents which matched the query

- **[Number of clusters]** The number of clusters created
- **[Used epsilon]** The epsilon in use
- **[SPARQL time (ms)]** The time in milliseconds the SPARQL query evaluation took
- **[Clustering time (ms)]** The time in milliseconds the clustering algorithm took
- **[result]** The result of the clustering
- **[message]** If an error occurred, this parameter shows the message
- **[stacktrace]** If debug enabled and an error occurred, this parameter shows the stacktrace.

The output of the **result** is a list of clusters with the following structure:

```
[{"representative": statement1, "rest" : [statement2, statement3, <...>]} ,
{"representative": statement4, "rest" : [statement5, statement6, <...>]},
<...> ] }
```

Note the representative/rest separation that stems the two algorithms. Statements (above denoted as statement<number>) are represented in the following form:

```
{ "uri" : "urn:document-cf25366f-f1d2-4715-8901-a03d2413fc11", "score":0.5,
"polarity" : "http://kdo.render-project.eu/kdo#negativePolarity", "topics" :
["http://www.dmoz.org/.../Shareware/Windows/Personal_Information_Managers",
"http://www.dmoz.org/Top/Computers/Internet/On_the_Web/Web_Applications",
"http://www.dmoz.org/Top/Shopping/Health/Pharmacy", <...> ] }
```

3.5 Evaluation of the Diversity-Aware Ranking Service

The diversity-aware ranking service makes use of semantic features that have been extracted automatically from text. In particular, the opinion mining toolkit (cf. D2.1.1, D2.1.2) and the fact mining toolkit (cf. D2.2.1, D2.2.2) process documents with natural language and store the extracted information in the data infrastructure of the render project. In order to present the stored documents to the user, we analyse the attached facts and cluster the documents accordingly. Thus, the diversity-aware ranking service helps to feature the effect of discovering new but related content. In the following, we will provide the evaluation of the diversity-aware ranking service including a discussion of the results.

We implemented a Web interface that visualizes the documents and their clusters. Particular attention has been paid to enable the users to understand the clusters. Thus we show attached sentiments and topics. The interface enables it even for users with not technical background to browse through the articles and clusters. A slider enables the user to modify the gamma value to the settings {1.0, 0.75, 0.5, 0.25, 0.0}. Figure 10 shows a screenshot of this interface. The interface is currently available at: <http://ranking.render-project.eu/vis.html>.

The evaluation of the diversity-aware ranking service faced particular problems that had to be overcome. The problems were:

1. Up to now, there are no standardized datasets that provide a reference for what is called “diversity-aware clustering of documents”.
2. The outcome of the diversity-aware ranking service is heavily dependent on the performance of the sentiment analysis and topic extraction tasks. Unfortunately, for this, no “clean” or “expert-generated” dataset was available.

The first problem was addressed by deciding for the application of quantitative rather than qualitative measures. For this, we extracted a random sample of 100 documents with the attached diversity

information from the render news dataset¹² and stored it in a dedicated triple store. On this endpoint, we applied the diversity-aware clustering service. The parameters for the service were set as follows:

- [endpoint] the dedicated endpoint where the random sample of 100 was stored.
- [restrictions] none
- [orderBy] none



Figure 10: Screenshot of the Web interface of that visualizes the documents and their clusters.

- [rank] none
- [algorithm] none
- [epsilon] none
- [gamma] varying in relation to the position of the slider {leftmost: 1.0, half left: 0.75, middle: 0.5, half right: 0.25, right: 0.0}
- [normalization] true
- [random] urn:document-news-29c19ccb-11a7-4269-a1dc-72361020d28b

Note that, implicitly, the rank parameter is overwritten by the given gamma value. Moreover, the maximum algorithm is chosen automatically as no pre-ranking was defined. As no selected epsilon was provided, the epsilon estimation described in Section 3.2 was put into place. A sample output is provided in Listing 2.

¹² This dataset is available at <http://rendernews.ontotext.com/>

```

{
  "status": "ok",
  "Selected algorithm": "maximum",
  "Number of documents": 100,
  "Number of clusters": 4,
  "Used epsilon": 0.4450836060132477,
  "SPARQL time (ms)": 80,
  "Clustering time (ms)": 5,
  "result": [
    {
      "representative": {
        "uri": "urn:document-news-29c19ccb-11a7-4269-a1dc-72361020d28b",
        ...
      }
    }
  ]
}

```

Listing 2: Excerpt of the output of the diversity-aware ranking service applied on the sample of 100 ($\gamma = 0.5$).

We conducted a survey with 49 participants being in their Bachelor, Master, or PhD studies in computer science. Here, in order to address the second problem, we decided also to ask about the subjective quality of extracted topics and sentiments in the selected sample of 100. The participants were asked to answer the following questions:

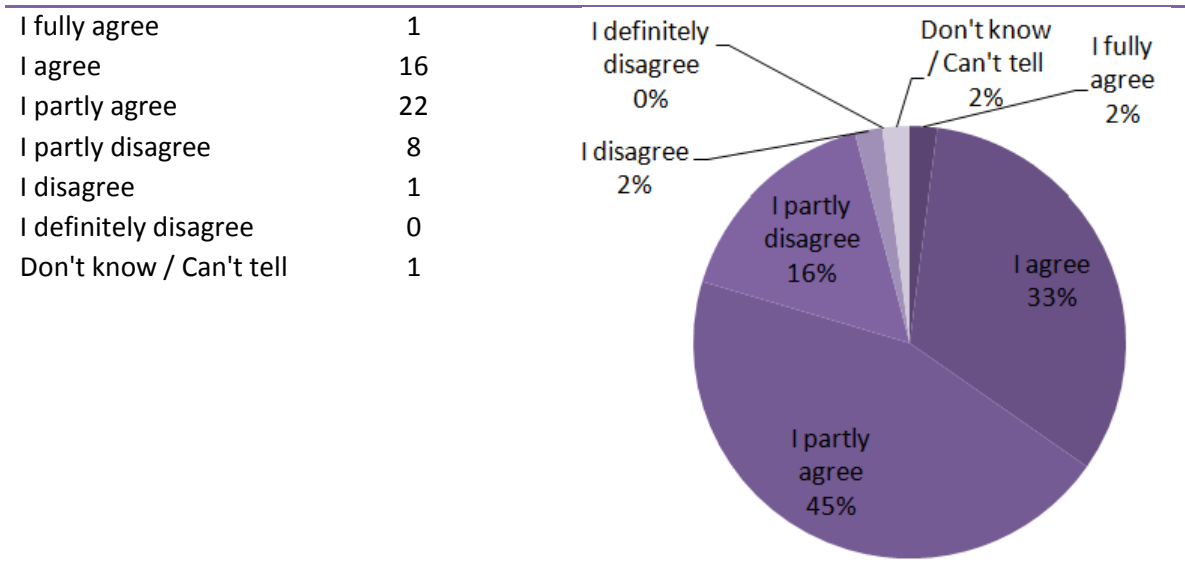
- *Do the extracted TOPICS fit to the articles (browse at least 10 articles before you provide your opinion)?**
- *Do the extracted SENTIMENTS fit to the articles (browse at least 10 articles before you provide your opinion)? **
- *Move the slider to the leftmost position. Do the clusters provide you with an overview about different topics (browse at least 4 clusters before you provide your opinion)? **
- *Move the slider to the rightmost position. Do the clusters provide you with an overview about different sentiments (browse at least 4 clusters before you provide your opinion)? **
- *Do you consider the other options (slider is in half left, middle, half right position) as useful combinations? **
- *Would you like to see your Tweets or Facebook posts organized a similar way (i.e. they are organized by extracted topics and sentiments)? **
- *Would you like to see the articles of your favourite online news portal organized a similar way (i.e. they are organized by extracted topics and sentiments)? **
- *How would you explain the difference in amount of clusters for the both extremes (slider at leftmost vs. slider at rightmost position): **
- *Further remarks, ideas, and feedback about the diversity-aware clustering:*

