M-Eco enhanced Adaptation Service (D5.3)

Lage, Ricardo Gomes; Leginus, Martin; Dolog, Peter; Durao, Frederico; Pan, Rong; Diaz-Aviles, Ernesto

Publication date:
2012

Document Version
Early version, also known as pre-print

www.meco-project.eu

M-Eco Adaptive Tuning and Personalization (D5.3)

Project title:	**Medical Ecosystem - Personalized Event-Based Surveillance**
Project acronym:	M-Eco
Project number:	247829
Project instrument:	EU FP7 Small or Medium-scale Focused Research Projects (STREP)
Project thematic priority:	Information and Communication Technologies (ICT)
Document type:	D (deliverable)
Nature of document:	R/P (report and prototype)
Responsible editors:	Peter Dolog
Authors:	Ricardo Lage[1], Martin Leginus[1], Peter Dolog[1], Frederico Durao [1], Rong Pan [1] and Ernesto Diaz-Aviles[2]
Quality Assessor:	Avare Stewart Email:`stewart@l3s.de`
Contributing participants:	AAU, L3S
Contributing workpackages:	WP5

Abstract

In this report, we present the final version of the Adaptive Tuning and Personalization (WP5) component of the M-Eco system. This component is focused on four main areas of interest to users of surveillance systems: presentation options for recommendation and adaptation, user and group models, user classification and modeling algorithms, and recommendation, adaptation and personalization strategies. In each of these areas, we propose improvements over those presented in the previous deliverable incorporating feedback from medical surveilance experts. The personalization component is enriched with several extensions. The multi-factor recommendation model is incremented with more factors, including time decay, tagging activity and location. Location is also improved with a method to predict event trajectories. Moreover, a new family of tensor-based recommenders is presented. The tag cloud model is improved with two new tags selection algorithms. We also introduce a propagation of the (ir)relevant terms obtained from tag clouds usage to WP3 and WP4 in order to improve their data collections algorithms. The evaluation part of the report summarizes the benefits and drawbacks of the models presented and describes possible directions for the future work.

Keyword List

social tagging, recommender system, tag cloud, group recommendation

M-Eco Adaptive Tuning and Personalization (D5.3)

Ricardo Lage[1], Martin Leginus[1], Peter Dolog[1], Frederico Durao [1], Rong Pan [1] and Ernesto Diaz-Aviles[2]

[1]Department of Computer Science, Aalborg University
Email:dolog,fred,ricardol,mleginus,rpan@cs.aau.dk
[2] L3S research Center / University of Hannover
Email:diaz@l3s.de

10 August 2012

Abstract
In this report, we present the final version of the Adaptive Tuning and Personalization (WP5) component of the M-Eco system. This component is focused on four main areas of interest to users of surveillance systems: presentation options for recommendation and adaptation, user and group models, user classification and modeling algorithms, and recommendation, adaptation and personalization strategies. In each of these areas, we propose improvements over those presented in the previous deliverable incorporating feedback from medical surveilance experts. The personalization component is enriched with several extensions. The multi-factor recommendation model is incremented with more factors, including time decay, tagging activity and location. Location is also improved with a method to predict event trajectories. Moreover, a new family of tensor-based recommenders is presented. The tag cloud model is improved with two new tags selection algorithms. We also introduce a propagation of the (ir)relevant terms obtained from tag clouds usage to WP3 and WP4 in order to improve their data collections algorithms. The evaluation part of the report summarizes the benefits and drawbacks of the models presented and describes possible directions for the future work.

Keyword List
social tagging, recommender system, tag cloud, group recommendation

Contents

1 Introduction **1**

2 Motivation **3**
2.1 Advantages and Motivations for the Personalization and Adaptation Methods . . 4
2.2 Improving Signal and Document Recommendations 5
2.3 Improving Navigation Elements 5

3 Design and Architecture **6**
3.1 Component Integration 7
3.2 Component Packaging 8

4 Adaptive Tuning and Personalization Models **10**
4.1 Analysis of Presentation Options for Recommendation and Adaptation 10
4.1.1 Introduction 10
4.1.2 Addressing User Feedback on UI Elements 11
4.1.3 Improving User Feedback Options 12
4.1.4 Addressing User Feedback on Tag Clouds 16
4.1.5 Explaining Recommendations 17
4.2 User and User Group Model Definition 19
4.2.1 Tagging Model 19
4.2.2 Location Preference Model 20
4.3 User Classification and Modeling Algorithms 20
4.3.1 Semantic Enhanced Tag-Based Recommendation 21
4.3.2 Expanding Tags with Neighbors For Improving Search Retrieval 21
4.3.3 Spectral Clustering of Tag Neighbors for Recommendation 22
4.3.4 Predicting Event Trajectories 22
4.3.5 Improving Multi-Factor Based Models 25
4.4 Recommendation, Adaptation and Personalization Strategies 26
4.4.1 Recommendations based on Signal Definitions 27
4.4.2 Tag-Based Recommendations 27
4.4.3 Recommendations based on Event Trajectories 27
4.4.4 Personalized Tag Cloud Adaptation and Navigation 28
4.4.5 Tensor Based Recommendations 33
4.4.6 Recommender System Techniques for Threat Assessment 39

5 Evaluation **40**
5.1 Evaluation of Multi-factor Recommendations 40
5.1.1 Introduction 40
5.1.2 Results 41
5.2 Personalized Tag Cloud Evaluation 41
5.3 Evaluation of Generative Model of User Taggings 47
5.4 Trajectory Prediction Evaluation 48

6 Deployment **52**
6.1 Installation Requirements 52
6.2 Web Services 53
6.2.1 Web Service Providers 53
6.2.2 Web Service Consumers 54

7 Related work **54**
7.1 Predicting Events 54
7.2 Tag Expansion in Recommendation 55
7.3 Personalized Tag cloud 56
7.4 Spatial Reasoning 56

8 Conclusion and Future Works **57**

1 Introduction

This deliverable presents enhanced methods of user and group modeling, and recommendation for event detection, which are part of the Adaptive Tuning and Personalization (WP5) component of the M-Eco system. It aims at adapting the information presented in the system based on what is likely to be more useful to a particular user or a group of users in different scenarios. In comparison to the previous deliverable D5.2 [11], we focus more on recommendation and navigation aspects of WP5 component as was requested by the feedback of M-Eco users and reviewers. We take into account suggestions to evaluate our recommendation methods and to improve GUI elements impacting navigation. We present an extended GUI for signal assessments, also improved tag cloud interface. These enhancements allow to collect users feedback and in consequence improve the following algorithms:

- Recommendations are able to generate more personalized results
- Signal generation process can be adjusted according to the users's signals assesments
- Data collection algorithms can incorporate users's feedback about (ir)relevant terms.

The recommendation aspect of this component is enriched with more developed multi-factor recommendation algorithm that incorporates users feedback from improved GUI. Moreover, tensor-based recommendations are introduced. The evaluations were conducted with more medical experts and majority of them consider algorithms from WP5 as useful. However, several improvements have to be incorporated in the future.

For a better readability of this report, in the following paragraphs, we explain commonly used terms. **Item**, in this case, is the general term used to represent anything that can be presented to a user, such as news articles, reports, blog posts as generated by methods from WP3 or signals in aggregated form as generated by methods from WP4. Adaptation is achieved through a process of modeling user interests in terms of items of the system, and recommending those that are more relevant.

In other words, the WP5 component adapts the M-Eco system to the needs of the user. It is an essential component in assisting users who have to synthesize an increasing number of facts, assess risks and react early to public health threats. Personalization and adaptation in the M-Eco project aims at customizing the items shown to the individual users, based on their use of the system.

In M-Eco, items used for recommendation are documents, tags and signals. **Documents** can refer to any source of information on the web identified by WP3 [37] as potentially relevant for health surveillance. These include medical blog posts, news articles or videos discussing epidemics, and personal messages in social networks describing one or more symptoms. **Tags** are labels that summarize a document succinctly. They can be assigned by users of M-Eco, as a way to let them organize the documents presented to them, or generated automatically, as a way to help the users browse through the documents. Signals are aggregations of documents corresponding to a specific set of medical conditions (e.g., symptoms or diseases) and locations.

Signals are produced to specific user needs when matched with one or more of their signal definitions. These definitions are a set of rules specified by the user in terms of explicit medical conditions and locations he or she is interested in having the system monitoring. Once the user specifies signal definitions, this component can provide a toolset which mainly helps in the following way:

- in presenting and ordering (recommendations of) signals and documents as results of signal definition matchmaking; and,
- exploring and navigating users for validation purposes.

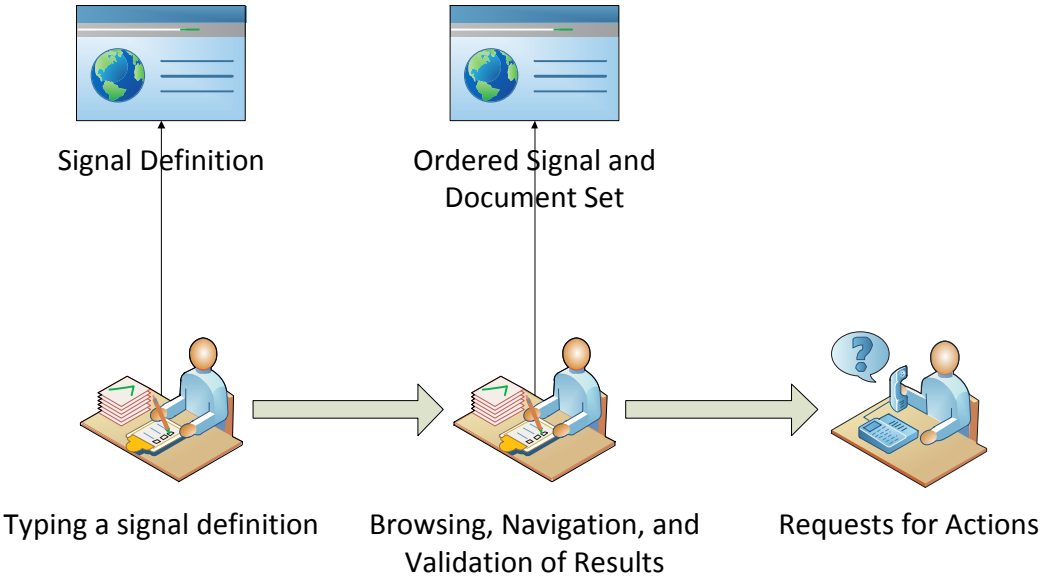


Figure 1: A medical surveillance user workflow concerning the signal definition, retrieval and navigation

The toolset follows a general workflow which is depicted in Figure 1. The workflow consists of the tasks where the medical personnel: 1) defines a signal definition, 2) review of retrieval realized by validating recommended results which match the signal definition at the signal and document level and finally 3) issue of requests for actions if needed.

The M-Eco WP5 services are targeted at the support for the second task and, therefore, the figure depicts a link between those tasks and web page symbols which will provide services not only from WP5 to support the task. WP5 methods rely on an availability of pre-processed data from WP3 and WP4. WP3 provides documents, messages and news from relevant chosen data sources indexed with disease, symptom and location information. WP4 aggregates the documents into signals and creates new items to be recommended. WP5 uses that information together with the signal definitions of the users to produce recommendations to them.

Signal definitions are the main source used to build a user or a group profile. Such a profile is used to encode the user preferences and needs in forms of models which will be introduced

later in this deliverable. It is then used to produce recommendations to the user. Besides the signal definition, WP5 also uses the user tagging activity and ratings as aspects to profile the user. These were already considered in methods described in the previous deliverable.

The main contributions of this deliverable are as follows. In addition to what we have already described in terms of the user profiling in previous deliverable, we extend the profiling with the *user's stated preferences for a set of locations*. We also propose *new approaches for predicting event trajectories* and *navigation through the items in the system using tags*. The extensions and new methods address diverse aspects of the M-Eco system such as motivation scenarios, configuration management, formal models, implementations and some real use case scenarios on how personalization can be applied to support medical surveillance. The result of personalization can be seen, for example, as a reordering on the ranking of recommendations or search results or a personalized tag cloud leading users to quick access the information fulfilling the users interests.

These contributions will be covered in the remainder of this deliverable. It encompasses motivations, use cases, and scenarios describing benefits of personalization in M-Eco, design and architecture of components, underlying models and their evaluation and deployment requirements. The content of this deliverable is organized as follows:

- Section 2 presents the motivations and assumptions for demonstrators, use cases and methods which utilize the personalization and adaptation techniques researched in this workpackage. It presents a concrete motivational medical scenario where users benefit from personalization.
- Section 3 presents design and architecture of personalization components including packaging. This section serves as an overview on what was delivered in the past deliverable, what is delivered now and what is planned to be delivered in forthcoming half a year.
- Section 4 introduces the personalization and adaptation models studied in this workpackage.
- Section 5 presents the evaluation of the personalization and adaptation components from user and performance viewpoints.
- Section 6 presents the requirements, installation guidelines and configuration for using M-Eco WP5 components and demonstrators either as a stand alone applications or through web services.
- Section 7 discusses the work delivered in the context of related work.
- Section 8 concludes the work, outlines the major achievements and points out future works.

2 Motivation

As discussed in the M-Eco Deliverable 5.2 (D5.2) [11], *section 2*, one of the main tasks of medical surveillance personnel is to identify whether there is a risk of an epidemic outbreak. M-Eco works towards the goal of supporting this task by providing means to detect and suggest those signals and documents which reflect upon such emerging situations from (social) web sources.

In the D5.2 deliverable, this task was addressed by WP5 methods in the following demonstrators and use cases:

- Signal and document recommendations;
- Modeling group work interactions;
- Navigating through documents for validation of signals.

In this deliverable we focus on recommendation and navigation aspects. Recommendations are used to help users deal with an increasing amount of items to be assessed by surveillance personnel. At the same time, personalized navigation schemes can assist them in browsing through these recommendations.

These two aspects are improved upon considering the feedback of M-Eco users and reviewers. We take into account suggestions to evaluate our recommendation methods and to improve GUI elements impacting navigation. Evaluations conducted and improvements implemented were performed in the context of an integrated prototype. Since this is the final version of our component, we wanted to consider the pipeline of the M-Eco system, assuming a streamlined feed of data from WP3, WP4 and WP5.

The remainder of this section is organized as follows. Each of the motivations stated in the previous deliverable will be summarized next (subsection 2.1). These motivations still represent the basis for the development of our component. The next sections detail the specific motivation for the new methods described in section 4 of this deliverable. We present first the motivation for improving upon the recommendations of signals and documents. Second, we present the motivation for improving navigation elements.

2.1 Advantages and Motivations for the Personalization and Adaptation Methods

In the previous deliverable [11], we presented the advantages and motivations for our methods in terms of **number of users**, **collaboration between users**, and **a need for navigation means**.

First, recommendations are important to individual users, regardless of the total number of them. Even considering the pre-processing done by WP3 and WP4, we still have to consider a large number of signals to filter and rank before presenting to the user. Figure 2 shows the number of signals from WP4 we fetched per day in the first semester of 2012. On two occasions, we had over 10,000 signals fetched in just one day. April was the month with the highest average, around 6,000 signals per day, while January was the lowest, around 700 signals per day. Even considering a month like January, 700 signals per day, with several documents each, is a large amount of data to go through.

Besides the large quantity of items, we have to deal with the different ways that the surveillance tasks are organized in each country and at the European and World level. This means that a specific health official as a potential user of the system looks only at specific diseases affecting specific locations. For this reason, WP5 deals with presentation and information overload issues by providing only those items (documents or signals), which are assumed to be relevant for particular users according to their interests. This also means that, even though we have a finite number of users provided by Robert Koch Institute (RKI) and the Governmental Institute of Public Health of Lower Saxony (NLGA), they can still benefit from recommendations in order to be able to assess relevant items.

Second, our component attempts to take advantage of the collaboration between users. In the previous deliverable, we looked at group work where several colleagues work together on

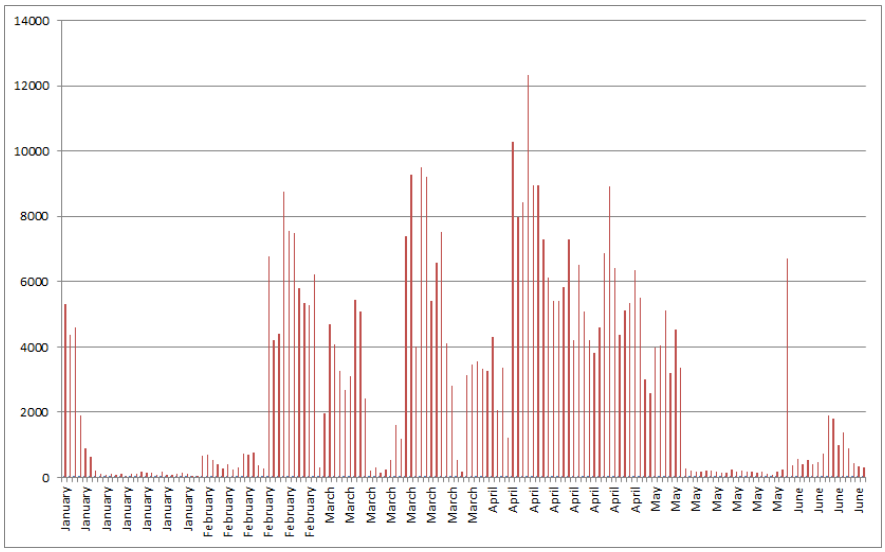


Figure 2: Number of signals WP5 fetched per day from WP4, from the beginning of January until June 15th

the same or similar surveillance task. We showed that such group settings can provide useful recommendations for these users. By monitoring their interactions in the context of a specific outbreak, the system can infer the group preferences and suggest related items accordingly.

Third, a personalized navigation scheme creates the means for the user to browse efficiently through recommended items. We consider tag clouds as a GUI element that can intuitively provide the user with snippets of content from the items the user is browsing. Tag clouds can, then, reduce information overload and also assist the user in discovering related contents.

2.2 Improving Signal and Document Recommendations

Recommendations can be improved by addressing the limitations of previous models and by complementing them with new methods. We consider mainly the limitations of the multi-factor model presented first in section 2.1 of our first deliverable [55]. Such a model has the potential to encompass together the results of different methods. By doing so, it can tune these methods according to the feedback given by individual users.

In this deliverable we show a multi-factor model expanded with more factors and also propose a new model using collaborative filtering. We also consider the weights given by the model as input to determine what influenced the recommendation the most. This information is used to explain the recommendation to the user.

2.3 Improving Navigation Elements

Tag clouds facilitates an exploration, browsing and validation process of a large number of documents. This was proved by several evaluation studies presented in the previous deliverables. However, users have raised the following drawbacks:

Figure 3: An example of a tag cloud that contains a document that is not related to a medical domain and is irrelevant for medical surveilance experts.

- Users should be able to assign new tags and also to remove the pregenerated taggings from the documents and in a such way change the structure of the tag cloud.
- Terms that were marked as (ir)relevant should be exposed to other partners so they can dynamically adjust retrieval process of documents.
- Tag cloud should present more relevant terms instead of general terms with high frequency.
- When a tag cloud reflects a large set of documents, a structure of the cloud can be biased and not necessarily capture all the relevant details of the underlying documents.

In order to better understand afore-mentioned problems, we provide an example of a tag cloud where one underlying document was harvested by WP3 because of a term *Euro fever* (please see Figure. 3). Obviously, this term does not have any medical meaning and it expresses a twitter user's excitment from European Football Championship. The document does not have any relevancy for medical surveilance experts. Therefore, there is a need to propagate this information to the data collection algorithms so that this noisy data can be filtered out in the beginning of the pipeline of M-eco system. Hence, a new tag cloud model allows users to remove inappropriate taggings generated by the system. These irrelevant terms are collected and consequently exposed via webservices to our partners so they can improve data collection process. The tag cloud model also allows users to annotate documents with more relevant taggings. Moreover, we delevoped new tags selection algorithms that optimize structure of tag clouds in terms of coverage and overlap (synthetic metrics).

