MULTISENSOR

Mining and Understanding of multilinguaLcontenT for Intelligent Sentiment Enriched coNtext and Social Oriented inteRpretation

FP7-610411

D5.3

Decision support techniques

Dissemination level:	Public
Contractual date of delivery:	Month 24, 31/10/2015
Actual date of delivery:	Month 25, 06/11/2015
Workpackage:	WP5 Semantic Reasoning and Decision Support
Task:	T5.4 Decision support
Type:	Report
Approval Status:	Final Draft
Version:	1.0
Number of pages:	32
Filename:	D5.3_DecisionSupportTechniques_2015-11-05_v1.0.pdf

Abstract

The goal of this document is to describe the technical details and motivation behind the implementation of a decision support system and a recommendation engine. It explains how these systems are developed, implemented and integrated, and how they cover the requirements described in T5.4.

Co-funded by the European Union

History

Version	Date	Reason	Revised by
0.1	15/9/2015	First draft sent to partners for comments	B. Simeonov (ONTO)
0.2	30/9/2015	Contributions by partners	M. Puigbo (PIMEC), E. Jamin (EVERIS)
0.3	20/10/2015	Integrated document for comments	B. Simeonov (ONTO), Georgi Dimitrov (ONTO).
0.4	26/10/2015	Document for internal review	B. Simeonov (ONTO)
0.5	30/10/2015	Internal review	C. Doulaverakis (CERTH)
1.0	6/10/2015	Final version	B. Simeonov (ONTO)

Author list

Organization	Name	Contact Information
Ontotext	Boyan Simeonov	boyan.simeonov@ontotext.com
Ontotext	Georgi Dimitrov	georgi.dimitrov@ontotext.com
PIMEC	Marti Puigbo	mpuigbo@pimec.org
EVERIS	Emmanuel Jamin	emmanuel.jean.jacques.jamin@everis.com

Executive Summary

The objective of this deliverable is to develop a Decision Support System (DSS) and a semantic recommendation engine as part of T5.4. These systems will support users in several aspects of important decision-making processes. They will also help professionals, such as journalist and media monitoring agents to drill down on specific topics and gain better understanding of content. The systems are implemented on top of GraphDB and are based on the reasoning techniques developed and described in D5.2.

For the purposes of MULTISENSOR, Ontotext has developed two different approaches to cover the T5.3 requirements:

- A Decision Support System
- A Recommendation Engine

The first section of this document provides an introduction to the topic.The second section focuses on the techniques for supporting the users' decision-making processes. These techniques were developed mainly in connection to Pilot Use Case 3 – SME Internationalisation, which is described in detail in D7.2. In order to do this, we have researched different statistical indicators, mainly provided by the World Bank and Eurostat. These indicators relate to information in different areas such as Social, Economic, Political, Sector, and Products. They build the foundation of the system and enable users to compare countries based on different parameters. Armed with this kind of information users such as entrepreneurs who want to expand their business abroad, will have a powerful supporting tool in their hands – a tool that will facilitate their decision-making process.

The third section of this document describes the implementation and integration of a semantic recommendation engine, which will offer users highly relevant articles. It will help journalists and media monitoring agents to drill down on specific topics and find relevant or hidden content, thus doing their daily job better. In our solution, a single list of recommendations involves a combination of a variety of factors: relevance of the article to the currently opened one, relevance with respect to the previous reads of the user, popularity of the article among readers, co-visitation and freshness.

Abbreviations and Acronyms

OWL	Ontology Web Language
RDF	Resource Definition Framework
W3C	World Wide Web Consortium
DSS	Decision Support System
GDP	Gross domestic product
NER	Named Entity Recognition
CF	Collaborative Filtering
GeoSPARQL	Geographic query language for RDF data
SPARQL	SPARQL protocol and RDF query language
SIMMO	Socially Interconnected and MultiMedia-enriched Object

Table of Contents

1 INTRODUCTION

The objective of this deliverable is to develop a Decision Support System (DSS) and a semantic recommendation engine, which will offer the users of the MULTISENSOR platform powerful tools for their jobs. These systems have to be implemented on top of GraphDB – an RDF graph database (a type of NoSQL graph database engine), also known as a triplestore. They also have to use the reasoning techniques we have developed and described in D5.2. As a result, these two systems will help professionals such as journalists, media monitoring agents and entrepreneurs to efficiently gain relevant information in support of their work.

Semantic Web technologies share many goals with DSSs. Both try to interpret data and deliver precise and relevant information to users. In the past decade, Semantic technologies have been used in DSSs to solve a number of different tasks such as information integration and sharing; web service annotation and discovery; knowledge representation and reasoning. If we have to make a distinction between them, DSSs have more specific goals, which depend on the type of the system and the area of usage. Therefore, DSSs are applied in business intelligence, information retrieval, knowledge management, situation awareness, emergency management, simulations systems, and many more, where they serve well-defined purposes.

The system we will implement and integrate in MULTISENSOR and more precisely for Pilot Use Case 3 will be based on statistical data indicators and will support entrepreneurs in some of their decision-making processes. For example, it will help them to compare different countries based on many parameters such as politics, social processes, economy, GDP, corruption levels, etc. So, if they want to expand their business abroad, they can use the system to choose a location that would offer the best conditions for their products or services to be successful.

Recommendation engines on the other hand are a subclass of information filtering systems, which try to predict the rating or the preference that a particular user will give to an item. In the last few years they have become so popular that all well-known web resources such as Google, Facebook, LinkedIn, Amazon, Financial Times, etc. use them.

There are three approaches for implementing such systems:

- Collaborative filtering (CF)
- Content-based filtering
- Hybrid recommender systems

The application we will implement and integrate belongs to the family of hybrid recommender systems. For a single list of recommendations we combine a variety of factors: relevance of the article to the currently opened one, relevance with respect to the previous reads of the user, popularity of the article among readers, co-visitation and freshness.

1.1 Requirements

In this section, we present the relation of the DSS and the Recommendation engine to the purpose of the project.

In MULTISENSOR, the DSS will mainly help the users of Pilot Use Case 3, which is described in more detail in D8.2. For example, entrepreneurs who want to expand their business abroad usually face a difficult process where many important decisions have to be made. If you are a yogurt producer and you want to expand your business, how to choose the country or geographical area where your product would be most successful? Before you can answer this question and arrive at an important decision, you need to consider many things. So, it will be very useful if there is a system that can provide all available information from proven sources that is relevant to each specific case.

To handle this task, we have researched different sources of statistical data for appropriate statistical indicators. As the system will be implemented on top of GraphDB, there is a technical restriction about the data format i.e., the data should be in one of the RDF formats recommended by W3C. Based on our research, the World Bank[1] and Eurostat[2] datasets seem to be most appropriate for our use case scenario. In addition, we have to implement a graphical user interface, which will handle every aspect of the available data. It will also provide flexibility and freedom to the users so that they can easily explore the data and get to know the big picture. Armed with such information, entrepreneurs will take their decisions based on facts, instead of assumptions.

