

## Research Article

# Two-Level Automatic Adaptation of a Distributed User Profile for Personalized News Content Delivery

**Maria Papadogiorgaki,<sup>1</sup> Vasileios Papastathis,<sup>1</sup> Evangelia Nidelkou,<sup>1</sup> Simon Waddington,<sup>2</sup> Ben Bratu,<sup>3</sup> Myriam Ribiere,<sup>3</sup> and Ioannis Kompatsiaris<sup>1</sup>**

<sup>1</sup> Centre for Research and Technology Hellas (CERTH), Informatics and Telematics Institute (ITI),  
1st Km Thermi-Panorama Road, Thessaloniki 57001, Greece

<sup>2</sup> Motorola Labs, Motorola Ltd., Jays Close, Viabes Industrial Estate, Basingstoke, Hampshire, RG22 4PD, UK

<sup>3</sup> Motorola Labs, Parc Les Algorithmes, Saint Aubin, 91193 Gif sur Yvette Cedex, France

Correspondence should be addressed to Ioannis Kompatsiaris, [ikom@iti.gr](mailto:ikom@iti.gr)

Received 28 February 2008; Accepted 24 June 2008

Recommended by Harald Kosch

This paper presents a distributed client-server architecture for the personalized delivery of textual news content to mobile users. The user profile consists of two separate models, that is, the long-term interests are stored in a skeleton profile on the server and the short-term interests in a detailed profile in the handset. The user profile enables a high-level filtering of available news content on the server, followed by matching of detailed user preferences in the handset. The highest rated items are recommended to the user, by employing an efficient ranking process. The paper focuses on a two-level learning process, which is employed on the client side in order to automatically update both user profile models. It involves the use of machine learning algorithms applied to the implicit and explicit user feedback. The system's learning performance has been systematically evaluated based on data collected from regular system users.

Copyright © 2008 Maria Papadogiorgaki et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. INTRODUCTION

The increasing popularity of mobile devices, such as laptops, mobile phones and personal digital assistants, and the advances in wireless networking technologies allow information to be accessed almost anywhere, at any time. As part of this trend, several personalized news services are emerging, such as systems that enable the distribution and delivery of news content to the individual users, from heterogeneous networks of devices. These environments raise challenging problems for the development of personalization applications. These problems concern the requirements of matching the user preferences while preserving privacy issues, as well as being aware of the limitations, for example, in network traffic.

The focus of this paper is to cover the personalization requirements of mobile users in the news domain, taking into account the user's personal preferences and interests but also attempting to preserve the privacy of the user preferences. To this aim, our system architecture performs a management of

a distributed user profile across client and server. The high-level user preferences reflecting the long-term user interests are stored in a skeleton (high-level) profile, which is managed by the server, while the low-level preferences representing the short-term user interests are stored in a detailed (low-level) profile in the handset. This distribution enables a two-level matching process between the user profile and the news content, which uses semantic metadata extracted from the textual content and aims at the same time at a minimal computational and communication cost. Thus the available content is initially filtered on the server to derive a list of recommended items in all preferred categories, while the matching of detailed user preferences in the handset results in the displaying of the items in a ranked order.

Apart from the distributed architecture, the novelty in the proposed approach lies in the fact that the distributed user profile on both sides is automatically updated by means of machine learning processes, which are performed in the handset and by exploiting both explicit and implicit user feedback. In addition, the paper emphasizes the exploitation

of named entities in the learning process. The motivation behind the automatic adaptation of the user profile is that the latter should be consistent with the user interaction, that is, it should follow the long/short-term changes of the user interests. For instance, assume that a user has denoted several categories as topics of high interest such as “Markets” and “Politics,” in order to receive interesting news items. If the user demonstrates stronger preference for the topic “Markets” and more specifically for the subtopic “Equity Markets” through her interaction with the system during a short time period, the news items from “Equity Markets” will be displayed higher than the other news items in the incoming list. On the other hand, if the user constantly selects to read news from another topic, which had been initially denoted with a low degree of preference, for example, “Society,” during a long time period, then the degree of preference of this topic will be automatically increased in order to receive more “Society” news items.

This paper is organized as follows. In Section 2, the server- and client-side components of our system architecture are briefly described. Following this, in Section 3, the distribution of the user modeling is presented along with the user profile initialization process. In Section 4, the semantic annotation of the incoming news items on the server is presented. The two-level matching processes, that is, the initial content filtering performed on the server and the low-level matching in the handset, are described in detail in Section 5. The short-term and long-term learning algorithms for the automatic adaptation of the user profile are presented in Section 6 and evaluated in Section 7. In Section 8, related work addressing issues raised in this paper is reported. Finally, in Section 9, conclusions regarding the proposed system architecture and the learning processes are drawn.

## 2. SYSTEM ARCHITECTURE

In this section, we present the distributed system architecture across the server and the client. A general diagram of the distributed system architecture is depicted in Figure 1, while the main server and client side components of the system are illustrated in detail in Figure 2. These components and the related processes are described below.

### 2.1. Server-side components

The incoming to the server news items, which in our application are articles that typically include a headline and a short abstract, is first stored in the content repository. Next, they are analyzed by the metadata generation module in order to be semantically annotated with the appropriate metadata. High-level semantic information about the content typically consists of relevant topic categories. Thus, the key step in the metadata extraction is the server-side classification of each news item according to a hierarchical news taxonomy. Low-level information comprises specific terms such as nouns and Named Entities and associated weights. Hence, the semantic annotation of each news item concerns its classification to a topic, as well as the extraction of the topic-related low-level

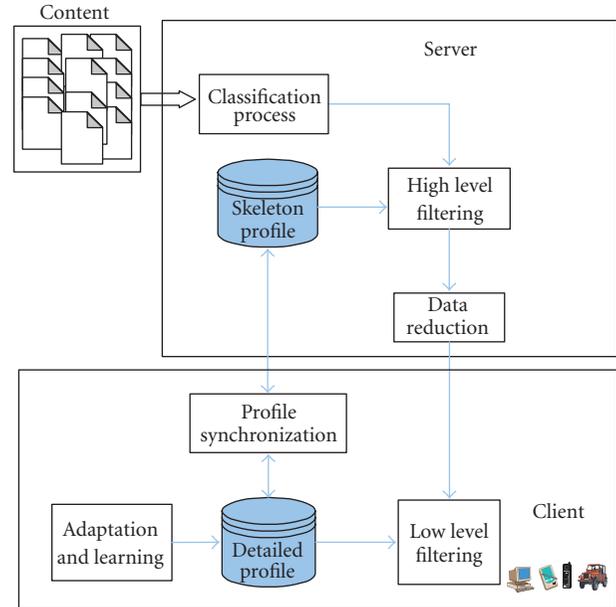


FIGURE 1: Distributed system architecture.

terms, that is, nouns and Named Entities contained in this item.

Following the classification process, a metadata reduction process is applied for each article, aiming to identify the most significant nouns according to the classification topic. Additionally, Named Entities are identified in each article and reduced by using a constantly evolving knowledge base. The reduction process is applied for both nouns and Named Entities of each news item, in order to significantly reduce the unnecessary metadata before their transmission to the client side, since contextual constraints exist, such as network overloading.

The news items are initially filtered on the server, based on the high-level general interests of the users stored on both the server and the handset. The high-level filtering algorithm matches entries in the skeleton user profile (long-term user interests) to the high-level metadata (e.g., topic categories, preferred sources of content). Thus, an initial set of recommended items for delivery to the user is computed, which will be subject to a further filtering step on the client. That set of news items is then transmitted to the client along with the corresponding final reduced metadata.

### 2.2. Client-side components

The main objective of client-side filtering apart from preserving the privacy of the user preferences (detailed profile) is to reduce the loading on the server infrastructure. The reduced metadata of each article are matched against the detailed user profile (short-term user interests) in the handset, in order to display the news items on the user screen of the mobile device with the appropriate ranking order (i.e., taking into account that the most interesting items should be displayed at the top of the screen).

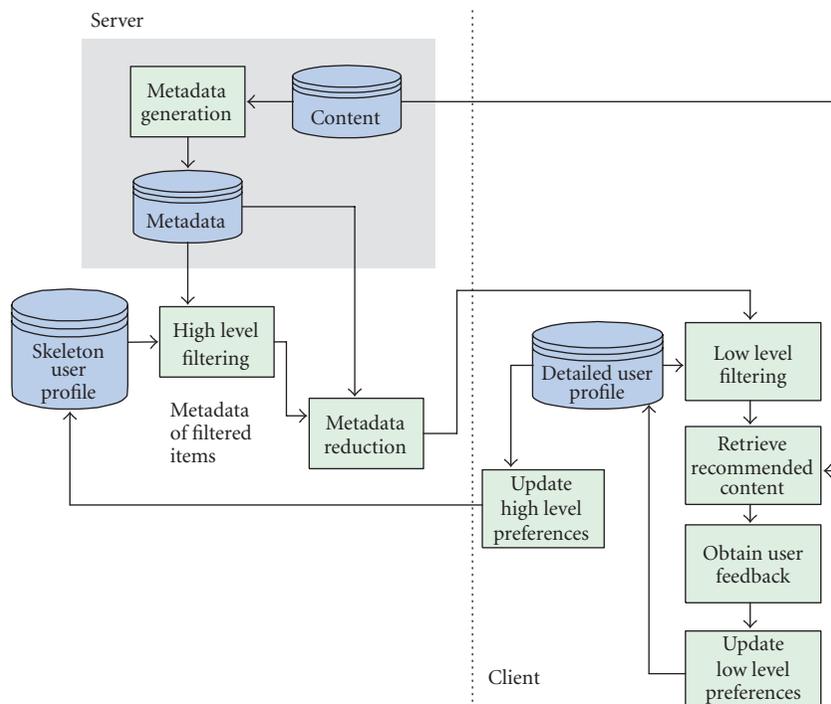


FIGURE 2: Diagram of client-server interactions.

The automatic adaptation of the distributed user profile concerns a two-level learning process, which is performed in the handset. More specifically, as far as the detailed user profile is concerned, usage tracking monitors the user interactions with content items and employs a short-term learning process to add significant terms (i.e., nouns and Named Entities) to the user profile. The usage tracking includes information on which items were read, what is the proportion of the length of a read item and the time spent on it. The updates of the high-level user profile by the long-term learning process are also employed in the handset, according to the classification topic of each read and nonread news item contained in a long time period set. The high-level profile can also be explicitly adapted by the user through an appropriate user interface at any time of the automatic learning process. Finally the high-level user preferences are transmitted to the server to, respectively, update the skeleton profile, whenever an adaptation takes place.

### 3. USER PROFILE MODELING

The main objective of the user profile modeling is to allow for the distributed semantic matching process that takes place both on the server and in the handset. This is enabled by the distribution of the user profile across client and server [1]. More specifically, the long-term user preferences are stored in a high-level (skeleton) profile on the server, and the short-term preferences in a low-level (detailed) profile in the handset.

The high-level user preferences refer to the broad news topics or categories that the users are interested in (e.g., sports, politics, business, etc.). They express the long-term

interests of the user that are likely to remain the same through a long time period and they are not subject to abrupt changes [2, 3]. In contrast, the low-level user preferences refer to the more detailed aspects of the news and represent the short-term user interests, which are subject to abrupt changes, depending on the daily news.