** mandatory*

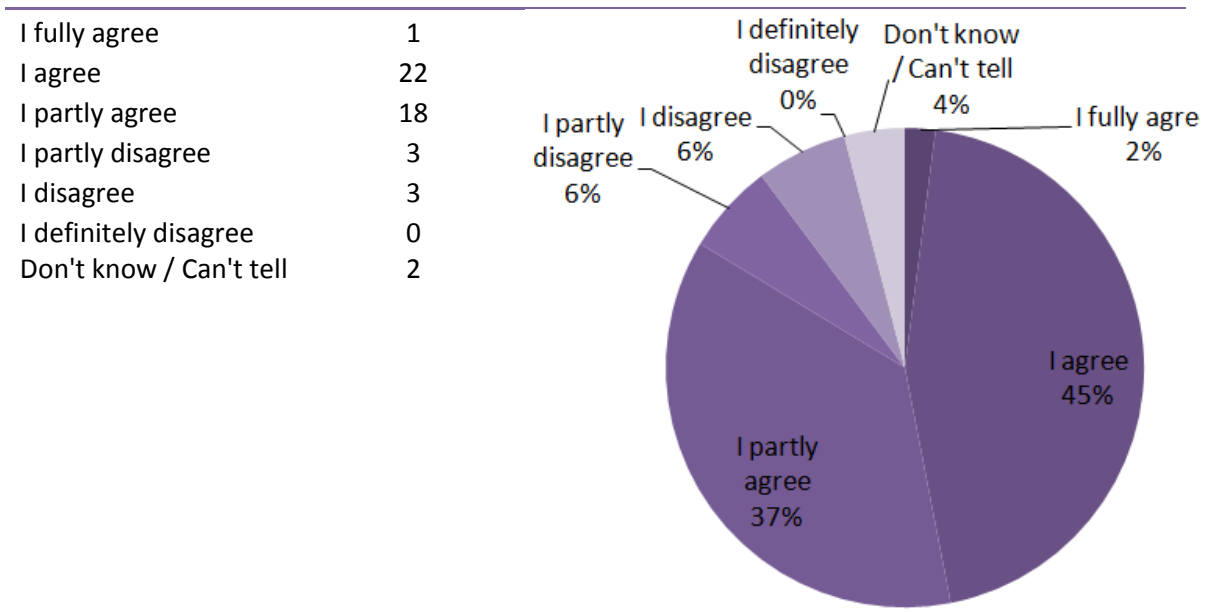
In the remainder of this section we will firstly present the results of the survey (cf. Section 3.6); in Section 3.7, we will discuss the results and provide ideas on how to improve ranking in the future.

3.6 Presentation of the survey results

- Do the extracted TOPICS fit to the articles (browse at least 10 articles before you provide your opinion)?

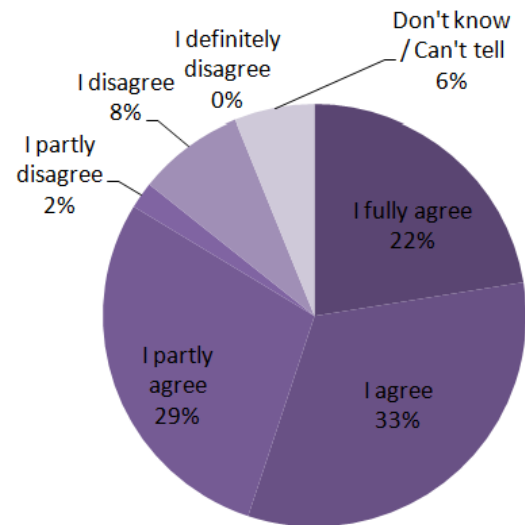


- Do the extracted SENTIMENTS fit to the articles (browse at least 10 articles before you provide your opinion)?



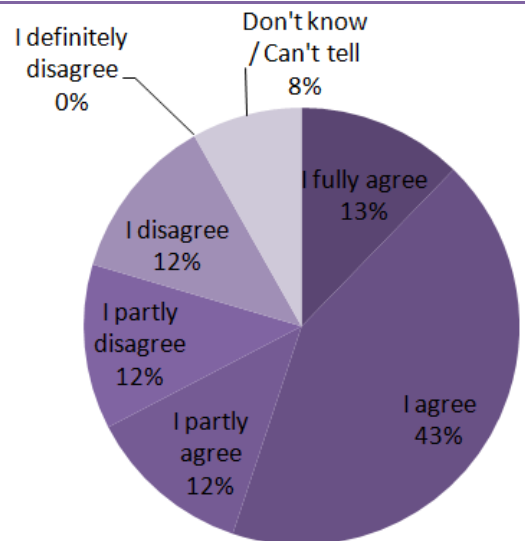
- **Move the slider to the leftmost position. Do the clusters provide you with an overview about different topics (browse at least 4 clusters before you provide your opinion)?**

I fully agree	11
I agree	16
I partly agree	14
I partly disagree	1
I disagree	4
I definitely disagree	0
Don't know / Can't tell	3



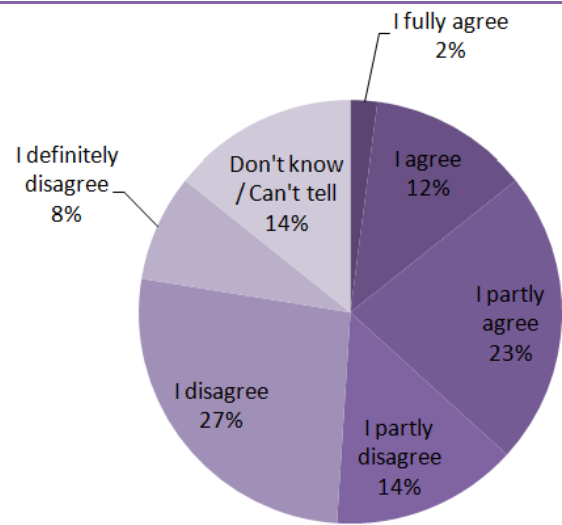
- **Move the slider to the rightmost position. Do the clusters provide you with an overview about different sentiments (browse at least 4 clusters before you provide your opinion)?**

I fully agree	6
I agree	21
I partly agree	6
I partly disagree	6
I disagree	6
I definitely disagree	0
Don't know / Can't tell	4



