NeOn: Lifecycle Support for Networked Ontologies

Integrated Project (IST-2005-027595)

Priority: IST-2004-2.4.7 – "Semantic-based knowledge and content systems"

D3.7.1 Context sensitive visualization of multiple ontologies (joint with D4.3.1)

Deliverable Co-ordinator: **Dunja Mladenić**

Deliverable Co-ordinating Institution: **J. Stefan Institute (JSI)**

Other Authors: **Blaž Fortunar (JSI), Marko Grobelnik (JSI), Martin Dzbor (OU)**

This deliverable provides a software prototype that enables context sensitive visualization of ontologies. The idea is to enable the user to gain a broad overview and abstract insight into the mesh of networked ontologies by projecting an ontology onto different landscapes and projecting different ontologies onto the same landscape. The software prototype also enables the generation of different landscapes based on a document collection and personalized styling of landscapes. Furthermore the generated landscape can be stored for later usage by other users. The system enables loading of arbitrary ontologies as well as predefined landscapes and the visualization of the ontology onto the landscape.

Document Identifier:	NEON/2008/D3.7.1/v1.0	Date due:	February 29, 2008
Class Deliverable:	NEON EU-IST-2005-027595	Submission date:	February 29, 2008
Project start date:	March 1, 2006	Version:	v1.0
Project duration:	4 years	State:	Final
		Distribution:	Public

NeOn Consortium

This document is a part of the NeOn research project funded by the IST Programme of the Commission of the European Communities by the grant number IST-2005-027595. The following partners are involved in the project:

Open University (OU) – Coordinator Knowledge Media Institute – KMi Berrill Building, Walton Hall Milton Keynes, MK7 6AA United Kingdom Contact person: Martin Dzbor, Enrico Motta E-mail address: {m.dzbor, e.motta} @open.ac.uk	**Universität Karlsruhe – TH (UKARL)** Institut für Angewandte Informatik und Formale Beschreibungsverfahren – AIFB Englerstrasse 11 D-76128 Karlsruhe, Germany Contact person: Peter Haase E-mail address: pha@aifb.uni-karlsruhe.de
Universidad Politécnica de Madrid (UPM) Campus de Montegancedo 28660 Boadilla del Monte Spain Contact person: Asunción Gómez Pérez E-mail address: asun@fi.upm.es	**Software AG (SAG)** Uhlandstrasse 12 64297 Darmstadt Germany Contact person: Walter Waterfeld E-mail address: walter.waterfeld@softwareag.com
Intelligent Software Components S.A. (ISOCO) Calle de Pedro de Valdivia 10 28006 Madrid Spain Contact person: Jesús Contreras E-mail address: jcontreras@isoco.com	**Institut 'Jožef Stefan' (JSI)** Jamova 39 SI-1000 Ljubljana Slovenia Contact person: Marko Grobelnik E-mail address: marko.grobelnik@ijs.si
Institut National de Recherche en Informatique et en Automatique (INRIA) ZIRST – 655 avenue de l'Europe Montbonnot Saint Martin 38334 Saint-Ismier France Contact person: Jérôme Euzenat E-mail address: jerome.euzenat@inrialpes.fr	**University of Sheffield (USFD)** Dept. of Computer Science Regent Court 211 Portobello street S14DP Sheffield United Kingdom Contact person: Hamish Cunningham E-mail address: hamish@dcs.shef.ac.uk
Universität Koblenz-Landau (UKO-LD) Universitätsstrasse 1 56070 Koblenz Germany Contact person: Steffen Staab E-mail address: staab@uni-koblenz.de	**Consiglio Nazionale delle Ricerche (CNR)** Institute of cognitive sciences and technologies Via S. Martino della Battaglia, 44 - 00185 Roma-Lazio, Italy Contact person: Aldo Gangemi E-mail address: aldo.gangemi@istc.cnr.it
Ontoprise GmbH. (ONTO) Amalienbadstr. 36 (Raumfabrik 29) 76227 Karlsruhe Germany Contact person: Jürgen Angele E-mail address: angele@ontoprise.de	**Food and Agriculture Organization of the United Nations (FAO)** Viale delle Terme di Caracalla 1 00100 Rome Italy Contact person: Marta Iglesias E-mail address: marta.iglesias@fao.org
Atos Origin S.A. (ATOS) Calle de Albarracín, 25 28037 Madrid Spain Contact person: Tomás Pariente Lobo E-mail address: tomas.parientelobo@atosorigin.com	**Laboratorios KIN, S.A. (KIN)** C/Ciudad de Granada, 123 08018 Barcelona Spain Contact person: Antonio López E-mail address: alopez@kin.es

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed to the writing of this document or its parts: JSI, OU.

Change Log

Version	Date	Amended by	Changes
0.1	05-02-2008	Dunja Mladenic	Overall structure of the report, abstract
0.2	7-02-2008	Dunja Mladenic	Approach description, summary
0.2	15-02-2008	Blaz Fortuna	Example usage
0.3	25-02-2008	Marko Grobelnik	Overall refinements
0.4	17-03-2008	Blaz Fortuna	Discussion
0.5	30-03-2008	Blaz Fortuna	Revision of Section 3
0.6	1.04.2008	Dunja Mladenic	Overall refinements (review comments)
0.7	2.04.2008	Marko Grobelnik	Finalization
1.0	3.04.2008	Dunja Mladenic	Overall refinements (review comments)

Executive Summary

Ontologies can be compared in different ways; one of them is by visualizing the content they cover. The main idea behind the proposed approach to context sensitive visualization of ontologies is the existence of a landscape (context) defined by the domain of interest onto which we can project ontologies. In that process we abstract from the structure of the ontology and focus on its content and coverage of different domain content as provided in the landscape. One can imagine that having a landscape that covers a rather broad topic area enables comparison of general as well as specific ontologies; the former covering a whole landscape and the latter covering just a part of it. A landscape that is focused on a specific topic would be of limited value when comparing general ontologies, as they would only be compared in the part that overlaps with the specific topic of the landscape.