#### 3.1. Initialization of the high-level profile

Several news web sites organize their content according to taxonomies, for example, Yahoo News <http://news.yahoo.com/rss>. In our system, the high-level (skeleton) user profile is represented as a three-level hierarchy of topics, which corresponds to our sample categorization of the news domain and is illustrated in Figure 3. The hierarchy was defined to be unambiguous and intuitive for users, through which they can reliably identify which category a news item falls under. However, it is not an exhaustive news topics hierarchy, but it contains only a subset of news domain topics, for purposes of fast prototyping the approach presented in this paper. The hierarchy consists only of three levels, in order to reduce the user overload when she tries to explicitly fill in her personal profile. Most of the topics correspond to the categories defined in the Reuters Corpus Volume 1 (RCV1) comprising news texts produced in 1996-1997, while the other topics were added for the completion of the hierarchy. RCV1 was chosen, since it provides a wide and adequate news categorization and additionally it was freely available in XML format, being appropriate for developing and evaluation purposes. The hierarchy consists of four general topics, namely, the “Business and Finance,” the “Lifestyle,” the “Government/Social” and the “World Crises.” Each of the

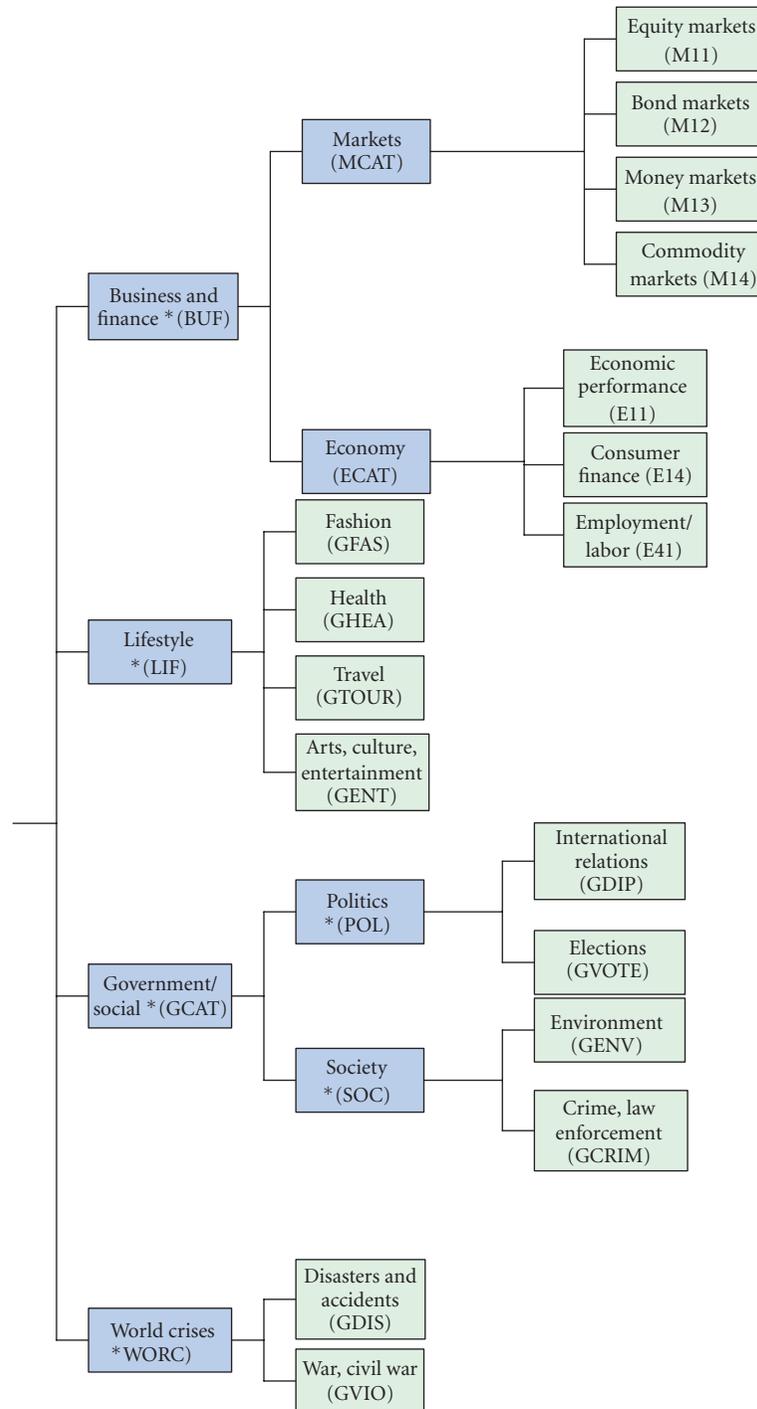


FIGURE 3: Hierarchy of topics. The topics, which do not belong to the Reuters corpus are marked with “\*.”

above first-level topics consists of subtopics, down to the last level. Hence the hierarchy consists of four trees each of which starts from a general (first-level) topic and ends to the last level (second- or third-level) topics, namely, the leaf topics.

The server transmits the three-level hierarchy to the client in order to be presented in the handset. The initialization of the high-level user profile is performed on the mobile device where the user should explicitly express her

degree of interest (i.e., high, medium, low, or none) for each particular topic based on the described hierarchy of topics. An appropriate user interface, which allows the user to browse the topics and denote her preference has been developed and is presented in Figure 4.

The initial (default) degree of interest for all the topics is “Medium” and the user may denote her preference either for an individual leaf topic, or for a higher-level topic. In the second case, all the subtopics down to the last level of

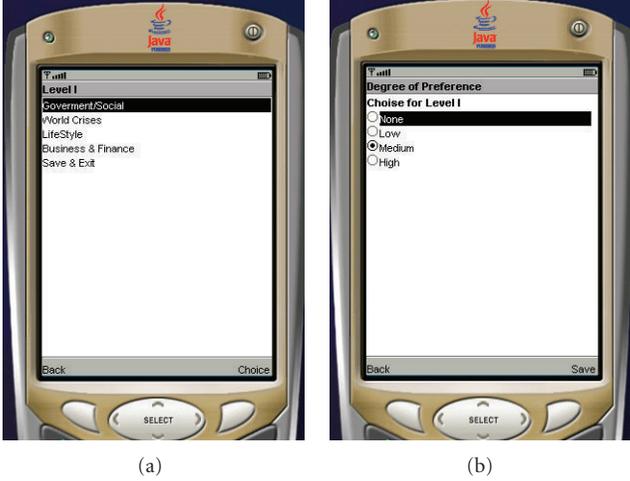


FIGURE 4: Snapshots from handset’s screens: (a) the user is allowed to choose a preferred topic at any level of hierarchy: using the scroll up/down arrows she can move to the several topics of the list, while with the “Select” button the user is able to navigate the respective next level of the hierarchy, that is, the list of the subtopics, which correspond to the selected topic. (b) This screen can be viewed through the “Choice” button of screen (a) for a marked topic. The user is allowed to denote a degree of preference for that topic using the “Select” button.

the hierarchy inherit the same degree of preference. If the user do not explicitly denote a specific degree of preference for one or more topics the default (“Medium”) degree is kept for the corresponding topics. The symbolic degrees of preferences are converted to numerical values in the 0-1 scale according to almost uniform intervals (Table 1) and are stored locally in the handset (in the corresponding vector) for all the leaf topics. Regarding the definition of the numerical values of Table 1, the “Medium” degree of preference was initially considered, to be equal to the middle weight value of the interval  $[0, 1]$ , that is, 0.5, and the “None” degree equal to 0. Then the “Low” and “High” degrees were approximately set, while their correspondence with the numerical values and the intervals of this table was validated through long-term learning experimental evaluation, described in Section 7.2.

The high-level user profile consists of the following vectors related to the leaf topics.

- (i) A Leaf Topics Vector  $\overrightarrow{LTopic}$ , which contains all the leaf topics.
- (ii) A Topic Weights Vector  $\overrightarrow{WLTTopic}$  containing the corresponding weights of the leaf topics.

The initialized high-level user profile (i.e., the Leaf Topics Vector  $\overrightarrow{LTopic}$  along with a vector containing the corresponding symbolic degrees of preference) is transmitted to the server and stored in a users’ database in order to formulate the skeleton user profile. It will then allow for the semantic matching process described in detail in Section 5.1. It is noted that only the Leaf Topics Vector is transmitted,

TABLE 1: Degrees of preference and the corresponding numerical values.

Degree of preference	Numerical value	Numerical interval
None	0	0
Low	0.3	(0, 0.3]
Medium	0.5	(0.3, 0.7)
High	0.7	[0.7, 1]

since as will be described in Section 4.2.1 the news items are finally classified only to leaf topics.

Additionally when the user browses the nonleaf hierarchical topics a bottom-up propagation of the degrees of preference is performed. More specifically, the adaptation of the degree of preference for all the nonleaf topics is determined from the average preference (numerical) value of their corresponding subtopics (as described in Section 6.2.3). This average value is quantized according to the existing symbolic degrees of preference as presented in Table 1.

Finally, the user is allowed to explicitly adapt her high-level profile whenever she wills, through the above-described user interface of the mobile device and following the same steps presented for the initialization of the profile.

### 3.2. Initialization of the detailed (low-level) profile

The detailed user profile, which is implicitly initialized and adapted by the personalization system, consists of textual low-level features, that is, nouns and Named Entities, that play a key role in the news personalization domain, as they capture a major part of the semantics in a news item. More specifically, the detailed user profile consists of the following vectors related to the nouns and Named Entities (Figure 5):

- (i) A Terms Vector  $\overrightarrow{T}$ , which contains the nouns and the Named Entities.
- (ii) A Weights Vector  $\overrightarrow{W}$  containing the corresponding weights.
- (iii) A Usage History Vector  $\overrightarrow{UH}$  containing the corresponding usage history of each term, which is a counter of how many times the term has been selected (integer number).
- (iv) A Terms Type Vector  $\overrightarrow{TT}$ , which contains for each term a character which expresses its corresponding type, that is, “N” for “Noun” is assigned to each noun and “P” for “Person,” “O” for “Organization” and “L” for “Location” are assigned to Named Entities (see example in Figure 5).

The initialization of the detailed user profile refers to a transitive period starting the first time the user interacts with the personalization system until the vector of low-level terms is sufficiently large according to a defined maximum number of terms (100 terms were used in our experiments). During this period, all extracted low-level features contained in the news items selected by the user will be inserted in the Terms

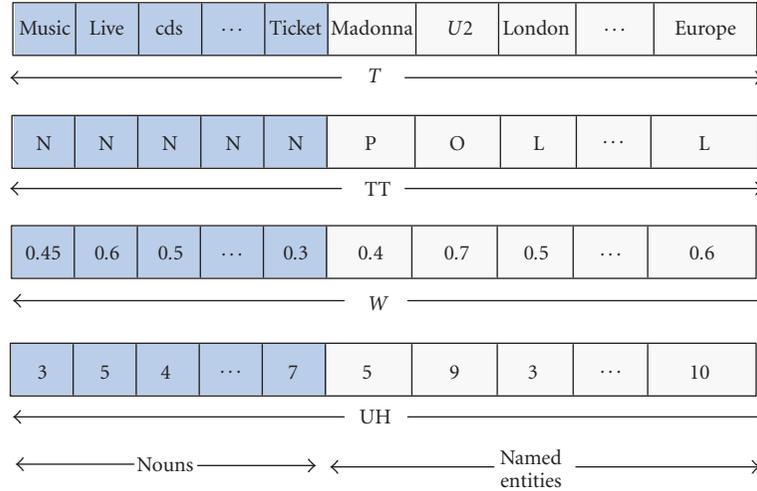


FIGURE 5: Example of detailed user profile vectors.