For the first two use cases – Journalism and Commercial Media Monitoring, we are going to implement and integrate a semantic recommendation engine. Today more than ever, we have access to large volumes of unstructured data. At the same time, there are many users searching for all kinds of information. If you are a journalist and you have to research a specific topic, first you need to find relevant information. On the other hand, if you are a media monitoring agent researching the European energy policy, you would have to go through content from many sources and different countries. We need tools that would help such professional to drill down on specific topics and find relevant content quickly and efficiently. Recommendation engines are perfect for this task as they can explore the data in depth and recommend articles depending on the topic, area of interest, popularity and other factors. The behaviour of the recommender also depends on the implemented algorithm and the approach that was used. Our solution will be built on top of GraphDB and will use the information from the inserted SIMMO objects[3]. This is a hybrid approach as it takes the best from the other two techniques and provides users with the necessary flexibility.

[1] http://worldbank.270a.info/.html

[2] http://eurostat.linked-statistics.org/

[3] Detailed information about the SIMMO object in D4.1

2 DECISION SUPPORT

2.1 Approaches overview

DSSs are computer-based applications that help people and organisations in some of their decision-making. For example, they can assist entrepreneurs by combining data, sophisticated analytical tools and friendly user interfaces into a single powerful systemprocesses as described in (Blomqvist, 2012).

Actually, we are all decision makers – both in the context of our jobs and as private individuals. So there is a great diversity of approaches as each area of knowledge has its own specifics. To be able to provide the most meaningful information and support to users, the system has to be tailored to the available content and the context of their work. This is why there are no universal solutions (Sprague et al., 1980).

Still, to be able to handle so many variations, DSSs are divided into several main categories – Model-driven, Data-driven, Communications-driven, Document-driven and Knowledge-driven Decision Support Systems:

- **Model-driven DSS –** uses a model of reality so that it can optimise or simulate outcomes of decisions based on the provided data. As long as the amount of data is not large, this model can be accessed and changed by the decision maker in order to analyse a certain situation. A classic example is a financial decision support system, which uses financial models to predict the impact of certain entrepreneurial decisions regarding the key economic indicators of the business. In general, this type of DSSs use complex simulation, optimisation or multi-criteria models to provide decision support (Power 2002). Representative of these category systems is ILOG JRules (Power 2006).
- **Data-driven DSS –** focuses on the access and manipulation of large amounts of data, for example Data Warehousing systems, or even more simple systems such as file systems with search and retrieval capabilities. Another example of Data-driven DSSs is the Geographical Information System - (GIS), which can be used to visually represent geographically dependant data using maps. WebFOCUS[4] is a system of this kind (Power 2002). Documentum ECM[5] is system of this kind (Power D.).
- **Document-driven DSS** – uses text or multimedia document collections as their basis for decision-making with respect to (Power 2002). This approach converts documents to valuable data. While Data-driven DSSs rely on documents, which are already standardised and inserted in the database, this approach deals with information that can be easily standardised and stored. There are three main types of documents used in this kind of systems:
    - Oral
    - Written

[4] http://www.informationbuilders.com/products/webfocus

[5] http://www.emc.com/enterprise-content-management/documentum/index.htm

- Video

There is no established methodology for working with such structures.

- **Communications-driven DSS -** focuses on the interaction and collaboration aspects of decision-making (Power 2006). Simple examples include groupware and video-conferencing systems that allow distributed and networked decision-making. At its basic form, this system can be a simple threaded e-mail and in its complex form, it can be an interactive video or a web communication application. All communications-driven DSSs have to have at least one of the following characteristics:
  - Support coordination and cooperation between two or more people.
  - Facilitate information sharing.
  - Enable communication between a group of people.
  - Support group decisions.

Systems using this approach are - WebEx[6], Polycom[7], Halo Collaboration Studio (Power 2006).

- **Knowledge-driven DSS –** these systems actually recommend or suggest actions to the users, rather than just retrieve information relevant to a certain decision, i.e., these systems try to perform some part of the actual decision-making for the user through special-purpose problem-solving capabilities.
- **Web-based DSS –** every DSS can be web-based. This term describes any system that can be accessed through a web browser.

The implemented DSS for MULTISENSOR belong to the category of Data-driven DSS.

2.2 Implementation in MULTISENSOR

The main goal of DSSs is to help entrepreneurs in the process of decision-making. In MULTISENSOR, this system is developed for the purposes of Use Case 3 - SME Internationalisation. For example, if you are an entrepreneur and you want to expand your business abroad, there are many important decisions you have to make. If your company is a yogurt producer, how do you choose the country where your product would be selling most successfully? Usually, before you can answer this question and arrive at an important decision, you need to consider many factors such as:

- Distance (if your products have an expiry date, how much time you will need to deliver your products to the target destination and if this will affect their quality)
- Weather (if your products depend on weather conditions)
- Transport (what kind of transport you will use)
- Regulations (what are the normative regulations and laws in the destination country concerning export/import, etc.)

[6] http://www.webex.com/

[7] http://www.polycom.co.uk/

- Social, Political, Economic and other indicators about this country

These are only a few of the questions that every entrepreneur needs to answer before making a final decision. Therefore, it would be very difficult to develop a DSS that can point directly to a decision. Our approach is different. The DSS we have developed and integrated will compare statistical indicators in many areas of the political, social, cultural and economic life in the potential countries. In this way, users will have access to such relevant information gathered in one place and supported by charts and diagrams and they will be able to base their decisions on solid facts rather than speculations. This will also provide them with the bigger picture and will significantly facilitate the whole process of decision-making.

In addition, the DSS will use the Geo-SPARQL technique developed for T5.2 and will provide information about specific regions in a country so that the user would be able to research the region's infrastructure, water resources, population, etc.

2.2.1 **Indicators**

In MULTISENSOR, an indicator is defined as a measurable quantity (e.g., a number or a ratio) or any piece of conceptual information (i.e., non-measurable) derived from a series of observed facts serving as a guide to the process of decision-making. In this case, we have mainly selected measurable indicators as they better serve the purpose of delivering an assessment support.

Measurable indicators are computable, measurable units derived by collections of data. Such examples include the GDP of a country, its total population or the percentage of people there who have access to the Internet. These indicators are very useful for establishing a comparison between countries and, thus, generating a preliminary assessment of the differences the countries present.

The decision support we have developed is for the purposes of Use Case 3 - SME Internationalisation. Here, in order to prioritise the market and also find out which SMEs are interested in exporting, we would need different SME indicators. Therefore, the system focuses on some general macroeconomic and macro-environmental factors, which would help users to get the bigger picture - what opportunities or dangers could be involved in the process of enlarging their business abroad.