3 Design and Architecture

This section briefly summarizes the architecture of WP5 framework presented in our previous deliverable (Section 3) [11] and extends it with the contributions from this deliverable. The

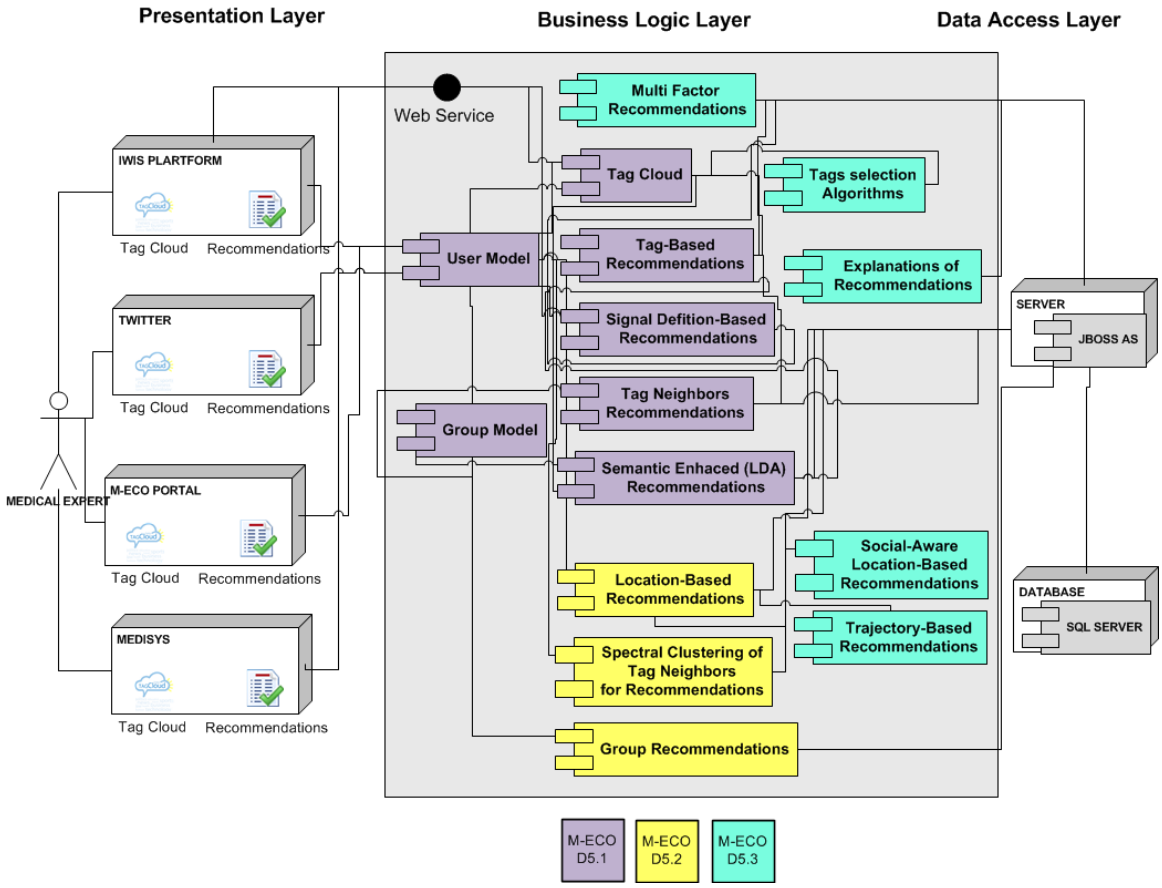


Figure 4: WP5 Component Integration.

component integration section shows how components integrate with the external system and how they connect between themselves. The component packaging provides a logical view of packages that implement the components. In this deliverable, the presentation and business logic layers with corresponding components (Recommendation, Tag Cloud, Web Service and Web Interfaces) are updated and improved.

3.1 Component Integration

Figure 4 shows the complete WP5 infrastructure as presented in the previous deliverable. It is divided in three layers: presentation layer, business logic layer and data access layer.

- **Presentation Layer** - is shown on the left part of Figure 4 and is represented by personalization functionalities (recommendations and tag cloud) implemented at the user interface of the following information systems: M-Eco portal [1], Medisys Health Informa-

[1] http://139.191.1.59/m-eco/

tion System [2], updated IWIS platform that incorporates users feedback [3] and Twitter [4]). An illustration of recommendations is shown in Figure 7 and the illustration of an improved version of tag cloud is shown in Figure 11. Worth mentioning that the services provided by the recommendations and personalized tag cloud are also available through REST web services (details in Section 6.2).

- **Business Logic Layer** - is shown in the middle part of the Figure 4 and is represented by components that generate recommendations and personalized tag clouds to the information systems at the presentation layer. It also shows how the components interact with each other. At first, the user and group model components build the user and group model with information gathered from the information systems in the presentation layer, then these user models are utilized by the recommendation and tag components for producing the personalized recommendations or tag clouds. The components are rendered in three different colors distinguished by the M-Eco deliverable where they are described. The components colored in *purple* were described in the M-Eco Deliverable D5.1 [55], the components colored in *yellow* are in implemented for the current M-Eco Deliverable D5.2 and the components colored in *turquoise* are implemented in the M-Eco Deliverable D5.3.

- **Data Access Layer** - is shown on the right part of the Figure 4 and is represented by components that maintain and access data on the database. The components in the business logic layer manage transactions to the database including storage, updates and removals. The components in the middle layer connect to the Microsoft SQL Server database through the JBoss Application Server. Worth mentioning that we are changing the current database software to PostgreSQL 8.3 due to license constraints.

The Figure 4 shows components in the Business Logic Layer as they were reported in different deliverables. It highlighted what was expected to be reported in this deliverable. Of the expected items, we disregard the social-aware recommendations, focusing instead on items requested by the feedback of M-Eco users and reviewers.

Multi-factor recommendations are discussed in Section 4.3.5. Tag-based methods are discussed in Section 4.4.2. We introduce a method for explaining recommendations to the user in Section 4.1.5. We also introduce a method for prediction of event trajectories in Section 4.3.4.

3.2 Component Packaging

The component package presents a logical view of packages (and their connections) that contain the actual model implementation of the components shown in Figure 4. This remains the same as in the previous deliverable (Section 3.2) and is presented here for consistency.

- **Model** - Package where the conceptual classes are placed such as Document, Indicator, User, Signal, etc. The individual and group models are explained in Section 4.2.

- **Recommendation** - Package where the classes that implement the personalized recommendations. This package utilizes classes from package Model and Database. The recommendation models implemented are *Signal Definition-Based Recommendation* (Section 2.2 of M-Eco Deliverable D5.1) [55], *Semantic Enhanced Recommendations* (Section

[2] http://medusa.jrc.it/
[3] http://demos.iwis.cs.aau.dk:8081/meco/. Use 'euro' as login and 'iwis' as password.
[4] http://twitter.com/

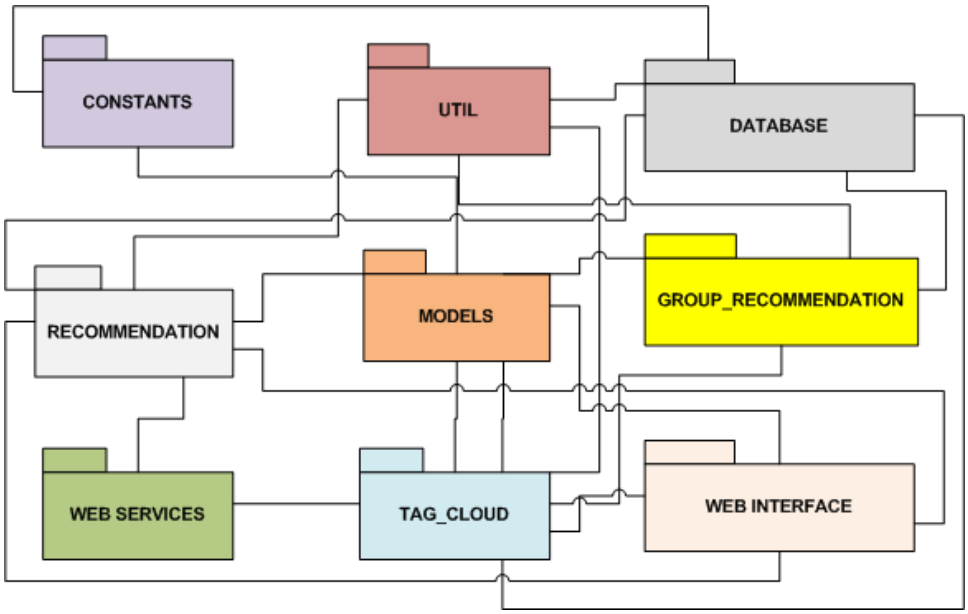


Figure 5: WP5 Component Packaging.

2.4 of M-Eco Deliverable D5.1 [55]), *Tag Neighbors-based Recommendations* (Section 2.3 of M-Eco Deliverable D5.1 [55]), *Spectral Clustering of Tag Neighbors for Recommendation* (Section 4.3.3), *Tag-Based Recommendations* (Section 4.4.2), *Tensor-Based Recommendations* 4.4.5) and *Location-Based Recommendation* (Section 4.4.3).

- **Group Recommendation** - Package where the classes that implement the recommendations addressed to groups. This package utilizes classes from package Model and Database. The group recommendation model is presented in our previous deliverable (Section 4.4.4 [11]).
- **Tag Cloud** - Package where the classes that implement the personalized tag cloud. This package utilizes classes from package Model and Database. The personalized tag cloud model and two different tag selections algorithms are explained in (Section 4.4.4). Moreover, the extended version of tag clouds supports removal of irrelevant terms.
- **Web Service** - Package where the classes that implement the RESTFUL web services providers and the web service clients (Section 6.2).
- **Util** - Package where the utilitarian classes are for string and file processing. Classes from all packages, except the Model one, access classes from the util package.
- **Constants** - Package where the widely constants are placed such as database url and credentials.
- **Database** - Package where the classes that handle the database access as well as queries are stored. Classes from all packages, except the Model, Util and Constants, access classes

from the Database package.

- **Web Interface** - Package where the classes that handle all assets related to the user interface. This package communicates with recommendation, group recommendation and tag cloud packages. The web interface for the tag cloud package was adjusted to allow users to remove irrelevant tags and provide their custom tags as presented in Section 4.1.4.

4 Adaptive Tuning and Personalization Models

This section presents the personalization methods proposed for solving the problems introduced in the Section 2. In order to address the issue of *users dealing with the huge amount of items in their daily activities*, we provide an analysis of presentation options so that the user interface is made up with intuitive and friendly components. In addition, we propose a number of recommendation methods for selecting the most appropriate piece of information by filtering out noisy and unimportant content. In order to support *groups of individuals that share similar interest with relevant information*, we provide means of combining individual preferences into group profiles so that they receive recommendation of common interest. In order to help users *navigating through vast collection of documents and finding new items*, we provide a visual representation of documents through a tag cloud component. Besides indexing documents in the corpus, each tag helps users to find new related information of interest. The following subsections present different models used to address each of the above problems with a particular contribution. These methods are also evaluated in user studies or controlled environments as shown in Section 5.

4.1 Analysis of Presentation Options for Recommendation and Adaptation

4.1.1 Introduction

In the previous deliverable [11] (Section 4.1), we discussed three aspects that user interfaces for the M-Eco system should consider:

- It should allow for browsing of relevant information and reduce information overload for users. For instance, a subset of relevant items should be displayed rather than a vast collection of possibly relevant documents.
- It should present a timeline of signals and documents in order to provide a time overview for the displayed resources and in a way that facilitate user orientation and exploitation of a large number of signals or documents.
- It should facilitate the exploitation of a large number of documents with alternative information views to provide an overview of a given set of documents. For instance, simple listing of related documents is sufficient when the number of such resources is small. When there is a need to explore a larger set of documents the interface should provide a navigational tool that presents a general picture and important properties of these documents. Such navigational component should lead a user only to important and relevant subsets of the documents and minimize an overhead from exploring all documents.

Figure 6: Top view of the current GUI of the M-Eco system

This analysis of presentation options is aimed at investigating techniques for adapting, structuring and browsing information in an intuitive and friendly way to the end user. For the M-Eco prototype this analysis is important because it provides means of easing the work of medical experts while navigating through documents in the system.

In this deliverable, we build on our previous work mainly considering the feedback given by users in the evaluations. We first discuss this feedback as reported in previous deliverables [3,11], highlighting some limitations raised. Next, we present the modifications we performed to the GUI with the aim of addressing these limitations. We do this in two steps, first discussing user feedback options and then tag clouds. We finish this section introducing the method by which we aim to give to the user an explanation of the recommendations received.

4.1.2 Addressing User Feedback on UI Elements

The Second Evaluation Report [3], describes the results of the first user assessment of functionality. It was conducted in Ispra, Italy, in May 2011. The aim of this assessment was to provide developers of the M-Eco system with feedback from the users. These users, after using the current M-Eco GUI shown in Figure 6[5], answered a questionnaire covering different aspects of it.

According to the Evaluation Report, users agreed only with specific properties of the GUI. For example, they considered the tag cloud useful and agree that rating of signals and documents can be useful. On the other hand, the process of searching for signals, the information they

[5]This GUI can be accessed at http://139.191.1.59/M-Eco/Default.aspx

provide, and the way content is organized were all points that should be further improved, according to the users.

All users agreed that the current listing of signals is not enough to search for signals. There should be some explicit functionality for that purpose. In general, there is some disagreement on the M-Eco GUI's ability to communicate information. They disagree, for example, that the initial M-Eco page provides a complete overview of what should be relevant to the user.

This seems to be mainly because the signals are not presented properly. Most users agree that this presentation can be improved. Similarly, some users showed concern with regard to understanding the summary found in each signal.

In addition to these points raised in the Evaluation Report, our previous deliverable [11] discussed in Section 4.1 the role of a timeline of relevant documents and signals. Such a feature was not readily available at the M-Eco prototype and, for this reason, we demonstrated it using Twitter instead. The interface of Twitter provides a timeline of short messages (less than 140 characters) called tweets that were posted by users. Interesting tweets can be marked as favorite and also shared by other users of the social system. We utilize Twitter as the presentation tool that provides a list of recommended documents and signals ordered by time.

We also discussed in Section 5.2.1 the suggestion given by users to add explanations to the recommendations given to them. Because the relevance of a recommendation is relative, an explanation of its reason could give a context. Just a "yes/no" rating, in this sense, may not represent the proper interests of the user.

4.1.3 Improving User Feedback Options

In our previous deliverable [11], in order to improve the performance of our personalization models, we analyzed a number of feedback options that were eventually applied either in our components or in the experimental evaluations of our components. The Section 5.2.1 shows a group discussion where we evaluate the assessment of our recommendations. Visual elements for users to give feedback to the recommendations include:

- *Thumbs-up or thumbs-down* is a hand gesture with the thumb extended upward or downward in approval or disapproval, respectively. This provides a binary assessment of an item recommended with the meaning positive for thumbs-up and negative for thumbs-down. The disadvantage is that a user may be ambivalent and none of the options will actually represent his feeling.

- *Rating Scale* is a means of assessment in terms of quality, quantity, or some mix of both. For example, one to five stars is commonly employed to categorize hotels. In our experiments, we extensively utilized a 5-star rating scale meaning 1-(irrelevant), 2-(little relevant), 2-(average), 2-(relevant), 2-(very relevant).

- *Comments* is a means of users describe their impression about the system itself textually. The advantage of this method is that they are not constrained by a set of alternatives. The drawback is that only a few users voluntarily provide comments.

- *Comments* is a means of identifying the user's interest in a given topic or tag by account the amount of clicks an item sums up in a period of time. This is an implicit way of collecting users preferences so that they are not instructed to act.

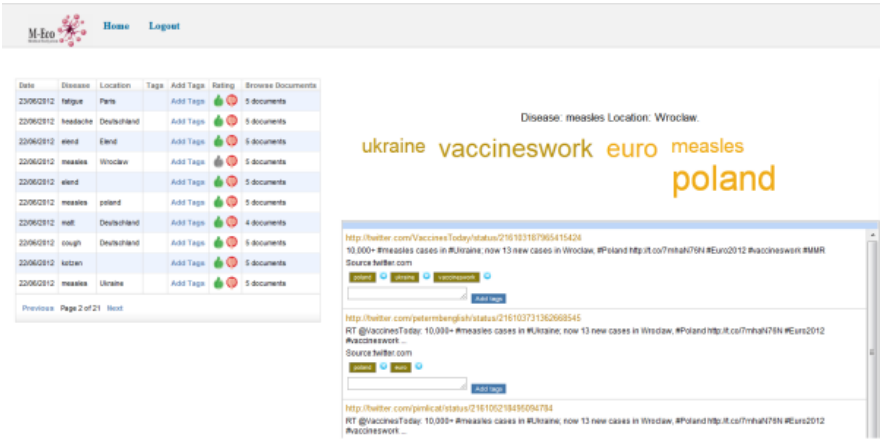


Figure 7: Small modifications to the signal navigation of the existing GUI

- *Bookmarking* is a direct indicator of a user's preferences once the user explicitly determines what his preferences are. This premise is more reliable when the bookmarks are labeled as "Favorite" or related meaning.

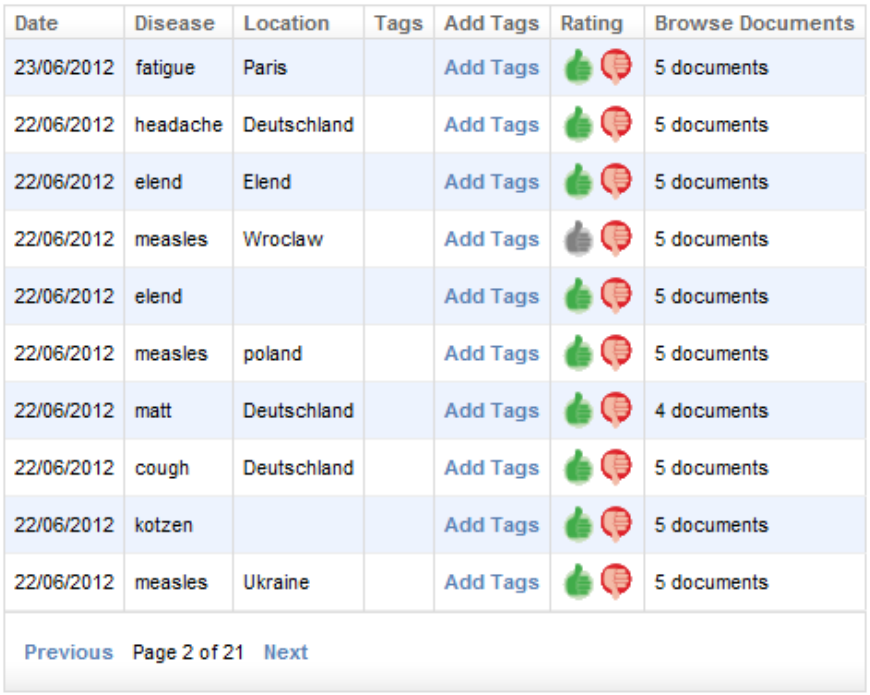
In this deliverable, we discuss changes in the interface taking into account these elements and the discussions with the users as described in Section 4.1.2. We built a new GUI as opposed to the current one presented in Section 4.1.2 with the goal of evaluating the changes, contrasting both versions. This new GUI was built in two steps. We first added small modifications to the signal navigation. This first version is a working GUI, although with limited functionality, that can be seen in comparison to the current GUI [6] Next, we improved further the GUI in a prototype version. In it, we tried to incorporate requests from users we considered relevant to augment their usage experience. The evaluation of this prototype should be presented in the third and last Evaluation Report.

Figure 7 shows a screenshot of the first version of the changed GUI. Figure 8 highlights the table elements to navigate through the signals. The changes made to the tag clouds are discussed in Section 4.1.4.

The first noticeable change is that the list of signals is now paginated. In the current GUI, the full list of signals for the logged user was typically fetched. This was overburdening our web services, forcing the list to be constrained: Only the more recent signals are currently being returned, since the GUI also does not support paginated requests to the web services. Pagination, then, allows the user, if he or she wants, to browse through all the signals relevant to him or her, without the overload of one big list. In the figure above, page 2 of 21 is shown. That means that the logged user has currently over 200 relevant signals.

The lists were also tabbed, separating recommended, predefined and other signals in the current GUI. We are now proposing one unified list where the ordering of the signals plays a more important role. We are experimenting with two different ordering approaches. One is the typical order by date already available in the current GUI. The other is an ordering by the score of the recommendation which takes the date into account but also other factors. When the user logs in, one of the two ordering approaches is chosen randomly. We expect to evaluate

[6] You can access it at http://demos.iwis.cs.aau.dk:8081/meco/. Use 'euro' as login and 'iwis' as password.



Date	Disease	Location	Tags	Add Tags	Rating	Browse Documents
23/06/2012	fatigue	Paris		Add Tags		5 documents
22/06/2012	headache	Deutschland		Add Tags		5 documents
22/06/2012	elend	Elend		Add Tags		5 documents
22/06/2012	measles	Wroclaw		Add Tags		5 documents
22/06/2012	elend			Add Tags		5 documents
22/06/2012	measles	poland		Add Tags		5 documents
22/06/2012	matt	Deutschland		Add Tags		4 documents
22/06/2012	cough	Deutschland		Add Tags		5 documents
22/06/2012	kotzen			Add Tags		5 documents
22/06/2012	measles	Ukraine		Add Tags		5 documents

Figure 8: New table to navigate through the signals

which ordering is best by evaluating the amount of positive ratings the signals receive in each of these approaches.

In each page of the table listing the signals, we tried to make the ability to tag or rate the signal more explicit. These two forms of feedback are now explicitly listed as columns in the table. The rating is dynamic so that when the 'thumbs up' or 'down' is selected, it is immediately greyed. We also added the total number of documents available for this signal in the last column. This information is a link to list those documents through the tag cloud. The addition of such link makes the access to the documents more explicit to the user, facilitating the navigation.