Vector  $\vec{T}$  of the detailed user profile, having an initial weight equal to 0.5 (corresponding to a “Medium” preference), in the Weights Vector  $\vec{W}$ . Additionally, the values contained in the Usage History Vector  $\vec{UH}$  for both nouns and Named Entities are updated throughout the initialization process according to the number of news items that the user selects.

In addition, during the initialization process, an important characteristic of the personalization system is initialized, which is the user behavior determined by the reading rate, that is, the number of news items selected per day. This is used in the proposed mathematical formulas (3) and (4), of Section 6.1.1, aiming to adjust the weights of the terms.

#### 4. SEMANTIC ANNOTATION OF NEWS CONTENT

In order to apply the personalization system based on the distributed user profile modeling described in the previous section, the news content should be semantically annotated with the appropriate metadata. More specifically, the analysis of the textual news content results in the extraction of the leaf topic to which each news item is classified and also of the significant (according to the classification topic) low-level terms, that is, nouns and Named Entities contained in this item. The semantic annotation of the news items takes place on the server side and the extracted metadata are transmitted to the handset following a reduction process that takes into account the communication and computational costs.

##### 4.1. Construction of training sets

Both the classification and the metadata reduction processes, described in Sections 4.2 and 4.3, respectively, require a training stage, which involves the use of the Reuters Corpus Volume 1 (RCV1) to provide the articles that can be used for training.

The training sets used for the semantic representation of each leaf topic in the proposed approach are generated according to the “Same Subtree” filtering. More specifically,

the news item is included in the training set of a leaf topic, only if it is categorized from Reuters, apart from the particular leaf topic, to sibling leaf topics (only if they belong to the third-level), or to its ancestors up to the first level. For example, a news item which according to Reuters is categorized to the topics MCAT (Markets), M11 (Equity Markets), and M13 (Money Markets), can be included in the training set of the leaf topic M11, while on the contrary, it is excluded from that training set if it is categorized to the topics MCAT (Markets), M11 (Equity Markets), E14 (Consumer Finance). The training set of each non-leaf hierarchical topic is constructed using the training sets of their subtopics. This criterion is applied for the semantic representation of the topics for the classification purpose.

The training sets, which are used for the data reduction process are constructed according to the “Only One Leaf Topic” filtering, that is, the news item is included in the training set of a leaf topic, if it is categorized from Reuters, only to this particular leaf topic and to any other higher-level topic. Since the data reduction process involves only the leaf topics’ representation, there is no training set construction for the non-leaf hierarchical Topics.

#### 4.2. Metadata generation

##### 4.2.1. News item classification according to topic category

The current classification method is based on machine learning along with vector representation techniques and uses the fixed hierarchical taxonomy of content categories, presented in Section 3.1. The automatic framework for the text classification is composed of an offline training process and an online classification process. An overview of the overall topic classification process is shown in Figure 6.

The textual analysis process includes the typical NLP preprocessing steps (i.e., tokenization, sentence splitting, parts of speech tagging, stemming), which are performed with the use of the appropriate GATE components [4, 5] and the text search engine Lucene <http://lucene.apache.org/>.

A statistic text analysis follows according to the vector space model, where each news article is represented as a vector of feature-value pairs. The features used are the extracted nouns in the text and the values are the corresponding weights based on their frequency of appearance in the text (term frequency—TF). Accordingly, each topic in the hierarchy is also represented as a feature-value vector that best expresses the semantics of this topic. The features are the semantically significant nouns while the values correspond to the term frequency-inverse document frequency (TF-IDF) weights [6]. This is referred to as the Topic Prototype Vector, statistically constructed from the appropriate training set.

The construction of the Topic Prototype Vectors is based on the relevance feedback algorithm originally proposed by Rocchio [7] for the vector space model. A generalization of the Rocchio algorithm that can be used for text categorization with more than two categories has been proposed by [8]. The Topic Prototype Vectors in our approach have been constructed using the Rocchio formula along with the topics hierarchy and the training sets generated according to the “Same Subtree” criterion described in Section 4.1. Thus, for each topic in the formula, as relevant articles (positive terms) are taken into account the training articles belonging to that topic, whereas as nonrelevant articles (negative terms) are considered the training articles belonging to all subtrees apart from the subtree of the topic.

Following the offline construction of the Topic Prototype Vectors, the online classification process depicted in Figure 6 results in the categorization of the news item in only one leaf topic. It involves the assignment of each incoming news article to the leaf topic with the shortest distance between the Topic Prototype Vector and the article’s vector of feature-value pairs. Thus the news item is classified to the leaf topic with which its noun terms vector has the highest cosine similarity, given by the following formula:

$$\text{Sim}(PV, I) = \frac{\overrightarrow{PV} * \overrightarrow{I}}{|\overrightarrow{PV}| \cdot |\overrightarrow{I}|}, \quad (1)$$

where  $\overrightarrow{PV}$  is the vector of the TF-IDF weights of the Topic Prototype Vector, while  $\overrightarrow{I}$  is the vector of TF weights of the noun terms extracted from the news item.

#### 4.2.2. Extraction of low-level metadata

The extracted low-level features include only the common nouns between those which were initially identified in the news item with the aid of GATE (the type of noun “N” was assigned to each of them) and the Prototype Vector of the topic category where it was classified. Along with these nouns their TF weights in the news item are extracted.

The analysis goes further by identifying the Named Entities contained in the news item, as well as their corresponding type (such as person, organization, location) both using GATE software. According to this, an additional term-frequency vector of Named Entities is generated for each content item in the online process. Furthermore, an enhanced semantic identification process for Named Entities is performed, as described in detail in Section 4.3.2.

### 4.3. Metadata reduction

Following the extraction of metadata, the next stage concerns their transmission to the client side in order to be used for the low-level filtering and learning processes. However, it is sensible to significantly reduce those metadata before the transmission, due to contextual constraints, such as the network overloading as well as, rarely in nowadays, limited memory space and processing capability in the client device. Thus, the unnecessary terms are eliminated for reducing the communication cost along with the computational cost in the handset as much as possible, aiming to allow the personalization process to be the most efficiently performed.

#### 4.3.1. Reduction of nouns using adapted TF-IDF method

The reduction of the noun terms is made based on the presumption that after different incoming documents are classified in a given topic, the differentiation between them can be made using only a subset of the extracted metadata, which were described in Section 4.2.2.

The reduction of nouns involves an offline training process aiming at a representation of the leaf topics, which is different from the ones used in the classification process in Section 4.2.1. During this stage, only the corpus of articles pre-classified in the same leaf topic is used in the training sets for each leaf topic, that is, the training set used has been constructed according to the “Only One-Leaf-Topic” criterion, described in Section 4.1. Hence, the above-mentioned corpus of each leaf topic is employed for the extraction of the new set of nouns along with the corresponding TF-IDF weights. In order to maintain the terms with the highest relevance a threshold has been defined, which corresponds to the percentage of the highest weighted nouns that will construct the reduced representation of the leaf topics. To this end, the 10% of the extracted nouns along with their TF-IDF weights will be contained in the resultant feature-value vector, namely the Adapted TF-IDF Prototype Vector.

The online step of the reduction process concerns the identification of the common nouns between the incoming document and the Adapted TF-IDF Prototype Vector of the leaf topic where it has been classified. Thus, the document’s metadata representation will be reduced only to those noun terms that are also present in the corresponding Adapted TF-IDF Prototype Vector.

The reduction aims for the new metadata to be able to identify the particular sub-area of the given leaf topic in which a user is interested. In this case, if a noun term is not relevant for the topic category in which the document was classified, it would be unlikely to be relevant for a particular sub-area from that topic. Additionally, the document’s metadata representation will eliminate all the noun terms that are not relevant for an intratopic classification. This reduction is made according to the fact that the Adapted TF-IDF Prototype Vector contains noun terms that are relevant in making a differentiation with other topics but may have a low differentiation value for the intratopic classification. For example, if an article was classified into the topic *Tennis* in order to determine the sub-area of the Wimbledon event,

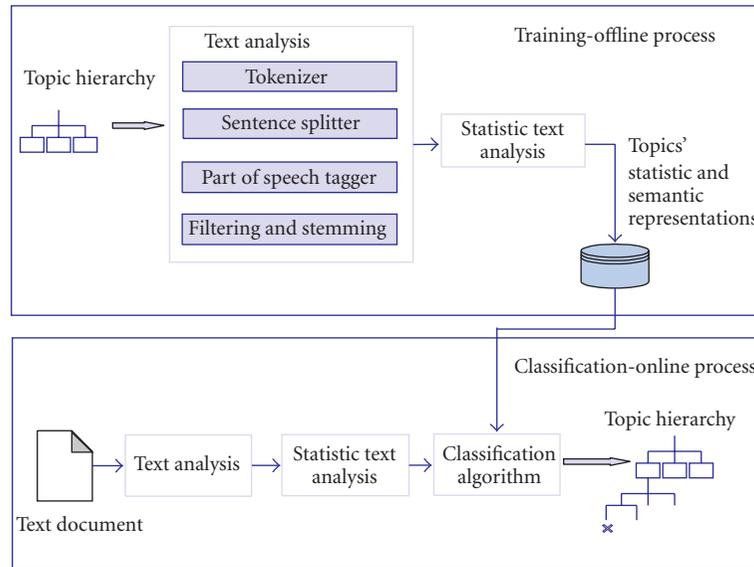


FIGURE 6: Diagram of news items classification process.

terms such as *tennis, set, game* will have less relevance than *grass, July, slam* and of course *Wimbledon*.

As the result of the data reduction process, the nouns which are contained in both the news item and the Adapted TF-IDF Prototype Vector of the corresponding leaf topic along with their TF weights in the news item will be sent to the user's device as the most significant topic-related nouns. These nouns will be called Adapted TF-IDF nouns.

#### 4.3.2. Reduction of named entities with a construction of a named entities knowledge base

During the metadata generation process there is no process aiming at the semantic identification of the Named entities. Thus, the output of the process is limited to Named Entities recognition and classification into a particular type (i.e., person, organization, location). To overcome this limitation, a methodology for constructing a Named Entities knowledge base has been defined, aiming both at semantically identifying Named Entities and also reducing the amount of data transmitted to the client.

More specifically, the semantic identification of a Named Entity concerns its association along with its corresponding type, with the particular topic where the news item, which contains the Named Entity has been classified. Additionally, the intended use of the knowledge base of Named Entities is to deal with cases of Named Entities having more than one representations. To this end, an ontology based learning approach has been followed to handle multiple interpretations such as follows.

- (i) Identify that two or more different representations refer to the same Entity. For example, identify that "Greenspan" and "The Federal Reserve Chairman" refer to the same person.

- (ii) Associate a Named Entity with its abbreviation. For example, "U.S." and "United States" refer to the same country.

In order to reach the aforementioned goals, the knowledge base of Named Entities is constructed following two complementary processes.