With the help of several export managers from PIMEC, we have categorised the selected indicators. In the economic category, we have picked data that can be easily compared between different countries despite of their size. Such indicators include data for GDP growth, GDP in Purchasing Power Parity, balance of trade and the time required for import and export according the laws of a given country.

In the social sector, we have decided to include data about total population, unemployment, degree of urbanisation (urban, rural), and percentage of households with access to the Internet. These indicators provide a clear picture of the situation in a specific country. For example, it is interesting to know the size of the population in a given country as it helps to evaluate the potential of the market.

The final group of indicators we use are cultural and political. It is very important to know the habits of the population in the economic sector. One of the World Bank indicators provides statistics about the business culture of the population in a region of interest. Also,

for a full picture of a given country, we need data about the political situation. Here, the respective indicators provide information about the current government, strategic plans for economic growth, etc. For example, it is important to know when the next elections will take place as they can change the political land shaft and bring risk for new enterprises.

2.2.2 SPARQL queries

The data that serve as a basis for the DSS is in RDF and are loaded into GraphDB – a triplestore developed by Ontotext. To interact with the knowledge base, the DSS needs appropriate mechanisms and queries. The standard query language for querying RDF data recommended by W3C and fully supported by GraphDB is SPARQL. To write appropriate queries that will serve the needs of the system, first we have to explore the data and find its schema and specifics.

Exploration of Eurostat datasets

Although the data in the Eurostat indicators has different shapes (depending on the dataset), there are some common fields that are found in all datasets:

- Every indicator has**Freq**(Annual, Quarterly, Monthly), which is consistent within the datasets.
- Every indicator has **Unit**.

Some indicators have additional fields - most of which are very important as they are part of the indicator definition. For example:

- **prop:indic*** - this group is build out of 6 different properties:
  - **prop:sex -** used to specify the gender of specific group of people
  - **prop:age -** years range of the statistical sample
  - **prop:incgrp -** Above 60% of median equivalised income
  - **prop:building -** in the data specification this is marked as **others**
  - **prop:deg_urb -** Densely-populated area (at least 500 inhabitants/Km²)
- **prop:s_adj** – we can ignore this property because of the type of the applied adjustment.
- **prop:isced97=ED5_6** – this property is fixed and it merely indicates the exact level of tertiary education attained: "First and second stage of tertiary education (levels 5 and 6)"@en.

2.2.2.2 Applicability of Indicator Values

To find the different values of the property **prop:indic_na** across the datasets, their meaning and number of occurrences, we have created the following query:

```
PREFIX qb: <http://purl.org/linked-data/cube#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
select ?dataset ?indic ?indicator (count(*) as ?c) {
        ?x qb:dataSet ?dataset.
        ?x prop:indic_na ?indic.
        ?indic skos:prefLabel ?indicator
} group by ?dataset ?indic ?indicator
```

The result of this query is the count of all available **prop:indic_na** in all datasets.

Age and Age Group

The next property we want to explore is **prop:age**. To find the different values of this property across the datasets their meaning and occurrences, we have used the following query:

```
PREFIX qb: <http://purl.org/linked-data/cube#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
select ?dataset ?ag ?age (count(*) as ?c) {
        ?x qb:dataSet ?dataset.
        ?x prop:age ?ag. ?ag skos:prefLabel ?age.
} group by ?dataset ?ag ?age
```

By investigating the results of this query, we reach the following conclusions:

- **edat_lfse_07** – uses age groups such as “From 15 to 64 years”
- **demo_mlexpec** – uses specific ages
- **demo_mlifetable**- uses specific ages.

Indicator Facet Value Combinations

The most complex dataset we will use is “Balance of trade - Current account - quarterly data” (ei_bpca_q), because it includes many factors. **prop:indic** has the largest variability with about 20 values (from BP-010 to BP-993).

By exploring this dataset with the following query:

```
PREFIX qb: <http://purl.org/linked-data/cube#>
select ?freq ?unit ?adj ?partner ?flow (count(*) as ?c) {
        ?x qb:dataSet eudata:ei_bpca_q;
        sdmx-dimension:freq ?freq;
        prop:unit ?unit;
        prop:s_adj ?adj;
        prop:partner ?partner;
        prop:stk_flow ?flow
} group by ?freq ?unit ?adj ?partner ?flow
```

we learn that the fixed factors are freq=freq-Q (Quarterly), unit=MIO-EUR (millions of Euro), s_adj=NSA (Not Seasonally Adjusted).

Table1 presents the other factors (?c is the number of observations with this particular combination of factors):

Partner	Flow	c
eu:partner#EXT_EU15	eu:stk_flow#CRE	168
eu:partner#EXT_EU15	eu:stk_flow#DEB	168
eu:partner#EXT_EU15	eu:stk_flow#NET	168
eu:partner#EXT_EU25	eu:stk_flow#CRE	196
eu:partner#EXT_EU25	eu:stk_flow#DEB	196
eu:partner#EXT_EU25	eu:stk_flow#NET	196
eu:partner#EXT_EU27	eu:stk_flow#CRE	427
eu:partner#EXT_EU27	eu:stk_flow#DEB	427
eu:partner#EXT_EU27	eu:stk_flow#NET	427
eu:partner#EXT_EU28	eu:stk_flow#CRE	432
eu:partner#EXT_EU28	eu:stk_flow#DEB	432
eu:partner#EXT_EU28	eu:stk_flow#NET	432
eu:partner#WORLD	eu:stk_flow#CRE	22097
eu:partner#WORLD	eu:stk_flow#DEB	22223
eu:partner#WORLD	eu:stk_flow#NET	22593

Table 1: Additional Factors

From this, we learn that flow is Debit, Credit and Net, and the partners are EU15, EU25, EU27, EU28, WORLD. (E.g., a source country may have positive balance of trade with EU15, but negative with WORLD.)

Indicator queries

After researching the indicators and the content of the datasets, we have developed relevant queries for each indicator. These queries will be in the core of the DSS. They will support the user interface by retrieving the necessary information from the knowledge base.

Our expectation is that with time, the list of loaded statistical datasets will increase, so we have created a Google document, where all developed SPARQL queries are listed together. For more details, see MULTISENSOR SPARQL Queries.

2.2.3 User Interface

To reflect the main important factors for the internationalisation process, four categories of indicators have been selected and integrated in the UC3 application - Political, Economic, Social and Cultural. In Table 2, the codes in the parenthesis represent the identifier of the dataset given by Eurostat.