In terms of feedback options, as presented above, we kept the 'thumbs up and down' approach. Since the users agreed with this method (see Section 4.1.2), we decided not to change it but, instead, to make it more explicit. We did add one other form of evaluation, though. We now count the clicks to access the documents of a signal and the order of a signal rated. We use the results of the experiment described in [11] (Sections 5.3 and 5.5) for counting the clicks. We showed that the clicks can provide supportive feedback for recommendations.

In addition, we raise a new hypothesis for storing the order of the rated signal. For instance, if the user rates positively the first signal in the table list, we believe that this signal should have a lower importance than, say, the signal in position ten which was also rated positively. The intuition behind this approach is that positive ratings in the first positions should be expected whereas positive ratings in lower positions need to be weighted up. Conversely, negative ratings rated in the first positions should receive a higher negative weight than negative ratings down the list.

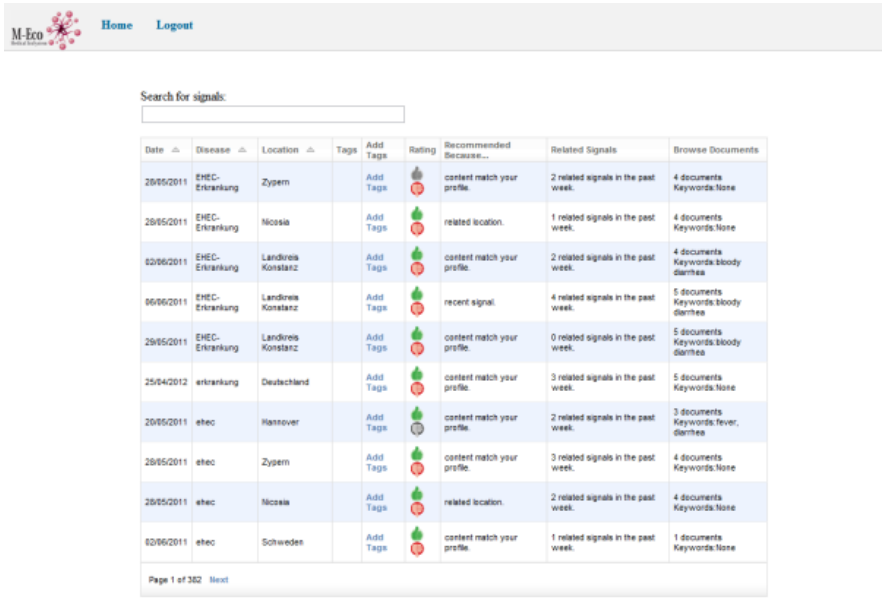


Figure 9: Redesign of the GUI, showing only a summary of signals in the main page

The score s of a signal given its positive (i.e, $+1$) or negative (i.e., -1) rating r and order o in the list of signals of a given user is calculated as:

$$s = r \times (log(n - o + 2))^{-r}, \tag{1}$$

where n is the total number of signals listed for a given user. We compute $n-o+2$ to ensure that $o < n - 1$ and avoid $log0$ and $log1$. Note also that this equation ensures that negative scores get higher weight than positive scores. Since the majority of users tend to only rate what they like, we wanted to emphasize the cases where a negative rating is actually given [14, 38].

In addition to these changes to a new working GUI, we also started to make further improvements in it where we address more of the points raised by the users as discussed in Section 4.1.2. Figure 9 shows a first sketch of these changes in the main page of the GUI. We focus on the first page because that was reported by the users as one that does not provide information in an adequate manner to them.

The main page now shows a more detailed summary of the signals to the user. If the user wants to see the details of a signal, he or she needs to click the corresponding row in the table to access a specific page. This main page also has a search feature, providing the user with the possibility to search for specific signals in his/her own terms. The table listing the signals also contains a number of changes. Figure 10 shows the details.

The first addition is the possibility to sort the list by date, disease or location. Notice the small arrow next to these headers. Once the user clicks one of those, the table is sorted by the elements of the corresponding column. We also added two new columns: 'Recommended Because...' and 'Related Signals'. The first provides an explanation for why the signal is being listed to the user. More details of how the approach works is given in Section 4.1.5. The second shows the number of signals that are related to the current signal in the last 7 days. Related here is considered to be signals that contain the same disease or symptom in the same or similar

Date	Disease	Location	Tags	Add Tags	Rating	Recommended Because...	Related Signals	Browse Documents
28/05/2011	EHEC-Erkrankung	Zypern		Add Tags		content match your profile.	2 related signals in the past week.	4 documents Keywords:None
28/05/2011	EHEC-Erkrankung	Nicosia		Add Tags		related location.	1 related signals in the past week.	4 documents Keywords:None
02/06/2011	EHEC-Erkrankung	Landkreis Konstanz		Add Tags		content match your profile.	2 related signals in the past week.	4 documents Keywords:bloody diarrhea
06/06/2011	EHEC-Erkrankung	Landkreis Konstanz		Add Tags		recent signal.	4 related signals in the past week.	5 documents Keywords:bloody diarrhea
29/05/2011	EHEC-Erkrankung	Landkreis Konstanz		Add Tags		content match your profile.	0 related signals in the past week.	5 documents Keywords:bloody diarrhea
25/04/2012	erkrankung	Deutschland		Add Tags		content match your profile.	3 related signals in the past week.	5 documents Keywords:None
20/05/2011	ehec	Hannover		Add Tags		content match your profile.	2 related signals in the past week.	3 documents Keywords:fever, diarrhea
28/05/2011	ehec	Zypern		Add Tags		content match your profile.	3 related signals in the past week.	4 documents Keywords:None
28/05/2011	ehec	Nicosia		Add Tags		related location.	2 related signals in the past week.	4 documents Keywords:None
02/06/2011	ehec	Schweden		Add Tags		content match your profile.	1 related signals in the past week.	1 documents Keywords:None

Page 1 of 382 Next

Figure 10: Redesigned table to navigate through the signals

locations (see our previous deliverable [11], Section 4.2.3) within the specified period of time.

Finally, in the column 'Browse Documents', we add a list of keywords. These are words extracted from the documents of the signal, that we believe can also represent the signal and give additional context to the user. These keywords are generated using the same method used to automatically generate tags for each document, as described in Section 4.4.6 of our previous deliverable [11]. The only difference is that we consider not only one but all documents of the signal to generate them.

4.1.4 Addressing User Feedback on Tag Clouds

Participants of several user studies related to tag clouds that were presented in the previous deliverables find a tag cloud model as useful information retrieval interface. According to them, it facilitates an exploration, browsing and validation process of a large number of documents. Tag clouds provide a general insight and overview about underlying set of documents. Moreover, depicted terms - tags describe additional context of the documents. In the medical domain, tags can be understood as additional medical conditions as is location, event (Euro 2012), groceries (cucumber) etc. Despite, these benefits, users raised a several suggestions and issues:

- Users should be able to assign new tags and also to remove the pregenerated taggings from the documents and in a such way change the structure of the tag cloud.
- Terms marked as (ir)relevant should be exposed to other partners so they can dynamically adjust retrieval process of documents.

In the following paragraphs, we present solutions to the above-described problems. This presentation is illustrated with screenshots of the tag cloud that was presented in the motivation

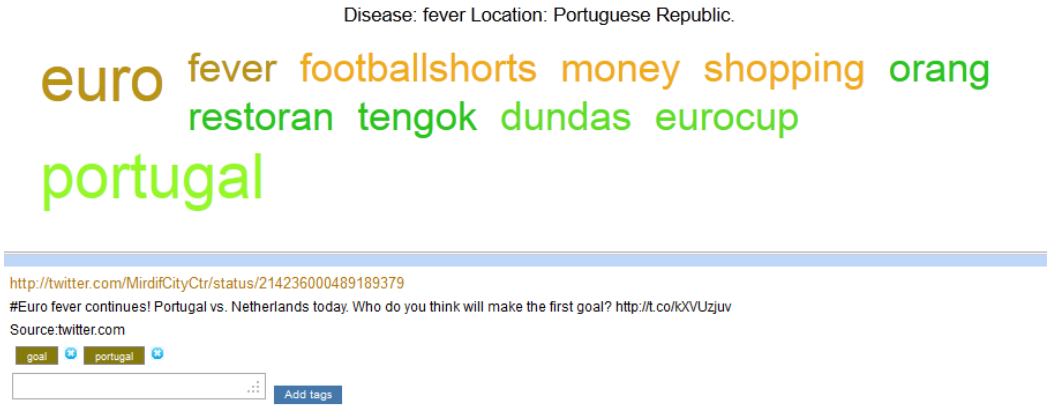


Figure 11: The tag cloud after removal of the irrelevant tag **Euro fever** (see original tag cloud in the Motivation section). The improved model also allows to add user tags to a selected document.

section. The tag cloud contained a document annotated with a term **Euro fever** hence, the document is irrelevant for surveilance experts. The intuitive solution would be to add a new feature into GUI for marking or rating a certain document as irrelevant. Consequently, the document would be propagated back to WP3 and WP4 in order to adjust a document retrieval and signal generation processes. However, a simple "irrelevancy" rating of documents does not provide a reason why the document is not useful for the medical surveilance tasks. Therefore, we proposed an improvement such that users can delete irrelevant terms e.g., **Euro fever** – for the illustration see the tag cloud in the Figure 11 which originally looked as showed in the Figure 3 in the Motivation section. Once, a user removes an irrelevant tag related to a certain document, the tag is removed from tag cloud and it is stored into the database. Partners from WP3 and WP4 can comfortably retrieve a list of irrelevant terms and automatically can extend their list of stop words in order to minimize a number of irrelevant documents in the system (see Figure 12).

Similarly, users can annotate a specific document with additional user tags. It provides benefits when more users view the same tag cloud and users want point out relevant documents to each other. Moreover, users taggings are exposed via webservices for other partners in order to extend their list of domain terms and consequenlty improve a recall of the system.

4.1.5 Explaining Recommendations

In the new GUI we are proposing, the table listing the signals now shows a brief explanation of why each signal is being listed. Section 4.1.3 presents this along other modifications to the GUI.

Explanations are defined based on the results of the multi-factor model discussed in Section 4.3.5. This model aggregates the different methods we use in our recommendation component. It does so by giving weights to the different methods according to the ratings given by each user to the signals he or she receives.

Once these weights are computed, we know which method received the highest, and therefore, was more likely to have influenced the final recommendation score. Currently, depending

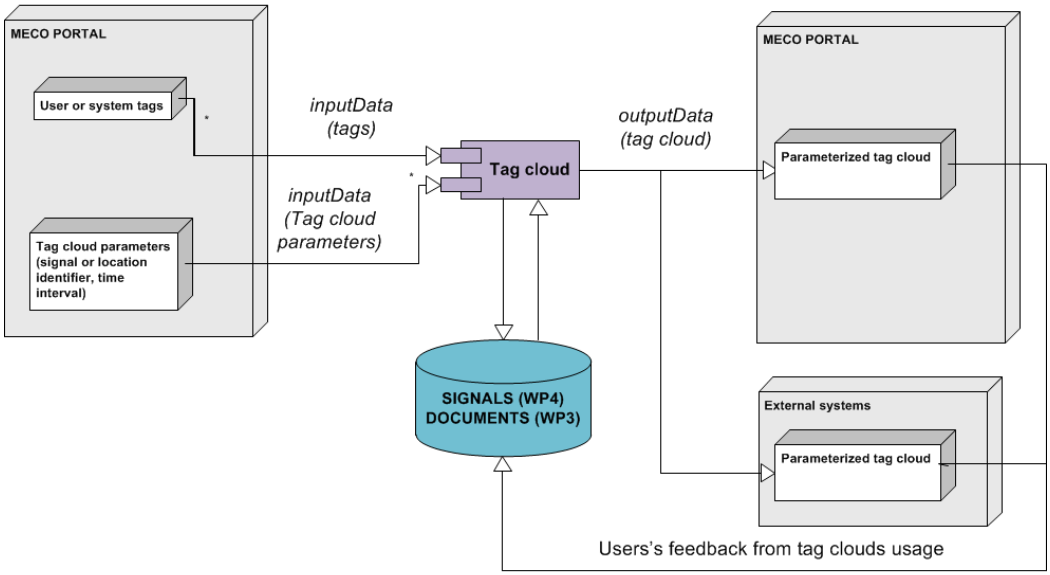


Figure 12: Users's feedback propagation process.

on the user interaction with the system, the following similarity scores are computed between:

1. the user signal definitions and the contents of each signal's documents;
2. the user signal definitions and the properties (i.e., disease and location values) of each signal;
3. the user defined locations together with similar locations and the contents of each signal's documents;
4. the user defined locations together with similar locations and the locations together with similar locations of each signal;
5. the tag set of the user and the contents of each signal's documents;
6. the tag set of the user and the properties of each signal;
7. the date of the signal and the current date when the recommendation is being computed.

Each of these items (we call them 'factors') have a similarity score which is fed to the multi-factor model to compute their respective weights. We then aggregate these weights in four categories: content, location, tag, and date. Items 1 and 2 above refer to the first category, items 3 and 4 to the second, items 5 and 6 to the third, and item 7 to the fourth category. We sum their respective scores and show a message to the user according to the category that received the highest score. The Figure 10 shows an example with three of these messages.

4.2 User and User Group Model Definition

User Modeling refers to the activity of maintaining information about the user's interest, abilities, knowledge and goals [6]. Information systems adjust their content, data, business rules, user interface according to the user's model information [54]. The performance of personalization components depends on how well elaborated the user model is. Information systems that personalize information for individuals sharing common interest need to create group models instead of single user models. In this case, the group needs to suppress the individual needs. Our personalization component develops user models for individual users and groups.

In general, our user models combine multiple indicators of user preference. For example, a system can learn a user's preferences based on the most frequent criteria of his signal definitions but also through his tagging activity. In this case, these two factors can be combined to generate more accurate recommendations to this user. The complete explanation on how this method is implemented and performs on real world data set is explained in *Section 2.2 of M-Eco Deliverable D5.1* [55].

4.2.1 Tagging Model

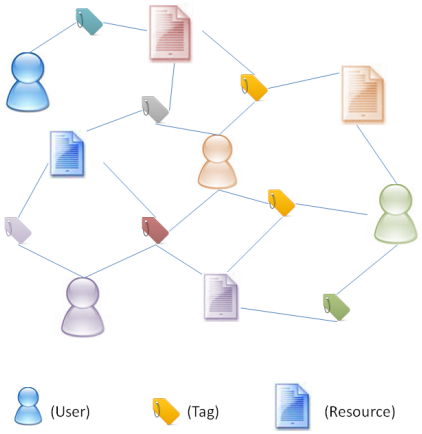


Figure 13: Graph that illustrates the social tagging activity.

Most of personalization in the M-Eco project relies on the user's tagging activity. We understand that tags are potential sources for learning user's interests and therefore can be utilized for selecting appropriate pieces of information respecting individual preferences. Meanwhile tags help solving the problem of users dealing with huge amount of items, tags can help users to navigate through documents of interest through personalized tag clouds. Before depicting the personalization models that address the problems listed above, it is necessary to understand how taggings are modeled and represent the user's interest.

Our general personalization model (based on [23]) realizes tagging systems as hyper graphs where the set of verticals is partitioned into sets: $U = \{u_1, ..., u_k\}, R = \{r_1, ..., r_m\}$, and $T = \{t_1, ..., t_n\}$, where U, R, and T correspond to users, resources, and tags. Figure 13 illustrates the

social tagging activity. In the context of M-Eco, the resources can represent signals, indicator, documents or any other entity which can be assigned with tag.

A tag annotation, i.e. a resource tagged by a user, is an element of set Y, where: $Y \subseteq U \times R \times T$. The final hyper graph formed by a tagging system is defined as G with: $G = \langle V, E \rangle$ with vertices $V = U \cup R \cup T$, and edges $E = \{\{u, r, t\} \mid (u, r, t) \in Y\}$. Particularly to understand the interests of a single user, our models concentrate on the tags and resources that are associated with this particular user, i.e. in a *personal* part of the hyper graph G. We then define the set of interests of a user as $P = (T_u, R_u, Y_u)$, where Y_u is the set of tag annotations of the user: $Y_u = \{(t, r) \mid (u, t, r) \in Y\}$, T_u is the tag set of the user: $T_u = \{t \mid (t, r) \in Y_u\}$, R_u is the set of resources: $R_u = \{r \mid (t, r) \in Y_u\}$.

4.2.2 Location Preference Model

In order to recommend users relevant information it is necessary to assess whether a document matches the user's interest. However, a number of related (and relevant) information still can be left behind. For instance, many documents matching a signal definition with the specified location *Munich* can be quickly retrieved while relevant information reported by neighbor locations (e.g., Augsburg, Rosenheim and Salzburg) are not considered. In order to improve the recall with related documents, we investigate spatial reasoning techniques as a means to help users discover related outbreaks in other locations, related to user's stated references.

In order to suggest items of related locations, we first model the user's location preferences. These preferences are initially modeled from the locations specified in the signal definition SD_u of the user u. We define the set L_u, the initial set of locations the user has an explicit preference as:

$$L_u = \{l | l \in SD_u\}, \tag{2}$$

where l is any location the user can choose from in the M-Eco system. Next, we introduce the concept of location similarity in order to find related locations. This similarity is computed as a function of two factors:

- Political hierarchy (e.g., city, state or country level); and,
- Distance and population.

Once the similarity score between two locations is computed, we expand the set L_u with the most similar locations. For each location $l \in L_u$, we expand the set L_u with the locations with similarity score higher than 0.9, where the maximum score is 1 when the location being compared is the same as l. In the end, the final set L_u contains the user locations together with their most similar ones. This information is then used to compute recommendations of items from related locations to the user. Section 4.4.3 details how the location similarity is computed and recommendations are produced.

4.3 User Classification and Modeling Algorithms

In this section we present a number of techniques for modeling user generated data such as tagging in order to generate proper recommendations. The recommendations are intended to reduce information overload of medical experts and facilitate their tasks while evaluating medical documents. In the following subsection, besides introducing the modeling algorithms, we highlight technical problem addressed and how they were solved.

4.3.1 Semantic Enhanced Tag-Based Recommendation

In order to lessening the problem of users when dealing with huge amount of information available, this approach aims at supporting users by selecting and retrieving relevant information that are particularly annotated with tags. For that, this method generates personalized recommendations from the analysis of the user's tagging activity. In order to recommend users with personalized information, the approach deals with common problems in social tagging systems: sparsity (when no or few tagging are available) and ambiguity (when a single tag refers to different concepts). The recommendations model tries to capture the semantic nature of social tagging data and then incorporates the semantic-enriched expression along with the traditional lexicon-based vector for improved recommendations. In the end, we combine a conventional tag-based recommender system with latent semantic analysis to generate personalized recommendations. The complete details of this model are demonstrated in *Section 2.4 of M-Eco Deliverable D5.1* [55]. The following subsection provides another method to support users with the selection and retrieval of pertinent information.

4.3.2 Expanding Tags with Neighbors For Improving Search Retrieval

This approach also addresses the problem of users dealing with huge amount of items. This method supports users with selection and retrieval of relevant documents so that they are not distracted with unsolicited content. From a different perspective, this method aims at assisting proactive users searching for items of interest instead of receiving recommendations in periods of time. Although search and recommendations are distinct mechanisms, the model for relevance assessment and retrieval is relatively common for both approaches. While recommendations look at the contextual information for suggesting information, the search engines depend on a query issue by a user. The technical problem addressed in this approach concerns under-specified queries that lead users to irrelevant search results. In order to tackle such an inconvenience, we proposed a method that utilize tags to augment the chances of relevant retrieval. In brief, when a user issues a query, it is extended with pre-computed related terms (called tag neighbors) aim at improving the retrieval of relevant items.

In order to test our approach, we crawled the articles from *MedWorm* repository system. The focus of our analysis was based on the observation of precision of our search engine. We compared our precision results with results from a baseline query search that rely on the simple user entry query (without expansion). The overall result was satisfactory, our results verify that our approach outperforms the baseline query search in terms of mean average precision (MAP). As a limitation of the work, we realized that the tag extension does not perform equally for all medical categories. As a future work, we aim at improving the quality of tag neighbors by comparing them against medical specialized dictionaries or domain ontology vocabularies. Further, we plan to realize more experimental studies necessary to validate the scalability and feasibility of the proposed approach in a broader scope. Finally, we aim at combining the current approach with other techniques previously explored such as collaborative filtering. The complete demonstration and evaluation on this method can be found in *Section 2.3 of M-Eco Deliverable D5.1* [55]. The following subsection provides an alternative method that also utilizes tag neighbors for improving recommendations.