- (i) The process dealing with abbreviated Named Entities connecting to external abbreviations databases. Two such databases have been investigated, one concerning Locations and the other concerning organizations. The locations database is actually a list of countries acronyms, whereas the organizations database consist of a number of organizations and companies.
- (ii) An ontology-based discovery of multiple representations. In order to handle different representations, they are initially regarded as distinct Entities and gradually identify their associations with other existing Named Entities, according to the learning criteria of the co-occurrence in the same context, and of the belonging to the same type.

The reduction process of Named Entities includes the assigning of all possible representations of a Named Entity to a particular code, corresponding to a unique character sequence. Additionally, the system recomputes the frequency of appearance of each Named Entity in the news item, with respect to the new reduced Named Entities vector (where each one is represented by a unique code).

#### 4.4. Metadata storage and transmission to the handset

Following the metadata extraction and reduction processes, the metadata are stored in a news items repository (Figure 7)

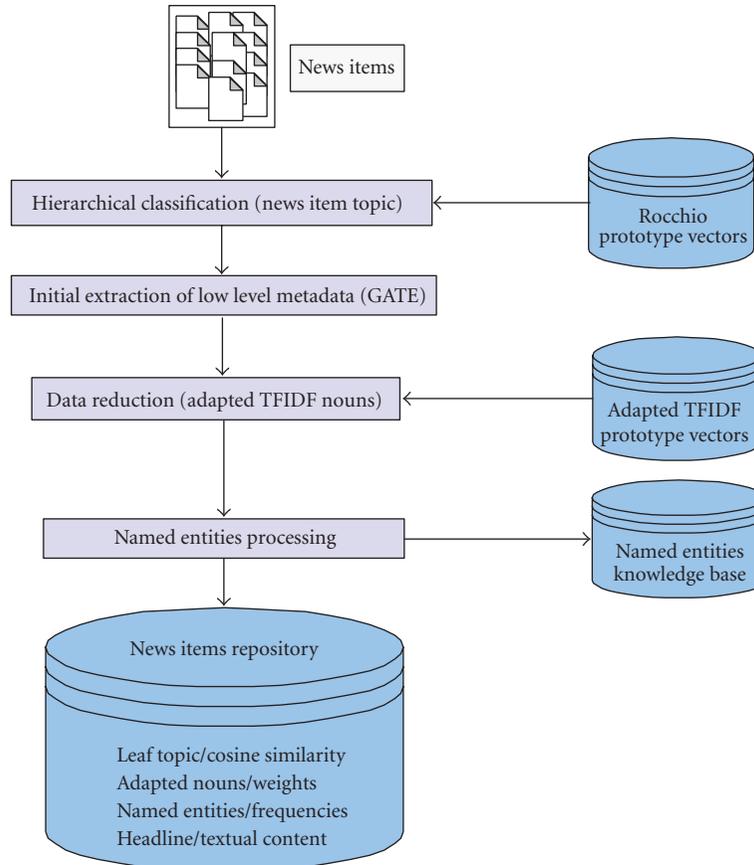


FIGURE 7: Metadata extraction and storage in the news items repository.

and are finally transmitted to the user device to be used in the low-level filtering and learning processes in the handset. The stored metadata are the following.

- (i) The Adapted TF-IDF nouns.
- (ii) The weights of the Adapted TF-IDF nouns corresponding to their frequency of appearance (TF) in the news item.
- (iii) The codes of Named Entities.
- (iv) The frequencies of the Named Entities' appearance in the news item.
- (v) The corresponding type of each term (i.e., the vector which assigns the type "N" to the nouns and "P," "O," or "L" to Named Entities).
- (vi) The leaf topic where the news item was classified.
- (vii) The cosine similarity value that has been computed for the news item and the classification topic; this is not transmitted to the handset but it is stored in the repository in order to be used in the server-side initial content filtering process, presented in Section 5.1.
- (viii) The headline and the textual content of the news item.

Apart from the above-mentioned metadata, a factor referring to the effect of each Named Entity type to a specific topic

is also transmitted. This is independent of each individual incoming news item and is calculated from the Named Entities knowledge base for each leaf topic. More specifically, it is measured by the percentage of the Named Entities belonging to a particular type (i.e., person, organization, or location) in a particular leaf topic, with regard to the total number of Named Entities belonging to this topic. This metric is used in the short-term learning process in the handset described in Section 6.1.

## 5. DISTRIBUTED SEMANTIC MATCHING

The main idea of this section is to present the high-level filtering of available content on the server, followed by matching of detailed user preferences in the handset. The output of the first filtering step is a list of recommended items for each user in all preferred leaf topic categories. Then, following the second filtering step, the content is displayed to the user in a ranked order. The distributed semantic matching process is described in detailed in this section.

### 5.1. Server-side initial content filtering

There are two inputs to the high-level filtering algorithm: the output of the topic classification process, that is, the only one leaf topic for the incoming to the server news items

(news items repository) and the explicit high-level user preferences stored in the skeleton profile (users' database). The main idea is that the content items are sent for further processing only to the users whose high-level profiles are related to the leaf topic of the content item. Hence, each user is assigned a set of news items that are classified to topics denoted as topics of interest. More specifically, the high-level filtering is based on a matching, implemented by simple queries, between the classification topic of each semantically annotated news item stored in the news items repository, and the preferences in the skeleton profile of each user in the users' database. Then according to simple rules which take into account this matching, the user will receive

- (i) the 100% (all) of the incoming to the server news items, which were classified to a leaf topic denoted with "High" degree of preference;
- (ii) the 50% of the incoming to the server news items, which were classified to a leaf topic denoted with "Medium" degree of preference;
- (iii) the 30% of the incoming to the server news items, which were classified to a leaf topic denoted with "Low" degree of preference; and
- (iv) none of the incoming to the server news items, which were classified to a leaf topic denoted with "None" degree of preference.

The 50% and 30% of the incoming news items in the cases of "Medium" and "Low" preference, respectively, are selected according to their ranking using the cosine similarity value estimated during the server-side classification. Namely, the 50% and 30% top ranked articles of the leaf topic are transmitted to the client. The motivation behind the rules regarding these 50% and 30% cases, lies in the following assumptions:

- (i) A user who has denoted high interest in a topic would read all the news articles related to that topic.
- (ii) On the other hand a user who has denoted medium or low interest to a topic would be satisfied to receive some news items concerning that topic, a percentage of them in the case of medium interest (i.e., the 50%) and a smaller one in the case of low interest (i.e., the 30%), in order to keep herself informed about the topic.
- (iii) Finally if the user is not interested at all in a particular topic, then she would be annoyed to receive related news items.

The output of the high-level filtering algorithm will be a list of content items for each user that will be submitted to the process of retrieval of the appropriate metadata from the news items repository.

### 5.2. Client-side low-level filtering-ranking

The semantic matching in the handset involves the semantic similarity between the detailed user profile stored in the handset and the significant low-level terms extracted from

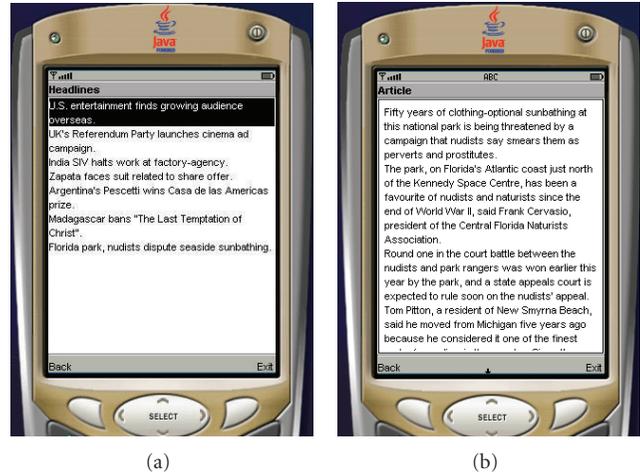


FIGURE 8: Snapshots from handset's screens (a) the ranked list of the headlines, and (b) the textual content of a selected article.

the article, that is, the Adapted TF-IDF nouns and the Named Entities, measured according to the cosine similarity metric.

The cosine similarity measure between the detailed user profile and the vector containing the terms of the document is calculated:

$$\text{Sim}(W, I') = \frac{\vec{W} * \vec{I}'}{|\vec{W}| \cdot |\vec{I}'|}, \quad (2)$$

where  $\vec{W}$  is the Weights Vector of the detailed user profile, while  $\vec{I}'$  is the vector of TF weights of the terms (i.e., the Adapted TF-IDF nouns and reduced Named Entities) extracted from the news item.

After calculation of the cosine similarity measures for all the incoming news items, the headlines of the news items are displayed on the user's screen based on the descending order of their corresponding cosine similarity measures, that is, the headlines of the articles with the highest cosine similarities results are displayed higher on the list, as illustrated in Figure 8. This ranked order corresponds to the short term user preferences, which are recorded in the detailed user profile. The user is able to view the textual content of each news item by selecting each headline in the list.

## 6. USER PROFILE LEARNING AND ADAPTATION

The user profile learning process is necessary for the personalization system, since user information needs are constantly changing, particularly in the context of the news domain. In this framework, the implicit user profile learning in the handset aims at identifying two different types of interest changes:

- (i) Abrupt interest changes: Abrupt interest changes may occur when new information needs arise due to user curiosity/immediate thoughts (internal) or motivation by the question of another person (external). Those changes refer to the need for adaptation of the short-term model.

- (ii) Gradual interest changes: User interests are widely recognized as changing slowly and gradually over time, for example, as conditions, goals and knowledge change. Gradual changes happen as consequences of continuous progress, for example, the user gaining experience or growing older. Those changes motivate the need for adaptation of the long-term model. However, as was already mentioned in Section 3.1, abrupt changes in the high-level profile can also be explicitly inserted by the user through the user interface employed in the initialization phase.

Our system performs a two-level learning process in order to automatically update the detailed and high-level user profile in the handset.

### 6.1. Short-term learning

The short term learning process exploits the implicit user feedback (i.e., monitoring of the user interactions with content items) for the adaptation of the detailed user profile, where the two main types of semantic metadata, that is, the nouns and Named Entities are involved. More specifically, the short-term user profile learning supports two main functionalities:

- (i) The adaptation of the values contained in the Weights Vector  $\vec{W}$
- (ii) The insertion and elimination of terms into and from the Terms Vector  $\vec{T}$  of the detailed user profile, respectively.

Both functionalities take place after the initialization process described in Section 3.2 has been completed, and thus the nouns and Named Entities corresponding to the news items selected by the user has been initially inserted in the Terms Vector  $\vec{T}$  of the detailed user profile.

#### 6.1.1. Weights adaptation

In several systems, which perform learning processes in order to update the user profiles, mathematical formulas are used for the adaptation of the different weight values [3, 9]. In our approach, the values contained in the Weights Vector  $\vec{W}$  of the detailed user profile, that is, the weight of each term in the Terms Vector  $\vec{T}$ , is updated according to a formula depending on whether the user selects or ignores news items that contain the term. This formula incorporates factors related to the particular term, such as the previous weight and the usage history of the term. Additionally, factors related to the selected content items where the term is contained are participating in the formula, such as the similarity measure of the item with the detailed user profile, the explicitly denoted weight of the leaf topic where it belongs, and the proportion of the amount of time spent to the item to its length [10]. Apart from the aforementioned factors the overall user behavior towards the personalization system is taken into account in order to adapt the weights of

the terms in the detailed profile, that is, the average number of read news item per day.