Category	Sub-category	Indicators	Graphical representation
Economic indicators	GDP	GDP growth	Line chart
		Real GDP growth rate – volume (tec00115)	Line chart
		GDP per capita in PPS (tec00114)	Line chart
		GDP per capita – quarterly Data (namq_aux_gph)	Line chart
		Exports of goods and services in % of GDP (tet00003)	Line chart
		Imports of goods and services in % of GDP (tet00004)	Line chart
		Export to import ratio (tet00011)	Line chart
		Inward FDI stocks in % of GDP (tec00105)	Line chart
	Importation / exportation	Customs and tariffs	Multidimensional lines chart
		Structure of taxes by economic function (gov_a_tax_str)	Multidimensional lines chart
		Export and Import	Multidimensional lines chart
		Current account – quarterly data (ei_bpca_q)	Line chart
		Harmonised indices – monthly data (ei_cphi_m)	Line chart
		Foreign Direct Investment	Line chart
Political indicators	---	Government type	Bar chart
		Political instability index	Bar chart
		Corruption perception index	Bar chart
		General government deficit (-) and surplus (+) – quarterly data (ei_nagd_q_r2)	Bar chart
Social indicators	Population	Life table (demo_mlifetable)	Bar chart vertical
		Human Development Index	Line chart
		Population with tertiary education attainment by sex and age (edat_lfse_07)	Bar chart with age groups
	Work	Unemployment rate	Line chart
		Harmonised unemployment rates (%) – monthly data (ei_lmhr_m)	Line chart
	Health	Life expectancy	Bar chart with age groups
		Life expectancy by age and sex (demo_mlexpec)	Bar chart with age groups
		Population distribution	Line chart

Cultural indicators	Urbanisation	Distribution of population by degree of urbanisation, dwelling type and income group (source: SILC) (ilc_lvho01)	Bar chart
	Consumption habits	Economic sentiment indicator (teibs010)	Line chart
		Households having access to the internet at home (isoc_pibi_hiac)	Histogram
		Easiness of doing business	Bar chart

Table 2: List of the indicators displayed per category

As shown in Figure 1 below, the user can select a specific category of indicators, which will be displayed for the country that was selected. In this example, we show the social indicators.

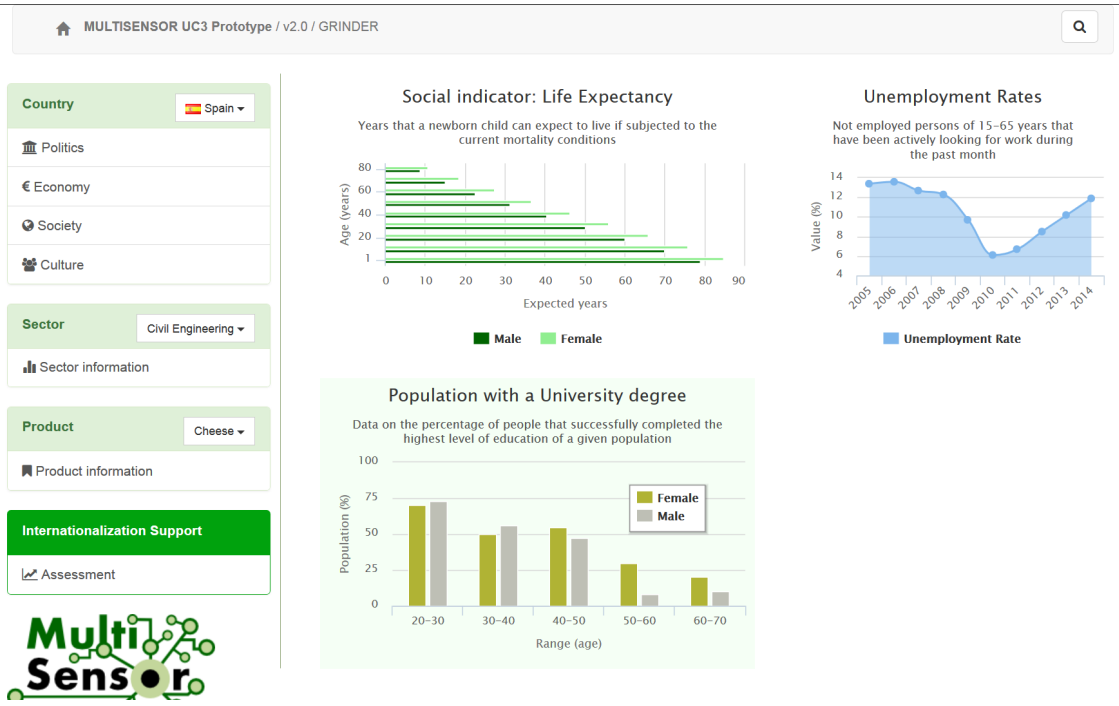

Figure 1: Per country, display of the social indicators

As some indicators take into account several dimensions (such as time frame, age, genre, etc.), there are different graphical representations that can used. We will focus on the following three main graphical representations.

The simplest one is the line chart, which represents two dimensions of the data. For example, to represent the unemployment rate, it displays how the rate (%) changes in different years (Cf. Figure 2).

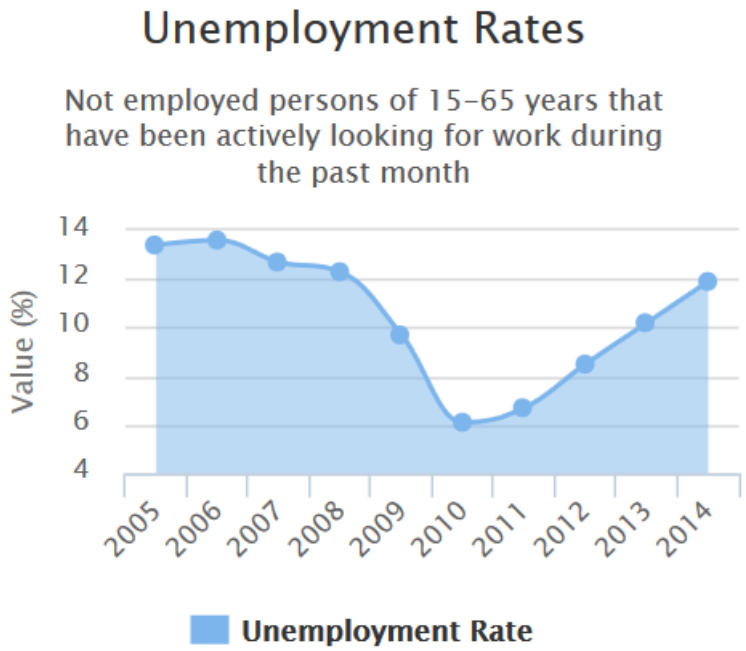

Figure 2: Line representation

Another graphical representation is the multidimensional line chart, which makes it easy to compare different values of the same units in different years. In the following example, you can compare a country's import and export of goods and services per year (Cf. Figure 3).

Figure 3: Multidimensional lines chart

The third graphical representation is the bar chart (Figure 4). It shows how one indicator changes according to different factors such as age group. For each factor, different values can be displayed. The example below presents the percentage of the population who have followed a tertiary education program for men and women of different age groups.