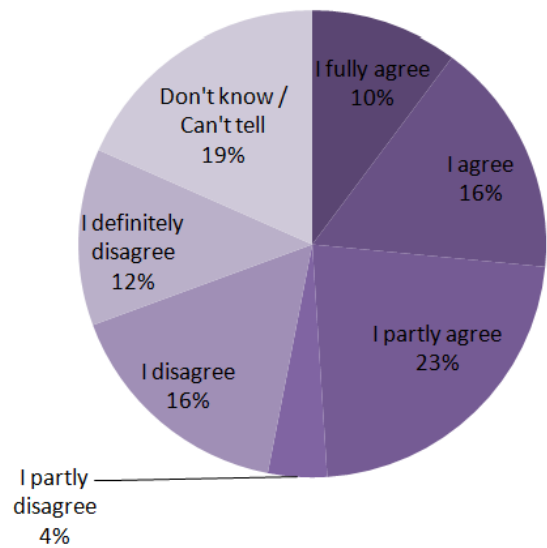
- **Do you consider the other options (slider is in half left, middle, half right position) as useful combinations?**

I fully agree	1
I agree	6
I partly agree	11
I partly disagree	7
I disagree	13
I definitely disagree	4
Don't know / Can't tell	7



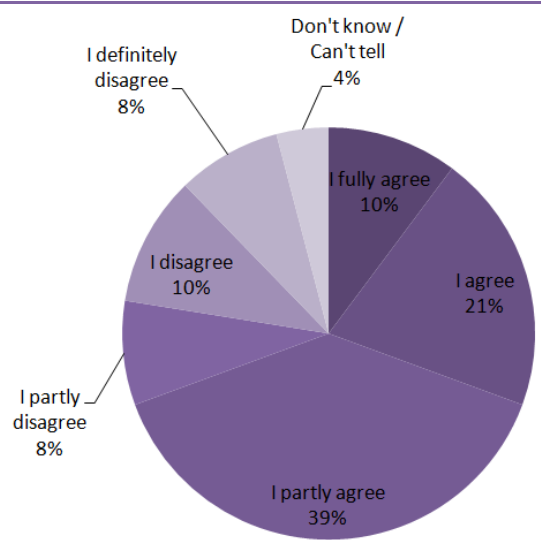
- **Would you like to see your Tweets or Facebook posts organized a similar way (i.e. they are organized by extracted topics and sentiments)?**

I fully agree	5
I agree	8
I partly agree	11
I partly disagree	2
I disagree	8
I definitely disagree	6
Don't know / Can't tell	9



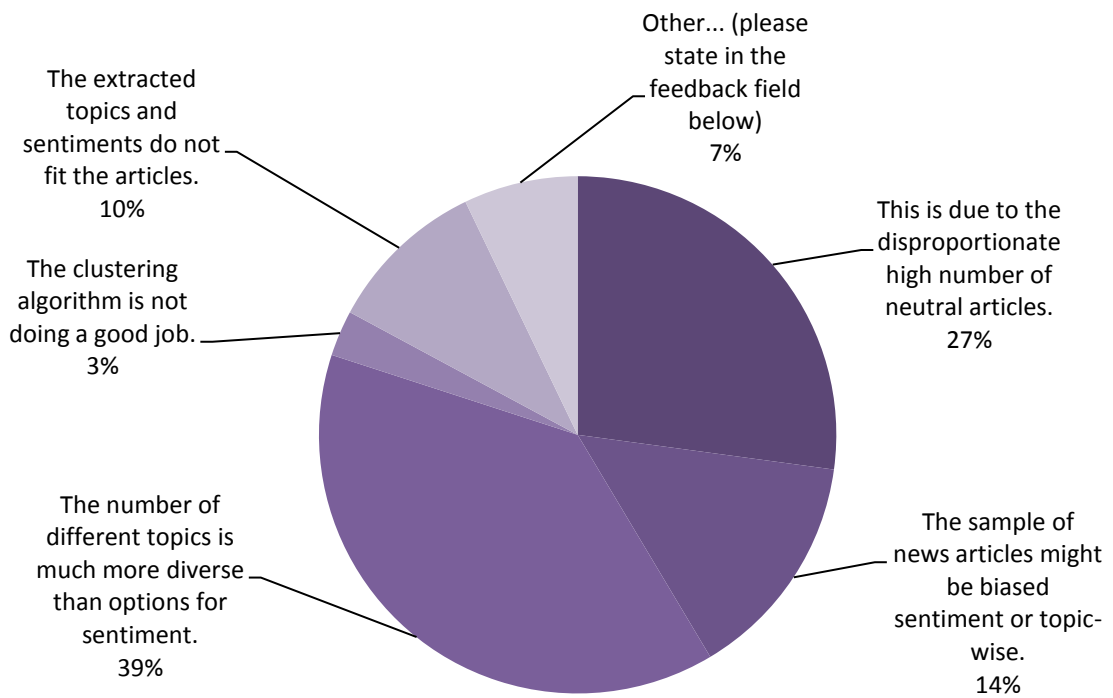
- **Would you like to see the articles of your favourite online news portal organized a similar way (i.e. they are organized by extracted topics and sentiments)?**

I fully agree	5
I agree	10
I partly agree	19
I partly disagree	4
I disagree	5
I definitely disagree	4
Don't know / Can't tell	2



- **How would you explain the difference in amount of clusters for the both extremes (slider at leftmost vs. slider at rightmost position):**

This is due to the disproportionate high number of neutral articles.	19
The sample of news articles might be biased sentiment or topic-wise.	10
The number of different topics is much more diverse than options for sentiment.	28
The clustering algorithm is not doing a good job.	2
The extracted topics and sentiments do not fit the articles.	7
Other... (please state in the feedback field below)	5



- **Further remarks, ideas, and feedback about the diversity-aware clustering:**¹³
 - *The algorithm seems to provide the same items if the slider is not on the leftmost, so it doesn't matter if he is in the center or rightmost (maybe not enough test data to get a good clue about if it works or not)*
 - *You cannot access other cluster while an other cluster is open.*
 - *I don't think it is a legitimate approach to assign "sentiments" to articles. Also i don't want some bogus algorithm providing me with information it thinks suits my preferences, i want the goddamn information i'm looking for in the most objective way possible. And i definitely don't want any further fiddling with my newsfeed etc.*
 - *While it might be that 'The number of different topics is much more diverse than options for sentiment' I see that as a shortcoming of the sentiment analysis, since articles could be grouped by sentiment values which would result in a more fine-grained clustering, possibly.*
 - *I can't see a reason why a cluster according to sentiment would make sense for any kind of information presentation.*
 - *sometimes the same words are used twice in the sentiments short articles sometimes have way too many sentiments*

3.7 Discussion of the results

The survey results presented in the previous section provide insights about the documents contained in the random sample as well as their cluster behaviour.