The idea of landscapes is operationalized here by providing a tool for landscape generation based on a collection of documents describing the landscape topic. Once the landscape is automatically generated and semi-automatically personalized by tuning several parameters, it is stored in a library of predefined landscapes. The developed prototype for context sensitive visualization enables loading of a predefined landscape and an arbitrary ontology. The system then automatically projects the ontology onto the landscape and enables browsing through the visual representation of the projection.

The prototype is currently a standalone system implemented on the top of the TextGarden C++ library. In the next phase it will become a loosely integrated Neon toolkit plug-in. The prototype can be downloaded from NeOn website.

The developed approach to context sensitive visualization of ontologies is a joint deliverable with WP4 (D4.3.1, report) where context and personalized visualization are put together.

Table of Contents

List of Figures

1. Introduction

Ontologies can be compared in different ways; one of them is by visualizing the content they cover while abstracting their structure. Content of ontologies is represented by names of concepts and relations as well as instances. Here we focus on using the textual part of ontology representation and providing a visualization of that text.

Visualization is commonly used in data analysis to help the user in getting an initial idea about the raw data as well as a visual representation of the regularities obtained in the analysis. In the case of textual data, visualization can contribute to the understanding, discovery and summarization of information captured in text. In this deliverable not only the ontology is abstracted to textual data but also the context that is used to provide the context sensitive visualization. We have introduced term landscape here for context in which the ontology is visualized. Landscapes are pre-calculated based on a document collection that captures content of the landscape that we want to use.

The rest of this deliverable describes the proposed approach, example usage of the system and provides discussion.

2. Approach Description

The main idea behind the proposed approach to context sensitive visualization of ontologies is the existence of a landscape (context) defined by the domain of interest onto which we can project ontologies. In that process we abstract from the structure of the ontology and focus on its content. Visualization is based on showing how the ontology content covers a domain of interest as captured in the landscape.

Landscapes are generated from a document collection describing the domain of interest. Once the landscape is automatically generated, it can be semi-automatically personalized via tuning several parameters, such as level of detail and font size. All the landscapes are stored in a library of predefined landscapes that can be shared by different users. Moreover via personalization we can have several landscapes describing the same domain but each personalized to a specific user's (or group of users') needs and preferences.

The developed prototype for context sensitive visualization enables loading of a predefined landscape and an arbitrary ontology provided in RDF format. The system then automatically projects the ontology onto the landscape and enables browsing through the visual representation of the projection.

2.1 Landscape generation

The first step of our approach to landscape generation is to map all the documents into a two dimensional vector space so we can plot them on a computer screen. Ideally they would be positioned in such a way that the distance between two documents would correspond to the content similarity between them. In the next subsections we give a sequence of methods that transform a text document into a two dimensional point.

2.1.1 Document representation

The most often used representation of text documents, which can be successfully used for topical modeling, is the bag-of-words (Salton 1991) representation. In this representation the order of the words in a document is ignored and only the presence or count of the words in the document is considered. More technically, documents are represented as vectors in a high-dimensional vector space, where each dimension corresponds to one word from the vocabulary. The value of a

specific dimension in the document vector reflects the count of appearances of the word in the corresponding document.

The similarity of two documents that are represented as vectors can be calculated using the cosine similarity measure, which is the cosine of the angle between the two vectors. Effectively this means that the more words the two documents have in common, the more similar they are to each other. Cosine similarity is a standard measure between documents used in the field of information retrieval.

However, not all the words are of equal importance for determining the similarity between two documents, e.g. two documents sharing word "the" are not necessarily topically similar. By weighting the elements of the vectors we can bias the cosine similarity so it disregards topically non-related words (e.g. "the", "and", etc.) and put more emphasis on the topically significant words. In the implementation we used the standard TFIDF weights where each element of the vectors (the count of the word or TF – term frequency) is multiplied by the importance of the word with regards to the whole corpus (the IDF weight – inverted document frequency). The IDF weight for the *i*-th word is defined as $IDF_i = \log(N/df_i)$, where N is total number of documents and df_i is the document frequency of the *i*-th word (the number of documents from the whole corpus in which the *i*-th word appears). The IDF weight was found to put lower weight on common words (the ones that appear in almost all the documents) and favor the words that appear scarcely in the documents.

2.1.2 Latent semantic indexing

A well known and used approach for extracting latent semantics (or topics) from text documents is Latent Semantic Indexing (Deerwester et al. 1990). In this approach we first construct a term-document matrix A from a given corpus of text documents. This is a matrix with vectors of documents from a given corpus as columns. The term-document matrix A is then decomposed using singular value decomposition, so that $A = USV^T$; here matrices U and V are orthogonal and S is a diagonal matrix with ordered singular values on the diagonal. Columns of matrix U form an orthogonal basis of a subspace in the bag-of-words space where vectors with higher singular values carry more information -- this follows from the basic theorem about SVD, which tells that by setting all but the largest k singular values to *0* we get the best approximation for matrix A with matrix of rank k).