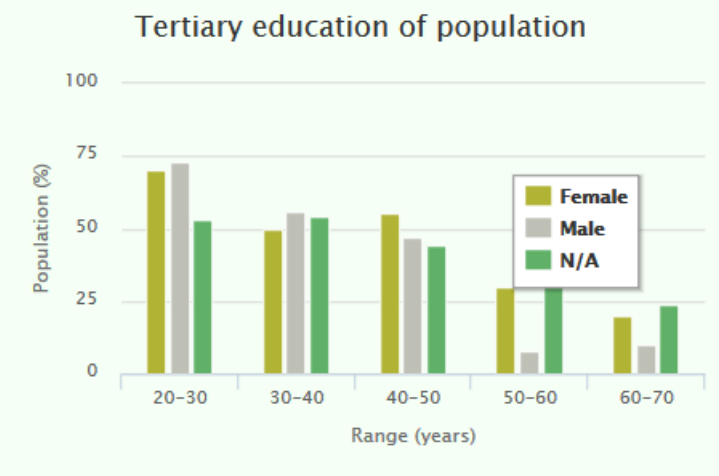

Figure 4: Bar chart (vertical) representation

Users can compare the values of the indicators for two countries, as shown below (Figure 5) and evaluate the conditions for export.

Table of indicators

The selection of the correct country for the international investment depends on a number of indicators. These indicators are very important in order to make a first analysis of the different options that can be presented in a global market.

#	Indicator	Spain	Ucrania
1	Merchandise Imports	3.17 (2011)	15.39 (2011)
2	Clear imports	5.5 (2005)	5.9 (2008)
3	Average days to import goods	10 (2011)	33 (2011)
4	GDP Growth	1,5 %	1,2 %
5	GDP-PPS	45	35
6	Unemployment	14 %	10 %
7	Internet households	80 %	75 %
8	Total number of inhabitants	10 M	42 M
9	Ease of doing business	12	6
10	Balance of trade	2002,6	4034,5

Decision Support

Based on the above information, the most suitable country is:

Figure 5: Internationalisation support

At the end, the system calculates which country has the highest number of favourable indicators for internationalisation of the goods and displays it as the most suitable country.

For more details about displaying the indicators in the UC3 application, please refer to D7.6 – Second Prototype.

3 RECOMMENDATION SYSTEM

3.1 Approaches overview

One of the greatest challenges nowadays is how to search and find useful information in the large volumes of available unstructured data. People search for information in their capacity as company employees as well as privately. Journalists are always looking for stories. Policy analysts have to sieve through compliance documents. But what if you could instantly find the exact document, the precise paragraph and the perfect reference to a specific topic? Applying semantic technology to this challenge accomplishes this goal in many ways. Internal users gain access to relevant content, which enables them to do their job faster and external users find relevant contextual content that is personalised for them. Such systems are called recommendation engines and they help users to find exactly what they need.

There are different approaches to implementing such systems. The three main techniques are Collaborative Filtering (CF) (Breese et al., 1998), Content-based filtering and Hybrid recommender systems.

3.1.1 Collaborative Filtering

The significant expansion of the Internet in recent years has made the task of finding and extracting information from all available resources much more difficult. The overwhelming amount of information increases the need for filtering tools that can help us find relevant content.

One of the popular techniques used for solving this problem is called Collaborative Filtering (CF) (Breese et al., 1998). It is a popular recommendation algorithm that bases its recommendations on the ratings or behaviour of other users in the system. The idea behind it is that people often get recommendations about different aspects of their lives from someone with similar tastes and experience - for example about food, music, movies, transport, etc. Let's say that a group of users like the same things as George, who is another user. In that case, it is very likely that George will like the same things that the group likes although he has not tried them yet (Ekstrand et al., 2011). This is how recommender systems help people to find the most appropriate and relevant information about products or articles on a given topic.

To work as expected, the CF algorithm has some minimal requirements:

- Active user participation
- Easy way to represent the user interests to the system
- Algorithms that can match people with similar interests.

The typical workflow of the CF approach is as follows:

1. A user expresses his preferences about different items by rating them. These ratings show an approximate representation of his interests in a specific area.
2. The system matches the ratings against other users and finds people with similar tastes.

3. The system recommends new items to the current user,which have not yet been rated by him but have been rated highly by other people with similar interests.

The core of CF is a database of items rated by users. Based on this kind of information, the algorithm tries to predict additional items that users will like. A typical scenario is when we have a group of users, a list of products or articles, and a list of items for each user with products or articles that he has rated. We can acquire knowledge about the users not only from their ratings, but also from their behaviour. Based on this information, we can create a user-item matrix and recommend items to the current user. Systems which are using this technique are - Facebook[8], MySpace[9], LinkedIn[10].

Main types of CF

As presented in the next diagram, there are three main types of CF:

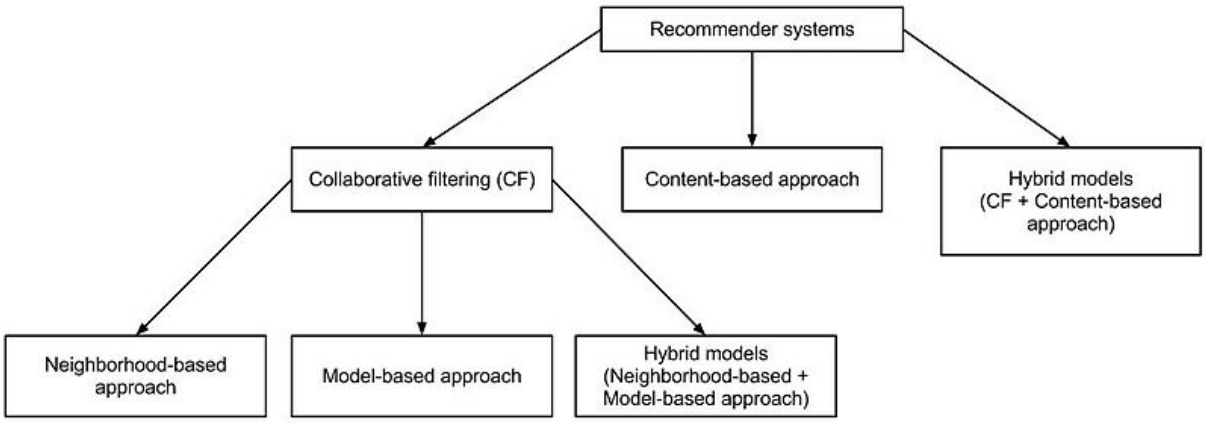

Figure 6: Types of CF[11]

- **Neighbourhood-based approach** (also known as Memory-based approach) - uses the ratings data to compute similarities between different users or items. Every user is a part of a group of people with similar interests. By identifying the neighbours of the current user, a prediction of preferences for new items can be made. It was one of the first approaches that were developed and it has been used in many commercial systems ever since (Altman, 1992).
- **Model-based approach** - uses models developed by data mining and machine learning algorithms. It allows finding complex patterns in the training data and making intelligent predictions for the collaborative filtering tasks (Su et al., 2009).