More specifically, the weights of the noun terms are adapted according to the following formula:

$$W_{\text{new}} = W_{\text{old}} \pm W_{\text{LT}} \cdot \text{Sim}(W, I') \cdot e^{-\beta * U_b * U_h} \cdot \log \frac{\text{time}}{\log \text{length}} \quad (3)$$

Correspondingly the weights of the Named Entities, are updated according to a similar formula:

$$W_{\text{new}} = W_{\text{old}} \pm W_{\text{LT}} \cdot W_{\text{NEType}} \cdot e^{-\beta * U_b * U_h} \cdot \log \frac{\text{time}}{\log \text{length}} \quad (4)$$

where:

- (i)  $W_{\text{old}}$ : represents the current term weight to be updated contained in the Weights Vector  $\vec{W}$ .
- (ii)  $\pm$ : is used to increase or decrease the current weight in case of positive or negative feedback, respectively. The articles that the user clicks to read are considered to be positive feedback for a term which exists in them, while the rest of the documents that contain the term but are not selected by the user, are considered to be negative feedback for the term.
- (iii)  $W_{\text{LT}}$ : is the explicitly denoted high-level weight of the leaf topic to which the news item has been classified (contained in the topic weights vector  $\vec{W}_{\text{LTopic}}$  of the high-level profile).
- (iv)  $\text{Sim}(W, I')$ : is the cosine similarity measure between the Weights Vector  $\vec{W}$  of the detailed user profile and the vector of TF weights of the terms (i.e., the Adapted TF-IDF nouns and reduced Named Entities) extracted from a news item.
- (v) The  $W_{\text{NEType}}$  is a factor referring to the effect of each Named Entity type (person, organization, or location) to a specific topic. It is calculated based on the Named Entities knowledge base and represents the semantic information, which is gathered from there.
- (vi)  $\log(\text{time}/\log \text{length})$ : incorporates the amount of time spent reading a news item in seconds and the length of the article in bytes, which operates as the normalizing factor. In the case of negative feedback, the time-length factor is set to 1, that is, it has no effect in the weight adaptation since the user does not spend time on the corresponding article.
- (vii)  $e^{-\beta * U_b * U_h}$ : is used to follow the personalized nonlinear change of the term weight according the usage history of the term. The changing rate of the weight is inversely proportional to the value of the parameter  $U_h$  that stands for the integer number of the selected articles where the term exists (contained in the Usage History Vector  $\vec{U}_H$ ) and  $U_b$ , which represents the

indicative mean number of articles that the user selects to read per day. The more articles a user reads per day, for example, the more slowly the weights increase in the low level profile.

- (viii)  $\beta$ : is a constant that is used to differentiate between the changing rate of the weight if the update is performed in relation with an interesting article or a non-interesting one. Thus it takes different values in the two opposite scenarios of positive/negative user feedback. More specifically, in the case of non-read articles (i.e., negative feedback from the user), the changing rate (i.e., the decreasing rate) should be much slower, since an unread news item does not constitute an explicit indication for non-interest. This is because an unread news item apart from considering it as not interesting, it can be interpreted as already read from another source, or it is possible that the user had no time to spend on it. On the contrary, in the case of read articles (i.e., positive feedback from the user) the changing rate (i.e., the increasing rate) should be faster, since a read news item demonstrates a strong indication for interest. Based on the numerical values produced by applying the formula, the proposed values for the beta constant were experimentally set to the following:

- (a)  $\beta = 0.01$  for read news items (positive feedback);
- (b)  $\beta = 0.02$  for non-read news items (negative feedback).

### 6.1.2. Insertion/elimination of terms into/from the detailed user profile

Apart from weights adaptation, a mechanism has been developed to update the terms (both nouns and Named Entities) contained in the Terms Vector  $\vec{T}$  of the detailed user profile. This ensures that the detailed user profile does not remain static after the initialization process but is constantly updated based on specific criteria.

When a user reads an article, which contains new terms (i.e., terms not existing in the current detailed profile), each of these terms is placed in a subordinate waiting stack as depicted in Figure 9. Then, each time the user selects a news item that contains any of those terms, the corresponding usage history value of each term in the waiting stack changes (i.e., it increases by one for each selection). The metric that determines the insertion of a new term into the Terms Vector  $\vec{T}$  in the detailed user profile is whether the term usage history exceeds a certain threshold. This threshold is determined by the user attitude towards the personalization system, namely it is proportional to the average number of news items that the user reads per day (e.g., for a user who reads approximately 20 news items per day, the usage history threshold of the terms in order to be inserted into the detailed user profile corresponds to 5). When a term is inserted into the Terms Vector  $\vec{T}$ , its initial weight in the Weights Vector  $\vec{W}$  has the default value of 0.5. Thus, the

default values for the initial entry into the system are similar to those used during the initialization process described in Section 3.2.

While the user interacts with the system, there may also be a need to remove terms from the detailed user profile, which imply low or non-interest from the user. In order to remove a term, both of the following two criteria should be satisfied:

- (i) whether the value of the term in the Usage History Vector  $\vec{UH}$  is lower than a certain threshold, which similarly to the insertion depends on the average number of read news items per day and it is lower than the insertion threshold, and
- (ii) whether the value of the term in the Weight Vector  $\vec{W}$  is lower than another certain threshold, which corresponds to a weight value around the medium preference (i.e., 0.5). It is additionally noted that if the weight of a term has turned to zero after several negative feedbacks, it is removed from the detailed profile anyway, that is, independently of its usage history.

## 6.2. Long-term learning

During the initialization process of the high-level user profile described in Section 3.1, the user explicitly denotes her high-level preferences, which are then transmitted to the server to allow for the initial content filtering. However, even the long-term user interests are subject to slow and gradual changes.

Thus, a long-term learning process has been developed to allow the system to follow any changes in the user preferences by automatically update the high-level user profile. This process involves a long-term user model, which consists of the following vectors, illustrated in Figures 10 and 11:

- (i) A Long-Term Noun Vector  $\vec{LTN}$ , which contains the nouns of a long-term set of articles.
- (ii) A Long-Term Correlation Values Vector  $\vec{LTCV}$ , which contains the corresponding calculated correlation values of the nouns in the long-term set of articles.
- (iii) The Prototype Nouns Vector  $\vec{PN}$  containing the Adaptive TF-IDF Prototype nouns for each leaf topic.
- (iv) The Prototype Weights Vector  $\vec{PW}$  containing the corresponding Prototype weights.
- (v) A Prototype Correlation Values Vector  $\vec{PCV}$ , which contains the calculated correlation values of the Prototype nouns for each leaf topic.
- (vi) A Long-Term Weights Vector  $\vec{LTW}$ , which contains long-term weights of the prototype nouns, which are constantly updated during the long-term learning process.

The long-term learning process involves three stages:

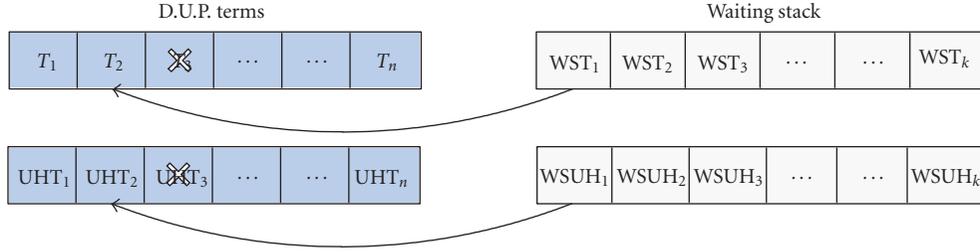


FIGURE 9: Insertion/elimination of terms into/from the detailed user profile.

- (i) The collection of nouns contained in a long-term set of articles.
- (ii) The association of the collected noun terms with the long-term user model and the adjustment of their weights according to a long-term learning formula.
- (iii) The updating of the skeleton profile on the server based on a client-server synchronization process.

6.2.1. Collection of nouns contained in a long-term set of articles

While the user is interacting with the system, all the articles displayed to the user (either selected or not), constitute a long-term set, on which the long-term learning process is based. The number of articles belonging to this set is predefined in the system to ensure that it covers a long-term period. All the noun terms that are contained in the long-term set of articles participate in the learning process.

In order to identify the effect of all those nouns on the long-term model, their correlation values have been investigated. More specifically, the relation between the change in the weights of the noun terms belonging both to the detailed user profile and a long-term set and their corresponding correlation values has been examined. After experimentation it has been found that:

- (i) When the correlation value is positive, there is an increase of the weight. Additionally, the greater the correlation value is, the larger the increase in the weight.
- (ii) When the correlation value is negative, there is a decrease in the weight.
- (iii) When the correlation value is zero, there is no change in the weight.

The motivation behind this investigation concerns the exploitation of the correlation value in order to update the long-term weights of the Prototype nouns corresponding to each leaf topic (Long-Term Weights Vector  $\vec{L}TW$ ), aiming at the automatic adaptation of the weight of each leaf Topic contained in the high-level profile, that is, in the topic Weights Vector  $\vec{W}LTopic$ . Therefore, by estimating the correlation values of the nouns in a long-term set of articles the weights of the leaf topics in the high-level profile can adapted.

The correlation values are calculated for all the terms contained in the long-term set of articles, inserted in the Long-Term Nouns Vector  $\vec{L}TN$ , and they are stored in the Long-Term Correlation Values Vector  $\vec{L}TCV$  (Figure 10). If the training set contains  $N$  incoming news items, the two binary discrete random variables taking 0 or 1 for values are defined:

- (i)  $X$  = Event that a randomly selected article contains the term  $w$ .
- (ii)  $Y$  = Event that the user selects to read an article from the incoming set.

If  $P_X$  and  $P_Y$  are the probability density functions of  $X$ ,  $Y$ , respectively, the joint probability density function is  $P_{XY}$ . The function to be computed is the correlation  $\rho(X, Y)$  between  $X$ ,  $Y$ , which produces a value between  $-1$  and  $1$ . The positive value for the term  $w$  indicates a dependency of the user selecting the article on the occurrence of  $w$ , while the negative value would tend to indicate that the user would not read articles containing  $w$ . A value of 0 would indicate that the two events of the user selecting an article and the occurrence of the term  $w$  in a news item are independent. The joint probability density functions for  $(x, y) = (0, 0), (1, 0), (0, 1), (1, 1)$  are considered as:

$$\begin{aligned}
 P_{XY}(1, 1) &= \{\text{articles containing } w \text{ that the user selects}\}/N \\
 P_{XY}(1, 0) &= \{\text{articles containing } w \text{ that the user does not select}\}/N \\
 P_{XY}(0, 1) &= \{\text{articles not containing } w \text{ that the user selects}\}/N \\
 P_{XY}(0, 0) &= \{\text{articles not containing } w \text{ that the user does not select}\}/N \text{ as well as the marginal probabilities } P_X(x) \text{ and } P_Y(y):
 \end{aligned}$$

$$\begin{aligned}
 P_X(1) &= P_{XY}(1, 0) + P_{XY}(1, 1) \\
 P_X(0) &= P_{XY}(0, 0) + P_{XY}(0, 1) \\
 P_Y(1) &= P_{XY}(0, 1) + P_{XY}(1, 1) \\
 P_Y(0) &= P_{XY}(0, 0) + P_{XY}(1, 0).
 \end{aligned} \tag{5}$$

The correlation coefficient between  $X$ ,  $Y$  is [11]:

$$\rho(X, Y) = \frac{E(XY) - E(X) \cdot E(Y)}{\sigma(X) \cdot \sigma(Y)}. \tag{6}$$

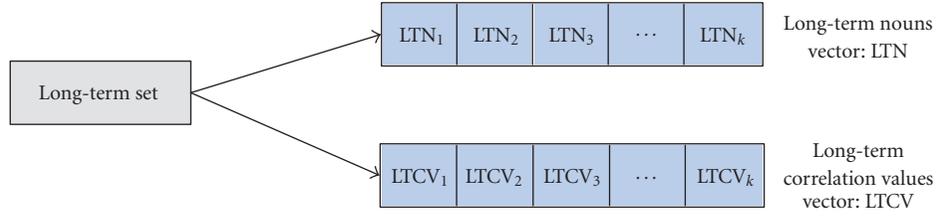


FIGURE 10: Vectors of nouns contained in the long-term set of articles.