The first two questions aimed at evaluating whether the extracted topics and sentiments are appropriate for the presented articles. In the case of topics, most of the participants (94%) selected "I agree", "I partly agree", and "I partly disagree" while in the case of "sentiment" most (82%) selected "I agree" or "I partly agree". Thus, for the sample of 100 we consider the extracted topics and sentiments to be appropriate (although not perfect) for clustering.

The clustering in accordance topics only ($\gamma = 1$) was judged as good with 84% of the answers in the range of "I fully agree" (22%), "I agree" (33%), and "I partly agree" (29%). This strong result might be due to the reason that grouping in accordance to topics is very common in the Web. Thus, the participants were already familiar with the outcome and no new way of interaction or perception was presented to them. A bit more controversial were the results in accordance to sentiment only ($\gamma = 2$). Here, the participants judged 68% as good (13% for "I fully agree", 43% for "I agree", and 12% for "I partly agree") while with 24% the judgments were negative ("I partly disagree", and "I disagree" each 12%). We assume this result to correlate with the general perception of the sentiment classification which was also judged not as accurate as topic assignments. Also, presenting sentiment classes is a very new and uncommon way of interaction. The uncertainty raised by this might be reflected in the weaker results. In general, the results of these two questions show that in either direction, the algorithm is capable of doing a reasonably good job.

The combined topic and sentiment clustering with the gamma value set to the values {0.75, 0.5, 0.25} received mixed feedback: 27% in the positive range, 49% in the negative range, and 14% for "Don't know / Can't tell". The question was whether the participant considers these options "useful". In fact, it is very unintuitive to blend two diversity aspects in such a way. However, as for the $\gamma = 0.5$ option, if we take two articles that do neither correlate sentiment nor topic-wise, they each get an own cluster and are presented to the user straight away. This might have an unsettling effect on the participants' perception as

¹³ These results include all meaningful comments and are presented unfiltered without correcting the wording or spelling.

the contrast of presented articles has not only been sharpened with respect to topics (this is what people are used to) but also with regard to sentiment.

The other group of questions focused on whether the participants would like to see a similar form of organization for their Tweets or Facebook posts and their favourite online news portal. The results are very diverse ranging from “I fully agree” (10%) to “I definitely disagree” (12%). The majority (23%) partly agrees. For the news portal, most (70%) participants agree in some form: 10% fully agree, 21% agree, and 39% partly agree. These numbers indicate that an extension of the scenario or adoption and trials by businesses should focus on the news use case rather than the Social Media one.

For the last mandatory question different answer options were provided with the option to select more than one. Most students would explain the shift from many clusters to few clusters either with

- “The number of different topics is much more diverse than options for sentiment” or
- “This is due to the disproportionate high number of neutral articles”.

Both answers are appropriate as, on the one hand, the (normalized) scores of the neutral articles are not varying strongly while, on the other hand, the topics and their similarities are much more diverse. Of course, this could change when a specific topic is pre-selected via the restrictions parameter of the ranking tool.

Finally, the participants were asked to provide comments and feedback also with respect to the preceding question. The answers were quoted in the previous section. Some of the answers describe appropriate feedback for the sentiment analysis as well as for the clustering while others are stark comments on why data and text processing in general is a very delicate and emotional topic.

4 Conclusion and Discussion

In this deliverable, two diversity-aware ranking tools were presented together with usage examples and evaluation of the results. The two tools operate at different levels and should be regarded as complementary and each of them can use techniques from the other. Thus they can be combined when necessary for a specific purpose and configuration.

A set of new features of the RENDER Diversity-Aware Ranking Service (described as prototype in D3.3.1 [9]), are presented like an HTML 5 Web interface, sentiment score normalization, and automatic algorithm selections. The Service uses efficient ranking algorithms and includes clustering that enables compact representation and visualization of larger amounts of posts, news articles or tweets. An extensive evaluation by users is presented with analysis and discussion of the results.

The CLAS OWLIM plug-in architecture and integration to OWLIM is presented in detail for the first time. The CLAS tools based on flexible and complex data selection and subsequent clustering and spreading activation for diversity-aware ranking are described. CLAS potential is demonstrated on the basis of several examples using a corpus of Google news articles, available at <http://rendernews.ontotext.com>.

The extension of CLAS based on enriched data models for CLAS outputs, addition of new algorithms and high quality semantic annotation is also discussed. A plan for development of new tools and adaptation of existing ones is proposed by Ontotext. This will allow exploring the full potential of CLAS for diversity-aware ranking and its evaluation by end users by the end of the project.

The RENDER Diversity-Aware Ranking Service and CLAS are tools aimed to operate at different levels of the Render technical architecture. The CLAS plug-in is designed to be part of the OWLIM services operating on RDF linked data and the RENDER Diversity-Aware Ranking Service provides specialized clustering and ranking algorithms related to the KDO based datasets and can provide a user interface to the ranking results obtained.

The integration of CLAS and the Service can easily be implemented, e.g. by accessing CLAS from the Service. CLAS on the other hand can add some of the clustering and ranking algorithms of the Service as new modules and extend their usage on richer datasets based on RDF graphs.

References

- [1] Reinier H. van Leuken, Lluís Garcia, Ximena Olivares, Roelof van Zwol: Visual Diversification of Image Search Results. Proceedings of the 18th international conference on World Wide Web (WWW '09), ACM, NY USA. 2009
- [2] Andreas Thalhammer, Ioan Toma, Antonio J. Roa-Valverde, Dieter Fensel (2012). Leveraging Usage Data for Linked Data Movie Entity Summarization. Proceedings of the 2nd International Workshop on Usage Analysis and the Web of Data (USEWOD2012) in the 21st International World Wide Web Conference (WWW2012), arXiv:1204.2718v1, Lyon, France, April 17th, 2012
- [3] Maurice Grinberg, Alex Simov, Simo Simov (2013). The CLAS plug-in for OWLIM: Spreading activation and clustering techniques for RDF ranking. In preparation.
- [4] Grinberg, M., Haltakov, V., Stefanov, H. (2010). Spreading Activation Mechanisms for Efficient Knowledge Retrieval from Large Datasets. In: Proceedings of WIRN 2010, COST 2102 Special session. IOS press.
- [5] Spreading activation components (v. 1-3) LackKC EC project deliverables D2.4.1, D2.4.2, and D2.4.3 <http://www.larkc.eu/resources/deliverables/>
- [6] Karypis, G. (2003). CLUTO: A Clustering Toolkit, <http://www.cs.umn.edu/~karypis/cluto/>.
- [7] Ying Zhao and Karypis, G. (2002). Criterion Functions for Document Clustering: Experiments and Analysis, technical report, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.6872>
- [8] Damova, M. and Russo, D. (2012). D1.1.3 Final Collection of Data. Render deliverable.
- [9] Andreas Thalhammer, A., Gagiou, A., Hangl, S. Toma, I., and Grinberg, M. (2012). D3.3.1 Prototype of diversity-aware ranking. Render deliverable.
- [10] N. Bell and M. Garland (2009). Implementing sparse matrix-vector multiplication on throughput-oriented processors, Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, New York, NY, USA: ACM, 2009, 1–11.
- [11] Harish, P. and Narayanan, P. (2007). Accelerating large graph algorithms on the GPU using CUDA, Proceedings of the High Performance Computing–HiPC 2007, Springer, LNCS, vol. 4873, 2007, 197–208.

