Noname manuscript No.
(will be inserted by the editor)

Automatic News Recommendations via Aggregated Profiling

Erik Mannens · Sam Coppens · Toon De Pessemier · Hendrik Dacquin · Davy Van Deursen · Robbie De Sutter · Rik Van de Walle

Received: date / Accepted: date

E. Mannens
Ghent University - IBBT
ELIS - Multimedia Lab
Ghent, Belgium
E-mail: erik.mannens@ugent.be

S. Coppens
Ghent University - IBBT
ELIS - Multimedia Lab
Ghent, Belgium
E-mail: sam.coppens@ugent.be

T. De Pessemier
Ghent University - IBBT
INTEC - WiCa
Ghent, Belgium
E-mail: tdpessem@intec.ugent.be

H. Dacquin
VRT
VRT-medialab
Brussels, Belgium
E-mail: hendrik.dacquin@vrt.be

D. Van Deursen
E-mail: davy.vandeursen@ugent.be Ghent University - IBBT
ELIS - Multimedia Lab
Ghent, Belgium
E-mail: davy.vandeursen@ugent.be

R. De Sutter
VRT
VRT-medialab
Brussels, Belgium
E-mail: robbie.desutter@vrt.be

R. Van de Walle
E-mail: rik.vandewalle@ugent.be Ghent University - IBBT
ELIS - Multimedia Lab
Ghent, Belgium
E-mail: rik.vandewalle@ugent.be

Abstract Today, people have only limited, valuable leisure time at their hands which they want to fill in as good as possible according to their own interests, whereas broadcasters want to produce and distribute news items as fast and targeted as possible. These (developing) news stories can be characterised as dynamic, chained, and distributed events in addition to which it is important to aggregate, link, enrich, recommend, and distribute these news event items as targeted as possible to the individual, interested user. In this paper, we show how personalised recommendation and distribution of news events, described using an RDF/OWL representation of the NewsML-G2 standard, can be enabled by automatically categorising and enriching news events metadata via smart indexing and linked open datasets available on the web of data. The recommendations – based on a global, aggregated profile, which also takes into account the (dis)likings of peer friends – are finally fed to the user via a personalised RSS feed. As such, the ultimate goal is to provide an open, user-friendly recommendation platform that harnesses the end-user with a tool to access useful news event information that goes beyond basic information retrieval. At the same time, we provide the (inter)national community with standardised mechanisms to describe/distribute news event and profile information.

Keywords News Modelling, Profiling, Recommendation

1 Introduction

In the PISA project[1] we have investigated how, given a file-based media production and broadcasting system with a centralized repository of metadata, many common production, indexing and searching tasks could be improved and automated. We have shown how such an up-to-date news bulletin can be dynamically created and personalized to match the consumer's *static* categories preferences [10,21] by merging different news sources and using the NewsML-G2 specification [15]. While the impact of file-based production indeed mainly affected the work methods of the news production staff - journalists, anchors, editorial staff, etc - [20,9], the added-value for the end-user was still marginal, id est, he might notice that news content is made available faster. Practically, our aim is to demonstrate the possibility of dynamically digesting an up-to-date news bulletin by merging different news sources, assembled to match the *real* individual consumer's likings, by recommending his favourite topics. In this paper, we build on our prior work and exploit further the semantic capabilities of NewsML-G2 and enhance it via an automatic recommendation system using the end-user's *dynamic* profile.

This paper is organised as follows. In Section 2, we briefly present the NewsML-G2 standard and its conceptualisation in an OWL [22] ontology being used in Section 3 as a unifying (meta)datamodel for highlighting the backend of the end-to-end news distribution architecture. Section 4 further elaborates on how the flow of news events can be categorised and automatically enriched with knowledge available in large linked datasets. Afterwards Section 5 and Section 6 unleash the dynamically harvested user profiles to the recommendation engine to harness the best-fit news items to individual user likings. Section 7 then distributes these recommended and enriched news events to the individual users. Finally, conclusions are drawn in Section 8.