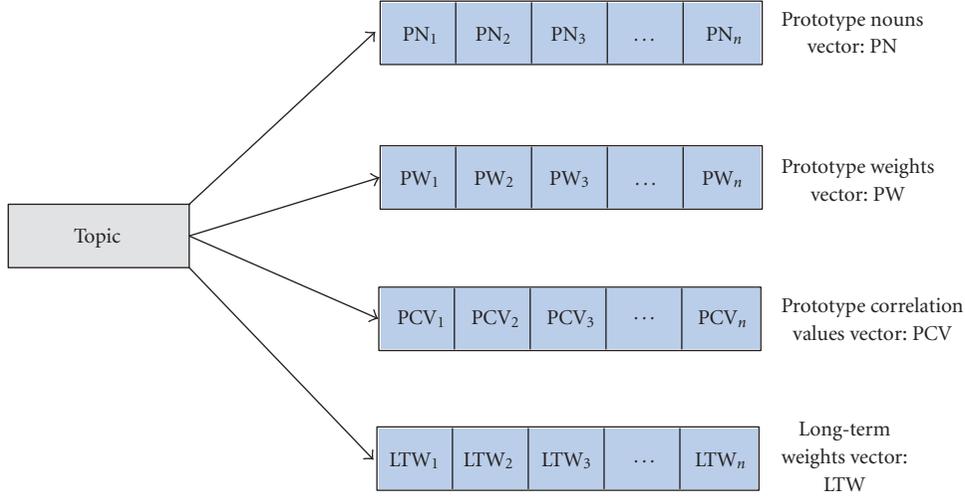


FIGURE 11: Vectors of nouns and weights corresponding to a particular leaf topic in the long-term learning model.

Analyzing the parts of this equation,

$$\begin{aligned}
 E(XY) - E(X) \cdot E(Y) &= \sum_{xy} xy \cdot P_{XY}(x, y) \\
 &\quad - \left( \sum_x x \cdot P_X(x) \right) \cdot \left( \sum_y y \cdot P_Y(y) \right) \\
 &= P_{XY}(1, 1) - P_X(1) \cdot P_Y(1)
 \end{aligned} \tag{7}$$

while

$$\sigma^2(X) = E(X^2) - E(X)^2 = P_X(1) - P_X(1)^2 = P_X(1) \cdot P_X(0). \tag{8}$$

Finally, the correlation between  $X$ ,  $Y$  is calculated as follows:

$$\rho(X, Y) = \frac{P_{XY}(1, 1) - P_X(1) \cdot P_Y(1)}{\sqrt{P_X(0) \cdot P_X(1) \cdot P_Y(0) \cdot P_Y(1)}}. \tag{9}$$

### 6.2.2. Association of low-level terms with the long-term learning model

Following the collection of nouns contained in the long-term set and the calculation of their correlation value, the next step is the association of those nouns with the leaf topics in the hierarchy of the high-level user profile. To this aim,

the handset receives from the server the Adapted TF-IDF Prototype Vectors containing the nouns for each leaf topic and their corresponding weights, which are stored in the handset's memory. These are the Prototype Nouns Vector  $\overrightarrow{PN}$  and the Prototype Weights Vector  $\overrightarrow{PW}$  depicted in Figure 11.

The long-term learning process for a particular leaf topic aims at adjusting the weight of the topic, which is initially specified from the user during the initialization of the high-level profile described in Section 3.1. This explicitly defined weight is propagated to all the nouns contained in the Prototype Nouns Vector  $\overrightarrow{PN}$  in order to initialize the Long-Term Weights Vector  $\overrightarrow{LTW}$  also displayed in Figure 11. Thus, the long-term learning process involves the adaptation of all the weights in the Long-Term Weights Vector  $\overrightarrow{LTW}$ . This is performed, according to the following steps:

- (1) All the Adapted TF-IDF Prototype nouns of this topic (in the Prototype Nouns Vector  $\overrightarrow{PN}$ ), which also belong to the Long-Term Nouns Vector  $\overrightarrow{LTN}$  are identified.
- (2) For the Adapted TF-IDF Prototype nouns that are NOT present in the Long-Term Nouns Vector  $\overrightarrow{LTN}$ , there is no change in the long-term weight, since the correlation value is assumed to be zero (due to the lack of information).

- (3) For the Adapted TF-IDF Prototype nouns that are also present in the long-term set, the new long-term weight is computed according to the following mathematical formula, after their corresponding correlation values are identified in the Long-Term Correlation Values Vector  $\overrightarrow{LTCV}$  and stored in the Prototype Correlation Values Vector  $\overrightarrow{PCV}$ :

$$W_{new} = W_{old} + CV \cdot W_{Prototype} \cdot U_b, \quad (10)$$

where:

- (i)  $W_{new}$  is the updated long-term weight to be stored in the Long-Term Weights Vector  $\overrightarrow{LTW}$ .
- (ii)  $W_{old}$  is the current value of the long-term weight contained in the Long-Term Weights Vector  $\overrightarrow{LTW}$ .
- (iii)  $CV$  is the computed correlation value of the noun contained in the Prototype Correlation Values Vector  $\overrightarrow{PCV}$ .
- (iv)  $W_{Prototype}$  is the topic related Adapted TF-IDF Prototype weight contained in the Prototype Weights Vector  $\overrightarrow{PW}$ .
- (v)  $U_b$  is the indicative mean number of articles that the user reads per day. This parameter represents the user's behavior towards the personalization system.

### 6.2.3. Updating the skeleton profile

After the computation of the adapted weights for all the (Adapted TF-IDF) Prototype nouns, the weight of each leaf topic in the topic weights vector  $\overrightarrow{WLTTopic}$  can now be updated using the following formula:

$$W_{LT\_new} = \frac{\sum_{i=1}^N W_{ad}}{N}, \quad (11)$$

where:

- (i)  $N$  is the number of the prototype nouns corresponding to the topic.
- (ii)  $W_{ad} = W_{new,i} \cdot (1 \pm (1 - C_p))$ , where:
  - (a)  $W_{new,i}$  is the long-term weight of each Prototype noun, which has been updated according to Formula (10) and stored in the Long-Term Weights Vector  $\overrightarrow{LTW}$ , that is,  $W_{new}$ .
  - (b)  $C_p$  is a coefficient used to increase the influence of the nouns, which are common to both the Prototype Nouns Vector  $\overrightarrow{PN}$  and the Long-Term Nouns Vector  $\overrightarrow{LTN}$ . This is equal to the percentage of those nouns in the Prototype Nouns Vector  $\overrightarrow{PN}$ .
  - (c) The + or - sign is applied when the correlation value of the Prototype Noun is positive, or negative, respectively.

- (iii) For the Prototype nouns that are not contained in the current Long-Term Nouns Vector  $\overrightarrow{LTN}$ , as well as for the Prototype nouns having zero correlation values, the following equation holds:  $W_{ad} = W_{new,i}$ , where  $W_{new,i}$  has not been updated.

Finally, the adapted weights of the leaf topics are stored in the handset's memory and then transmitted to the server to allow for the high level news content filtering.

When the user browses the non-leaf topics in the hierarchy, the changes in the weights of the leaf topics are propagated to their corresponding supertopics. Thus, the weights of the non-leaf topics are determined using the adapted weights of the leaf topics according to the following formula:

$$W_{T\_new} = \frac{\sum_{i=1}^M W_{T\_new,i}}{M}, \quad (12)$$

where:

- (i)  $W_{T\_new,i}$  is the updated weight of each subtopic of the non-leaf topic ( $W_{T\_new,i}$  corresponds to  $W_{LT\_new}$  if the subtopic is a leaf topic).
- (ii)  $M$  is the number of subtopics corresponding to the non-leaf topic.

## 7. EVALUATION OF THE PERSONALIZATION ENGINE

In this section, the experimental results are presented following the evaluation of the personalization engine, which includes the automatic adaptation for both the detailed and the high-level user profile. The evaluation tests concern each of the distinct learning processes performed in the handset, that is, the short-term learning and the long-term learning process. The evaluation experiments were conducted using news content from the Reuters corpus, and collecting data from regular system users. It should be noted that the user is aware of the system's personalization capabilities:

- (i) of automatically updating the high-level profile according to her long-term interests (thus she can explicitly alter the adapted symbolic degree of preference according to her choice when she does not approve the system's changes).
- (ii) of ranking the headlines of the incoming news items based on the user implicit feedback so she expects that the higher a headline is displayed in the list the more the corresponding news item falls under her interest.

### 7.1. User evaluation of short-term learning component

The evaluation of the short-term learning process is performed in order to demonstrate the overall performance of the short-term learning component. Moreover, the effectiveness of the low-level filtering that results in the ranking of the news items, is shown through this evaluation. Two versions of the short-term learning component have been

compared, namely the complete approach, which uses nouns and Named Entities, and a variant of this approach that uses only nouns. This comparative evaluation aims in demonstrating the contribution of Named Entities in the learning performance. In the variant of the system, Named Entities do not participate neither in the low-level matching process nor in the learning of the detailed user profile, since they do not exist at all in the profile.

For the evaluation of the two learning versions, 500 articles were semantically annotated and their metadata were stored in a news items repository. The user evaluation group consisted of 25 individuals. Each user was asked to manually rank according to his/her preferences a test set of 20 articles (5 articles from each topic) that belong to 4 different leaf topics:

- (i) 2 leaf topics chosen by the users belonging to 2 different trees in the hierarchy. A tree is defined as a group of topics sharing the same first-level topic.
- (ii) 2 other leaf topics chosen randomly from the 2 remaining trees.

For each user, a set of 100 articles was collected (4 topics of 25 items) that were used in a short-term learning process involving the interaction of the user. During this process, the user receives 4 different sets of 25 articles per day and the system constructs a detailed user profile for the current user, exploiting the user implicit feedback. The created profile is used to automatically rank the initial manually ranked test set of the 20 articles for which the user has provided explicit feedback. The above-described process has been repeated by each user twice, namely once for each variation of the short-term learning system.