Vectors that form the basis can be also viewed as concepts and the space spanned by these vectors is called the *Semantic Space*. Each concept is a vector in the bag-of-words space, so the elements of this vector are weights assigned to the words coming from our documents. The words with the highest positive or negative values form a set of words that is found most suitable to describe the corresponding concept.

2.1.3 Multidimensional scaling

Multidimensional scaling (Carroll and Arabie 1980) enables dimensionality reduction by mapping original multidimensional vectors onto two dimensions. Here the points representing documents are positioned into two dimensions so they minimize some energy function. The basic and most common form of this function is

$$E = \sum_{i \neq j} \delta_{ij} - d(x_i, x_j))^2, \qquad (1)$$

where x_i are two dimensional points, $d(x_i, x_j)$ denotes Euclidian distance between the two points (we will refer to this with d_{ij}) and δ_{ij} represents the similarity between two vectors (in our case documents *i* and *j*). An intuitive description of this optimization problem is: the better the distances between points on the plane approximate real similarity between documents, the lower the value of the energy function E. Notice that function E is nonnegative and equals zero only when distances between points match exactly the similarity between documents.

A common way of applying multidimensional scaling is by using gradient descent for the optimization step. The problem with this approach is that the energy function is not convex: it usually has many local minima which are not that interesting for us. One could start this method repeatedly with different initial states and then choose the results with the lowest energy.

We choose a slightly different approach which is based on reformulation of the energy function. Given a placement of points, we calculate for each point how to move it to minimize the energy function. We denote the current positions of points with (x_i,y_i) and the desired position with $(x_i',y_i') = (x_i + \delta x_i, y_i + \delta y_i)$. Then we have

$$d_{ij}'^2 - d_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2 - (x_i + \delta x_i - x_j - \delta x_j)^2 + (y_i + \delta y_i - y_j - \delta y_j)^2 \approx$$

$$\approx (x_i - x_j)\,\delta x_i + (x_j - x_i)\,\delta x_j + (y_i - y_j)\,\delta y_i + (y_j - y_i)\,\delta y_j =$$

$$= [(x_i - x_j), (x_j - x_i), (y_i - y_j), (y_j - y_i)][\,\delta x_i, \delta x_j, \delta y_i, \delta y_j]^T.$$

By writing this for each pair (i,j) and substituting d_{ij}' with the original distance δ_{ij} between i-th and j-th document we get a system of linear equations which has a vector of moves (δx and δy) for a solution. This is an iteration which finds a step towards minimizing the energy function and is more successful at avoiding local minima. Each iteration involves solving a linear system of equations with a very sparse matrix. This can be done very efficiently using the Conjugate Gradient (CG) method. Finally, the points are normalized to lie in the square $K = [0,1]^2$.

2.1.4 Visualization using dimensionality reduction

The proposed visualization approach is based on a sequential combination of linear subspace methods and multidimensional scaling for reducing document space dimensionality. Both methods can be independently applied to any data set that is represented as a set of vectors in some higher dimensional space. Our goal is to lower the number of dimensions to two so that the whole corpus of documents can be shown on a computer screen.

Linear subspace methods like Latent Semantic Indexing focus on finding projections from the original bag-of-words space into a lower dimensional space while trying to preserve as much information as possible. By projecting data (text documents) only on the first two directions we can get the points that live in the two dimensional space. The problem with linear subspace methods is that only the information from the first two directions is preserved. In case of LSI it would mean that all documents are described using only the two main concepts.

In (Fortuna et al. 2006) we have proposed combining the two methods (linear subspace and multidimensional scaling) in order to take advantage of both. What follows is a description of that algorithm:

Input: Corpus of documents to visualize in form of TFIDF vectors.

Output: Set of two dimensional points representing documents.

Procedure:

1. Calculate k dimensional semantic space generated by input corpus of documents,
2. Project documents into the semantic space,
3. Apply multidimensional scaling using energy function on documents with Euclidian distance in semantic space as similarity measure.

There is a parameter k in the first step which has to be given to this procedure. In order to select this parameter we use following heuristic. Let $\Sigma_k = S_1^2 + S_2^2 + \ldots + S_k^2$, where S_i is i-th singular value. We know that $\Sigma_n = \mathrm{Trace}(A^TA)$, where n is the number of the documents in the corpus and A is the term-document matrix. From this we can guess k by prescribing the ratio Σ_k / Σ_n to some fixed value (e.g. 75% which was found to produce the best results).

2.2 Projecting ontology onto semantic landscape

The procedure defined in the previous sections can be used to make a map of topics from a given collection of documents. The map is calculated for each document collection separately with no connection to the other document collections. In our case we would like to visualize a given ontology on a predefined set of topics, e.g. for comparing two ontologies on a predefined set of topics. To handle such scenarios we developed a notation of *semantic landscape* which defines the set of topics over which the ontology will be visualized. Essentially it is a pre-given set of documents, which talk about the desired topics (e.g. biology, fishing industry, pharmaceutical products) projected onto a two dimensional map. The documents which define the semantic landscape are called *landmarks*.

Notice that the landmarks do not appear in the visualization, they are just used for positioning the given collection of documents on the map as follows. In order to position a document on a map, the top N most similar landmarks are calculated and the position of the document is then a convex combination of these landmarks' positions. More specifically, let μ_i be the cosine similarity of the document to the i-th most similar landmark and A_i be the location of the landmark on the 2D map. Than the location of the document is calculated as

$$\left(\frac{1}{\sum_{i=1}^{N}\mu_i}\right)\sum_{i=1}^{N}\mu_i A_i. \qquad (2)$$

The positioning is computationally very inexpensive which in turn means that once the semantic landscape is calculated the visualization of new documents can be done in near real time.