[1] http://www.ibbt.be/en/projects/overview-projects/p/detail/pisa/

2 News Modelling

The IPCT News Architecture framework (NAR[2]) is a generic model that defines four main objects (*newsItem*, *packageItem*, *conceptItem* and *knowledgeItem*) and the processing model associated with these structures. Specific languages such as NewsML-G2 or EventsML-G2[3] are built on top of this architecture. For example, the generic *newsItem*, a container for one particular news story or *dope sheet*, is specialized into media objects (textual stories, images or audio clips) in NewsML-G2.

Within a *newsItem*, the elements *catalog* and *catalogRef* embed the references to appropriate taxonomies; *rightsInfo* holds rights information such as who is accountable, who is the copyright holder and what are the usage terms; *itemMeta* is a container for specifying the management of the item (e.g. title, role in the workflow, provider). The core description of a news item is composed of administrative metadata (e.g. creation date, creator, contributor, intended audience) and descriptive metadata (e.g. language, genre, subject, slugline, headline, dateline, description) grouped in the *contentMeta* container. A news item can be decomposed into parts (e.g. shots, scenes, image regions and their respective descriptive data and time boundaries) within *partMeta* while *contentSet* wraps renditions of the asset. Finally, semantic in-line markup is provided by the *inlineRef* container for referring to the definition of particular concepts (e.g. person, organization, company, geopolitical area, POI, etc).

NAR is a generic model for describing news items as well as their management, packaging, and the way they are exchanged. Interestingly, this model shares the principles underlying the Semantic Web: *i)* news items are distributed resources that need to be uniquely identified like the Semantic Web resources; *ii)* news items are described with shared and controlled vocabularies. NAR is however defined in XML Schema and has thus no formal representation of its intended semantics (e.g. a *NewsItem* can be a *TextNewsItem*, a *PhotoNewsItem* or a *VideoNewsItem*). Extension to other standards is cumbersome since it is hard to state the equivalence between two XML elements. EBU (amongst others) have proposed to model an OWL ontology of NewsML-G2[4] to address these shortcomings and we have discussed the design decisions regarding its modelling from existing XML Schemas [33, 21].

3 News Gathering

News broadcasters receive news information from different sources. The *Vlaamse Radio- en Televisieomroep* (VRT), the public service broadcaster of the Flemish part of Belgium, in particular gathers its material from its own news crews and from several trusted international news agencies and/or broadcasters, like *Reuters*, *EBU Eurovision*, and *CNN*, as can be seen in the *Back-end* part of Figure 1. The rough-cut and mastered essence created by the news crews is stored into VRT's Media Asset Management (MAM) system. Reporters use AVID's iNews application to enrich the essence by adding descriptive information, such as captions, anchor texts, etc. This application

[2] http://www.iptc.org/NAR/

[3] http://iptc.cms.apa.at/std/EventsML-G2/

[4] http://www.ebu.ch/metadata/ontologies/NML2/

is also used to create and organize the rundown of a classical television news broadcast. Within our presented architecture, the essence is retrieved from the VRT's MAM and copied into a separate MAM for demo purposes. The rundown information is extracted from iNews in the *Standard Generalized Markup Language* (SGML) format, upconverted into a NewsML-G2 instance, and pushed to the NewsML-G2 Parser. The news items (essence and metadata) from international news agencies are received via satellite communication. More and more providers also structure their metadata in this NewsML-G2 standard which can directly be pushed to the NewsML-G2 parser. The essence just needs to be packed into an *Material Exchange Format* [32] (MXF) instance before the MAM can process it. Afterwards, the essence is transcoded into a consumer format, such as H.264/ AVC [17], to be seen as the *Automated Production* component in Figure 1.

4 News Enrichment

The NewsML-G2 Parser then takes as input a NewsML-G2 instance (XML format) and produces an enriched NewsML-G2 instance (RDF triples) compliant to the NewsML ontology. First, the incoming XML elements are parsed and converted to instances of their corresponding OWL classes and properties within the NewsML ontology. Second, plain text contained in XML elements such as *title* and *description* is sent to the metadata enrichment service. The latter extracts named entities from the plain text and tries to find formal descriptions of these found entities on the Web. Hence, the metadata enrichment service returns a number of additional RDF triples containing more information about concepts occurring in the plain text sections of the incoming NewsML-G2 instance. Finally, the resulting RDF triples are stored in the AllegroGraph RDF store (see Figure 1).