4.3.3 Spectral Clustering of Tag Neighbors for Recommendation

In line with previous method (Section 4.3.2), the primary goal of this approach is to support users with relevant recommendations. In order to improve recommendations with quality and quantity of tag neighbors, we investigate the spectral clustering algorithm to filter out noisy tag neighbors. The tag clustering is used to find the tag aggregates with similar functions or topics. The spectral clustering is based on the graph partition which maps the original co-occurrence observations onto a new spectrum space with a reduced dimensionality. The obtained partition guarantees the disjoint clusters with minimum cut optimization. The basic idea of clustering filtering is: the neighboring tags from the same tag cluster of the target tag contribute collaboratively to specific function or topic, being kept as the appropriate tag neighbors for tag expression expansion; otherwise be discarded. So the next processing step is to filter out the noisy tags according to the discovered tag clusters. Worth mentioning that each tag has an expanded tag neighborhood, which might belong to different clusters. To ensure all neighboring tags are from the same tag cluster, each tag in the expanded neighborhood will be compared with all the tags from the tag cluster where the target tag is assigned. If the expanded neighbor appears in the same cluster, it then can be considered as the appropriate neighbor of the tag tag, making it kept in the expanded tag set; otherwise, it should be filtered out. After such steps above, the left elements could be defined as the tag neighbors for the target tag, and the quality of the tag neighborhood will be accordingly improved. Also in such way the density in the integrated tag-user-document matrix could be increased substantially.

After the tag neighbor expansion is completed, we get updated user profiles and document profiles in the forms of tag vector expression with expanded tag neighbors. We then utilize the similarity measure between users and documents to make tag-based recommendations. A complete description of this work was presented in the previous deliverable [11] Section 4.3.3 and Section 5.2 or it can be seen in our research article [42].

4.3.4 Predicting Event Trajectories

Introduction to Event Trajectories In this section, we consider the task of reconstructing trajectories of an event from a social networking service. That is, assuming an identified event in a set of messages, we attempt to identify where the discussion is moving geographically. By reconstructing such trajectories it could be possible to anticipate the impact of an event on a certain location. Early warning systems, such as M-Eco, could then benefit from the estimated trajectory of a particular epidemic outbreak in order to take proper action in affected areas or potential areas still to be affected.

In this task, we use Twitter to conduct our experiment. We attempt to:

- Extract spatial information from Twitter relative to the context of one particular event;
- Use the extracted information to reconstruct the trajectories of the event around the world.

We also discuss the possibility of predicting future locations based on the identified trajectories. It complements this discussion with potential context-sensitive information (e.g., location importance or social graph of users) that could be used to augment the prediction. The overall task of reconstructing trajectories is not yet integrated to the WP5 Component. It is presented here as a result of conducted research that could be used in later developments of the M-Eco project.

The reconstruction of trajectories is a two-step process. First, spatial information is extracted from Twitter messages or user profiles. This step involves crawling Twitter data, extracting the messages related to a specific event and their specific spatial information when available, and matching the location strings from the user profile. Once this is done, in the next step we estimate the main locations in time where the event took place. With the coordinates of locations defined, we compute the trajectories of the event. The next sections describe each of these steps.

Extracting spatial information The first step in the process of reconstructing the trajectories of an event is to extract the messages from Twitter related to a particular event. Assuming a dataset of crawled Twitter data, we obtain these messages by searching the hashtags associated with an event. A hashtag (or simply tag) is label used by the user to characterize a particular message in a certain way. On Twitter, a hashtag is composed by the character # followed by a single word.

On Twitter, events are typically associated with a hashtag because it turns the message easily searchable. Twitter has a feature that allows you to see all messages matching a specific hashtag up to 15 days old. Examples of hashtags matching an event include #occupy or #OWS, referring to the Ocuppy Wall Street and related movements, #H1N1, referring to the 2009 flu pandemic, and #Lybia, referring to the protests that took place in that country.

By searching the crawled Twitter dataset for a set of hashtags H related to a specific event, we obtain the set of messages M associated with that event. For each message $m \subset M$ we:

1. Extract the text, timestamp and user profile;
2. Verify whether the message is geolocated and extract the corresponding latitude and longitude if present;
3. Extract the location from the user profile and match it to the GeoNames database[7].

Most messages from Twitter are not geolocated. Only 1% of all messages on Twitter contain geolocation information. For this reason, in most cases we associate the location informed on the user profile as the location of the message. However, this location is informed by the user as free text, hence a matching with geolocation data is needed.

We perform the matching of the locations extracted from the user profiles with the locations from the GeoNames database. This database covers more than 8 million geolocations on all countries. It also contains name variations for the same location, increasing the possibility of a match. Still, since the locations from Twitter are free text, there is no guarantee that they match any location. For this reason, we developed an algorithm to perform a fast matching beyond the exact text comparison.

First, we load all GeoNames records into a trie structure [46]. A trie is a tree structure where each node holds as key the prefix of all the descendants. That is, the first nodes will contain, each, the first letters of all subsequent strings, and so on. Then, in a naive approach, a string to be matched navigates through the tree, letter by letter, until the full string is found or is not present. According to [46], the Patricia trie, the implementation we use, has a $O(nlogn)$ pre-processing time that has to be done only once, and $O(m)$ time for the worst case exact match.

[7] http://www.geonames.org/

Once all GeoNames records are loaded, for each location extracted from Twitter we initially perform an exact match on the trie. If no match is found, we use the Hamming distance in order to find an approximate match bitwise. This always ensure that a match will be returned but does not guarantee that it corresponds to the location extracted from Twitter. This approach helps retrieve locations efficiently at a city level or, at most, neighborhood level, which represents the majority of the locations informed by the users.

Estimating locations in time and reconstructing the trajectories Once the messages extracted from Twitter are associated with geolocation information, we build on [33] to reconstruct the possible trajectories of the event over time. We assume the trajectories to be across cities. The objective is to understand how an event expands over time to different regions instead of reconstructing the many micro trajectories that are likely to exist within different cities. We also consider first the cities with more activity on Twitter. We assume that the more messages on Twitter a certain location has, the more likely that the event has a presence there. Finally, we take into account that an event is likely to spread in different directions and that it can emerge independently in different locations. An event can then have different trajectories.

We define a trajectory as a sequence, L, of triples ordered by the timestamp t:

$L =< lat_0, lon_0, t_0 >, ..., < lat_n, lon_n, t_n >$

where lat_i and $lon_i (i = 0...n)$ are respectively the latitude and longitude coordinates of a location in $\mathbb{R}^2$ space. With this in mind, given a trajectory L, we decide to add a new location l to it according to the following parameters:

- *Temporal gap between the location and the last point of the trajectories*: We consider the minimum and the maximum interval between two consecutive timestamped locations. If the location's timestamp is below the minimum gap, we assume that it is related to the current state of the event rather than a new activity related to it. If the timestamp is above the maximum gap, we assume that the location is independent of existing trajectories and, therefore, use it as a starting point for a new trajectory.

- *Spatial gap between the location and the last point of the trajectories*: We consider the minimum and the maximum distance between two consecutive timestamped locations. Distance is computed using plane approximation according to the formula:

$$d = \sqrt{R_{eq}^2 \times (lat_1 - lat_2)^2 + R_p^2 \times (lon_1 - lon_2)^2 \times cos((lat_1 + lat_2)/2)^2} \quad (3)$$

  where R_{eq} is the equatorial radius, R_p is the polar radius, lat_1 and lat_2 are the latitudes for locations 1 and 2, and lon_1 and lon_2 are the longitudes for locations 1 and 2.

  Similarly to the temporal gap, locations below the minimum distance are assumed to be part of the existing trajectory and, therefore, are ignored. This minimum gap works in the same manner as the parameter *tolerance distance* from [33]. Locations above the maximum distance are assumed to be unrelated to existing trajectories and become the starting point of new trajectories.

We do not consider the maximum speed explicitly because we assume it to be a function of distance and time. However, for each new location l in L we consider the parameters above for each location in each trajectory. We do this to account for the possibility of multiple trajectories. That is, a new location l might be a fork from an existing point in a existing trajectory. For

example, if a new location is closer to a previous point from a trajectory, other than the last one, we use these two points to create a new trajectory, a fork from the existing one.

Predicting future trajectories [35] proposes a four-step process in order to predict future locations of a moving entity: Data selection, local models extraction, T-Pattern Tree Building, and prediction. Data selecion is a process to determine a spatial area and a time period of relevance, in order to detect the parts of a trajectory crossing it in a particular time period. In the next step, models are built using Trajectory Patterns (or T-patterns). T-patterns are defined as a set of regions linked by the time to reach each of them in a trajectory.

Next, the authors present a method to build a prefix tree called T-pattern Tree defined as $PT = (N, E, Root(PT))$, where N is a finite set of nodes containing information related to a region in a T-pattern, E is a set of labeled edges containing the time interval between a parent and a child node, and $Root(PT)$ represents the root node of the tree. The T-pattern Tree is used for efficient predictions of future locations. Instead of listing and comparing all possible combinations of T-patterns independently, the tree helps navigate the patterns to find the best one that matches a given trajectory.

Given a T-pattern Tree, PT, the prediction algorithm computes the path score for each path of PT relative to a trajectory T. The algorithm visits all nodes in the tree and computes all possible paths. Once this is completed, the algorithm returns the best score and the candidate regions as the prediction of next locations in the trajectory.

4.3.5 Improving Multi-Factor Based Models

Introduction We introduced the multi-factor model in section 2.1 of our first deliverable [55]. In this deliverable we introduce improvements to the original model. We add a time decay factor, the date score, to account for the recency of the signal. We also improved the performance by caching the data used for the regression model.

The date score is simply computed as the date difference between the current date and the date the signal was generated. That is,

$dateScore = MAXDAYS - (currDate - signalDate)/MAXDAYS$,

where $MAXDAYS$ is a constant determining the maximum amount of days from the current date allowed for a signal to be considered for recommendation. We set dynamically the maximum number of days as the difference between the current date and the date of the oldest signal rated by the user in his or her last interaction with the system. For example, if the user interacted with the system yesterday, and the oldest signal rated was one week old, then, $MAXDAYS = 8$ for the recommendations being processed today.

We use this date score in the model together with other factors. These include the location of the signal, the tagging activity of the user, and the contents of the documents of the signal. Because the signal is a fixed entity, that is, their documents and features remain unchanged once they are created, we cache the contents of documents to increase the performance of the computation. We use a Java component called JDBM [8] aimed at storing Java objects directly to the disk storage. We use this to cache to the disk the result of building the vector-space model of the documents of each signal to be recommended.

Comparing with a Collaborative Filtering method. We plan to compare ours with a collaborative filtering method. Sawar et al. describe such a item-based collaborative filtering

[8] https://github.com/jankotek/JDBM3

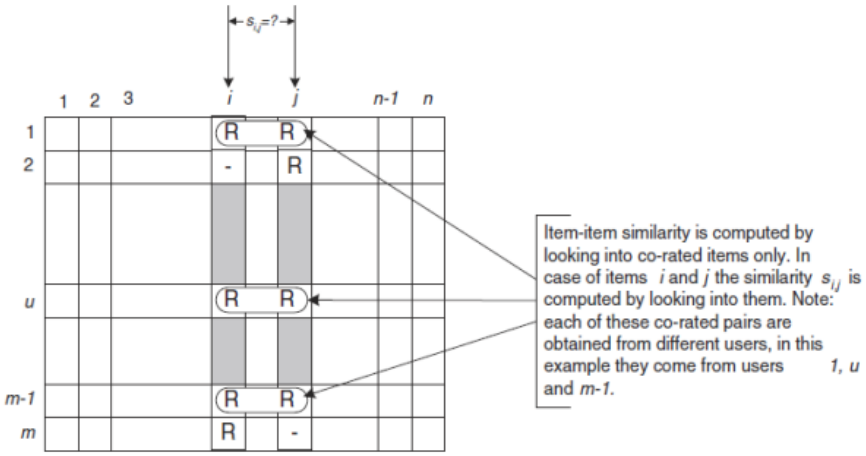


Figure 14: Isolation of co-rated items and similarity computation [45]

recommendation in their work [45]. Given an active user u_a, the authors describe the task of finding an item likeliness in two forms:

- **Prediction** is a given numerical value for the active user. We modify the range of this prediction according to the range of our ratings as described in Section 4.1.3.
- **Recommendation** is a list of items that the active user will like the most, according to the predicted score of each item.

The process of item-based collaborative filtering recommendation occurs as follows. First, we filter out signals that do not contain at least one of the diseases or symptoms specified by the user. Next, we compute the pairwise similarity between each two remaining signals. This is done by building a vector of ratings given by different users to these signals. Only pairs of co-rated signals are considered, that is, we only include in the vector the ratings of users that rated both signals. Figure 14 from [45] (page 289) illustrates this.

Last, once the similarity between signals are computed, we determine the prediction value $P_{u,i}$ for each item i that the user u the has not rated. This value is computed as follows:

$$P_{u,i} = \frac{\sum_{j \in S}(sim_{i,j} \times r_{u,j})}{|S|} \tag{4}$$

Where j is a signal in the set S of signals rated by the user, $sim_{i,j}$ is the previously computed similarity between signal i and j, and $r_{u,j}$ is the rating given by user u to signal j.

Once the prediction is computed for all signals in S, we recommend the top N signals to the user.

4.4 Recommendation, Adaptation and Personalization Strategies

This section introduces a number of personalization models focused on providing users with information that match their interest. The personalization models utilize the user (and group) models (see Section 4.2) for adapting information to heterogeneous individuals. This adaptation

is represented by a recommendation, i.e. a list of items ordered by relevancy or tag clouds, a cluster of terms that helps users to navigate through a collection of documents.

4.4.1 Recommendations based on Signal Definitions

As shown in Section 2, users create signal definitions to perform surveillance of specific diseases in determined locations. These inputs can be understood as fundamental source for personalization once they expose the user's interest. In addition to the personalization model that relies on the user's tagging as shown in Section 4.3.1, we also proposed a method that utilizes signal definition information to provide personalized recommendations. The limitation of this method is that the recommendations are constrained solely by the parameters (e.g., location, medical conditions) and related information that may be still important are not retrieved. The complete details of this model can be seen in *Section 2.2 of M-Eco Deliverable D5.1* [55].

4.4.2 Tag-Based Recommendations

Tags are potential indicators of user preference. For instance, a medical expert that has exhaustively assigned the tag "*swine flu*" to the documents he evaluates, seems to be interested in that disease. Therefore, this knowledge can be utilized to filter out irrelevant recommendations unrelated to "*swine flu*". For recommending items to the user, we compare his tags, i.e. tags assigned by him to his documents of interest against the tags assigned to candidate and unknown documents. The comparison is realized by the cosine similarity of two tag vectors, one corresponding the user's tag vector and the other corresponding to the document's tag vector. The documents with highest similarity to the user's tag profile is then selected to be recommended.

Worth mentioning that although the tag-based recommendation component appears as a single component in Figure 4, it is not utilized as a stand alone application. Instead, this component is adapted to some extent and reused by other components such as the *Semantic Enhanced Tag-Based* component, *Tag Cloud* component, *Tag Neighbors* component and *Spectral Clustering of Tag Neighbors* component.

4.4.3 Recommendations based on Event Trajectories

In this deliverable, we present in Section 4.3.4 a method to predict the trajectory of events based on their location. In this Section we summarize the method of recommendation based on location presented in Section 4.4.3 of our previous deliverable [11]. We present it here because we still use this method to generate the recommendations. In Section 4.2.3 of the previous deliverable we presented the model for user location preferences. It was built from the set of user locations defined in his or her signal definitions and from similar locations. Location similarity is computed as a function of two factors:

- Political hierarchy (e.g., city, state or country level); and,
- Distance and population.

Political hierarchy is used to compute the political distance. This distance is a function that closely associates how "connected" two cities are based on their hierarchy. The motivation for this distance is the fact that two locations within the same political hierarchy are more likely to be of interest to a health official which is restricted to actions within his or her administrative

region (e.g., a state or country). The political distance, then, increases the importance of cities within Lower Saxony instead of cities in other regions or countries even though the actual geographic distances of the latter might be lower.

In addition to the political distance, the actual distance and the population of the locations are used to compute the proximity distance. This distance is initially computed as the geodesic distance, i.e. the shortest distance between two points on earth. It is then weighted according to two factors: population size and airport connections.

To compute the final location similarity score (LSS), we determine empirically the relationship between the political, and proximity distance. We computed their values pairwise for a set of 160 locations related to the EHEC outbreak in Germany. LSS can be pre-computed for all pairs of locations available on M-Eco. Once this process is completed, the score is normalized to range from 0 to 1. With Location Similarity Scores pre-computed, we use them in cosine similarity measures to find out items that are of related locations.

4.4.4 Personalized Tag Cloud Adaptation and Navigation

The complete description of tag cloud generation process is presented in the M-Eco Deliverable 5.2 (D5.2) [11], *section 4.4.4*. In the following section we present only improvements of the tag cloud model. We describe the algorithms that optimizes the structure of tag clouds in terms of coverage and overlap. This contribution was accepted at the 12th International Conference on Web Engineering ICWE 2012 in Berlin [34].

Methodologies for Improved Tag Cloud Generation with Clustering

Clustering techniques In the following paragraphs, we present several clustering techniques that are utilized in the process of tag cloud generation. These techniques can be utilized also for tensor based recommendations (see Section 4.4.5) Firstly, we introduce syntactical pre-clustering based on Levenhstein distance. Then, we present three different clustering techniques. The first two proposed approaches(Correlated Feature Hashing and Complete linkage hierarchical clustering) cluster tags according to their co-occurrence based similarities. The third (K-means) algorithm considers each tag from a tag space as feature vector. These techniques were proposed and described in [32].

Syntactical Pre-clustering Syntactical pre-clustering filters out items with typographical misspellings unnecessary plural and singular forms of the same item and also compounded items from two different terms connected with some separator. These redundant items would occupy an item set unnecessarily as they would have the same semantical meaning. Levenhstein distance is first computed for each term pair from the initial term space. The distance between two terms measures the number of required changes (substitution, insertion and deletion of a character are allowed operations) to transform one term into another. We justify its use because it attains significantly better results than Hamming distance as shown in [13].

Correlated Feature Hashing We propose to reduce a tag space with hashing function that is similar to the proposed technique in [4] where authors successfully reduced dictionary size by utilizing hashing. The idea is to share and group tags with similar meaning.

We sort the tags used within the system according to the frequency of usage such that t_1 is the most frequent tag and t_T is the least frequent. For each tag $t_i \in 1, \ldots, T$ is calculated

DICE coefficient with respect to each tag $t_j \in 1, \ldots, K$ among the top K most frequent tags. The *DICE* coefficient is defined as:

$$\text{DICE}(t_i, t_j) = \frac{2.\text{cocr}(t_i, t_j)}{\text{ocr}(t_i) + \text{ocr}(t_j)} \tag{5}$$

where $\text{cocr}(t_i, t_j)$ denotes the number of co-occurrences for tags t_i and t_j, $\text{ocr}(t_i)$ and $\text{ocr}(t_j)$ is the total number of tag t_i and t_j assignments respectively. For each tag t_i, we sort the K scores in descending order such that $S_p(t_i) \in 1, \ldots, K$ represents the tag of the p-th largest DICE score $\text{DICE}(t_i, S_p(t_i))$. We can then use hash kernel approximation defined as:

$$\bar{\Phi}_{t_j}(x) = \sum_{t_i \in T : h(t_i) = t_j} \Phi_{t_i}(x) \tag{6}$$

and given by a hash function:

$$h(t_i) = S_1(t_i) \tag{7}$$

The described approach is replacing each tag t_i with the tag $S_1(t_i)$. We have reduced tag space from all T tags to the K most frequent tags.

Complete linkage hierarchical clustering In the second approach we utilize Complete linkage agglomerative hierarchical clustering technique [25]. In the beginning, each entry that should be clustered is considered as single cluster. For each cluster is computed Dice similarity (see Formula 5) with all other clusters. The cluster with the highest similarity to the considered cluster is merged with the cluster. When clusters contain more tags, the lowest similarity between two tags from those clusters is considered for the merging step. The aggregation of clusters repeats until the single cluster is obtained. The final clustering structure is denoted also as dendrogram. The required number of clusters is obtained by cutting a dendrogram at a certain level such that a given number of clusters is obtained.

K-means The following clustering technique differs from the previous in such a way that each tag is expressed in n-dimensional vector space where the i-th dimension corresponds to the i-th item res_i (in a similar way as in [24, 39]).

We denote $T = \{t_1, t_2 \ldots, t_{|T|}\}$ as the set of all distinct tags that are clustered and $R = \{res_1, res_2 \ldots res_n\}$ the set of all items that are tagged with tags from T. Let $f(t, res_i)$ be equal to a frequency of a tag t assigned to item res_i otherwise it is equal to 0. Then, the vector representation of tag t is:

$$t = (f(t, res_1), f(t, res_2), \ldots, f(t, res_n)) \tag{8}$$

Once, tags from T are expressed as n-dimensional vectors, we proceed with the cluster analysis. The K-means is a simple well known clustering technique that groups objects from a given set into k clusters (given a priori). The clustering of a tag space with the K-means algorithm is computed as follows:

1. Each tag from a tag space T is expressed as n-dimensional vector. According to the size of the tag space and user requirements an amount of clusters is set to k.