The semantic landscape can be also be used for a more efficient visualization of larger corpora, when the multidimensional scaling (MDS) algorithm would take too long to finish. In such cases a semantic landscape is generated from landmarks, which are extracted from the corpora. In our experiments we used a *k*-means (Jain and Flynn 1999) clustering algorithm for clustering the corpora in a larger number of clusters, usually several hundreds. Since *k*-means is more efficient than MDS, this significantly improves the time performance.

2.3 Visualization enhancements

Several additions, besides the positions of ontology concepts, can be added to the visualization to increase the amount of information the user can get from it at a fast glance or while exploring the map. First of all, the background can be used to depict, for example, the density of landscape documents or ontology documents in a particular part of the map. Second, the content of the landscape documents can be used for generating and displaying the keywords which are relevant for a particular part of the map.

The background depicting the density can be either generated based on the documents which were provided for the background landscape or the ontology's concepts. The density is estimated as follows. Each point from the landscape is assigned height using the formula

$$h(x, y) = \sum_i \exp(-\sigma \, ||(x,y) - (x_i, y_i)||^2), \qquad (3)$$

where σ defines how wide is the influence of one point.

Similar to density, a set of most important keywords can be assigned to each point of the map. The keywords are selected by averaging the TFIDF vectors of documents which appear within a predefined distance to the point.

Currently, the keywords are used in two distinct scenarios. First, when the user moves the mouse to a specific point, the list of most important keywords, calculated in real-time, is displayed (Figure 1). Second, for each point from a set of random points, uniformly distributed over the map, the

most important keyword is computed and displayed on the map, using white font. These keywords help the user to see the main topics by just glancing over the map. We call these keywords *common words*.

There are several parameters which control the calculation and appearance of keywords. First is the radius, defining which documents are used when averaging the TFIDF vectors. In the case of a dynamically calculated list of keywords, the radius is selected using the mouse scroll wheel. The dark circle shows the area from which the keywords are extracted. Figure 2 shows the effect of this parameter. Second, the density of common words is controlled by sparsity parameter, which defines the minimal distance between any two common words. And finally, the minimal distance between repeating common words can also be specified. Figure 3 shows the effect of the parameter. These three parameters can be given live through the GUI or can be predefined when the landscape or ontology visualization is created.

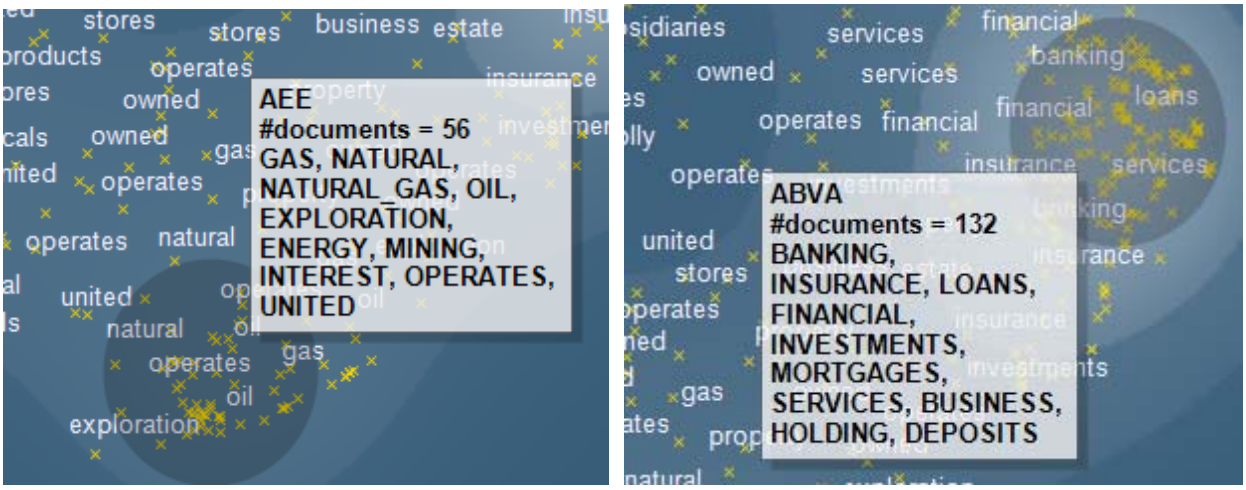

Figure 1. List showing the main keywords for the area marked with the dark circle.