In order to evaluate the learning system's performance, the ranking output of the system is compared to the manual ranking of the user. For this purpose one or more performance measures are needed. The standard IR performance measures precision and recall, rely for their calculation on the identification of each retrieved result as either a positive or a negative one. However, in our case only the ranking of the 20 articles for the different users is known; an item that has been ranked, for example, 8th by our algorithm is only known to have been ranked, for example, 10th by a user or a pool of users. Hence it can be clearly identified as relevant (positive) nor irrelevant (negative) to a given subject. Consequently, precision and recall are not the most suitable measures for quantifying the agreement between these two ranked lists. Instead a standard IR metric is used, which measures the correlation between two ranked lists, in our case the manually ranked by the user list and the automatically ranked by the system one. This metric is Spearman's rank correlation coefficient [12]:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}, \quad (13)$$

where  $d_i$  represents the difference of each article's ranking between the two lists, and  $n$  the number of articles in each list. In our case  $n = 20$ . Indicative correlation

TABLE 2: Correlation between the two (user's and system's) ranked lists.

User	Correlation of ranked lists using nouns and NEs	Correlation of ranked lists without NEs
1	0.85	-0.09
2	0.67	0.35
3	0.83	0.38
4	0.61	0.22
5	0.79	0.67
...	...	...
25	0.83	0.15
Average	0.70	0.48

results concerning the two ranked lists for both our short-term learning approaches, that is, the complete short-term learning system and the variant of the system with absence of Named Entities, are depicted in Table 2.

Additionally, the percentage error for the position of each of the ranked articles by the system, according to the manual ranking per user, is defined::

$$\text{error} = \left( \frac{|N_{\text{Manual Order}} - N_{\text{System Order}}|}{\max [N_{\text{Manual Order}} - 1, N_{\text{Total Articles}} - N_{\text{Manual Order}}]} \right) \cdot 100\%, \quad (14)$$

where  $N_{\text{Total Articles}} = 20$ . Indicative results along with the average percentage error per user are shown in Table 3 for both our short-term learning approaches, that is, the complete short-term learning system and the variant of the system with absence of Named Entities.

Precision can be applied for measuring the learning system's performance for the  $N$  top recommendations of the system, that is, the percentage of the  $N$  top ranked articles according to the system ranking, which were manually ranked also within the  $N$  top ones. In this case recall is equal with precision. Hence indicative results for the precision of the complete short-term learning system and the variant of the system without Named Entities, for the 10 top recommendations are depicted in Table 4.

The results of the evaluation process through all of the three different metrics seem promising. Additionally, they demonstrate the strong contribution of Named Entities in the high short-term learning performance, since the results of the complete learning component are much better than the ones of the variation without Named Entities. However, they could be further improved if certain limitations are handled, which do not concern the personalization system, but are mostly related to user perception. More specifically, the limitations arising from the user feedback are the following:

- (i) The users belong to a specific "social" group and most of them are not familiar with certain topics (particularly the economic related ones).

TABLE 3: %Error for the exact ranking position of each article in comparison to the manual ranking per user (25 users).

	User	1	2	3	4	5	...	25
Articles ranking position	1	0%	63.16%	15.79%	15.79%	21.05%	...	5.26%
	2	5.56%	22.22%	5.56%	27.28%	0%	...	11.11%
	3	0%	41.18%	23.53%	82.35%	0%	...	17.65%
	4	0%	37.50%	12.50%	6.25%	18.75%	...	12.50%
	5	6.67%	6.67%	13.33%	6.67%	6.67%	...	13.33%
Short-term Learning using nouns & NEs (Formula (3), (4))	...	...	...	...	...	...	...	...
Average	16	6.67%	13.33%	20%	33.33%	13.33%	...	26.67%
	17	37.50%	18.75%	6.25%	18.75%	12.50%	...	12.50%
	18	11.76%	17.65%	11.76%	11.76%	11.76%	...	11.76%
	19	5.56%	0%	44.44%	0%	11.11%	...	16.67%
	20	21.05%	21.05%	15.79%	31.58%	47.37%	...	10.53%
Articles ranking position	1	10.53%	44.44%	62.50%	26.67%	33.33%	...	83.33%
	2	33.33%	27.28%	6.67%	12.50%	34.50%	...	67.40%
	3	41.18%	55.70%	39.20%	37.50%	33.33%	...	34.50%
	4	62.50%	22.30%	58.67%	34.50%	22.22%	...	17.65%
	5	13.33%	10.53%	12.50%	33.33%	6.67%	...	42.80%
Short-term Learning using only nouns (Formula (3))	...	...	...	...	...	...	...	...
Average	16	6.67%	10.53%	39.20%	59.70%	27.28%	...	37.50%
	17	93.75%	44.44%	27.28%	55.70%	6.67%	...	22.22%
	18	100%	12.50%	28.35%	44.44%	26.67%	...	17.23%
	19	83.33%	23.10%	31.58%	13.33%	58.67%	...	12.50%
	20	36.84%	83.33%	21.05%	28.35%	10.53%	...	39.20%
Average		46.60%	33.33%	29.20%	41.18%	25.75%	...	44.44%

TABLE 4: Precision of the short-term learning system for the 10 top ranked articles.

User	Correctly ranked articles using nouns and NEs	Correctly ranked articles without NEs
1	100%	60%
2	90%	70%
3	90%	60%
4	80%	50%
5	80%	70%
...	...	...
25	90%	50%
Average	83.20%	61.50%

- (ii) Some users raised the issue that they do not find the articles consistent with some topics (i.e., different “interpretation” of the topics from the users). Hence some topics are harder to predict than others.
- (iii) The users have chosen certain topics, but the corresponding existing articles are not interesting for them, so this has affected their manual ranking to the test articles. Additionally their interaction with the system and consequently the final ranking of the system has been affected.

## 7.2. Experimental evaluation of long-term learning component

In order to evaluate the long-term learning process, 500 articles were offline annotated and their metadata were stored in a news items repository. The articles constituted 5 long-term sets of 100 articles each, used for evaluation purposes. These sets were constructed using articles that are classified to all the different leaf topics of the hierarchy. The 20 articles in each set belong to a specific leaf topic, which was selected for the evaluation purposes. More specifically, the “Disasters & Accidents” (“GDIS”) category was selected in order to observe its weight adaptation, which is induced by the interaction with the personalization system on the news items contained in the long-term sets.

Initially the topic “GDIS” was explicitly denoted with a “Medium” degree of preference. Thus, its initial weight corresponds to the 0.5 value. The following experiments were conducted for evaluating the long-term adaptation (i.e., increase or decrease) of the initial weight during the 5 sets.

- (i) All the “GDIS” along with several news items from other topics were selected in each set. An increase of the “GDIS” weight is expected.
- (ii) Approximately half of the “GDIS” along with several news items from other topics were selected in each set. A small increase of the “GDIS” weight is expected.

TABLE 5: Weight adaptation of “GDIS” topic after the completion of 5 long-term evaluation sets concerning the user selections of all, half, and none of the “GDIS” news items contained in these sets.

Long-term evaluation set	All (100%)	Half (50%)	None (0%)
1	0.5623	0.5197	0.4705
2	0.6203	0.5405	0.4325
3	0.6308	0.5654	0.4047
4	0.6762	0.5598	0.3520
5	0.7015	0.5740	0.2982

- (iii) None of the “GDIS” but only news items from other topics were selected in each set. A decrease of the “GDIS” weight is expected.

The experiments showed that when the 100% of the “GDIS” articles were selected, there was a constant increase of the weight of the topic after the completion of each set. When the last set was completed, the final weight just exceeded the 0.7 value, that is, the degree of preference changed from “Medium” to “High.” In the second case, when half of the “GDIS” articles were selected, there was also a constant increase of the weight, but in a reduced rate, so that the degree of preference did not finally change to “High.” Finally, when none of the “GDIS” articles was selected, there was a constant decrease of the weight of the topic until the last set where the weight became lower than the 0.3 value, that is, the degree of preference changed from “Medium” to “Low.” In Table 5, the weight adaptation of the topic “GDIS” during the 5 long-term sets is depicted.

As a conclusion, the changing rate of the weight of a particular topic is sufficient in order to change its symbolic degree of preference (e.g., from Medium to High, or from Medium to Low). This happens when the user demonstrates strong interest for this topic, or she keeps ignoring it through her interaction with the personalization system during a large number of news items.

## 8. RELATED WORK

In this section, related work is described addressing issues raised in this paper, such as distributed personalization architectures, methods for acquiring/adapting user profiles from implicit/explicit feedback and user modeling in the news personalization domain.

In recent years, machine learning techniques have been developed for application to a distributed architecture consisting of a server and a client machine (i.e., a cell phone, or a pocket personal computer). Reference [1] presents a distributed architecture for personalized news access, consisting of a central server, which handles a variety of functions and two clients, a web-based adaptive news service that learns from users’ explicit feedback, and another, which is geared towards wireless information devices (i.e., wireless organizers, PDAs, cell phones) and learns by observing the user. The learning process still resides, contrary to our

approach, for both clients, on the server. A distributed learning approach in a PDA, which uses a Bayesian classifier for the selection of articles of interest according to the user profile, is presented by [13]. The articles are extracted from web pages and displayed in a zoomable interface-based browser on a PDA. For keeping the profile up to date, the user provides implicit feedback to the system, which monitors her reading behaviors. The [14] approach uses a two step filtering, with a first filter on the server, and a second filter on the device. However, the server filter in that case is often reduced to a simple filtering linked to content sources.

Several systems that have recently been developed for personalized news access, use the explicit or implicit feedback that the user provides for the construction and the updating of the user profile [15]. In the implicit user input, the user has no direct access to the information in the user profile or its construction. The acquisition of the profile as terms, categories or sets of relevant documents must be made implicitly by interpreting user actions on the system such as the number of key clicks in a document, the amount of scrolling through the document, or the amount of time spent reading the document. The types of implicit feedback that can be reliably extracted from observed user behavior in web search are investigated by [16]. Furthermore, [17] explores different approaches for ranking web search results by exploiting user interactions with the search engine. On the other hand, in explicit profile construction, the user has the responsibility to give the required information to the personalization engine for the construction of the user profile representation, normally through a graphical user interface. The acquisition of the profile can be made by asking the user to enter “terms” or “categories” corresponding to her preferences [18, 19], or by applying a supervised learning algorithm on a training set of “documents,” which the user regards as relevant. Reference [20] proposes an adaptive personalized web browser, monitors the user’s access behavior such as history, bookmarks, content of pages and access logs to model her interests. A user model dealing with an explicit definition provided by the user through a profile editor, and an implicit part maintained by intelligent services is presented by [21]. Explicit feedback provides more accurate estimates of user interest [22], since there are many reasons why a user would spend time on a particular document other than being interested in it, for example, the user decides that she is not interested in a document after the careful analysis of it. In our work, both explicit and implicit user feedback have been exploited. The explicit feedback is being used to ensure that the user profile is being properly initialized, while the implicit feedback provides the means to reduce the user overload by exploiting a combination of different types of metadata, that is, hierarchical topic categories, and low-level terms.

Recently, the advances in the Semantic Web technologies have enabled the representation of user profiles in a variety of ways. Semantic annotation of content with domain concepts, combined with semantic user preferences, enable inferring user preferences for content. Content semantics are typically based on hierarchies (taxonomies) of categories. The majority of the wireless content providers adopt this

type of hierarchical structures. References [23, 24] also use concept hierarchies for user profiles. References [25–27] create a list of concepts of interest, while [23, 28, 29] create a hierarchically-arranged collection of concepts, or ontology. References [10, 30] build user profiles consisting of specific concepts of a hierarchy, which is represented by an ontology. The system automatically monitors the user's browsing habits in requested web pages from search engines. The initial profile is constructed by assigning the visited web pages to specific concepts of a predefined reference hierarchy-ontology. Semantic user preferences often form the basis of user profiles and they may be divided in two categories, namely records of thematic categories indicating user preference for specific categories or classification schemes of content, and records of simple concepts or weighted sets, indicating the level of the user interest for each concept [31–33].