2. It randomly places k centroids such that a distance from each other is maximized.

3. Each tag from the tag space is bound to the nearest centroid.

4. New centroids are computed as the mean value of tags vectors grouped with a given centroid. It continues with the step 3, until new centroids are identical with the centroids from the previous iteration.

We obtained k disjoint clusters of tags so we can proceed with the selection of tags for the tag cloud generation. The results of K-means algorithm depend on used distance measure - we exploit only Cosine distance as it attains the best results [32].

Tag cloud metrics In this paragraph, we present common metrics that measure different aspects of a tag cloud. Next, we introduce our two methodologies for tag cloud generation that attempts to improve the tag clouds according to the presented metrics.

The quality of tag clouds is usually assessed by the users that subjectively rate the structure and arrangement of tags in the cloud. However, such users based assessments are expensive and hardly available. To overcome this limitation, we use synthetic metrics for evaluation of different aspects of a generated tag cloud. Such metrics allow to measure the quality of tag clouds and, as a consequence, various tag selection algorithms can be utilized to maximize considered metrics. In this work, we consider 2 well-known metrics, *coverage* and *overlap*, introduced in [52]. Furthermore, we introduce a new metric *chained coverage* which is utilized in the proposed methodologies. For the following definitions consider D as a set of exiting documents, T as the whole set of existing tags and D_t as the set of documents assigned to a tag $t \in T$.

The first metric is *coverage*, defined as:

$$\text{Coverage}(t) = \frac{|D_t|}{|D_a|}, \tag{9}$$

where $|D_t|$ is the number of documents assigned to a tag t and $|D_a|$ is the number of all documents that are considered during a tag cloud generation process. The metric ranges between 0 and 1. When a coverage for a particular tag t is close to 1, the majority of considered documents was annotated with a tag t. We utilize this metric during the selection process to maximize number of documents that can be accessed directly by exploring a tag cloud.

Overlap of T_c: Different tags in T_c may be assigned with the same item in D_{T_c}. The overlap metric captures the extent of such redundancy. Thus, given $t_i \in T_c$ and $t_j \in T_c$, we define the overlap $over(T_c)$ of T_c as:

$$\text{Overlap}(T_c) = avg_{t_i \neq t_j} \frac{|D_{t_i} \cap D_{t_j}|}{\min\{|D_{t_i}|, |D_{t_j}|\}}, \tag{10}$$

If $over(T_c)$ is close to 0, then the intersections of documents annotated by depicted tags are small and such tag clouds are more diverse.

There exist different selection techniques that try to optimize a given metrics which result into enriched tag clouds. In this work, we propose two new methodologies that improve introduced metrics. Furthermore, we introduce a new metric *chained coverage* that captures how many documents are covered by a considered tag given that documents covered by previously selected tags are not considered. This metric combines coverage and overlap altogether and

provides simpler decision-making during the tag selection for the tag cloud. *Chained coverage* is given as:

$$\text{Chained coverage}(t|T_s) = \frac{|D_t \setminus D_{T_s}|}{|D_a|}, \tag{11}$$

where D_{T_s} is a set of documents covered by previously selected tags T_s. The proposed metric can be understood as combination of the classical coverage with the zero overlap with the respect to the previously selected tags. We assume that the diversity of the tag cloud is desired property as users are not interested in retrieving redundant documents covered by different tags. Therefore, the goal is to maximize a chained coverage of each tag used for the tag cloud generation. The metric simplifies a selection process of tags as instead of optimizing two independent metrics i.e., coverage and overlap we maximize only the chained coverage.

Syntactical Pre-clustering of Tags Social tagging systems collect heterogeneous tags assigned by the users to the resources of the system. Tags in these systems can have the same semantical meaning however they are syntactically different i.e., typos, singular and plural forms and compounded tags.

Syntactical pre-clustering filters out tags with typographical misspellings (E.colli, E.coly) unnecessary plural and singular forms of the same tag (cucumber, cucumbers) and also compounded tag from two different terms connected with some separator (E.coli, E-coli, E coli). These useless tags would occupy a tag cloud's space and in consequence it would result in the semantically redundant tags in the cloud. Tags like *E.colli*, *E.coly*, *E-coli* or *E coli* can be aggregated and represented only with the most frequent tag *E.coli*.

To remove from the tag cloud syntactically different tags with the same semantical meaning, we propose a methodology that aggregates syntactically similar tags into clusters. In the tag cloud generation process, obtained clusters can be represented only with the most frequent tag which can have the following benefits:

- The coverage of the depicted tag in the tag cloud improves as it covers all documents annotated with the syntactically different tags from the given cluster
- Generated tag cloud does not contain syntactical variations of the same term as only the most frequent tag from each cluster is considered. Therefore, it allows to create a more diverse tag cloud, i.e., lower overlap between depicted tags.

In our method, syntactical pre-clustering introduced in Section 4.4.4 is used in the following manner. Levenhstein distance is first computed for each tag pair from the initial tag space. The edit distance between two tags measures the number of required changes (substitution, insertion and deletion of a character are allowed operations) to transform one tag into another. We justify its use because it attains significantly better results than Hamming distance as shown in [13].

Once, an edit distance is calculated, the tag space is divided into clusters. Each group contains only tags where the Levenhstein distance is equal or lower than a defined threshold (a number of maximum changes to transform a tag from the tag pair into a second tag). Then, the most frequent tag for each cluster is selected and is used in all further computations. It represents all other tags from a considered cluster.

In the end, our goal is that syntactical pre-clustering will affect the structure of the generated tag cloud in the sense that depicted tags are semantically more diverse.

Improving coverage and diversity of tag clouds with clustering The second methodology aggregates semantically related tags into a disjoint group. Each cluster can be perceived as a latent topic described with the related tags. The goal of cluster analysis is to cover all available topics in the tag space and as a consequence map it into a generated tag cloud to achieve maximal diversity of depicted tags. The methodology is motivated due to the drawback of the usual approach (denoted also as a baseline approach) where only the most frequent tags are considered. The selection of the most popular tags results into a tag cloud with terms that have too broad meaning. Therefore, depicted tags cover redundantly a certain set of documents i.e, the overlap of such tag cloud is unnecessary high. For instance a tag cloud generated from the top-25 most frequent tags from Bibsonomy dataset [27] contains tags as *public*, *video*, *Media*, *books*, *blog* or *search*. Obviously, such tags have general meaning or no information value for users. Moreover, often are assigned to the documents in combination with other frequent tags. The possible solution is to minimize a number of tags with the general meaning and additionally select popular but more specific tags as the objective is to preserve the coverage and minimize overlap of the tag cloud.

The aforementioned drawbacks of tag clouds generated from the most popular tags are addressed with the combination of cluster analysis of tags and maximization of the introduced metric – chained coverage. The former one provides basis for a diversity of a generated tag cloud by assuring that all latent topics within the tag space are captured. The latter one suppresses tags with the general meaning and instead selects popular but specific tags. The maximization of the chained coverage promotes (specific) tags with the high coverage of not yet covered documents by previously selected tags. On the other hand frequent tags with low chained coverage (general meaning) are omitted.

We explore different approaches of tags selection from the created clusters. The method based on selecting one tag with the highest coverage from each cluster generates more diverse tag clouds. However, the coverage is lower or comparable to the baseline approach as the chained coverage of generated clusters follows a power law distribution. Thus, majority of clusters belong to the long tail of such distribution.

Therefore, we propose a technique (see Algorithm 1) that selects tags proportionaly from each cluster. The provided tags are syntactically grouped and subsequently semantically clustered by one of the introduced clustering technique. The number of clusters is equal to the tag cloud size. The obtained clusters are sorted by chained coverage in descending order. The chained coverage of each cluster is given by previously explored clusters starting from cluster with the highest coverage. The method computes the number of tags to be selected from the cluster based on the chained coverage of a given cluster given the tag cloud size. From each cluster is selected a number of tags with the highest chained coverage. The goal is to cover a given cluster as good as possible in terms of coverage and overlap. The selection based on maximization of chained coverage satisfies such requirements. The method terminates when the number of selected tags is equal to the tag cloud size.

Tag Cloud Generation Once, the tags are selected according to our proposed methodologies, they are depicted in the tag cloud. Semantically related tags from the same cluster are displayed with the same color which is specific for each cluster. Such tags are also located near each other and it allows to explore tags in more convenient way. Location and particular color of tags from the identical cluster results into a tag cloud which is semantically structured and as was shown in [19]. This presentation structure differs from the most common visualization of tag clouds where tags are alphabetically sorted. It allows to differentiate main topics in

Algorithm 1: The methodology for tag cloud generation.

```
   Input: tags, tagCloudSize
   Output: selectedTags
 1 tags ← syntacticalClustering(tags);
 2 clusters ← semanticalClustering(tags,tagCloudSize);
 3 clusters ← sortClustersByChainedCoverage(clusters);
 4 foreach cluster in clusters do
 5 |   tagsToSelect ← cluster.chainedCoverage(exploredClusters) · tagCloudSize
 6 |   for i=1 to tagsToSelect do
 7 |   foreach tag in cluster do
 8 |   |   if tag.chainedCoverage(selectedTags) is highest in the cluster then
 9 |   |   |   if tag.chainedCoverage(selectedTags) > threshold then
10 |   |   |   |   selectedTags ← selectedTags + tag;
11 |   |   |   end
12 |   |   end
13 |   end
14 |   if size of selectedTags > tagCloudSize then
15 |   |   return [selectedTags]
16 |   end
17 |   exploredClusters ← exploredClusters + cluster;
18 end
```

the tag cloud and also users can perceive and notice semantic relations between tags in neighbourhood [19]. Moreover, it helps to understand connections between tags, for example, tags cucumber and Spain are hardly interpretable in alphabetically sorted tag cloud. However, if they are depicted together with the tag E.coli a user can easily assume that these tags are related to E.coli outbreak.

Personalized Tag Cloud Generation: Another benefit of performed cluster analysis is a possibility to generate personalized tag clouds. Such tag cloud is an adapted version of the above-mentioned general tag cloud model. User's preferences are incorporated into the tag cloud such that tags related to user's tags are preferred over others. The selection of similar tags is performed by retrieving tags from the clusters that contain at least one of the user's tags.

4.4.5 Tensor Based Recommendations

In the following section, we present the state of the art tag-based recommender based on tensor factorizations [40, 50]. Tensor based recommenders build 3-dimensional matrix (tensor) by reflecting relationships between all users, items and tags from STS. Afterwards, a factorization technique is performed on the constructed tensor. The tensor approximation usually reveals latent relations between the involved objects. They outperform other tag-aware state-of-the-art recommendation algorithms as was shown in [40,50]. They generate recommendations of items, tags or users from the same approximated tensor [50] – a factorization needs to be computed only once for all types of recommendations.

However, there are many practical difficulties that restrict usage of tensor based recommenders in real world applications. Below is a list of the most significant problems that are addressed by this algorithm:

- A factorization is computationally demanding process and most of the tensor based recommenders [40, 50] calculate tensor approximation in the offline mode.
- When new users, items or taggings are inserted into a system there is a need to recompute tensor approximation so the appropriate recommendations are generated.
- Sparse STS data restricts factorization technique to detect latent relations hence a recommender is not always able to generate appropriate recommendations. A particular topic in STS is represented with various tags assigned by different users. Therefore, user preferences can be too specific due to the nature of tags diversity. In consequence, a factorization technique cannot correctly detect important interests of a user as these preferences can be defined differently by other users.
- Excessive memory demands when large datasets are used.
- Time-consuming tuning of factorization parameters to generate accurate recommendations.

Our approach addresses above mentioned problems with exploitation of the clustering techniques that reduce the size of a tag space. The proposed method is motivated by the fact that majority of the STS contain a lot of semantically related tags which can be grouped. We also introduce a heuristic method to speed-up parameters tuning process for HOSVD recommenders. The main contribution is twofold:

- Precision of recommendations is improved.
- Execution time of a tensor approximation is significantly decreased.

In consequence, memory requirements are significantly decreased. Also, the factorization can be recomputed more often and recommendations will embrace the new entered objects.
To the best of our knowledge, we are the first who introduce clustering of tag space to reduce a dimension of the tensor to improve precision and execution time of the factorization. This work was accepted at the User Modeling, Adaptation, and Personalization - 20th International Conference, UMAP 2012 [31].

HOSVD A higher-order singular value decomposition (HOSVD) is an extended version of the SVD applied to the multi-dimensional matrices. SVD computes matrix approximation for any matrix $F_{D_1 \times D_2}$ in the following way:

$$F_{D_1 \times D_2} = U_{D_1 \times D_1} . S_{D_1 \times D_2} . V^T_{D_2 \times D_2} \tag{12}$$

where $U_{D_1 \times D_1}$ contains left singular vectors (eigenvectors of FF^T), $V_{D_2 \times D_2}$ contains right singular vectors (eigenvectors of $F^T F$) and $S_{D_1 \times D_2}$ is diagonal matrix with singular values – square roots of the non-zero eigenvalues of FF^T sorted in descended order. In the area of information retrieval a well-known technique Latent semantic indexing (LSI) also utilizes SVD – it reveals latent relations between words and documents from a corpus. LSI addresses synonymy

of words - which is also crucial for the HOSVD. It is common to process only first c top singular values and corresponding singular vectors ($c \leq min(D_1, D_2)$) to achieve better approximation of the matrix F – it removes noise and preserves only the most important information from the original matrix.
Tensor of n-th order – is multidimensional array with N indices, denoted as $\mathcal{A} \in R^{I_1 \times I_2 \times \ldots, \times I_N}$. In this work we consider only 3-rd order tensors.
Tensor fiber – is one dimensional fragment of a tensor (column vector), such that all indices are fixed except for one. For 3rd order tensor there are column, row and tube fibers. A tensor can be converted into so called mode matrices by arranging particular fibers of a tensor as columns of mode matrices. Tensor of 3rd order can be unfolded into three different mode matrices with the following dimensions:

$$\begin{array}{ll} A_1 \in R^{I_1 \times I_2 I_3} & \text{– column fibers of } \mathcal{A} \text{ as columns of } A_1 \\ A_2 \in R^{I_2 \times I_1 I_3} & \text{– row fibers of } \mathcal{A} \text{ as columns of } A_2 \\ A_3 \in R^{I_3 \times I_1 I_2} & \text{– tube fibers of } \mathcal{A} \text{ as columns of } A_3 \end{array}$$

Mode-n multiplication of tensor by matrix – a mode-n multiplication

$$\mathcal{Y} = \mathcal{A} \times_n F \tag{13}$$

of a tensor $\mathcal{A} \in R^{I_1 \times I_2 \times \ldots \times I_N}$ by a matrix $F \in R^{D_n \times I_n}$ is a tensor $\mathcal{Y} \in R^{I_1 \times \ldots \times I_{n-1} \times D_n \times I_{n+1} \times \ldots \times I_N}$ with elements:

$$y_{i_1, i_2, \ldots, i_{n-1}, d_n, i_{n+1}, \ldots, i_N} = \sum_{d_n=1}^{D_n} = a_{i_1, i_2, \ldots, i_N} f_{i_n, d_n} \tag{14}$$

HOSVD of 3rd order tensor is defined as:

$$\mathcal{A}' = \mathcal{S} \times_1 U^1_{c1} \times_2 U^2_{c2} \times_3 U^3_{c3} \tag{15}$$

where U^1_{c1}, U^2_{c2} and U^3_{c3} are matrices with the top c_i left singular vectors from the SVD of $1, 2, 3$ mode matrices respectively. Core tensor S is obtained according to:

$$\mathcal{S} = \mathcal{A} \times_1 (U^{(1)}_{c1})^T \times_2 (U^{(2)}_{c2})^T \times_3 (U^{(3)}_{c3})^T \tag{16}$$

The factorized tensor $\mathcal{A}'$ is the approximation of the initial tensor $\mathcal{A}$.

HOSVD in Social Tagging Systems

The usage data of a recommendation system are represented by 3rd order tensor – $\mathcal{A}$ where for a particular user with a selected information item and an assigned tag is stated a weight 1 and for all other cases where is not created relation a weight is 0 :

$$a_{u,i,t} \in \mathcal{A}, a_{u,i,t} = \begin{cases} 1, \text{ exists a relation for } (u, i, t) \\ 0, \text{ no relation between } (u, i, t) \end{cases} \tag{17}$$

The tensor is unfolded into the three mode matrices, denoted as the 1-mode, 2-mode and 3-mode respectively. The unfolded mode matrices from the initial tensor $\mathcal{A}$ are subject of the SVD. It results into creation of U^n, S^n, V^n matrices. The most important are U^1, U^2, U^3 as they contain the left singular vectors of the 1-mode, 2-mode and 3-mode matrices.

Users	Information items	Tags	Weights
U_1	I_1	T_1	1
U_2	I_1	T_1	1
U_2	I_2	T_2	1
U_3	I_3	T_3	1

Table 1: The associations between the objects

$$\begin{pmatrix} -0.53 & 0.00 & -0.85 \\ -0.85 & 0.00 & 0.53 \\ 0.00 & 1.00 & 0.00 \end{pmatrix}$$

Figure 15: Application of the SVD to the 1st mode matrix - U^1 matrix

We are including an illustrative example to better describe the HOSVD factorization. Let us assume a social tagging system with 3 different users, 3 different information items – articles and 3 tags. The associations between these objects are shown in the Table 1. The initial tensor A is constructed according to the usage data (Table 1). The tensor is unfolded into the three mode matrices, denoted as the 1-mode, 2-mode and 3-mode respectively.

The unfolded mode matrices from the initial tensor $\mathcal{A}$ are subject of the SVD. It results into creation of U^n, S^n, V^n matrices (see Figures 15 and 16) with the U^1 and S^1 matrices respectively, $V^{1\,T}$ is not depicted due to the huge size and is not required in the further computations), the most important are U^1, U^2, U^3 as they contain the left singular vectors of the 1-mode, 2-mode and 3-mode matrices.

The algorithm stores top c_i singular values of $i-th$ mode matrices with corresponding left singular vectors in order to construct the core tensor and the approximated tensor $\mathcal{A}'$ From the new tensor $\mathcal{A}'$, the recommendation system is able to suggest tags or information items with the highest weights to a given user.

Clustered tag space and Genetic algorithm We propose to utilize a cluster analysis on the tag space to group similar tags into clusters. Such reduced tag space causes smaller initial tensor and in consequence better time performance, lower memory demands are achieved while the quality of recommendations is preserved. Before describing technical details of our method, we provide motivation for clustering and describe our approach with the illustrative example.

Motivation The majority of tags used within the social tagging systems are assigned and used rarely. These infrequently used tags cause unnecessary high memory demands e.g. 48122 tags from the M-eco system were assigned just once, and an initial tensor will contain 48122 slices (each slice of the tensor corresponds to a particular tag) of the size $|U| \times |I|$ however

$$\begin{pmatrix} 1.62 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 0.62 \end{pmatrix}$$

Figure 16: Application of the SVD to the 1st mode matrix - S^1 matrix

Dataset	Number of tags at each frequency	Tag frequency
M-eco system 05/07/2012	48122	1
M-eco system 05/07/2012	8393	2
M-eco system 05/07/2012	3700	3

Table 2: Amount of rarely used tags in the M-eco system.

User	Web item	Tag	Weight
u1	www.ecoli.org	e.coli	1
u1	www.ecoli.org	e-coli	1
u1	www.ecoli.org	Escherichia coli	1
u1	www.bbc.co.uk/news/world-europe-13597080	cucumber	1
u1	www.ecoli.org	pictures	1
u2	www.ecoli.org	e-coli	1
u3	cnn.com	news	1

Table 3: Tagging posts of the motivational example.

each such slice will contain only one value. We explore the M-eco system and evaluate the amount of such rarely used tags in the Table 2. Our findings are supported also by [26] where authors have shown that around 30 % of all distinct tags were used only once within the Delicious system. Their analysis also shows that synonyms, acronyms and spelling variations occur frequently among tags. Many tags differ only in the spelling variations – upper or lower case initial letters of tags, singulars or plurals, spelling mistakes.
Majority of the rare tags can be grouped with the more frequent ones because of the mentioned reasons. On the other hand, there are rarely used tags with the unique and specific meaning that cannot be clustered. It is the task of the clustering to appropriately distinguish which tags can (not) be clustered.
Let us present the following motivational example with the 3 users (*u1, u2, u3*), 3 web items (*www.ecoli.org, cnn.com, www.bbc.co.uk/news/world-europe-13597080*) and 6 distinct tags (*e.coli, e-coli, Escherichia coli, cucumber, pictures, news*). The tagging posts of the users assigned to the web items are depicted in the Table 3.
Obviously, the tags *e.coli, e-coli, Escherichia coli* are semantically related and our approach groups them into one cluster *e.coli cluster* = {*e.coli, e-coli, Escherichia coli*}. Grouping the similar tags into the cluster provides new relations as follows: Before the HOSVD factoriza-

User	Web item	Tag	Weight
u1	www.ecoli.org	e.coli cluster	3
u1	www.bbc.co.uk/news/world-europe-13597080	cucumber	1
u2	www.ecoli.org	e.coli cluster	1
u3	cnn.com	news	1

Table 4: Tagging relations when tags about e.coli are grouped into one cluster.

tion is computed, we find the number of preserved top singular values for each mode matrix with GA method. Once the tensor is approximated, the following scores are obtained for the given triplets. HOSVD factorization reveals a latent relation between the user *u2* and the web

User	Web item	Tag	Weight
u1	www.ecoli.org	e.coli cluster	3.0435
u1	www.bbc.co.uk/news/world-europe-13597080	cucumber	0.9287
u2	**www.bbc.co.uk/news/world-europe-13597080**	**cucumber**	**0.2572**
u2	www.ecoli.org	e.coli cluster	0.8429
u3	cnn.com	news	1

Table 5: Results of the factorization with the revealed relation between user *u2* and web item *www.bbc.co.uk/news/world-europe-13597080*.

item *www.bbc.co.uk/news/world-europe-13597080*. Therefore, the recommendation system recommends item *www.bbc.co.uk/news/world-europe-13597080* to the user *u2*. The similar result would be obtained if the tags *e.coli, e-coli, Escherichia coli* would not be merged however our approach removes 2 slices of the tensor and in consequence it improves time performance of the factorization and decreases memory requirements.