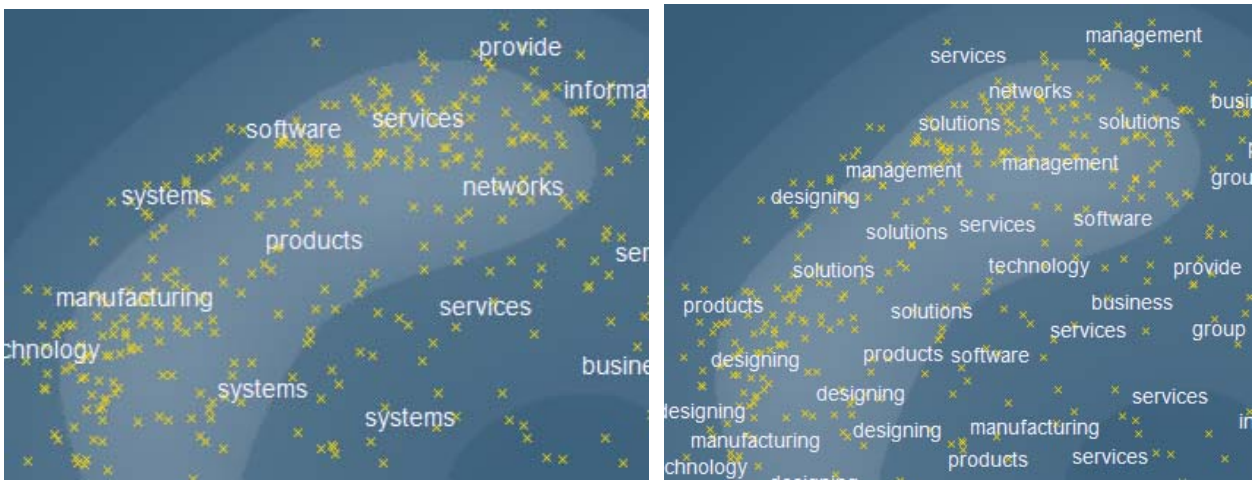

Figure 2. Effect of the sparsity parameter. Common words on the right are more densely spread around the map than the ones on the left.

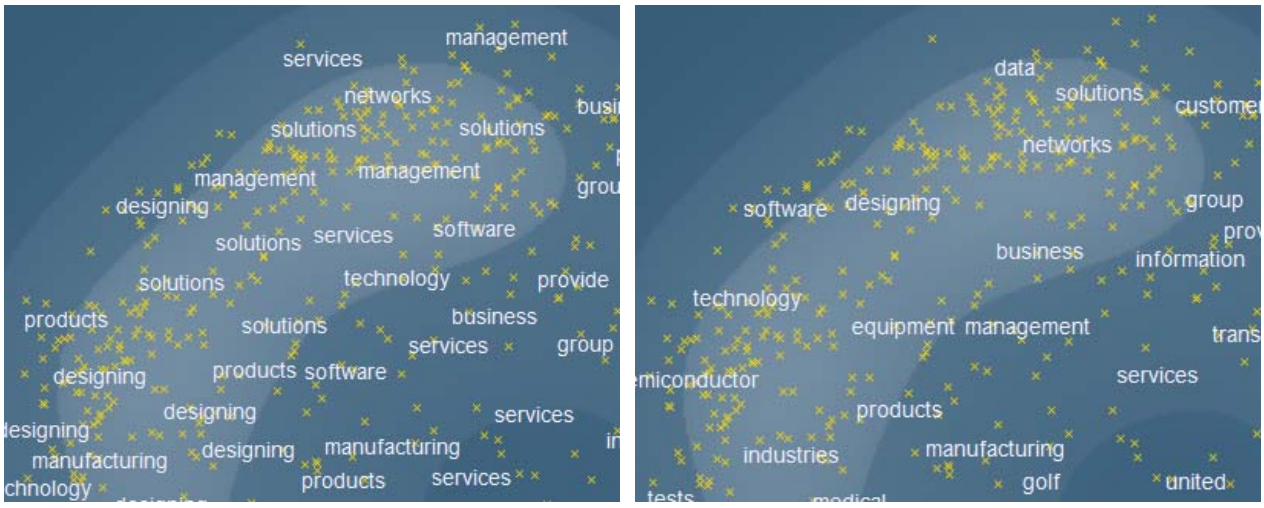

Figure 3. Effect of the minimal distance between repeating common words. Note that increasing the distance decreases the number of common words, since it decreases their selection.

3. Usage of the system

3.1. Example visualization

The proposed approach is implemented as an extension to our system named Document Atlas (Fortuna et al. 2006; http://docatlas.ijs.si). The system is implemented on the top of the Text Garden library (Grobelnik and Mladenić 2006; http://www.textmining.net).

As an example you can find below a projection of a sample ontology describing economical areas on a landscape generated from descriptions of companies taken from Yahoo! Finance website. Figure 4 shows the visualization of the ontology from OntoGen (Fortuna et al. 2006; http://ontogen.ijs.si) and visualization of companies' descriptions using Document Atlas.

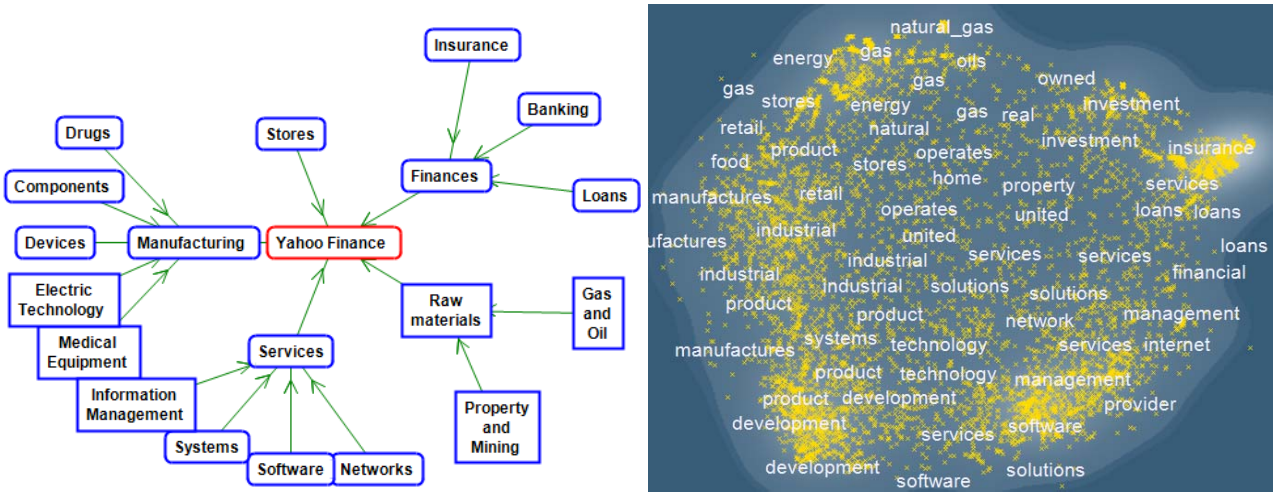

Figure 4. Visualization of ontology (left) and companies' descriptions (right).