[8]https://www.facebook.com/

[9]https://myspace.com/

[10]https://www.linkedin.com/

[11]https://en.wikipedia.org/wiki/Collaborative_filtering#/media/File:Collaborative_Filtering_in_Recommender_Systems.jpg

- **Hybrid model -** combines the Model-based and Memory-based approaches (Ekstrand et al., 2011), thus overcoming the weaknesses of the CF methodology. Usually, this is the hardest and most expensive method to implement, but it also gives the best results. Many commercial systems such as Google use this approach (Das et al. 2007).

Problems and weaknesses

The CF algorithm has many problems and weaknesses:

- **Data Sparsity**– regarding (Su et al., 2009), this is a challenge when a new product, a new article or a new user enters the system, the so-called cold start problem. In such situations, it is difficult to give recommendations, because the system does not have enough information i.e., before some users start rating the new item, nothing can be recommended. In the same way, no meaningful recommendations can be given to a new user before he starts rating items, thus showing his preferences. There is also another aspect of this problem, called coverage. When the number of articles or products in the database is much bigger than the number of user ratings, the system may not be able to generate recommendations for these items.
- **Scalability** – when the number of existing users and items grows significantly, the computational resources will go beyond the acceptable levels. For example, with millions of users and millions of articles, we have O(n) complexity, which will be too much (Su et al., 2009).
- **Synonymy** – this refers to the tendency to have the same or similar products or articles featured with different names. Most recommendation systems cannot make such a distinction and threat these items as different (Su et al., 2009).
- **Gray sheep** – this is a problem when there is a group of people whose behaviour does not consistently agree or disagree with any group of people in the system and consequently does not benefit from CF with respect to (Su et al., 2009) and (Claypool et al., 1999).

3.1.2 Content-based filtering

Content-based recommendation systems analyse a set of products or articles, previously rated by a user and build a model or a profile of this user based on the features of the rated items (Brusilovsky 2007). The profile is a representation of the user interests and it is used for generating future recommendations. The process of recommending basically consists of matching the user attributes against the attributes of the product or article. The resulting matching is the relevance of the user interest to this item. Whether the user profile reflects his preferences correctly or not has a tremendous effect on the accuracy of the recommended data. For example, if the profile is reliable, the user preferences for other items can be easily guessed. Based on this, we can filter and show only products or articles that interest the user and omit the ones that do not. Content-based recommendation systems need a good approach for representing the items and generating the user profile. In addition, there has to be a methodology for comparing the items with the profile (Van Meteren et al., 2000). The recommendation process consists of three steps:

1. **Analysing content** – if the information has no structure, we need some kind of a pre-processing step to extract structured content from the data. There are different techniques to achieve this goal such as parsing the documents according to some methodology, machine learning, etc. The main responsibility of the component is to represent the content of the items coming from the information sources in a form that is suitable for the next processing step with respect to (Lops et al., 2011).
2. **Profile learner** – this step is responsible for collecting the user preferences and for generalising them in order to construct the user profile. Usually, at this point machine learning is used to infer a model of user interests based on items that he/she liked or disliked in the past (Lops et al., 2011).
3. **Filtering** – during this step, the items collected from the user profile are matched against the data items in order to find matching elements and recommend a matching product or article (Lops et al., 2011).

Compared to CF, the Content-based approach has many more advantages:

- **User Independence** – Content-based recommendation systems use the ratings provided by an active user to build his profile. In contrast, CF needs ratings from other users to find the "nearest neighbour" of the active user (Lops et al., 2011).
- **New item** – Content-based recommender systems can recommend new items that are still not rated by any users, which solves the cold start problem (Lops et al., 2011).
- **Transparency** – it is very important to show how each item has been chosen and recommended, so that the user can judge if the recommendation is relevant to his interests or not. In contrast, CF is a black box where the only information we have is that unknown users with similar interests have liked this product or article (Lops et al., 2011).

The Content-based approach also has some weaknesses:

- **New User** – when a new user appears, the system needs a certain number of ratings in order to build his profile and understand his preferences. Therefore, when only a small number of ratings are available, it is unable to provide reliable and relevant recommendations (Lops et al., 2011).
- **No surprise** – content-based algorithms do not have mechanisms for finding new items. Their recommendation is based on matching the user preferences against the product or article. Consequently, the system will recommend only items similar to those that the user has liked in the past. For example, if the user has rated science fiction movies, the algorithm will suggest only movies related to this category.

For example, systems that are using this approach are - Rotten Tomatoes[12], Internet Movie Database[13], Jinni[14], Rovi Corporation[15] andJaman[16].

[12]http://www.rottentomatoes.com/

[13]http://www.imdb.com/

[14]http://www.jinni.com/

3.1.3 Hybrid recommender systems

Hybrid recommender systems combine two or more approaches according to (Burke 2002). Usually, they represent a combination of CF and some other techniques. Research in this area shows that the combination between CF and Content-based filtering gives better results than each of these systems on its own. There are many different ways to combine and merge these two techniques and the following is a list of some of them:

- **Mixed** – this approach generates an independent set of recommendations and then joins the ranking candidates, before showing them to the user. Because it merges the results and it is hard to see the improvement over each component separately, there is no easy way to evaluate the system (Burke 2002).
- **Weighted** – this is the most straightforward approach. The items are scored separately by both systems and the final results are a linear combination of the intermediate results. Content-based recommenders are able to make prediction for any item, while CF can only score an item if there are peer users who have rated it (Burke 2002).
- **Switching** – some recommender systems are a hybrid of more than two approaches. In Switching, the different techniques follow a certain order and if the first one does not produce high confidence results, the next one is tried. The difficulty with this approach is developing a reliable methodology of when and how to switch the different techniques (Burke 2002).
- **Cascade** - cascade models make candidate selection exclusively with a primary recommender and employ a secondary recommender simply to refine item scores. For example, items that were equally scored by the main component might be re-ranked employing the secondary component (Burke 2002).
- **Feature combination** - systems that follow the feature combination approach only employ one recommendation component, which is supported by a second passive component. Instead of processing the features of the contributing component separately, they are injected into the algorithm of the actual recommender (Burke 2002).

Each of these approaches has its strong and weak points. The benefit of combining more than one algorithm in a recommendation system is that they can compensate each other's weaknesses and provide better results to the user. Which one to choose would depend on each individual use case. Therefore, many studies compare the different techniques with the same set of data.

The developed and integrated Recommendation System for MULTISENSOR belongs to the category of Hybrid Recommenders.

[15] http://www.rovicorp.com/

[16] https://jaman.com/

3.2 Implementation in MULTISENSOR

3.2.1 Overview

In its current state, the recommendation engine combines four main components/objectives:

- Relevance:
  - The relevance of the news item with respect to the user (user history)
  - The relevance with respect to the user's momentary context (the news article he/she is currently reading).
- Collaborative Component – measuring popularity among similar/peer users.
- Popularity – how popular the articles are. Popularity may be two-fold: popularity among peer users (this is the collaborative component) or popularity with respect to other articles (similar to the PageRank scheme, which reflects the popularity of a web site amongst other web sites).
- Freshness – the recency of the proposed articles.