Annex A CLAS Plug-in Example: Google News Corpus Analysis and Datasets Generation

The following prefixes should be used in the SPARQL queries presented below if needed.

SPARQL 25: Prefixes used in all SPARQL queries.

```
PREFIX clas: <http://www.ontotext.com/owlim/plugin/CLAS#>
PREFIX kdo: <http://kdo.render-project.eu/kdo#>
PREFIX pkm: <http://www.ontotext.com/proton/protonkm#>
PREFIX sioc: <http://rdfs.org/sioc/ns#>
PREFIX dbp-prop: <http://dbpedia.org/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

A.1 Statistics about dmoz labels (sioc:tag)

SPARQL 26: News articles and the respective number of dmoz labels for them.

```
SELECT ?s (count(?d) as ?dmoz_tags)
WHERE { ?s a kdo:NewsArticle ;
        sioc:tag ?d .
}
group by ?s
order by desc(?dmoz_tags)
```

Results for [PREFIX pkm:...](#) (100 of 297946)

s	dmoz_tags
urn:document-news-e6c1830d-b78a-49ec-95d2-b177743a7b02	19
urn:document-news-7a1e0868-e2aa-473b-88c4-abac7829b550	19
urn:document-news-fca04174-c376-437c-a025-7f96928d3885	19
urn:document-news-c875491d-26fe-46ac-871a-87be330875b8	18
urn:document-news-c0f14d82-4836-4b5a-b7d2-dd67eb498df9	18

SPARQL 27: Dmoz labels and number of news articles with that label.

```
SELECT ?d (count(distinct ?s) as ?news_per_dmoz_tags)
WHERE { ?s a kdo:NewsArticle ;
        sioc:tag ?d
}
group by ?d
order by desc(?news_per_dmoz_tags)
```

Results for [PREFIX pkm:...](#) (100 of 3571)

d	news_per_dmoz_tags
Regional	132102
Society	97341
North_America	93621
United_States	78734
Business	70369
Society_and_Culture	36889
Sports	35693
Issues	34941

A.2 Statistics about dmoz topics (sioc:topic)

SPARQL 28: News with dmoz labels and number of dmoz topics per news, for news which have DBpedia entities extracted.

```
SELECT ?s (count(distinct ?dt) as ?dmoz_topics_per_news)
WHERE {
  {SELECT ?s
   WHERE { ?s a kdo: NewsArticle> ;
            sioc:topic/pkm:refersToInstance ?db .}
   group by ?s
  }
  ?s sioc:topic ?dt .
  FILTER (!isBlank(?dt))
}
group by ?s
order by desc(?dmoz_topics_per_news)
```

Results for [PREFIX pkm:...](#) (100 of 36388)

s	dmoz_topics_per_news
urn:document-news-02b28ef3-de2f-4b58-9a93-d02efb7aefc2	11
urn:document-news-160c2d89-2173-4b6e-bde2-6b21c7120be9	11
urn:document-news-367dd81d-df45-4c7c-9969-c7cc7729178e	11
urn:document-news-5a05f998-7b5f-453b-ab25-509031ce6f64	11

SPARQL 29: dmoz labels and number of news articles per label, for news with DBpedia entities extracted.

```
SELECT ?dt (count(distinct ?s) as ?news_per_dmoz_topics)
WHERE {{SELECT ?s
        WHERE { ?s a kdo: NewsArticle> ;
                sioc:topic/pkm:refersToInstance ?db .}
        group by ?s
       }
       ?s sioc:topic ?dt .
```

```

        FILTER (!isBlank(?dt))
    }
    group by ?dt
    order by desc(?news_per_dmoz_topics)

```

Results for PREFIX pkm:... (100 of 6748)

dt	news_per_dmoz_topics
http://www.dmoz.org/Top/Games/Video_Games/Recreation/Brow...	2522
http://www.dmoz.org/Top/Games/Video_Games/Recreation	2055
http://www.dmoz.org/Top/Games/Video_Games/Browser_Based/C...	2036
http://www.dmoz.org/Top/Society/History/By_Region/North A...	1986

A.3 News with DBpedia URIs attached

SPARQL 30: Number of news articles with DBpedia entities extracted.

```

SELECT (count(?s) as ?news_number)
WHERE {
  {SELECT ?s (count(?dt) as ?dmoz_topics)
   WHERE { ?s a kdo: NewsArticle> ;
            sioc:topic/pkm:refersToInstance ?dt .
   }
  group by ?s
  order by desc(?dmoz_topics) }}

```

news_number

36390

SPARQL 31: News articles and number of DBpedia entities extracted.

```

SELECT ?s (count(distinct ?dt) as ?dburis_per_news)
WHERE { ?s a kdo: NewsArticle> ;
        sioc:topic/pkm:refersToInstance ?dt .
}
group by ?s
order by desc(?dburis_per_news)

```

Results for PREFIX pkm:... (100 of 36390)

s	dburis_per_news
urn:document-news-503ae6ad-9c24-4481-ac30-5a60f88f5cb6	133
urn:document-news-c7a9f734-db8c-4485-b0ab-a99def77a754	132
urn:document-news-2144a82f-f67e-46a7-9c70-3ef9a87fe67d	112
urn:document-news-357da306-1c89-4821-beaf-44b672a78ba4	109
urn:document-news-25556731-dc81-4233-9332-80515f924ecb	105
urn:document-news-98f2238d-7c5d-4fe8-84f5-e0bb24fe35e5	100
urn:document-news-f9cad3bd-1061-401f-9dc3-4eb84c67b2c9	92

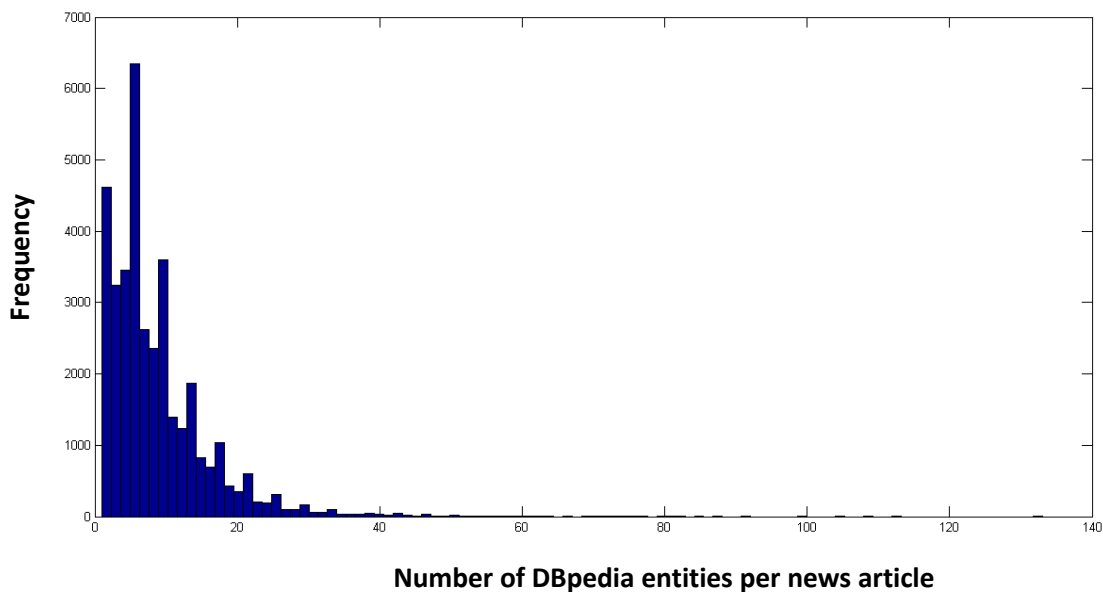


Figure 11: Frequency of the number DBpedia entity per news article.

SPARQL 32: News articles per DBpedia entities extracted.