The above-mentioned approaches, which use Semantic Web tools in order to represent user profiles, are closely related to our work with respect to the use of hierarchical long-term user models, and the classification of content to topic categories. However, there are two main limitations compared to our work. First, they focus on the detection only of long-term user interests and second, they do not propose how these methods could be applied on constrained environments such as mobile devices.

Regarding user interests, [34] distinguishes between short-term interests, which are determined by a particular user query and long-term interests, which are determined by the user preferences over a long time period. He argues that longer-term user properties should also be taken into account when a system filters the content to be delivered. The two separate user models, that is, a long-term and a short-term user model are applied in several systems. In [2], a user interest hierarchy is learnt from a set of web pages visited by the user. The higher-level interests (more general), correspond to long-term interests, while the lower-level ones (more specific), correspond to short-term interests. Reference [3] describes a scheme for dynamic learning of user interests from user feedback in an automated information filtering Internet system using a 3-descriptor scheme for the representation of each category of interests in a profile, which also allows learning of long/short-term interests. Reference [9] captures user interests in order to build and update user profiles exploiting low-level features such as keywords, extracted from text using language processing techniques. Generally, the user profiles are adapted using various learning techniques including the exploitation of vector space model [28, 35], genetic algorithms [36], the probabilistic model [25], or clustering [37].

Our research combines aspects from several systems regarding the separation of the user model into short-term and long-term and the user profile learning. The novelty in our approach, apart from the distributed nature of the architecture, is that the learning process for both models is employed exclusively on the client side following the user explicit and implicit feedback. Additionally, the short-term model is not limited to the use of terms as keywords, but it also exploits the semantic information arising from the

association of the noun terms with the topic classification of the news articles.

## 9. CONCLUSIONS

In this paper, a distributed architecture for personalized news content delivery has been presented. It consists of a two-stage semantic matching process, enabling a high-level filtering of available content on the server, followed by matching of detailed user preferences in the handset. This is enhanced with a learning and adaptation process based on explicit and implicit user feedback. The learning process for both the short-term and long-term models takes place in the handset and the adaptation in the long-term model is also transmitted to the server through a client-server synchronization process.

Both user models exploit the semantic annotation of the news content with different types of metadata such as the topic category of the news item, the identified Named Entities and the most significant noun terms according to the classification topic.

The evaluation results of both the short-term and long-term learning processes are very promising for the implementation of the system in a commercial environment, not only because they are consistent with the user expectations, but also because they are achieved with a minimal user overload and taking into account the communication and computational cost.

In the future, another challenge would be to automatically learn topic hierarchies from the textual content rather than use the manually constructed ones, as in the current case. Furthermore, the learning process in the handset could be extended to take into account the contextual information of the user, such as time and location, which are key inputs in the current mobile environments.

## ACKNOWLEDGMENT

The work presented in this paper was fully supported by the project “Distributed Knowledge Management for Personalized Content Delivery” funded by Motorola UK Ltd.

## REFERENCES

- [1] D. Billsus and M. J. Pazzani, “User modeling for adaptive news access,” *User Modelling and User-Adapted Interaction*, vol. 10, no. 2-3, pp. 147–180, 2000.
- [2] H. R. Kim and P. K. Chan, “Learning implicit user interest hierarchy for context in personalization,” in *Proceedings of the 8th International Conference on Intelligent User Interfaces (IUI '03)*, pp. 101–108, ACM Press, Miami, Fla, USA, January 2003.
- [3] D. Widyantoro, T. Ioerger, and J. Yen, “An adaptive algorithm for learning changes in user interests,” in *Proceedings of the 8th International Conference on Information and Knowledge Management (CIKM '99)*, pp. 405–412, Kansas City, Mo, USA, November 1999.
- [4] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, “GATE: a framework and graphical development environment for robust NLP tools and applications,” in *Proceedings of the*

- 40th Anniversary Meeting of the Association for Computational Linguistics (ACL '02), Philadelphia, Pa, USA, July 2002.
- [5] H. Cunningham, D. Maynard, K. Bontcheva, et al., "Developing language processing components with GATE (a user guide)," University of Sheffield, Sheffield UK, 2005.
  - [6] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.
  - [7] J. Rocchio, "Relevance feedback in information retrieval," in *The Smart Retrieval System—Experiments in Automatic Document Processing*, G. Salton, Ed., chapter 14, pp. 313–323, Prentice-Hall, Englewood Cliffs, NJ, USA, 1971.
  - [8] T. Joachims, "A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization," in *Proceedings of 14th International Conference on Machine Learning (ICML '97)*, pp. 143–151, Nashville, Tenn, USA, July 1997.
  - [9] E. Bloedorn and I. Mani, "Using NLP for machine learning of user profiles," *Intelligent Data Analysis*, vol. 3, no. 2, pp. 3–18, 1998.
  - [10] S. Gauch, J. Chaffee, and A. Pretschner, "Ontology-based personalized search and browsing," *Web Intelligence and Agent Systems*, vol. 1, no. 3-4, pp. 219–234, 2003.
  - [11] J. Cohen, P. Cohen, S. G. West, and L. S. Aiken, *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*, Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 3rd edition, 2003.
  - [12] E. L. Lehmann and H. J. M. D'Abrera, *Nonparametrics: Statistical Methods Based on Ranks*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1998.
  - [13] R. Carreira, J. M. Crato, D. Gonçalves, and J. A. Jorge, "Evaluating adaptive user profiles for news classification," in *Proceedings of the 9th International Conference on Intelligent User Interfaces (IUI '04)*, pp. 206–212, Madeira, Portugal, January 2004.
  - [14] W. Matz, "System and method for filtering content," Bellsouth Intellect PTY Corp, Application no. US2003726727A, Filed 20031202 Published 20050602, 2005.
  - [15] A. Kobsa, "Generic user modeling systems," *ACM User Modeling and User Adapted Interaction*, vol. 2, no. 1-2, pp. 49–63, 2001.
  - [16] T. Joachims, L. Granka, B. Pang, H. Hembrooke, and G. Gay, "Accurately interpreting clickthrough data as implicit feedback," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05)*, pp. 154–161, Salvador, Brazil, August 2005.
  - [17] E. Agichtein, E. Brill, and S. Dumais, "Improving web search ranking by incorporating user behavior information," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '06)*, pp. 19–26, Seattle, Wash, USA, August 2006.
  - [18] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," in *Proceedings of the 13th International Conference on World Wide Web (WWW '04)*, pp. 403–412, New York, NY, USA, May 2004.
  - [19] L. Kerschberg, W. Kim, and A. Scime, "WebSifter II: a personalizable meta-search agent based on weighted semantic taxonomy tree," in *Proceedings of the International Conference on Internet Computing (IC '01)*, pp. 14–20, Las Vegas, Nev, USA, June 2001.
  - [20] P. K. Chan, "Constructing web user profiles: a non-invasive learning approach," in *Proceedings of the International Workshop on Web Usage Analysis and User Profiling (WEBKDD '99)*, pp. 39–55, San Diego, Calif, USA, August 1999.
  - [21] L. Razmerita, A. Angehrn, and A. Maedche, "Ontology-based user modeling for knowledge management systems," in *Proceedings of 9th International Conference on User Modeling (UM '03)*, pp. 213–217, Springer, Johnstown, Pa, USA, June 2003.
  - [22] L. M. Quiroga and J. Mostafa, "Empirical evaluation of explicit versus implicit acquisition of user profiles in information filtering systems," in *Proceedings of 4th ACM Conference on Digital Libraries*, vol. 37, pp. 238–239, Berkeley, Calif, USA, August 1999.
  - [23] T. Kurki, S. Jokela, R. Sulonen, and M. Turpeinen, "Agents in delivering personalized content based on semantic metadata," in *Proceedings of the AAAI Spring Symposium on Intelligent Agents in Cyberspace*, pp. 84–93, Stanford, Calif, USA, March 1999.
  - [24] Y. Labrou and T. Finin, "Yahoo! as an ontology—using Yahoo! categories to describe documents," in *Proceedings of the 8th International Conference on Information Knowledge Management (CIKM '99)*, pp. 180–187, Kansas City, Mo, USA, November 1999.
  - [25] D. Mladenic, "Text-learning and related intelligent agents: a survey," *IEEE Intelligent Systems and Their Applications*, vol. 14, no. 4, pp. 44–54, 1999.
  - [26] M. Pazzani, J. Muramatsu, and D. Billsus, "Syskill & Webert: identifying interesting web sites," in *Proceedings of the 13th National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference (AAAI '96)*, vol. 1, pp. 54–61, Portland, Ore, USA, August 1996.
  - [27] S. E. Middleton, D. C. De Roure, and N. R. Shadbolt, "Capturing knowledge of user preferences: ontologies in recommender systems," in *Proceedings of the 1st International Conference on Knowledge Capture (K-Cap '01)*, pp. 100–107, Victoria, Canda, October 2001.
  - [28] A. Pretschner, *Ontology based personalized search*, M.S. thesis, The University of Kansas, Lawrence, Kan, USA, 1999.
  - [29] F. Tanudjaja and L. Mui, "Persona: a contextualized and personalized web search," in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS '02)*, pp. 1232–1240, Big Island, Hawaii, USA, January 2002.
  - [30] S. Gauch and J. Trajkova, "Improving ontology-based user profiles," in *Proceedings of Computer Assisted Information Retrieval (RIAO '04)*, pp. 380–389, Vaucluse, France, April 2004.
  - [31] K. Bradley, R. Rafter, and B. Smyth, "Case-based user profiling for content personalisation," in *Proceedings of International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH '00)*, pp. 62–72, Trento, Italy, August 2000.
  - [32] R. Prestes, G. Carvalho, R. Paes, C. J. P. Lucena, and M. Endler, "Applying ontologies in open mobile systems," in *Proceedings of the Workshop on Building Software for Pervasive Computing (OOPSLA '04)*, Vancouver, Canada, October 2004.
  - [33] A. Cali, D. Calvanese, S. Colucci, T. Di Noia, and F. M. Donini, "A description logic based approach for matching user profiles," in *Proceedings of the 17th International Workshop on Description Logics (DL '04)*, vol. 104, pp. 196–202, Whistler, Canada, June 2004.
  - [34] A. Jameson, "Modeling both the context and the user," *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 29–33, 2001.
  - [35] L. Chen and K. Sycara, "Web mate: a personal agent for browsing and searching," in *Proceedings of the 2nd International Conference on Autonomous Agents and Multi Agent Systems*

(*AGENTS '98*), pp. 132–139, ACM Press, Minneapolis, Minn, USA, May 1998.

- [36] M. Mitchell, *An Introduction to Genetic Algorithms*, A Bradford Book, The MIT Press, Cambridge, Mass, USA, 1996.
- [37] B. Mobasher, H. Dai, T. Luo, M. Nakagawa, Y. Sun, and J. Wiltshire, “Discovery of aggregate usage profiles for web personalization,” in *Proceedings of Web Mining for E-Commerce Workshop, in Conjunction with the ACM-SIGKDD Conference on Knowledge Discovery in Databases (KDD '00)*, Boston, Mass, USA, August 2000.