The resulting projection of the sample ontology to the landscape from companies' descriptions is shown on Figure 5. The content of the ontology (concept names in the example in Figure 5) are presented using yellow font (for the major concepts) or yellow cross (for lower-level concepts). The links between concepts are shown using green lines and the density is shown as a texture in the background of the map (the lighter the color, the higher the density). Landscape is represented by keywords and their position. Keywords are displayed using white color font.

When the user moves a mouse on the map a set of the most common keywords is computed in real-time for the area around the mouse pointer. The area, from which these keywords are extracted, is marked darker on the map, and the list of keywords is shown in the semi-transparent window next to the mouse as shown in and Figure 6. The user can also zoom-in to see specific areas in more details. By clicking on a document (yellow crosses on the map), more information about the selected ontology content is shown on the left side of the screen (see Figure 5).

Figure 7 shows the same ontology projected on a different landscape. The landscape is generated from Reuters news articles. Note that the part of the ontology describing financial corporations gets spread over a larger area, while the technological concepts get densely grouped in the area of technological news. This shows the bias of then selected news articles towards news related to financial sector.

Figure 8 shows another ontology, generated using OntoGen from country descriptions taken from CIA World Fact Book, projected on the landscape generated from companies' descriptions. Note that most of the concepts get placed on around the same area, with some notable exceptions. For example, Middle East and Central Asia get placed on the area describing oil companies and East Asia gets placed in the area of manufacturing (the concept includes China, Korea, Japan).

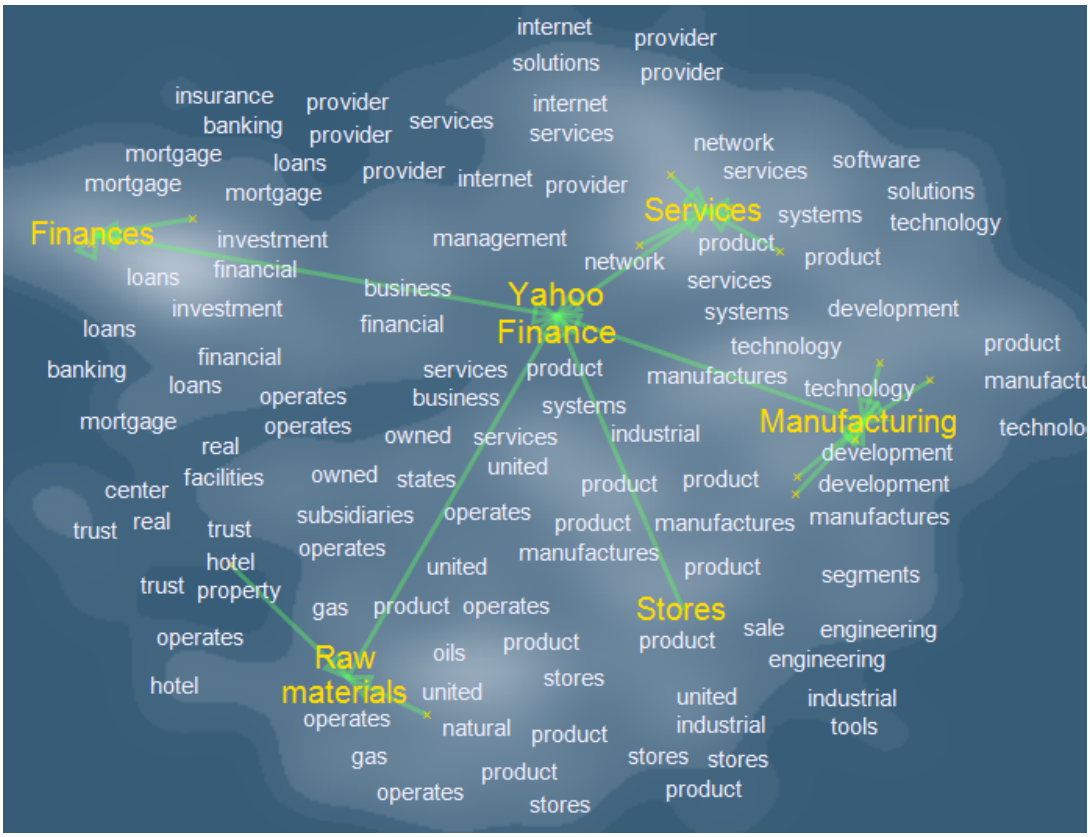

Figure 5. Ontology projected on the landscape of company descriptions.