```

SELECT ?dt (count(distinct ?s) as ?news_per_dburis)
WHERE { ?s a kdo: NewsArticle> ;
        sioc:topic/pkm:refersToInstance ?dt .
}
group by ?dt
order by desc(?news_per_dburis)
    
```


Results for PREFIX pkm:... (100 of 39759)

dt	news_per_dburis
dbr:U.S.	4148
dbr:Reuters	2237
dbr:United States	2215
dbr:Washington	2155
dbr:New York	2143
dbr:Congress	1659
dbr:Associated Press	1332
dbr:America	1331
dbr:Senate	1319

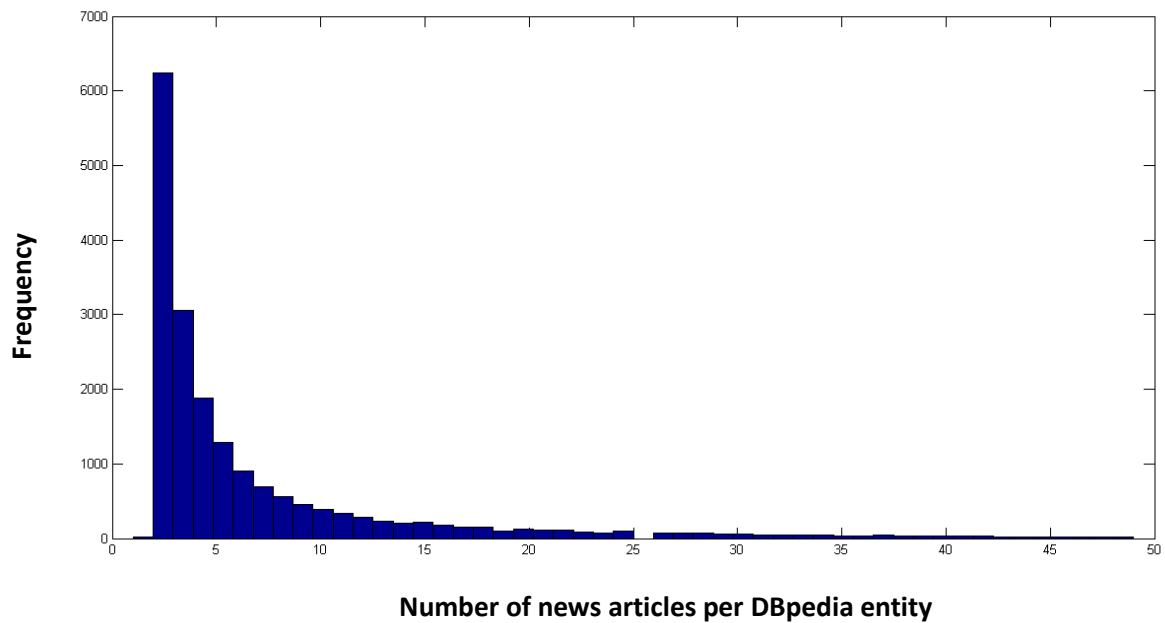


Figure 12: Frequency of the number news articles per DBpedia entity. (Only frequencies for the interval [2,50) are shown.)

Annex B CLAS Plug-in: Diversity-aware Ranking of Google News Articles Based on dmoz Labels (sioc:tag)

B.1 Dataset generation

SPARQL 33: Three SA iterations activating the DBpedia entities shared news similar to the seed.

```
SELECT ?s ?p
FROM <http://www.ontotext.com/owlim/plugin/CLAS#dump>
WHERE { ?s a kdo: NewsArticle ;
        sioc:tag ?p
}
```

NB: Press XML/JSON download to create and download the data

```
ASK { clas:L clas:hasParamValue "-zscores -showtree" }
```

```
ASK { clas:N clas:hasParamValue "1000" }
```

```
ASK { _:b1 clas:doClustering "direct" }
```

```
SELECT ?s ?cluster
WHERE { ?s a kdo: NewsArticle ;
        clas:inCluster ?cluster }
```

Results for [SELECT ?s...](#) (100 of 297946)

s	cluster
urn:document-news-1dff73f0-bccf-43b8-af56-a09ce9656fbc	900
urn:document-news-9e54e539-9ca2-4c44-9a56-4d0c81e0e2e4	315
urn:document-news-bcc92aba-2dd8-46f5-8600-5d4e9361e201	574
urn:document-news-6d97dca7-d230-4813-9485-469116c04d44	170
urn:document-news-a091c341-cfa1-4249-a255-2affa4269297	160
urn:document-news-f1aaeddd-abef-48d4-9410-4509d6db0347	491
urn:document-news-1899ff94-8c2c-42fe-9566-dce6b3381d3f	287
urn:document-news-945494cd-49a2-42b8-a57e-daf3a4188911	491

SPARQL 34: Generation of cluster No. 97.

```
SELECT ?s ?clusterDistance
WHERE { ?s a kdo: NewsArticle ;
        clas:inCluster 97 ;
        clas:hasClusterDistance ?clusterDistance }
```

Results for PREFIX... (47)

s	clusterDistance
urn:document-news-751a6243-dcdc-4cd9-8fef-d3a069691f15	0.420699
urn:document-news-6c616544-43d6-4e29-afe0-8b0324ef7a80	0.340957
urn:document-news-99fcd91a-8b74-4f5b-8b85-bc7e3fa7178c	0.340957
urn:document-news-6b1c493e-c60d-4d13-9428-f4272c5b344d	0.369103
urn:document-news-4a83b158-283a-4bad-89f5-3b00fa9937b4	0.43762
urn:document-news-3f0ddaf2-2853-43ac-9ccf-0a4cde926001	0.43762
urn:document-news-71120948-f37c-4504-8864-0a9210513df1	0.341424
urn:document-news-6a99fb40-a561-4148-9490-48d160ea3caa	0.340957

SPARQL 35: SA and similarity evaluation from a node in cluster No. 97.