GA - tuning tensor based recommenders In this section, we propose a heuristic method for estimating optimal parameters for tensor based recommenders. The approach is based on Genetic Algorithm (GA) [18] and it identifies the optimal parameters for HOSVD based recommender so that the best possible accuracy is attained. GA is adapted to our search problem in the following way:

- The parameters c_1, c_2, c_3 are genes that are together encoded in a chromosome.
- A population of possible solutions consists of k different chromosomes.
- Fitness function is the average precision for considered users and for the provided chromosome (parameters c_1, c_2, c_3).

Below are described steps of the process of searching for the optimal parameters with GA.

1. A population is randomly generated in a way that each chromosome consists of 3 genes. Each gene represents a particular parameter and it is randomly initialized (parameter c_i belongs to the defined interval $[0.4, 0.8]$).
2. Calculate the fitness function of each chromosome from the population. The result of the fitness function is average precision for the given parameters.
3. New population is created such that chromosomes that produce best average precision are reproduced. New chromosomes are created with mutation and cross-over operations.
4. Go to step 2, repeat until fixed number of iterations is reached.

The method can be accelerated that only some users are considered when recommendations are generated. The selection of users should statistically represent all types of users that occur in a given dataset. Once the searching process is finished, GA returns a chromosome with the best average precision (result of the fitness function). The given chromosome contains the optimal parameters c_1, c_2, c_3.

Cluster analysis of tags Different clustering techniques can be utilized to group similar tags into a cluster. The general proposed approach consists of the following steps:

1. Perform cluster analysis of a tag space with the selected clustering method from the 4 proposed techniques.

2. Build an initial tensor where a tag dimension has the same size as the amount of obtained clusters. A tagging performed by a user u to a item i with a tag t is a triplet *(u, i, t)*. All such triplets are encoded to corresponding positions in the initial tensor with the initial weight 1 and a tag t is mapped to the matching cluster. When two or more triplets share the same item and user – differ only in tags and these tags belong to the same one cluster a final weight in the tensor is the amount of such triplets, e.g. given two triplets: *(u, i, t_1), (u, i, t_2)* and tags t_1, t_2 belong to the same tag cluster, then an initial tensor will contain weight 2 at the position (a row for a user u, a column for a item i and slice corresponding to tag cluster with tags t_1, t_2).

3. Find the optimal parameters for the recommender with the proposed GA based heuristic method.

4. Compute tensor factorization for the constructed initial tensor. Finally, items recommendations are generated according to the sorted weights from the factorized tensor for the given user and all not observed items.

4.4.6 Recommender System Techniques for Threat Assessment

Besides the approaches explored in this deliverable for *Adaptive Tuning and Personalization*, within the M-Eco project we have also exploited Recommender System techniques for *Threat Assessment* (Task 4.5 according to the Description of Work).

The motivation is that for public health officials, who are participating in the investigation of an outbreak, the millions of documents produced over social media streams represent an overwhelming amount of information for risk assessment. To reduce this overload we explore to what extent recommender systems approaches can help to filter information items according to the public health users' context and preferences (e.g., disease, symptoms, location). In particular, we propose two methods to facilitate the task of Threat Assessment, namely: *Personalized Tweet Ranking for Epidemic Intelligence* and *Online Topic Discovery in Social Streams*.

Below we present a summary of such methods and discuss the technical details and evaluation in Deliverable 4.3 under the chapter *Monitoring and Support for Risk Assessment*.

Personalized Tweet Ranking for Epidemic Intelligence Personalized Tweet Ranking algorithm for Epidemic Intelligence (PTR4EI) provides users a personalized short list of tweets that meets the context of their investigation. PTR4EI exploits features that go beyond the medical condition and location (i.e., user context), but includes complementary context information, extracted using LDA and the social hash-tagging behavior in Twitter, plus additional Twitter specific features. Our experimental evaluation showed the superior ranking performance of PTR4EI.

The main advantage of PTR4EI is that can discover new relationships based on a limited context in order to help filtering the large amount of data.

Online Topic Discovery in Social Streams Recently, modern disease surveillance systems have started to also monitor social media streams, with the objective of improving their timeliness to detect disease outbreaks, and producing warnings against potential public health threats. The real-time nature of Twitter makes it even more attractive for public health surveillance.

For example, an epidemiologist monitoring Twitter to enhance his capabilities for epidemic detection and control. Social media data has to be processed in real-time, additionally, the surveillance models must be updated online, otherwise the timeliness required for such a critical system will be heavily impacted. Any time lag in modeling the data could render the outcome of the modeling obsolete and useless.

In the presence of a continuous stream of incoming tweets, arriving at a high rate, our objective is to process the incoming data in bounded space and time and recommend a short list of interesting topics that meet users' individual needs.

The high rate makes it harder to: (i) capture the information transmitted, (ii) compute sophisticated models on large pieces of the input, and (iii) store the input data, which can be significantly larger than the algorithm's available memory.

This problem setting fits a streaming model of computation by Muthukrishnan [36], which establishes that, by imposing a space restriction on algorithms that process streaming data, we may not be able to store all the data we see. The impact is that the data generated in real-time carries high-dimensional information which is difficult to extract and process.

We propose to use *Stream Ranking Matrix Factorization* – `RMFX` –, an approach for recommending topics to users in presence of streaming data. `RMFX` represents a novel principled approach for online learning from streams, that selects a subsample of the observed data based on the objective function gradients, and uses it to guide the matrix factorization.

The online nature of the algorithm can certainly benefit time-sensitive filtering for threat assessment and outbreak detection.

5 Evaluation

In this section, we present the experimental evaluations of the personalization models (see Section 4) that address the problems identified in Section 2. The experimental evaluation aims at testing whether our proposed approach performs as expected so that we can claim our methods as contributions to solve or lessening the problems in the motivation of this work. The evaluations are represented by *user study*, in which participants assess the performance our developments (recommendations and tag clouds); *interviews and group discussions*, in which the participants answer pre-defined questions or provide feedback of our methods; and *controlled experiments*, in which public medical datasets are utilized for the assessment of our methods.

5.1 Evaluation of Multi-factor Recommendations

5.1.1 Introduction

In this section, we describe an evaluation of the multi-factor approach presented in Section 4.3.5. We use a list of previously identified outbreaks provided by RKI. This list contains over 400 outbreaks ranking from 21/05/2011 to 13/07/2011. The outbreaks are all related to the EHEC epidemic that occurred last year. The list covers small specific cases in smaller cities up to larger, countrywise reports. These outbreaks were matched to existing signals produced by WP4, producing an overlap of signals identified by M-Eco and outbreaks reported by RKI.

Precision	Multi-factor	Signal-based
@3	0.294	0.085
@5	0.315	0.081
@10	0.333	0.142
@20	0.264	0.201

Table 6: Precision values for the multi-factor and signal-based approaches

These signals are part of larger database of signals with aproximately 7000 signals and 20000 documents up to 13/07/2011. It is this database that we use in our evaluation.

In addition, although we use a case from last year, the specific dates are irrelevant since we simulate the system as events unfolded. That is, we start our experiment as if the current date were 21/05, when the first outbreaks were detected by RKI. For each day, we run our recommendation component and compare which signals recommended to the user are also in the list provided by RKI. We repeat the process every day and, after all days are processed, we compute the average precision and recall.

We assume a user with a signal definition set for EHEC in Berlin. At the end of each day, we rate positvely up to the top 10 signals recommended to the user that overlap with outbreak cases provided by RKI. Conversely, we rated negatively up to the top 10 signals recommended that do not overlap with these cases. Although we can simulate the rating, we discard the tag factors in our model for this evaluation because they would require more explicit and personal user feedback. The factors we consider, therefore, are signal definition, location, date and content of documents.

We compare the results of our multi-factor approach with the approach of recommendations based on signal definitions cited in Section 4.4.1. Unfortunately, we cannot use any of the tag-based or collaborative filtering methods in this evaluation because they require user feedback which is not available.

5.1.2 Results

Figure 17 shows the Precision x Recall curve for both the multi-factor and the signal-based approaches. Precision is always much higher for our multi-factor approach, indicating that the addition of other factors to model can help in matching the user preferences. The highest precision of the multi-factor approach was 0.506 when the recall was 0.232. 217 signals were returned and 110 overlapped with the list provided by RKI. On the other hand, in the content-based approach, the highest precision was 0.344 on a much higher recall, 0.439. 606 signals were returned and only 209 overlapped.

The multi-factor approach has also better results when looking at the first returned signals. Precision @3, @5, @10 and @20 is always around 0.30 for the multi-factor as opposed to values lower than 0.1 for the signal-based approach. The table 6 shows these values.

5.2 Personalized Tag Cloud Evaluation

In the following section we present all the evaluations related to the tag cloud component that were conducted during the M-eco project. Firstly, we will summarize two experiments *Tag Cloud Evaluation for E.coli Outbreak in Germany* and *Preliminary Tag Cloud Feedback.*

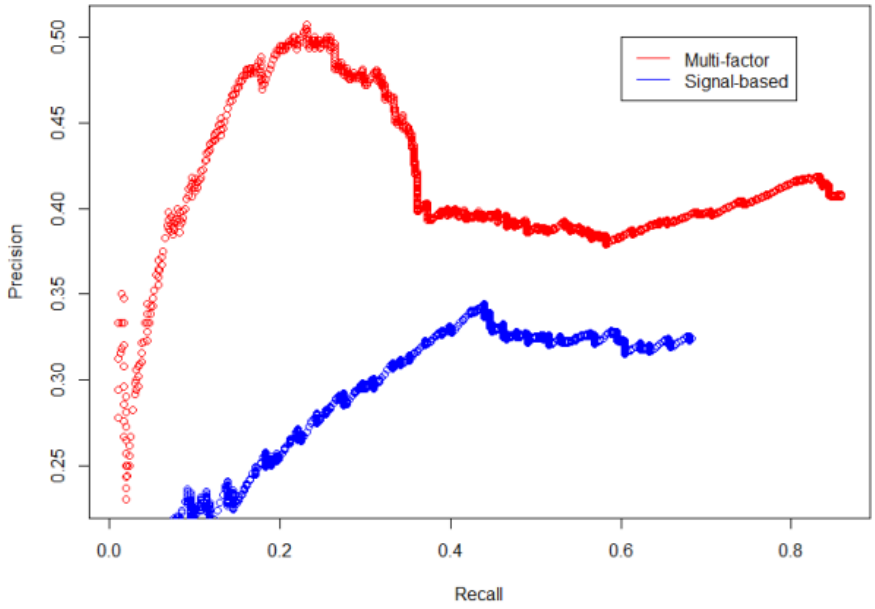


Figure 17: Precision and recall graph of the multi-factor and signal-based recommendation approaches

These two evaluations are shortly summarized as they are more precisely described in the M-Eco Deliverable 5.2 (D5.2) [11], *section 5.2*. Last two evaluations were conducted during the evaluation meeting in Berlin. The former one was organized in collaboration with a partner from WP4. The objective of the session was to evaluate various GUI elements including tag clouds. The last evaluation is completely dedicated to tag cloud usefulness, advantages and drawbacks. In the end of this section, we provide a discussion about tag cloud usefulness for medical surveilance systems, also summarize current drawbacks and provide directions for the future work.

Tag Cloud Evaluation for E.coli Outbreak in Germany. The objective of the experiment was to assess relevancy and utility of parametrized tag clouds. The evaluation was conducted with the group recommendations component (D5.2 [11], *Section 5.5*). where the aim of the experiment was to simulate EHEC outbreak in Germany. News recommendations were presented to users through Twitter interface where each suggested item contained a link to the parametrized tag cloud.

For the assessment, the participants in the experiment were asked to assess a relevance of the corresponding tag clouds for the recommended items in Twitter interface. The rating scale was between 1 (dislike) and 5 (excellent) for the relevancy of a given tag cloud. This rating covers the following aspects:

- Is a given tag cloud's structure appropriate?
- Are depicted tags relevant and meaningful?
- What is the general impression from a given tag cloud?

The participants were not overloaded with more questions as the experiment was lasting for almost a week. For each round participants assessed and rated five different tag clouds. We have obtained 149 user ratings of tag clouds relevance. The average rating was 3.244 and ratings with evolving time were increasing. The main reason of average relevancy of generated tag clouds was that tags depicted in the tag cloud link to the documents which are related to a given location but their content is too general. Participants consider tag clouds as an useful retrieval tool however, the identified drawbacks should be solved in future development. The complete setup and further details about this user study can be found in the *Section 5.2. of M-Eco Deliverable D5.2* [11].

Personalized Tag Cloud Feedback. The second experiment was conducted with 6 medical experts to assess the usability of our personalized tag cloud. This preliminary evaluation scenario assumes a hypothetical medical expert that created a signal definition matching “cholera" as a medical condition and “Haiti" as a location. The set of documents belonging to signals that match this signal definition is provided. The participants are then asked to find out more about this outbreak based on the documents found through the interface of the tag cloud provided. Once the exploration process is finished, the participants are asked to answer a set of open-ended sub-questions that correspond to the following evaluation questions:

- Does the tag cloud improves a navigation process through documents?
- Does the structure of the tag cloud correspond sufficiently to the defined signal definition?
- Are the documents retrieved through the tag cloud an appropriate match to the signal definition?

As a result, participants found the personalized tag cloud as useful tool that can improve browsing and retrieving of documents within M-Eco system. A query refinement mechanism clearly improves tag-based information retrieval. Tag cloud should contain more semantically useful tags and not relevant tags should be not displayed. Users preferred a single language tag cloud, therefore it should be avoided of mixing tags with different languages. The complete setup and further details about this user study can be found in the *Section 3.3.3. of M-Eco Deliverable D2.2* [3].

Visualization tools - a joint evaluation with WP4 During the Berlin’s 2012 user workshop, we have conducted a joint usability study with WP4 in order to evaluate different visualization tools. A partner from WP4 presented the following tools:

- Signals Cluster Map
- Medical Conditions Timeline

Moreover, the evaluation was extended with a tag cloud component. In total, nine experts participated in the study, four of them from RKI, three from NLGA, one from ECDC, and one from WHO. The study was designed based on five scenarios related to the following medical conditions: Scharlach [Scarlatina], Measels [Masern], Windpocken [Chickenpox], Keuchhusten [Whooping cough], EHEC [EHEC]. For each scenario participants were asked a set of questions regarding their understanding how the outbreak evolves. The participants were using the presented visualization tools in order to answer the questions. The most preferred tool is a

timeline visualization. However, users find all the presented visualization components useful for the medical surveilance tasks. Participants consider a tag cloud component as a beneficial tool because it provides the following benefits:

- it provides an additional context about considered signals
- it provides an insight, overall picture about considered signals
- simplifies a browsing process as it is easier to retrieve a specific subset of relevant documents related to the specific signal

The list of questions that were presented to the participants, a detailed description of WP4 visualization tools and other details about this joint evaluation study can be found in M-Eco Deliverable 4.3 (D4.3) [49], *section 9*.

Tag cloud evaluation from Berlin's 2012 user workshop Similarly as in the previous evaluation, in total, nine experts participated in the study, four of them from RKI, three from NLGA, one from ECDC, and one from WHO. The objective of this evaluation was to assess the quality of the tag clouds related to the signals presented to the users within the M-Eco project. Participants should consider the initial period of the EHEC outbreak in Germany from 15 to 23 of May, 2011 (see Figure. 18). The assumption is that the participant is responsible for monitoring reports from the web with the goal of understanding how the disease is spreading. Given this scenario, please access the web page http://meco.l3s.uni-hannover.de:8080/WP4WS/jsp/vix.jsp?mc=ehec&startDate=20110515&endDate=20110523. You will see signals of various kinds that should be related to the given outbreak. Please, browse through them, explore related tag clouds in order to assess their relevance to the outbreak. As participants finished the exploration of the signals, they were instructed to answer the following questions:

1. On a scale from 1 to 5, how would you rate the overall **usefulness** (how relevant, interesting are tag clouds for a user interests and their decision-making.) of the tag clouds?
2. On a scale from 1 to 5, how would you rate the overall **correctness** (how tags do match to the given outbreak) of the tags shown in the tag clouds?
3. On a scale from 1 to 5, how would you rate the overall **usefulness** of the documents retrieved through the tag clouds?
4. On a scale from 1 to 5, how would you rate the overall **correctness** of the documents retrieved through the tag clouds?
5. What are the main benefits of tag clouds?
    - Simplifies browsing of documents
    - Presents additional context for the signal
    - Provides better overview about signals
    - No particular benefits
    - Others, specify:

6. What are the limitations / drawbacks of tag clouds?

   - Useless when the amount of documents is low
   - Do not simplify browsing process of documents
   - No particular drawbacks
   - Others, specify:

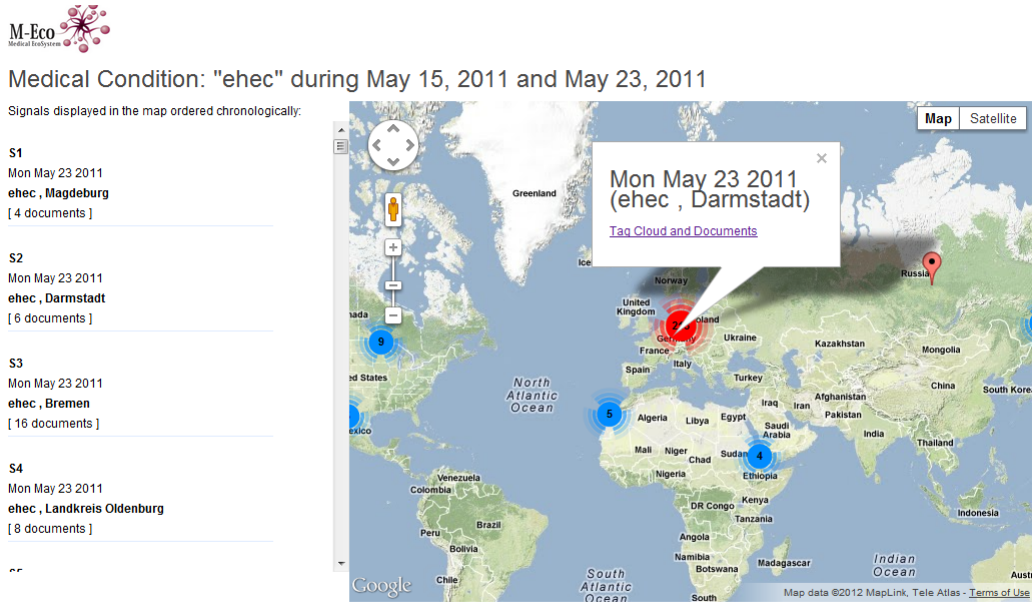


Figure 18: Signals for the initial period of the EHEC outbreak in Germany from 15 to 23 of May, 2011 with links to the corresponding tag clouds.

Majority of participants consider tag clouds as useful for a decision-making in medical surveilance tasks. The average rating for the first question is 3.44. The second question regarding how tags do match to the given outbreak – an overall corectness of tags in the tag clouds has attained an average rating 4. The overall usefulness of underlying documents for considered tag clouds has an average rating 3.55 and the overall corectness of these documents has an average rating 3.44.

Participants find the following benefits as the most important (sorted according to total number of votes):

- Presents additional context for the signal (6 out of 9).
- Simplifies browsing of documents (4 out of 9).
- Provides better overview about signals (4 out of 9).

However, participants also found some drawbacks or issues that should be addressed in the future development of tag clouds for medical surveilance tasks:

- Useless when the amount of documents is low.
- Useless when there are too many tags (irrelevant tags should be filtered out).

Besides the above-described results of the evaluation study, we have obtained a valuable feedback and comments from participants that is presented in the following paragraphs. An participant from NLGA suggested to decrease the number of tags in the cloud. Second proposal from RKI participant is to allow users to easily differentiate between medical tags and other terms. Similar suggestion came from the ECDC participant, to differentiate a tags types with distinct colors e.g., medical tags would be represented with a specific color, other color for tags with other but relevant content and different color for probably irrelevant terms. Another suggestion is to allow tag clouds users to hide/remove specific irrelevant tags from tag clouds and in a such way remove noisy documents.