The recommendation engine comprises the top *n* results from a list of news articles ordered by the above listed criteria, where the parameterisation of the precise relative importance of each of these criteria is still to be specified.

Each of the components (relevance, popularity, collaborative, freshness) represents a different aspect of the recommendation objective. The ranks in the final ordered list of news are controlled by the weight parameters *wUserProfile (wUP), wContext (wCT), wRelevance (wR), wFreshness (wF), wCollaborative (wCol)* and*wPopularity (wP),* which determine the effect each objective has on the final outcome.

One of the features of our recommender is to provide recommendations based on similarities. For this task, the system needs to process the content of the articles through the Named Entity Recognition (NER) pipeline. Here, the quality of the recognised entities depends on the available corpora as the candidates for recognition are matched against it. One of the hardest aspects of this task is disambiguation. To provide high quality disambiguation, we need an understanding of the surrounding world provided by the knowledge base. Thanks to the reasoning techniques developed and described in D5.2, our understanding of the world is improved by generating knowledge from the existing knowledge and by finding hidden connections in the data. As a result, the recommendation algorithms can better understand the similarities between different articles and provide high quality recommendations.

3.2.2 Scoring Formula

The score for a user U, reading an article $A_{current}$ and a candidate article A_{target} is given by:

$$S(U,A_{current},A_{target}) =$$

$$(1 + wUP*userProfileSimilarityScore + wCT*contextSimilarityScore)^{wR} *$$

$$*(1 + freshness)^{wF} * collaborativeScore^{wCol} * popularity^{wP}$$

where the parameters satisfy the following identifiability conditions:

wR + wF + wCol + wP = 4,

wUP + ***wCT*****= 2.**

Modifying the parameters *wUP, wCT, wR, wF, wCol, wP* allows us to change the relative weighting of the different components and to obtain recommendations that are most suited for a particular use case.

There are different scenarios for the parameters. One is to use the click through rates of the users to automatically find the best set of parameters. The other is to allow the user to choose his preferred set through an intuitive user interface.

Relevance Score

Our definition for relevance is two-fold:

- Relevance with respect to the *short-term history* of the user (mapped by the momentary context in terms of the current article and the previous one or two news items viewed). We call this *Context*.
- Relevance with respect to the *long-term history* (mapping the interests of the user as measured by his previous behaviour). We call this *User Profile*.

For calculating the relevance, we use article representations *R(A)* of the article *A*:

R(A) = vector of the top N concepts, weighted with their tf-idf scores

Boost factors are typically applied for terms belonging to particular document fields (e.g. title, body, summary, or the fields storing the URIs of the extracted concepts).

Good quality article representations are crucial for the performance of the recommender. This is achieved by using our proprietary state of the art concept extraction algorithm for DBPedia-based concept disambiguation and inference, which is also used on the now.ontotext.com site.

Relevance with respect to the current context

The current context is simply the representation $R(A_{current})$ of the article $A_{current}$being currently viewed (it is possible to generalize this for the last 2 or 3 articles). The *context relevance score*, ***contextSimilarityScore***, is the (weighted) cosine similarity between the weighted tf-idf representations $R(A_{current})$ and $R(A_{target})$ of the current and target articles.

At the moment, per-field boost factors are assigned when looking up articles similar to the one from which the transition is made:

Document Field Name	**Boost Factor**	**Comments**
Content	1	The baseline boost factor for the content of the articles is set to 1. This is serving as the reference value, with respect to which the other values are determined.
Titles	0.25	Titles are assigned lower values, as they often include the names of

		popular rubrics, e.g. "Q&A", which cover a great diversity of mostly unrelated topics. Boosting the title term weights would lead to a bigger number of recommendations based solely on rubric names. Therefore, a greater weight is assigned to the "summary" field, which is regarded as a better source of important terms.
Concepts	2	Tend to improve the quality of recommendations.
Keyphrases	2	Tend to improve the quality of recommendations.
Summary	2	Tends to include the most important terms associated with a document.

Table 3: List of boost factors per document field

Relevance with respect to the user history

This aspect of relevance, or *User Profile (UP*), has two components - *static (UPstat)* and *dynamic (UPdyn)*.

A. Static component

The static component is a time decayed combination of the article representations $R(A_i)$ of the articles A_i read by the user.

The user profile *UPstat(n)* at day *n* is updated as follows:

$$Upstat\,(n) = \gamma * Upstat\,(n-1) + \sum_{A_i today} R(A_i)$$

It takes top *N* keywords everywhere.

We have decided to use:

γ= 0.8 + 0.15 * #articlesInProfile/(#articlesInProfile +10)

The constant of 10 has the meaning that the time decay γ starts to stabilize above 90% as long as the user profile accumulates 20 articles. Obviously this is a dampening of the parameter γ that can be easily adapted.

When there are fewer articles in the user profile, the decay factor γ is smaller, which effectively means that the history is quickly forgotten. The reason for this setting is to be able to quickly forget the cold start initialisation.

Currently, an update is performed every time a notification about a given user accessing an article is received (no daily updates are performed).

The static component of the user profile relevance score, ***staticUserprofileSimilarityScore***is the (weighted) cosine similarity between *UPstat(n)* andthe representation of the target article $R(A_{target})$.

B. Dynamic component

In addition to the static user profile, we also support a dynamic transition matrix *UT* for the user where the transitions between the concepts in the static user profile are recorded. The dynamic component provides boosts for the concepts belonging to the static user profile *UPstat*.

UT is a transition matrix, where all elements are nonnegative and the sum of the elements in each row is 1. The dimensions of *UT* are *#UPstat(n)* x *#UPstat(n)*. In *UT,* we keep track of the transition between concepts in the static profile *UPstat(n)*.

At inception, the matrix *UT* is initialised as a unit diagonal matrix. Then, with each transition of the user from news article *Ai* to article *Aj,* it is updated as follows:

1. Get the tf-idf vector of the target article *Aj* restricted to the concepts in *UPstat(n)* and normalise it, so that its elements (tf-idf scores and boosts) sum to 1. Call this vector *wj*.
2. For the top 5 concepts (based on their tf-idf scores) *Ci* from the outbound article *Ai,* which are also in *UPstat(n),* replace the corresponding row *UT[Ci,.]* in the transition matrix *UT* with the row

$$UT[Ci,.] = \alpha * UT[Ci,.] + (1 - \alpha) * wj,$$

where*α* is an update parameter that controls the stickiness/momentum of transition matrix.

The *α* parameter can either be computed in a way that makes it increase gradually with the growth of the number of recorded transitions (*numTrans*):

$$\alpha = min(0.95, max(0.05, numTrans / (numTrans + 1)),$$

or be assigned a constant value. Currently, *α* is fixed to the value of 0.95.