```
ASK { [] clas:activateNode
      <urn:document-news-751a6243-dcdc-4cd9-8fef-d3a069691f15> }
```

```
ASK { [] clas:setClusterFocus
      <urn:document-news-751a6243-dcdc-4cd9-8fef-d3a069691f15> }
```

```
SELECT ?news ?activity ?similarity ?typicality ?cluster
WHERE { ?news a kdo:NewsArticle ;
          clas:hasActivity ?activity ;
          clas:inCluster ?cluster ;
          clas:hasClusterFocusDistance ?similarity ;
          clas:hasClusterDistance ?typicality .
        FILTER ( ?similarity > "0.0"^^xsd:double )
      }
order by DESC(?similarity)
```

Results for PREFIX... (47)

View as [Exhibit](#) Download

news	activity	similarity	typicality	cluster
urn:document-news-751...	9.0	1.0	0.420699	97
urn:document-news-6d8...	8.0	1.0	0.420699	97
urn:document-news-f2...	8.0	0.9428090415820635	0.334971	97
urn:document-news-6c6...	7.0	0.9354143466934853	0.340957	97
urn:document-news-99f...	7.0	0.9354143466934853	0.340957	97
urn:document-news-4a8...	7.0	0.9354143466934853	0.43762	97
urn:document-news-3f0...	7.0	0.9354143466934853	0.43762	97
urn:document-news-6a9...	7.0	0.9354143466934853	0.340957	97

SPARQL 36: SA and similarity evaluation from a node in cluster No. 97.

```
SELECT ?news ?similarity ?typicality
WHERE { ?news a kdo:NewsArticle ;
          clas:hasClusterFocusDistance ?similarity ;
```

```

        clas:hasClusterDistance ?typicality .
        FILTER ( ?similarity > "0.0"^^xsd:double )
    }
    order by ASC(?similarity)

```

Results for PREFIX_ (47)

Vi

news	similarity	typicality
urn:document-news-114cde95-ac03-462f-...	0.75	-3.066795
urn:document-news-c9f9f959-7c21-4226-...	0.75	-3.066795
urn:document-news-778a15b0-f468-4b90-...	0.7826237921249264	-2.033721
urn:document-news-933a2b0a-159a-460c-...	0.8249579113843055	-0.995104
urn:document-news-c6a6ea1f-ec24-4961-...	0.8249579113843055	-2.607712
urn:document-news-6b1c493e-c60d-4d13-...	0.8660254037844387	0.369103
urn:document-news-0996227c-b171-4c30-...	0.8660254037844387	0.369103
urn:document-news-40892012-9ff4-44f8-...	0.8660254037844387	0.369103
urn:document-news-bac0ac4e-4edb-40fd-...	0.8660254037844387	0.369103
urn:document-news-7fb22952-269c-448d-...	0.8660254037844387	0.369103

SPARQL 37: SA and similarity evaluation from a node in cluster No. 97.

```

SELECT ?news ?activity ?cluster
WHERE {
    ?news a kdo:NewsArticle ;
        clas:hasActivity ?activity ;
        clas:inCluster ?cluster
    FILTER ( ?activity > "3"^^xsd:double )
}
order by DESC(?activity)

```

news	activity	cluster
urn:document-news-751a6243-dcdc-4cd9-...	9.0	97
urn:document-news-d0ce6981-e3a5-462b-...	8.0	500
urn:document-news-cb072f83-e71f-47fd-...	8.0	637
urn:document-news-3c47a62e-c3a4-4a74-...	8.0	226
urn:document-news-f6ea8917-1e64-4f70-...	8.0	206
urn:document-news-49daaacf-5a0d-4eb7-...	8.0	206
urn:document-news-b5e640ca-e9ea-4d35-...	8.0	638
urn:document-news-350f6089-4e48-40a0-...	8.0	316
urn:document-news-dc514ad5-2bc8-48d4-...	8.0	638

SPARQL 38: SA and similarity evaluation from a node in cluster No. 97.

```

SELECT ?cluster ?avg_activity_per_cluster ?sum_news

```

```

WHERE {
  {SELECT ?cluster (count(?news) as ?sum_news)
             (sum(?activity) as ?sum_activity)
   WHERE {?news a kdo:NewsArticle ;
            clas:hasActivity ?activity ;
            clas:inCluster ?cluster
          }
   group by ?cluster
  }
  BIND((?sum_activity/?sum_news) as ?avg_activity_per_cluster)
  FILTER(?avg_activity_per_cluster > "1.0"^^xsd:double)
}
order by desc(?avg_activity_per_cluster)
    