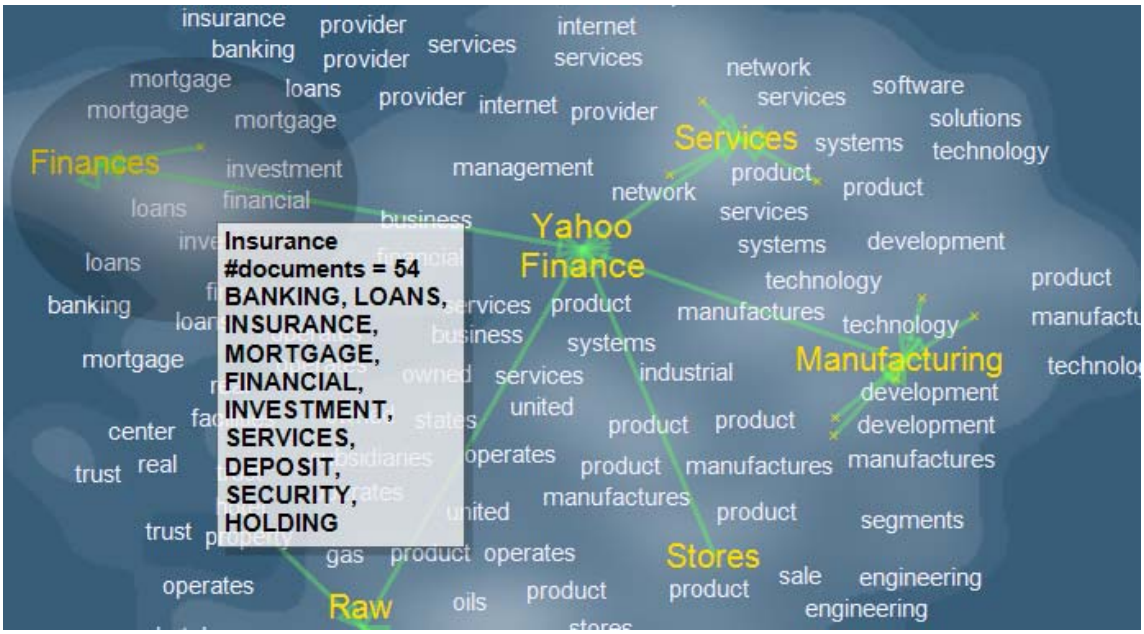

Figure 6. Projected ontology with focus on the list of keywords describing financial area.

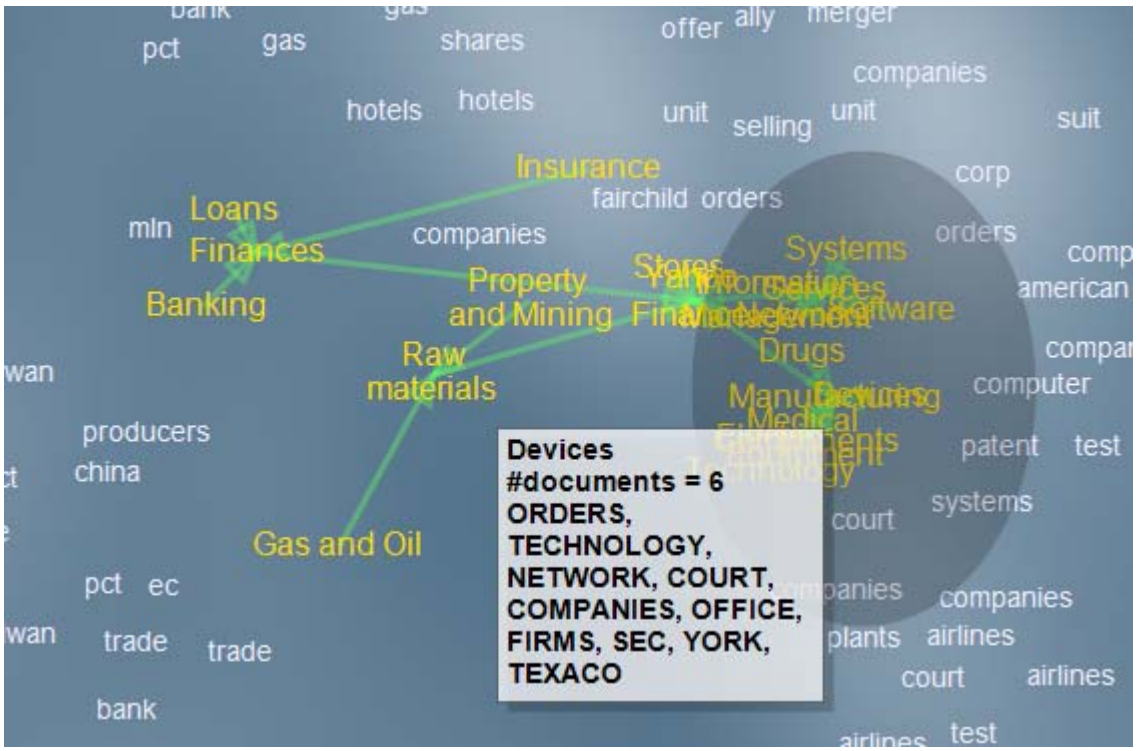

Figure 7. Ontology projected on the landscape generated from Reuters news articles.