The linguistic processing consists in extracting named entities such as persons, organisations, companies, brands, locations and other events. We use both the i.Know's Information Forensics service[5] and the *OpenCalais* infrastructure[6] for extracting these named entities. For example, the processing of the headline "Tom Barman and his band dEUS opening their latest album Vantage Point in Rock Werchter" will result in five named entities: 'Tom Barman','dEUS', 'Vantage Point', 'Rock Werchter', and 'Werchter' together with their type (i.e. Person, Music Group, Music Album, Event, Location, etc). Once the named entities have been extracted, we map them to formalised knowledge on the web available in *GeoNames*[7] for the locations, or in *DBPedia*[8]*/FreeBase*[9] for the persons, organisations and events. The string 'Tom Barman' is therefore mapped to its URI in *DBPedia*[10] that provides *i)* a unique identifier for the resource and *ii)* formalised knowledge about this person such as his biography, career and genealogy in multiple languages. Therefore, the use of the *OpenCalais* web service allows us to populate the knowledge base by providing a list of possible instances for all named entities discovered.

[5] http://www.iknow.be/

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

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

[8] http://dbpedia.org/

[9] http://www.freebase.com/

[10] http://dbpedia.org/resource/Tom_Barman

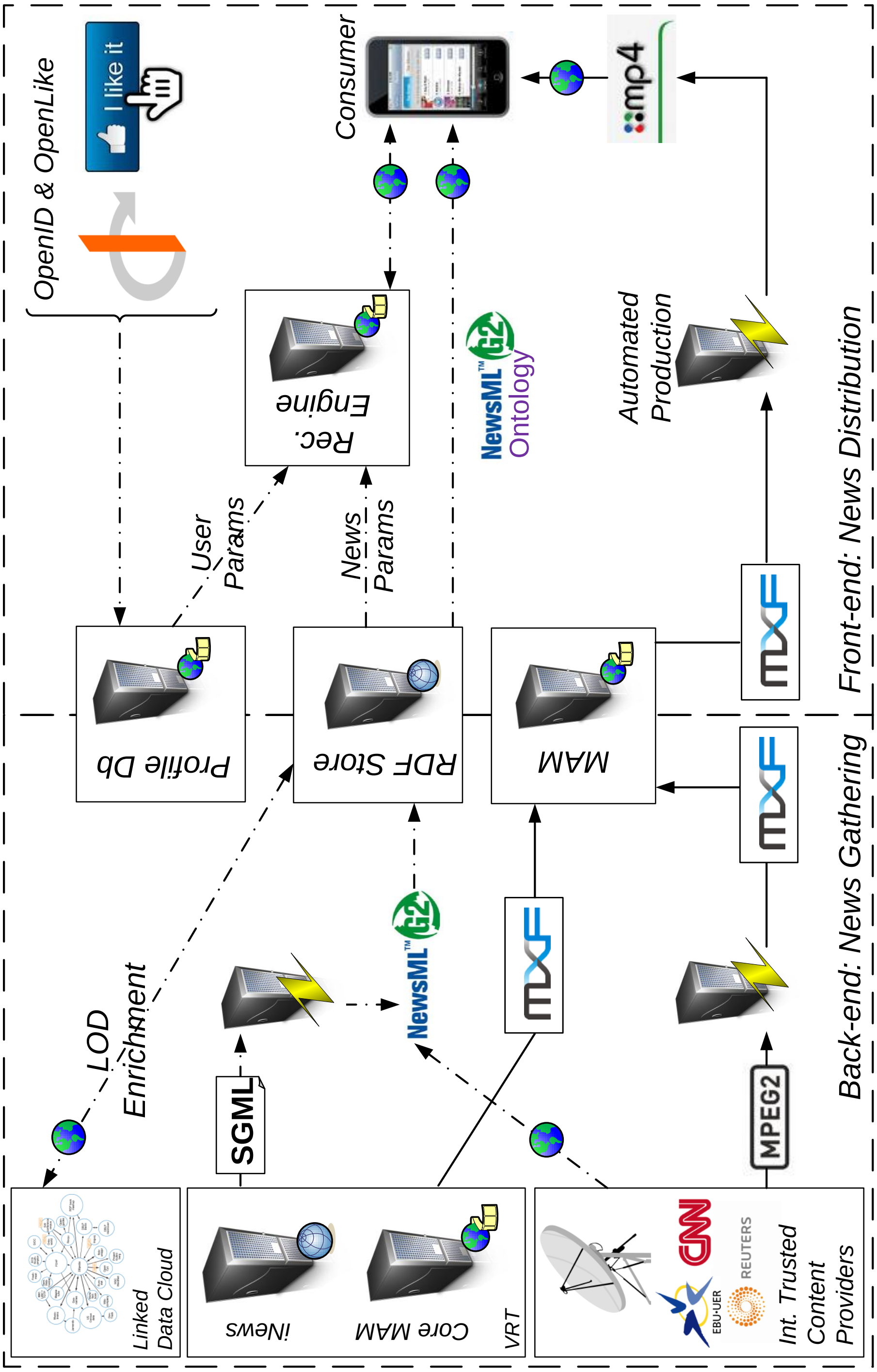

Fig. 1 End-to-end News Distribution Architecture

5 User Profiling

Recent years technologies have appeared that empower the user to have more control over his experience. E.g., RSS[11] was designed to empower the user to view the content he or she wants, when it's wanted, not at the behest of the information provider [25]. Beyond this control over the view, the user needs adequate filtering mechanisms in order to work through this real-time stream of (news) data. Recommendation systems offer content tailored to the user's needs. A common way of serving selected content is to relate it to the user's profile information. *Amazon*[12] is considered a leader in online shopping and particularly recommendations. They have built a smart set of recommendations that tap into a user's browsing history, past purchases and purchases of other shoppers [16]. *Pandora*[13] is a music recommendation system that leverages similarities between pieces and music and thus a recommendation system based on genetics. While both *Amazon* and *Pandora* offer an excellent service, they do not have access to the massive amount of information about a user that is stored in his preferred social network.