```

cluster	avg_activity_per_cluster	sum_news
97	6.7021275	47
206	5.980769	52
374	5.904762	42
227	5.3012047	83
637	3.1329114	316

Annex C Node Selection Based Spreading Activation (NSbSA)

This Annex summarizes, closely following [4], the approximate SA algorithm used in the CLAS for all the examples presented.

C.1 Standard SA

NSbSA is obtained from the original knowledge dataset by extracting a connectivity matrix. This matrix can be derived off-line or on-line SPARQL queries.

To introduce NSbSA, it is useful to first introduce the basic notation and rules for the standard SA case. SA requires a vector with initial activities (input vector) and a connectivity matrix (weight matrix) in which each connection corresponds to a predicate from the dataset and is taken equal to 1 for simplicity, although weights different from 1 can be introduced as well. If we assume that the activities of the query elements are 1's and that they are part of the dataset, SA can be implemented as follows:

- $\mathbf{t}(n_n) = \mathbf{I}_{n_n}[k_{i_1}, k_{i_2}, \dots, k_{i_n}]$, is the query (target) represented as a vector, where n is the number of nodes in the query, which are identified within the dataset and $\mathbf{I}_{n_n}[l_1, l_2, \dots, l_n]$ is defined as a n_n -dimensional vector with n elements equal to 1, with indexes listed within the square brackets, and zeros elsewhere;
- $\mathbf{k}(n_n)$ is a vector of all n_n nodes' activations for the dataset;
- $\mathbf{W}(n_n \times n_n)$ is the nodes' connectivity matrix for all nodes in the dataset, with 1's as weights for any statement connecting two nodes.

SA is schematically given by the following iteration process (assuming identity activation function and no decay):

$$\begin{aligned}
 \mathbf{t}(n_n) &= \mathbf{I}_{n_n}[k_{i_1}, k_{i_2}, \dots, k_{i_n}] \\
 \mathbf{k}^{(0)}(n_n) &= \mathbf{t}(n_n) \\
 \mathbf{k}^{(1)}(n_n) &= \mathbf{W}(n_n \times n_n) \mathbf{k}^{(0)}(n_n) \\
 \mathbf{k}^{(2)}(n_n) &= \mathbf{W}(n_n \times n_n) \mathbf{k}^{(1)}(n_n)
 \end{aligned} \tag{1}$$

$$\mathbf{k}^{(it)}(n_n) = \mathbf{W}(n_n \times n_n) \mathbf{k}^{(it-1)}(n_n) = [\mathbf{W}(n_n \times n_n)]^{it} \mathbf{k}^{(0)}(n_n)$$

where $\mathbf{t}(n_n)$ is the initial query (considered to be the target for the retrieval); and \mathbf{W} is the nodes' connectivity matrix. $\mathbf{k}^{(it)}(n_n)$ is the vector with nodes activated after iteration it .

The SA process given by Eq. (1) involves matrix-vector multiplication which for large matrices is highly resource demanding.

C.2 SA as Non-Zero Elements Search

Instead of matrix-vector multiplication used in exact SA approaches, NSbSA implements SA by searching for non-zero elements in a sparse vector. This is a process of following the connections of the nodes from the query (the seeds), finding the nodes they are connected to, then repeat this procedure with the newly found nodes. This will lead to newly selected nodes and the process is repeated iteratively. The efficiency of this method, apart from the possibility for efficient implementation and storage (see [4] for discussion and evaluation), is related to the use of each node as a source of activation only one time. Once a node has been used for SA, its connections are never used again. This is not the case in standard SA, in which all the connections of active nodes are used over and over again. The latter of course is computationally extremely resource demanding. In NSbSA, the process of node selection can come across a node which has been selected in a previous iteration. In this case, a counter attached to each node is incremented. The larger the value of this counter is, the more relevant the node is considered to be. The number of times a node is selected is a measure of the number of independent paths to this node from the query nodes.

The above approach can be formalized similarly to standard SA, given by Eq. (1). The process starts, as in SA, as an iteration process. The nodes from the query which are initially the only selected nodes are stored in a vector with dimension equal to the number of nodes in the dataset with 1's for the nodes in the query and zeros otherwise. The nodes' connectivity matrix contains only 1's. The process starts with the nodes of the query. For each of them the corresponding rows in the connectivity matrix is searched for non-zero elements. The non-zero elements point to other nodes. These nodes are taken as seeds and the process is repeated until the preset number of iterations is reached. At each iteration step, the new nodes found are compared to the previously selected. Only the new nodes are used to proceed further. If a node has been already selected, a counter for this node is incremented so that the node is not used in subsequent iterations, as its connections have been already followed.

This process can be expressed by an iteration process similar to the one expressed in Eq. (1):

$$\begin{aligned}
\mathbf{t}(n_n) &= \mathbf{I}_{n_n} [k_{t_1}, k_{t_2}, \dots, k_{t_n}] \\
\mathbf{f}^{(0)}(n_n) &= \mathbf{t}(n_n) \\
\mathbf{k}^{(0)}(n_n) &= \mathbf{t}(n_n) \\
\mathbf{f}^{(1)}(n_n) &= \mathbf{f}^{(0)}(n_n) + \sum_{c=1}^{n_n} \mathbf{W}^T(\mathbf{k}^{(0)}(n_n), c) \\
\mathbf{k}^{(1)}(n_n) &= \Theta \left(\sum_{j=1}^{n_n} \mathbf{W}^T(\mathbf{k}^{(0)}(n_n), j) \right) - \mathbf{k}^{(0)}(n_n) \\
&\dots \\
\mathbf{f}^{(it)}(n_n) &= \mathbf{f}^{(it-1)}(n_n) + \sum_{j=1}^{n_n} \mathbf{W}^T(\mathbf{k}^{(it-1)}(n_n), j) \\
\mathbf{k}^{(it)}(n_n) &= \Theta \left(\sum_{j=1}^{n_n} \mathbf{W}^T(\mathbf{k}^{(it-1)}(n_n), j) \right) - \sum_{i=0}^{it-1} \mathbf{k}^{(i)}(n_n) \\
&\text{or} \\
\mathbf{k}^{(it)}(n_n) &= \Theta \left(\sum_{j=1}^{n_n} \mathbf{W}^T(\mathbf{k}^{(it-1)}(n_n), j) \right) - \Theta(\mathbf{f}^{(it-1)}(n_n))
\end{aligned} \tag{2}$$

where $\mathbf{t}(n_n)$ is the initial query (considered to be the target for the retrieval); $\mathbf{W}^{(0)}$ is the original nodes' connectivity matrix; the notation $\mathbf{W}^T(\mathbf{k}^{(it-1)}(n_n), j)$ means taking the row indexes corresponding to the non-zero elements of $\mathbf{k}^{(it-1)}(n_n)$ and the summation over j is summation over the columns of the nodes' connectivity matrix \mathbf{W} ; Θ is a Heavyside function producing a vector (matrix) with the same dimension as the argument vector (matrix) but with elements equal to 1 for any element of the argument larger than 0, and 0 otherwise; $\mathbf{f}^{(it)}(n_n)$ is accumulating the nodes that have been activated and the number of times it happened via independent connections (each connection is used only once).

C.3 NSbSA Algorithm

The nodes' connectivity matrix is stored using the CRS (Compressed Row Storage) format (e.g. see [10]), which allows for a very efficient (constant in time) retrieval of the non-zero elements in a specified row, which is crucial for the good performance of the described algorithm. The algorithm implementing NSbSA (see Eq. (2)), is a modified breadth-first-search algorithm [11] (see

Table 2). In the deliverable, it is implemented of for a MP CPU.

In

Table 2, $G(N, L)$ is the graph corresponding to the matrix of connections with N nodes and L connections. It works iteratively with a frontier array that holds the nodes to be processed on the next iteration. The processing is divided between a grid of threads. The algorithm makes use of one Boolean array as a frontier, an integer array to store the result V and another array to hold all the currently processed links D .

Table 2: Modified breadth-first-search algorithm implementing NSbSA according to Eq. (2).

NSbSA Algorithm	
CUDA_SA (Graph $G(N, L)$, Source Vertex S , Iterations I)	CUDA_SA_KERNEL($D, F, V, csize$)
1: Create index array X from all nodes in $G(N, L)$ 2: Create links array P from all links in $G(N, L)$ 3: Create buffer array D 4: Create frontier array F and results' array V , both of size N 5: Initialize F to <i>false</i> 6: Initialize V to 0 7: $F[S] \leftarrow true$ 8: for $i = 1$ to I do 9: for $nid = 1$ to N do 10: if $F[nid]$ then 11: Append links ($P[X[nid]]$; $P[X[nid+1]]$) to D array 12: end if 13: end for 14: $F \leftarrow false$ 15: Copy F, D to device 16: Compute chunk size $csize$ from the size of D 17: Invoke CUDA_SA_KERNEL($D, F, V, csize$) on grid 18: Copy F back to host 19: end for	1: $tid \leftarrow getThreadID$ 2: $first \leftarrow D[tid * csize]$ 3: $last \leftarrow first + csize$ 4: $nid = first$ 5: while $nid < last$ do 6: if NOT $V[nid]$ then 7: $F[nid] \leftarrow true$ 8: end if 9: $V[nid] \leftarrow V[nid] + 1$ 10: $nid \leftarrow nid + 1$ 11: end while

Before each iteration, the frontier is examined and only the links for the corresponding nodes are copied to the device. If the memory on the device is not sufficient for the links, multiple transactions are performed. At each iteration, each thread operates on its own chunk of nodes. If a node is activated for the first time, it is scheduled for activating on the next iteration by marking it in the frontier array. When a node is activated, its activation count in V is incremented. The process stops when the desired number of iterations is reached or no more nodes are candidates for activation. In this algorithm, performance drops down when there are multiple transactions between the host and the CUDA¹⁴ enabled device or due to memory latencies.

¹⁴ <http://en.wikipedia.org/wiki/CUDA>