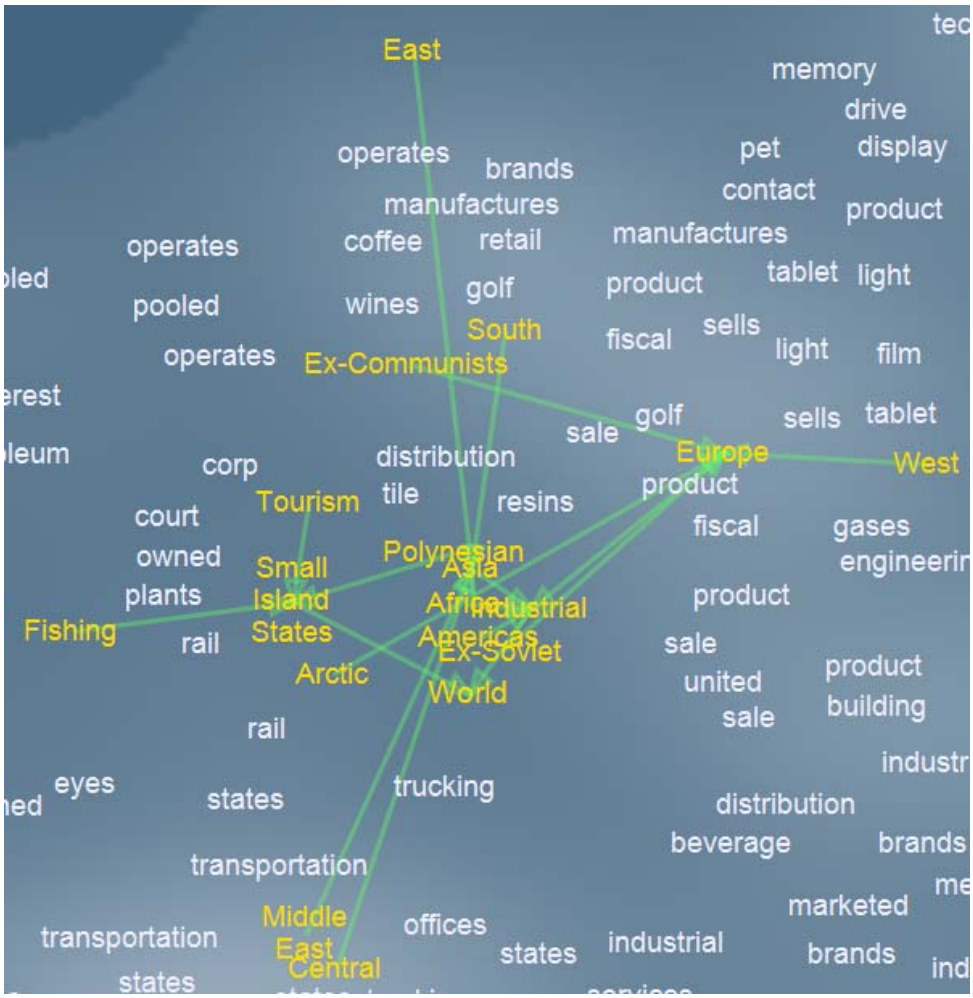

Figure 8. Ontology built from CIA World Fact Book projected on the landscape generated from the description of companies.

3.2. Input format and usage of the tools

The whole visualization pipeline consists of a command-line utility `VizMapContext.exe` which prepares the visualization and Document Atlas tool which displays the visualization.

The command-line utility gets as input the documents, from which it generates the landscape, and the ontology. The documents can be given as a text file (parameter `-ctx:`), where each line corresponds to one document and the first word in each line is a unique identifier for the document (e.g. the name or ID of the document). An ontology can be passed in several ways. First, it can also be passed as a text file (parameter `-idoc:`), where each line corresponds to one concept from the ontology, the first word being unique name of the concept and the rest of the line corresponding to the text grounding the concept (e.g. textual description, keywords, etc.). Second, the input can be a OntoCfier structure (parameter `-iol:`), which was defined in (Grobelnik et. al 2007). And finally, the input can be RDF encoding of an ontology (parameter `-irdf:`), stored using the PROTON schema. This format can be produced using the OntoGen tool.

The outputs of the command line utility are two files. First, the landscape used for visualization is stored for future reuse (parameter `-octx:`). Second, the visualization is stored (parameter `-oviz:`). This is the file, which can be opened in Document Atlas.

4. Discussion

In (Fortuna et al. 2006) we have proposed a system for text corpora visualization which is based on a similar visualization pipeline and techniques as the ones presented here. The main difference is that the methods presented here can operate on larger corpora to generate semantic landscapes, support personalization of the landscapes and enable projection of ontology onto the predefined landscapes.

Somewhat related to our work on landscape generation is an approach to automatic topic identification in a document collection described in (Wang and McCallum 2006). It uses Latent Dirichlet Allocation (LDA) for modeling the topics. The result of the method is a set of topics, where each topic is depicted by a set of keywords and their importance.

The prototype is currently a standalone system implemented in the .Net environment. In the next phase it will become a loosely integrated Neon toolkit plugin.

References

Salton G (1991) Developments in Automatic Text Retrieval. Science, Vol 253: 974-979

Deerwester S, Dumais S, Furnas G, Landuer T, Harshman R (1990) Indexing by Latent Semantic Analysis. Journal of the American Society of Information Science

Carroll JD, Arabie P (1980) Multidimensional scaling. In M.R. Rosenzweig and L.W. Porter (Eds.), Annual Review of Psychology, 31: 607-649

Fortuna B, Mladenić D, Grobelnik M (2006) Visualization of text document corpus. Informatica 29: 497-502

Fortuna B, Grobelnik M, Mladenic D. Semi-automatic Data-driven Ontology Construction System. Proceedings of the 9th International multi-conference Information Society IS-2006, Ljubljana, Slovenia.

Grobelnik, M., Mladenic, D., Text Garden, Springer 2008

Grobelnik M, Brank J, Fortuna B, Mozetic I. Contextualizing ontologies with ontolight: a pragmatic approach. Proceedings of the 10th International multi-conference Information Society IS-2007, Ljubljana, Slovenia.

Jain AK, Murty MN, Flynn PJ (1999) Data Clustering: A Review, ACM Comp. Surv.

Wang X, McCallum A (2006) Topics over time: a non-Markov continuous-time model of topical trends. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 424 – 433, Philadelphia