The user transition matrix *UT* accounts for the transition between concepts common for the user. If a concept drops off *UPstat(n)* and a new one enters for the first time, we have a cold start for the corresponding row/column in *UT*, i.e. set them to (0,0,...,0,1,0,...0) where the 1 is precisely the position of the new concept in the row/column space.

The transition part of the relevance score is obtained upon multiplying the transition matrix UT with the representation of the current article $R(A_{current})$: *UPtrans(Acurrent)* is then the activation applied to the current article the user is looking at:

$$UPtrans(Acurrent) = UT * R(A_{current})$$

This is simply a spreading activation vector indicating which topics the user is likely to navigate to, given his current context which is the current article he is reading.

The dynamic component of the user profile relevance score, ***dynamicUserprofileSimilarityScore***is the (weighted) cosine similarity between *UPtrans(Acurrent)* andthe representation of the target article $R(A_{target})$.

Finally, the ***userProfileSimilarityScore***is obtained as an average of the dynamic and the static components:

userProfileSimilarityScore=(staticUserprofileSimilarityScore+dynamicUserprofileSimilarityScore)/2

Other combinations, different from the equally weighted convex can also be used, for example:

userProfileSimilarityScore *= δ ** ***staticUserprofileSimilarityScore*** *+ (1 - δ) ** ***dynamicUserprofileSimilarityScore,***

for *δ* in[0,1].

If the user tends to jump a lot to sports news after reading politics, then the *UPtrans(Acurrent)* will capture this to some extent and would produce fresh suggestions.

Popularity

Currently, we use a daily popularity rank: All articles from some date *D* are ordered according to the count of the unique user's views. Article *A* from date *D* becomes the total visit boost:

totalVisitBoost of A = 1 + #articles from date D with less views than A / #articles from date D

Thus *totalVisitBoost* measures the relative popularity of the article with respect to the other articles from the same date (week, etc.).

Clearly, it is easy to modify this boost by widening the time frame (from 1 day) to influence the way in which the popularity rank is calculated. The ***popularity*** score in the final formula is precisely the *totalVisitBoost*and the parameter *wP* controls the importance of the boost.

Collaborative component

At this stage, the collaborative component is realised by a one-step co-visitation matrix. An important part of our future work will be to improve the collaborative component of the recommender.

Article-article CovisitationBoost: For all news in the set recentA of all news not older than *N* days (currently 30 days), we create the article-article matrix *articleSim* with dimensions *#recentAx #recentA,* which is initialised by zeros.

When an arbitrary user transitions from article *Ai* into *Aj,* the elements *sim(Ai, Aj)* and *sim(Aj, Ai)* of the matrix *articleSim* are incremented by one. Hence, the element *sim(Ai, Aj)* contains the number of all transitions from *Ai* to *Aj,* and from *Aj* to *Ai* .

The collaborative covisitation boost, *covisitBoost* for article *Aj*with respect to a user viewing article *Ai* is given by:

covisitBoost = 1 + sim(Ai, Aj) / maxj sim(Ai, Aj).

As already mentioned, currently the collaborative score is just the co-visitation boost.

Freshness

The freshness factor is based on a function *fR* for an article *Ai* with time stamp *ti*(using the Solr*recip()* function):

$$freshness=fR(Ai) = a / (m * (t_{now} – ti) + b),$$

with*a, m, b* pre-set to sensible values: in our current setting, the freshness amounts to 1.0 for today's news and drops in half after a month. Obviously, due to scaling invariance these are just 2 effective parameters and their values depend on the time unit. In the case it is days to have a value of 1.o for today's news and value of 0.5 for news that are 30 days old we need to have (again up to a scaling factor): a=1.0, b=1.0 and m=1.0/30.0.

As a result, the system does not recommend articles that the user has already viewed.

4 CONCLUSIONS

In this report, we have presented the development of a Decision Support System and a Recommendation engine as part of T5.4. These systems are implemented on top of GraphDB.

The developed DSS will be used mainly for the purposes of Use Case 3 – SME Internationalisation. It will help entrepreneurs in small and medium enterprises with some of their decision-making processes. As a foundation, the system uses statistical data provided by the World Bank and Eurostat. This data comprises indicators from different categories – Economy, Health, GDP, Corruption, etc. Based on these categories, the system will provide a comparison between different countries with the help of interactive visualisations, charts and graphics.

The Recommendation engine will help journalists and media monitoring agents in their daily jobs by providing additional information about specific topics. Semantic reasoning will be used to increase the percentage of recognized entities by the NER service and in this way, the recommender will provide more clear and accurate results. We will continue to develop this system during the next period and the improvements will be reported in D5.4.

5 REFERENCES

Altman, N. S., "An introduction to kernel and nearest-neighbor nonparametric regression".The American Statistician 46 (3): 175–185, 1992.

Blomqvist, E. 2012. “The Use of Semantic Web Technologies for Decision Support - A Survey”, Linköping University, p. 1-24.

Breese, J. S., Heckerman, D., andKadie, C. 1998. “Empirical analysis of predictive algorithms for collaborative filtering”, In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, p. 43-52, Morgan Kaufmann Publishers Inc.

Brusilovsky, P.2007. “The Adaptive Web”, p. 325, ISBN 978-3-540-72078-2

Burke, R. 2002. “Hybrid Recommender Systems: Survey and Experiments”, California State University, Fullerton Department of Information Systems and Decision Sciences.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin M., “Combining content-based and collaborative filters in an online newspaper,” in Proceedings of the SIGIR Workshop on Recommender Systems: Algorithms and Evaluation, Berkeley, Calif, USA, 1999.

Das, A., Datar, M., Rajaram, S., “Google News Personalization: Scalable Online Collaborative Filtering", in Proceedings of WWW ’07, 2007.

Ekstrand, M. D., Riedl, J. T., and Konstan, J. A. 2011.“Collaborative filtering recommender systems”, Foundations and Trends in Human-Computer Interaction,vol 4 (2), p. 81-173.

Lops, P.,De Gemmis, M., and Semeraro, G. 2011. “Content-based Recommender Systems: State of the Art and Trends”, Chapter 3.

Power D. J., "Decision Support Systems: Concepts and Resources for Managers", Westport, CT: Greenwood/Quorum Books, 2002.

Power, D. J., "What are examples of decision support systems in global enterprises?", DSS News, Vol. 7, No 7, 2006.

Sprague, R. 1980. “A Framework for the Development of Decision Support Systems”, MIS Quarterly, vol. 4 (4), p.1-25.

Su, X., and Khoshgoftaar, T. M. 2009. “A Survey of Collaborative Filtering Techniques”, Polytechnic Institute of NYU, Bell Labs, Alcatel-Lucent, Netflix Inc., Los Gatos.

Van Meteren, R., and Van Someren, M. 2000. “Using Content-Based Filtering for Recommendation”, NetlinQ Group, University of Amsterdam.