The final conclusions and discussion from this evaluation study and also all previous user studies are presented in the following paragraph.

Final conclusions and discussions Tag cloud as a retrieval interface is considered as useful tool for medical surveilance tasks. Users that participated in the conducted evaluations listed the following benefits:

- Tag clouds provide simplified browsing and exploration process of a large set of documents as the tool allows to retrieve a required subset of documents in a fast and easy way. Tag clouds enable a query refinement and in a such way clouds can deliver to users various subsets of documents.
- Tag clouds present an additional context about an underlying set of documents. In the medical domain these additional context can be understood as extended medical conditions of a considered outbreak.
- Tag clouds allow users to make serendipitious discoveries, realize new interesting facts about a given outbreak that would not be possible with a tradition keyword based search.
- Tag clouds have a huge potential for collecting user feedback about (ir)relevant tags and documents. Such collected data can be exposed to data collection processes. In a way data collection process can be adjusted and improved.

Besides many benefits of tag clouds there are still many unresolved research questions and problems that should be considered in the future. In the following part, we summarize main issues and drawbacks of the concept of tag clouds:

- **Selection process of underlying documents for a tag cloud generation**. Obviously, when irrelevant and not user interesting documents are passed to tag cloud generation algorithms then the generated tag clouds do not satisfy users requirements and needs.
  - Classical symptoms based data collection process is not sufficient. It is required that user feedback about (ir)relevant terms, documents will be passed to data collection algorithms in order to incrementally improve a quality of collected documents.

  - Another harvesting techniques have to be explored in order a precision of retrieved documents.

- **Signal generation process affects a tag cloud generation**. Obviously, when signals aggregate a low number of documents tag clouds start to loose their meaning and become useless.
  - A future development of signal generation algorithms should concentrate on a research question how to aggregate a reasonable number of relevant documents such that a generated signal structure will deliver to medical experts useful and relevant information and a tag cloud structure will be appropriate.

- **Annotation process of short messages**. Obviously, a number of retrieved tweets and news per day is enormous. Therefore is absolutely impossible to rely on users to annotate these messages and in consequence generate tag clouds. There is a clear need to develop robust algorithms that will be able to annotate short messages with reasonable and relevant terms - tags.
  - A huge number of tweets contains a link to the external webpage. A content of this external webpage should be considered during the annotation process of the original tweet.
  - As vocabulary of social networks users differs there is a need to propose algorithms that will be able to aggregate and normalize various terms and in a such way better aggregate similar messages.

- **Filtering out irrelevant tags**. Participants of the conducted evaluation studies complained about irrelevant tags. Therefore, there is a need to address the following research questions:
  - There is a need to define when and in which context a given term is (ir)relevant.
  - Tag clouds should be able to adjust their structure during the user interaction in order to optimize tag clouds structure to present the most relevant tags to the user.

- **Tags selection algorithms**. This research problem is closely connected to the previous problem. During tag clouds generation process there is a need to select only the most relevant tags that cover the maximal subset of underlying documents. The future development should concentrate on developing new metrics that will combine state-of-the-art metrics such are coverage, overlap with terms relevancy.

5.3 Evaluation of Generative Model of User Taggings

The goal of the generative model is to assign tags to documents when no or few tagging is available. As mentioned previously, low tagging activity makes difficult to build user profiles and, as a result, it generates inaccurate recommendations. The proposed model (see Section 4.4.6 in the M-eco deliverable D5.2 [11]) was evaluated in order to verify the quality of generated taggings. The evaluation was conducted in a such way that users considered generated tags and removed those that were not relevant for given documents. Each participant was asked to evaluate 10 different documents and to assess generated tag assignments by removing not relevant tags (such tags that do not describe or reflect the content of a document).

In total there were 6 different participants, all of them are medical or surveillance experts. Each participant was assessing a different set of documents in order to evaluate all types of documents and make a sample of evaluated documents as representative as possible i.e. evaluated documents should represent a whole collection of available documents in the M-Eco system. The generative model was evaluated by computing precision for each evaluated document. Precision is a common metric in the area of information retrieval which was computed in the following way:

$$Precision(doc_i) = \frac{(|tags_{doc_i}| - |not\ relevant\ tags_{doc_i}|)}{|tags_{doc_i}|}, \tag{18}$$

where $|tags_{doc_i}|$ represents a number of tags of the document doc_i and $|not\ relevant\ tags|$ is the number of not relevant tags stated by the participant. Precision of evaluated documents ranges between 0.5 and 1. The average precision of all evaluated documents is **0.72**. This means that **72**% of generated tags by the generative model were considered by users as relevant. Such precision is considered as acceptable and the generative model replaces an expensive user taggings in a sufficient way. The majority of not relevant tags belongs to the following groups:

- General terms that do not have any descriptive value for a given document e.g., articles, bathroom, buddy, component and country.
- Not meaningful abbreviations that do not have any descriptive value for a given document e.g., dvo, gbm, gdhg, lyhmkbdd, oct., ughh and ucv.
- Names that do not have any descriptive value for a given document e.g., Albert, Hitler, Lauren and Lucy.
- Terms that express some time period and do not have any descriptive value for a given document e.g., days, month, century and time.

Because of the above mentioned problems, the model should be improved to avoid a generation of tags that do not have any descriptive value. Another drawback of this method is a limitation of generating only tags that occur as terms in the given document. To address this problem the model should be extended to produce also tags that are relevant for given documents but do not have to occur within the text of documents. This can be achieved by integrating some medical ontologies or by querying a relevant search engine with generated tags and extract new relevant terms from the retrieved documents. Such extracted terms could be used as new tags of considered documents.

5.4 Trajectory Prediction Evaluation

Methodology To evaluate our approach of trajectory reconstruction presented in section 4.3.4, we attempt to reconstruct the trajectories of the event Occupy Wall Street[9], using data from Twitter. We use this event because it was the most readily available at the time of crawling. To obtain specific data about it, we first crawled a larger set during the one month period of the start of the event, from September 17 until October 16 2011. The crawler was built in a distributed configuration to increase performance since Twitter limits the number of requests per IP to 350/hour. In total, 72,000 user profiles and 7 million messages were obtained. Out

[9] http://occupywallst.org/

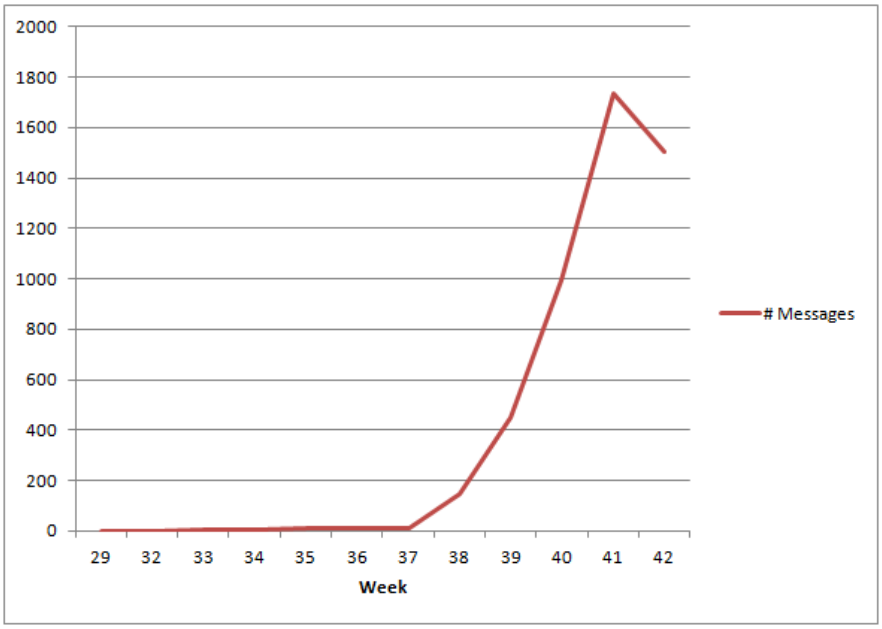


Figure 19: Number of Twitter messages per week related to the Occupy Wall Street or similar events

of these, 4,905 messages were related to the Occupy Wall Street or similar events (e.g., Occupy Boston) during the period.

Figure 19 shows the number of messages obtained per week. The event started officially on September 17 2011 (end of week 38). The small number of messages before that are initial discussions about it or advertising the official start. At the end of the one month period of messages crawled, the event was still going on in different cities across the United States (US) and the world. It started in New York City but by October 15 the event had spread to 951 cities in 82 countries[10]. Our dataset contains messages from 526 different cities in 71 countries.

To perform an initial evaluation of our approach, we compare the trajectories built by our approach with different dates of occurrence of the event in different cities in the US. According to a Wikipedia webpage[11], which aggregates different sources to report the chronology of the event, many differents cities organized similar protests. In this work we assume that all these protests are part of the same “Occuppy" event. One of the earlier ones to follow was Occupy Chicago, which occurred initially on September 24. Cities in California (e.g., San Francisco or Los Angeles) or central parts of the country only started organize similar protests by the beginning of October. We use these dates in order to compare the results of the trajectories built.

Results We show the distribution of all messages over the world on Figure 20. The points represent the location of at least one message and the circles around some of them represent the number of messages in that particular location. The larger the number of messages the larger

[10] http://www.france24.com/en/20111015-indignant-protests-go-global-saturday
[11] http://en.wikipedia.org/wiki/List_of_ “Occupy"_protest_locations

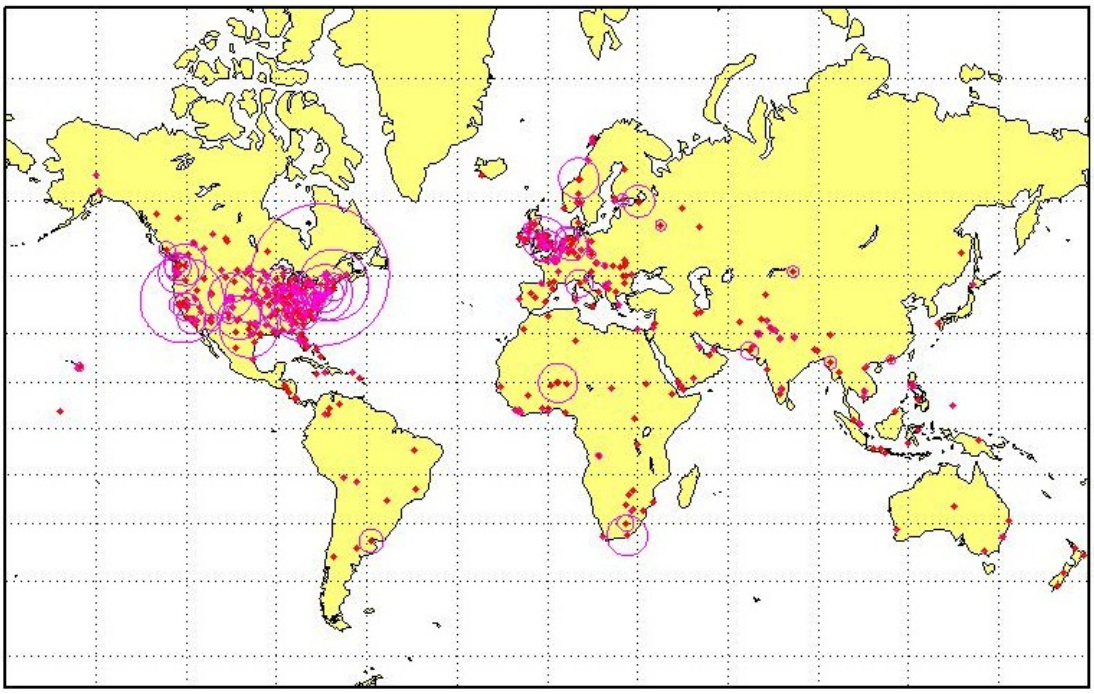


Figure 20: Twitter messages around the world

is the diameter of the circle. Note that most activities are on the east and west coast of the US, specially around New York City and San Francisco. Some more intense activity can be seen in the cities of Texas and Colorado on Central US and in the state of Washington, particularly in Seattle. In other countries, activity on Twitter in general also corresponds to locations where protests took place. London and surrounding areas are particularly active.

The analysis of the trajectories also resemble how the event spread over the US. Figure 21 shows the trajectories built after the first week, until September 24. The first trajectory started from New York City on September 17. Each arrow from there represents one day later in time. Note that two different trajectories emerge from the city, one going north and another south. Independently, another trajectory emerged from Lansing and Chicago on September 19. These two cities would late organize protests, the first ones happening on October 15 and September 24 respectively. Similarly, the trajectories south already indicate regions that later organized protests. It also goes in the direction of the states of Texas and Colorado which witnessed a stronger wave of protests. However, there is nothing yet to be seen on the west coast.

Figure 22 shows the trajectories from all messages in the period analyzed. Trajectories in red have 5 or more locations in them. Note how specific locations become clusters from where just one more location is related. This happens because of the 'forking' scheme that makes one location a node from the nearest location in a existing path, creating a new path between them. In this final figure it is also possible to see some paths emerging from San Francisco and Los Angeles in the west coast. A small trajectory also emerges from Seattle in the northwest.

In general, the trajectories resemble and sometimes anticipate the occurrence of protests in different locations. However, the algorithm is yet not accurate in detecting more important

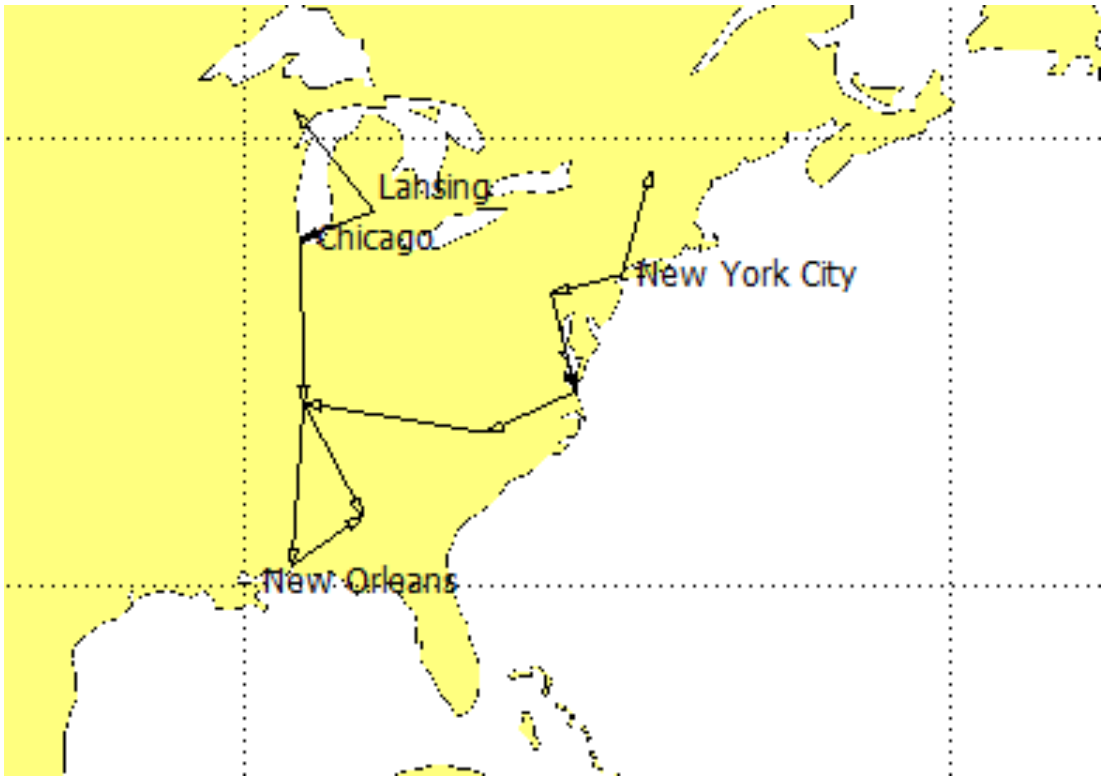


Figure 21: Trajectories after one week

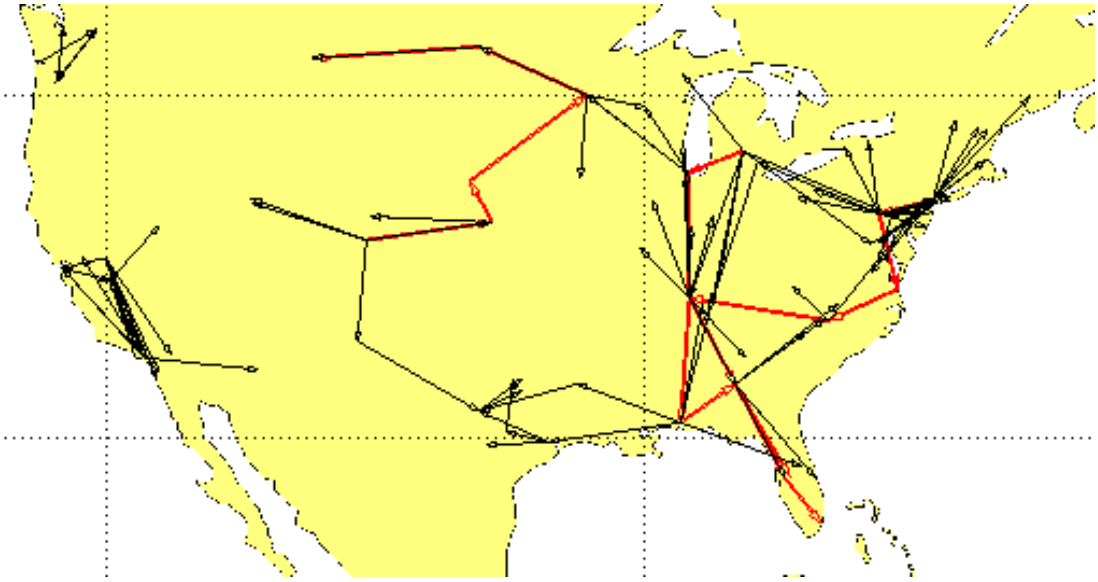


Figure 22: Final trajectories in the US

cities. Sometimes it added to a trajectory a smaller city even though a more important one was nearby. The 'forking' scheme also needs improvements. Currently it sometimes forks a path to a longer new one even though there existed a path with a shorter distance.

This has to do with the greedy implementation of the algorithm. In its current implementation, whenever a first candidate is found that matches the parameters presented on section 4.3.4 it will take action accordingly without looking for alternative candidates. Some of the problems with the locations also have to do with problems with the matching algorithm also presented on section 4.3.4. Because the location informed by the user is free text, sometimes the locations are identified incorrectly.

6 Deployment

In this section we introduce the two ways of accessing our personalization components, either through a stand alone application or using our web services.

6.1 Installation Requirements

In order to run the components as a stand alone application the following software requirements must be fulfilled:

- Java SDK 6;
- Eclipse Java EE edition;
- Jboss 4.2.3 or later from `http://www.jboss.org/jbossas/downloads.html`;
- Apache Ant (it should come with eclipse);
- Subversion plug-in for Eclipse @ `http://subversion.tigris.org`;
- Checkout the project @ `https://iwis.svn.sourceforge.net/svnroot/iwis/WP5`;
- SQL Server with Microsoft SQL Server Management Studio version 10.0.1600.22 or later. This temporary solution because we are changing to PostgreSQL 8.3 due to license constraints.

Configuration and Deployment. After software installation, some configuration must take place:

- Edit your environment variables JBOSS–HOME, ANT–HOME and JAVA–HOME. These names for the variables are suggestive;
- Set your "JBOSS–HOME" in the "build.properties" file and 'seam-gen.properties' file;
- The database configuration should be set in the class database. *IDatabaseConstants.java* and WP5-dev-ds.xml;
- To deploy the system, simply run the task "build.xml" in the target deploy;
- The system should be accessible from the URL `http://localhost:8080/WP5/home.seam`. There you will find a link to the applications.

Data loading. This deployment guideline does not ensure the data loading, which should be gathered primarily by web services of the system. However, for applications that are going to access location-based recommendations there is a requirement of loading the GeoNames database [12]. The article Loading GeoNames Data Into SQL Server 2008 provides a detailed guide line on how to load GeoNames data into SQL Server 2008.

Calling Components Methods. The personalization services are primarily intended to be accessed through our web services (see Section 6.2), however our components can be directly called through the following methods:

- Class:*SignalDB*, Method:*getSignalsByUser(userId,startDate,endDate)* - it returns a signals relevant for a given user ID within a period of time;
- Class:*TagCloudService*, Method:*getTagCloud(signalId)* - it returns a list of tags for a given signal ID;
- Class:*IndicatorDB*, Method:*getRecommendedIndicators(userId)* - it returns a list of recommended indicators for a given user ID;
- Class:*DocumentDB*, Method:*getDocumentsByIndicatorId(indicatorId)* - it returns a list of documents for a given indicator ID.

6.2 Web Services

The major goal of the WP5 web services is to support integration between the project partners, and provide outsiders with personalized information processed by our components. In order to reach such an integration, we implemented web services providers and consumers. The web services providers are interface where personalization services such as recommendations are available for external access. The web services consumes are client applications that gather information provided by others web services, usually from other partners.

We publish our personalization services as REST Web Services, to which external applications only need to make a HTTP request to access the desired service. The REST Web Service call was implemented to return objects either in XML or in JSON format. The web service is called using the basic URL path:

http://{server:port}/{workpackage}/{institution}/rest/{serviceName},

e.g.,

http://localhost:8080/WP5/aau/rest/recommendations

6.2.1 Web Service Providers

In the following we present the web services which provide personalized information.

- **getUserSignalList()** - it returns a list of user signals that were already matched to one of his/her signal definitions in a given period(start date, end date). No documents go attached to the signal list. `http://demos.iwis.cs.aau.dk:8081/WP5/aau/rest/services/userSignalNoDocuments/{userId}/{startDate}/{endDate}/`

[12] http://www.geonames.org/

- **getListRecommendedSignalList()** - it returns list of recommended signals for a given user in a given period (start date, end date). `http://demos.iwis.cs.aau.dk:8081/WP5/aau/rest/services/recommendedSignalList/{userId}/{startDate}/{endDate}/`
- **getUserSignal()** - it returns a list of user signals along with documents that were already matched to one of his/her signal definitions in a given period (start date, end date). `http://demos.iwis.cs.aau.dk:8081/WP5/aau/rest/services/userSignalWithDocuments/{userId}/{startDate}/{endDate}/`
- **tagCloudBySignal()** - it returns a tag cloud as JSON object for a given signal identifier. `http://demos.iwis.cs.aau.dk:8081/WP5/aau/rest/services/tagCloudBySignal/{signalId}/`
- **tagCloudByLocation()** - it returns a tag cloud for all tags assigned to a set of all documents for a given location identifier. The service returns a tag cloud structure encoded as JSON object. The set of considered documents can be restricted with a given time interval. `http://demos.iwis.cs.aau.dk:8081/WP5/aau/rest/services/tagCloudByLocation/{locationId}/{startDate}/{endDate}/`
- **getIrrelevantTerms()** - it returns a top-k most recent irrelevant terms with their frequency of irrelevant ratings made by users. `http://demos.iwis.cs.aau.dk:8081/WP5/aau/rest/services/getirrelevantterms/{k}/`
- **getRelevantTerms()** - it returns a top-k most recent relevant terms with their frequency of irrelevant ratings made by users. `http://demos.iwis.cs.aau.dk:8081/WP5/aau/rest/services/getrelevantterms/{k}/`

6.2.2 Web Service Consumers

For the matter of integration with other parties that provide information, we also implemented web services clients that consume information. They are:

- **getDocumentList()** - it returns a list of documents extracted from Twitter by WP3;
- **getDocument ()** - it returns a single document by its ID extracted from Twitter by WP3;
- **getSourceList()** - it returns a list of sources from the where the documents are retrieved.

7 Related work

This section reviews and compare a number of related works that contributed to the comparison against our own models. In each related work of this section, we compare technical details, identify differences and outline our improvements over baseline approaches.

7.1 Predicting Events

We use Twitter in our experiments with predicting events trajectory described in Section 4.3.4. Twitter is a social network service categorized as a microblogging platform. These platforms allow their users to share short messages in a simple manner, creating a scenario for fast communication. Twitter in particular allows their users to share messages with, at most, 140

characters. Other users can then opt to subscribe to an user's account and follow more closely the message updates. Users following or being followed by others generate a large social graph on the service.

For example, [29] analyzed the topological characteristics of Twitter, crawling more than 41 million user profiles and 1.47 billion social relations. In the study, the authors show that Twitter deviates from known characteristics of human social networks. In one key aspect, this difference means that information is disseminated faster and to a broader number of users. Additionaly, unlike other services on the web and real-life social networks, messages shared by users on Twitter are public to everyone and can even be accessed by other systems through a public API[13].

This made Twitter an ideal candidate for analyzing on the web how ongoing events unfold. During recent years, a number of studies investigated different aspects of events, including predicting flu trends [1] and typhoons [43]. However, few of these considered the spatial information available to attempt to understand the trajectories of events. By reconstructing the trajectories of an event from the location of user profiles or user messages on Twitter discussing it, it could be possible to antecipate the impact of an event on a certain location.

7.2 Tag Expansion in Recommendation

Tags have been recently studied in the context of recommender systems due to various reasons. Recommendations of relevant documents should be based on the sufficient occurrences for similar signals expressed by tags. We review the related literatures from the perspectives of tag expansion and tag clustering. K. R. Bayyapu and P. Dolog in [5] try to solve the problems of sparse data and low quality of tags from related domains. They suggest using tag neighbors for tag expression expansion. However the tag neighbors are based on the content of documents. We propose another approach to extend the tag set for the user profile by collaborative filtering approach. [56] proposes a collaborative filtering approach TBCF (Tagbased Collaborative Filtering) based on the semantic distance among tags assigned by different users. That is, two users could be considered similar not only if they rated the items similarly, but also if they have similar understanding over these items. To calculate the semantic similarity, the WordNet dictionary is being accessed to find the shortest path connecting a tag and its synonym in the graph synsets. In [7], an interesting approach was proposed to model the documents in social tagging systems as a document graph. The relevance of tag propagated along edges of the documents graph is determined via a scoring scheme, with which the tag prediction was carried out. Heymann et al. [22] addressed the same problem of tag prediction based on the anchor text, web page content and the surrounding hosts. A binary classifier was trained on a set of very popular bookmarks to differentiate the closest tags. [17, 47] demonstrated how tag clusters serving as coherent topics can aid in the social recommendation of search and navigation. In [21] topic relevant partitions are created by clustering documents rather than tags. By clustering of documents, it improves recommendation by distinguishing between alternative meanings of a query. While in [8], clusters of documents are shown to improve recommendation by categorizing the documents into topic domains.

[13] https://dev.twitter.com/

7.3 Personalized Tag cloud

Sinclair et al. [48] considers tag clouds as useful retrieval interface when a user's searching task is not specific. Users by exploration of tag clouds get familiar with a domain of the system. According to [41] a tag cloud allows users to perform 4 different tasks: i) search - retrieving matching content to the selected term in the tag cloud, ii) browsing - user can browse available documents, not necessarily to search for some particular topic or task, iii) impression formation - user gains an impression which topics are dominant for the documents associated with the tag cloud and iv) recognition - user can recognize which of different documents a tag cloud is more likely to visualize. This retrieval interface also supports a query refinement during the search task as by addition or deletion of tags a placed search query changes. This strategy was considered our tag neighbor expansion of user queries as shown in Section 4.3.3.

In medical surveillance systems there is a need to assess a huge number of documents in a short time. A surveillance personnel has to search, browse and create an impression from the explored documents. Therefore, a tag cloud is suitable retrieval interface which can improve a validation process of documents. The majority of tag clouds visualize tags in alphabetical order however, there is a lot of ongoing research about depicting tags in some semantical manners. [12, 20] propose to group semantically related tags and depict them in a tag cloud near by with similar color. Such approach provides better orientation in the tag cloud as related tags can be easier identified by users. Tags are clustered based on their co-occurrences. Similarly, the proposed tag cloud generation method also groups semantically similar tags. However, our approach is more robust as syntactically similar tags are firstly pre-clustered (grouping singular, plural or misspellings of tags) similarly as proposed in [51]. It leads to tag space reduction as resulted tag cloud does not contain syntactically similar tags. In the second phase, tags are similarly clustered based on tags co-occurrences but our proposed approach also considers retrieved semantical distances from WordNet dictionary if available. A folksonomy contains a huge number of tags therefore there is a need to select only the most important that will be depicted in the tag cloud. [53] proposes different tags selection algorithms and our method utilizes a similar selection technique where tags with higher coverage are preferred. However, our method selects tags with higher coverage from different clusters. It results into a semantically more diverse tags thus a user can explore and browse more topics from the generated tag cloud.

In comparison to all related work the proposed method can generate parameterized tag clouds based on different parameters such as location, time, signal which allows to restrict set of considered documents and in consequence also tags. Such Moreover, a tag cloud can be personalized if user profile is available.

7.4 Spatial Reasoning

Spatial reasoning has been constantly studied in the literature in applications there orientation is the primary value driver. Spatial reasoning encompasses of two main abilities: the ability of calling up images in mind and the ability to reason with these images. Spatial Reasoning typically applies to Geographic Information Systems (GIS) and often assumes that exact coordinates are known and inferences can be carried out to make a decision. In computer science, lots of effort have been put on *collision detection* to tell whether two objects occupy the same space at the same time and *path planning* to plan the best trajectories so that two objects avoid collisions [15, 28].

Our location-based method (see Section 4.4.3) presents the model for user location preferences, which is built from the set of user locations defined in his or her signal definitions and from similar locations. In particular, the location similarity is computed as a function of two factors such as *political hierarchy* (e.g., city, state or country level) and *distance and population*. To the best of our knowledge, almost no report in the literature performs spatial reasoning considering the factors addressed in our model. For this reason, we will discuss a number of techniques and compare them without our approach at the technical level regardless the medical domain. [16] present a new approach to represent qualitative spatial knowledge and to spatial reasoning. This cognitive considerations motivates the approach and is based on relative orientation information about spatial environments. The approach aims at exploiting properties of physical space which surface when the spatial knowledge is structured according to conceptual neighborhood of spatial relations. In our method, the notion of conceptual neighborhood is also considered once the outbreak warning are target to cities nearby the outbreak focus. In line with our method [16] proposes a formal method for qualitative reasoning about distances and cardinal directions in geographic space. The main problem addressed is how to infer the distance and direction from point two points in a space. Our method differs from [16] in the sense that we look into the neighborhood location by considering additional factors to determine the minimal distance between two points.

The exposure of user data in social web applications has attracted attention of researchers that try to estimate the location of web content or people on the web based on the analysis of geo-related terms. Some applications focused on extracting explicitly geographic information from web pages such as address or points of interest [2, 30]. [9] propose and evaluate a probabilistic framework for estimating a Twitter user's city-level location based purely on the content of the user's tweets, even in the absence of any other geospatial cues. The framework relies on three key aspects i) tweet content, ii) a classification component for automatically identifying words in tweets with a strong local geo-scope; and iii) a lattice-based neighborhood smoothing model for refining a user's location estimate. The framework aims at estimating k possible locations for each user in descending order of confidence. Our model converges with [9]' study in the sense that we also explore Tweeter data for predicting neighborhood locations, however we are not focused on finding people. In addition our method. Crandall et al. [10] investigate how to organize a large collection of geotagged photos. They try to combine textual and visual features to place images on a map. They have restrictions in their task that their system focuses on which of ten landmarks in a given city is the scope of an image. As to methods that support social surveillance, [44] investigates the detection of earthquakes with real-time Twitter data. In order to make such a prediction, they make use of location information for tracking the how of information across time and space. For predicting earthquake, their algorithm needs to maintain a knowledge base of where and when the earthquake is reported. Our work also location-based method also relies on a knowledge base but a medical one.

8 Conclusion and Future Works

In this report we presented the final version of WP5 personalization component implemented for the M-Eco project addressing diverse aspects such as motivation scenarios, formal models, design and the requirements necessary to install and realize the benefits of personalization in M-Eco.

Recommendation and navigation aspects of WP5 component are improved according to the

feedback from M-eco users and reviewers. The new version of GUI provides improved signals assesments, the tag cloud interface allows users to remove or annotate documents with (ir) relevant tags. These users's feedback is utilized for more precise personalization proces.Moreover, the feedback can be utilized by other partners in order to improve signal generation and data collection algorithms.

A future work is influenced by the feedback from the conducted evaluations. The tag cloud model should provide an active learning of user's preferences in order to adjust structure of tag cloud according to user's needs. We plan to focus on a selection algorithms for tag cloud generation in order to distinguish irrelevant, noisy terms from relevant tags.

References

[1] H. Achrekar, A. Gandhe, R. Lazarus, S. Yu, and B. Liu. Predicting flu trends using twitter data. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 702–707. IEEE, Apr. 2011.

[2] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-where: geotagging web content. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 273–280. ACM, 2004.

[3] G. Backfried, E. Diaz-Aviles, P. Dolog, J. Dreesman, F. Durao, T. Eckmanns, G. Kirchner, M. Krieck, R. G. Lage, M. Leginus, L. Otrusina, A. Stewart, and E. Velasco. D2.2 evaluation report. Technical report, August 2011. EU FP7 Small or Medium-scale Focused Research Projects (STREP).

[4] B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, O. Chapelle, and K. Weinberger. Learning to rank with (a lot of) word features. *Information retrieval*, 13(3):291–314, 2010.

[5] K. R. Bayyapu and P. Dolog. Tag and Neighbour Based Recommender System for Medical Events. In *Proceedings of MEDEX 2010: The First International Workshop on Web Science and Information Exchange in the Medical Web co-located with WWW 2010 conference*, 2010.

[6] P. Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11:87–110, March 2001.

[7] A. Budura, S. Michel, P. Cudré-Mauroux, and K. Aberer. Neighborhood-based tag prediction. *The Semantic Web: Research and Applications*, pages 608–622, 2009.

[8] H. Chen and S. Dumais. Bringing order to the web: automatically categorizing search results. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 145–152, New York, NY, USA, 2000. ACM.

[9] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 759–768, New York, NY, USA, 2010. ACM.

[10] D. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. In *Proceedings of the 18th international conference on World wide web*, pages 761–770. ACM, 2009.

[11] P. Dolog, F. Durao, R. Lage, M. Leginus, and R. Pan. M-eco enhanced adaptation service. Technical report, December 2011. EU FP7 Small or Medium-scale Focused Research Projects.

[12] F. Durao, P. Dolog, M. Leginus, and R. Lage. Simspectrum: A similarity based spectral clustering approach to generate a tag clound. In *Proceedings of (accepted for) the 11th International Conference on Web Engineering*, 2011.

[13] F. Echarte, J. Astrain, A. Córdoba, and J. Villadangos. Pattern matching techniques to identify syntactic variations of tags in folksonomies. *Emerging Technologies and Information Systems for the Knowledge Society*, pages 557–564, 2008.

[14] R. Farmer. Ratings bias effects. Available at: `http://buildingreputation.com/writings/2009/08/ratings_bias_effects.html`. Accessed on June 22, 2012.

[15] K. Forbus, J. Mahoney, and K. Dill. How qualitative spatial reasoning can improve strategy game ais. *Intelligent Systems, IEEE*, 17(4):25–30, 2002.

[16] C. Freksa. Using orientation information for qualitative spatial reasoning. *Theories and methods of spatio-temporal reasoning in geographic space*, pages 162–178, 1992.

[17] J. Gemmell, A. Shepitsen, M. Mobasher, and R. Burke. Personalization in folksonomies based on tag clustering. In *Proceedings of the 6th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*, July 2008.

[18] D. Goldberg. Genetic algorithms in search, optimization, and machine learning. 1989.

[19] Y. Hassan-Montero and V. Herrero-Solana. Improving tag-clouds as visual information retrieval interfaces. In *International Conference on Multidisciplinary Information Sciences and Technologies*, pages 25–28. Citeseer, 2006.

[20] Y. Hassan-Montero and V. Herrero-Solana. Improving tag-clouds as visual information retrieval interfaces. In *MERIDA INSCIT2006 CONFERENCE*, 2006.

[21] C. Hayes and P. Avesani. Using tags and clustering to identify topic-relevant blogs. In *International Conference on Weblogs and Social Media*, March 2007.

[22] P. Heymann, D. Ramage, and H. Garcia-Molina. Social tag prediction. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 531–538, New York, NY, USA, 2008. ACM.

[23] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In Y. Sure and J. Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, chapter 31, pages 411–426. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[24] A. Huang. Similarity measures for text document clustering. In *Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand*, pages 49–56, 2008.

[25] S. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[26] M. Kipp and D. Campbell. Patterns and inconsistencies in collaborative tagging systems: An examination of tagging practices. *Proceedings of the American Society for Information Science and Technology*, 43(1):1–18, 2006.

[27] Knowledge and U. o. K. Data Engineering Group. Benchmark folksonomy data from bibsonomy, version of january 1st, 2010.

[28] L. Kotovsky and R. Baillargeon. The development of calibration-based reasoning about collision events in young infants. *Cognition*, 67(3):311–351, 1998.

[29] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM.

[30] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: social honeypots+ machine learning. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 435–442. ACM, 2010.

[31] M. Leginus, P. Dolog, and V. Zemaitis. Improving tensor based recommenders with clustering. In *Proceedings of (accepted for) the 20th conference on User Modeling, Adaptation, and Personalization UMAP 2012*, 2012.

[32] M. Leginus and V. Zemaitis. Speeding up tensor based recommenders with clustered tag space and improving quality of recommendations with non-negative tensor factorization. Master's thesis, Aalborg University, 2011.

[33] G. Marketos, E. Frentzos, I. Ntoutsi, N. Pelekis, A. Raffaetà, and Y. Theodoridis. Building real-world trajectory warehouses. In *Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*, MobiDE '08, pages 8–15, New York, NY, USA, 2008. ACM.

[34] L. Martin, D. Peter, L. Ricardo, and D. Frederico. Methodologies for improved tag cloud generation with clustering. In *Proceedings of the 12th International Conference on Web Engineering*. Springer, 2012.

[35] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Wherenext: a location predictor on trajectory pattern mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 637–646, New York, NY, USA, 2009. ACM.

[36] S. Muthukrishnan. *Data streams: Algorithms and applications*. Now Publishers Inc, 2005.

[37] L. Otrusina, P. Smrz, and G. Backfried. Speech recognition and content classification subsystems. Technical report, December 2010. EU FP7 Small or Medium-scale Focused Research Projects (STREP).

[38] S. Rajaraman. Five stars dominate ratings. Available at: `http://youtube-global.blogspot.fr/2009/09/five-stars-dominate-ratings.html`. Accessed on June 22, 2012.

[39] D. Ramage, P. Heymann, C. Manning, and H. Garcia-Molina. Clustering the tagged web. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 54–63. ACM, 2009.

[40] S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD conference on Knowledge discovery and data mining*. ACM, 2009.

[41] A. W. Rivadeneira, D. M. Gruen, M. J. Muller, and D. R. Millen. Getting our head in the clouds: toward evaluation studies of tagclouds. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07, pages 995–998, New York, NY, USA, 2007. ACM.

[42] G. X. Rong Pan and P. Dolog. Improving recommendations in tag-based systems with spectral clustering of tag neighbors. In *The 3rd FTRA International Conference on Computer Science and its Applications (CSA-11)*, 2011.

[43] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 851–860, Raleigh, North Carolina, USA, 2010. ACM. ACM ID: 1772777.

[44] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

[45] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 285–295, New York, NY, USA, 2001. ACM.

[46] H. Shang and T. H. Merrettal. Tries for approximate string matching. *IEEE Transactions on Knowledge and Data Engineering*, 8(4):540–547, Aug. 1996.

[47] A. Shepitsen, J. Gemmell, B. Mobasher, and R. Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 259–266, New York, NY, USA, 2008. ACM.

[48] J. Sinclair and M. Cardew-Hall. The folksonomy tag cloud: when is it useful? *J. Inf. Sci.*, 34:15–29, February 2008.

[49] A. Stewart, E. Diaz-Aviles, N. Kanhabua, and M. Fisichella. Deliverable D4.3: M-Eco Event Detection and Analysis. Technical report, L3S Research Center / Leibniz University of Hannover (LUH). Project: Medical Ecosystem - Personalized Event-Based Surveillance (M-Eco). Project Number: 247829. EU FP7 Small or Medium-scale Focused Research Projects (STREP)., 2012.

[50] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis. *IEEE Transactions on Knowledge and Data Engineering*, 2009.

[51] J. van Dam, D. Vandic, F. Hogenboom, and F. Frasincar. Searching and browsing tag spaces using the semantic tag clustering search framework. In *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*, pages 436–439. IEEE, 2010.

[52] P. Venetis, G. Koutrika, and H. Garcia-Molina. On the selection of tags for tag clouds. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 835–844, 2011.

[53] P. Venetis, G. Koutrika, and H. Garcia-Molina. On the selection of tags for tag clouds. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 835–844, New York, NY, USA, 2011. ACM.

[54] W. Wahlster and A. Kobsa. User models in dialog systems. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 4–34. Springer, Berlin, Heidelberg, 1989.

[55] G. Xu, K. Bayyapu, R. Pan, R. Lage, and P. Dolog. Personalization and adaptation component for the m-eco system. Technical report, December 2010. EU FP7 Small or Medium-scale Focused Research Projects (STREP).

[56] S. Zhao, N. Du, A. Nauerz, X. Zhang, Q. Yuan, and R. Fu. Improved recommendation based on collaborative tagging behaviors. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 413–416. ACM, 2008.