Together with the evolution of recommendation engines, social networks are growing, according to a recent *Forrester Research* study [6]. The giant in the space remains *Facebook*[14], which gets 87.7 million unique viewers per month, according to *ComScore*[15]. And although *Facebook* is the most popular social network at the moment, users don't limit themselves to one dedicated network. There are a plethora of popular social networks with more than 1 million monthly visitors: *Myspace*[16], *Twitter*[17], *LinkedIn*[18] and *Netlog*[19] are among the more popular ones. There are in fact already also a number of very popular social news websites, a.o. *Reddit*[20], *Digg*[21], and *Propeller*[22]. Generally, a user's profile consists of three types of information: 1) *static information*, e.g., the user's birthdate, address, favourite books, etc; 2) *dynamic information*: this is information coming from the user's activity stream, e.g., what is the user listening to, what is the user's current location, feedback of the user on offered content, etc, using the *OpenLike* paradigm[23] (see Figure 1); 3) *the social graph*: this contains all the user's connections to other users, e.g., a friendlist.

Current recommendations are mostly offered within the closed context of a single community: e.g. *Facebook* recommends events based on RSVP event invitations[24] from other users connected to the *Facebook* user's social graph. *Facebook* does not automatically recommend events based on the static and dynamic profile information of a user,

[11] RSS:ReallySimpleSyndication,alsoseehttp://www.rss-specifications.com/

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

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

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

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

[16] http://www.myspace.com/

[17] http://www.twitter.com/

[18] http://www.linkedin.com/

[19] http://www.netlog.com/

[20] http://www.reddit.com/

[21] http://digg.com/

[22] http://www.propeller.com/

[23] http://openlike.org/

[24] http://wiki.developers.facebook.com/index.php/Events.rsvp

nor does it take into account possibly interesting data coming from other social networks. That is why we created a *global profile*, which is aggregated from other profiles the user has in different user communities. This global profile is consumed together with the news events information by the recommendation system to yield recommendations of news items.

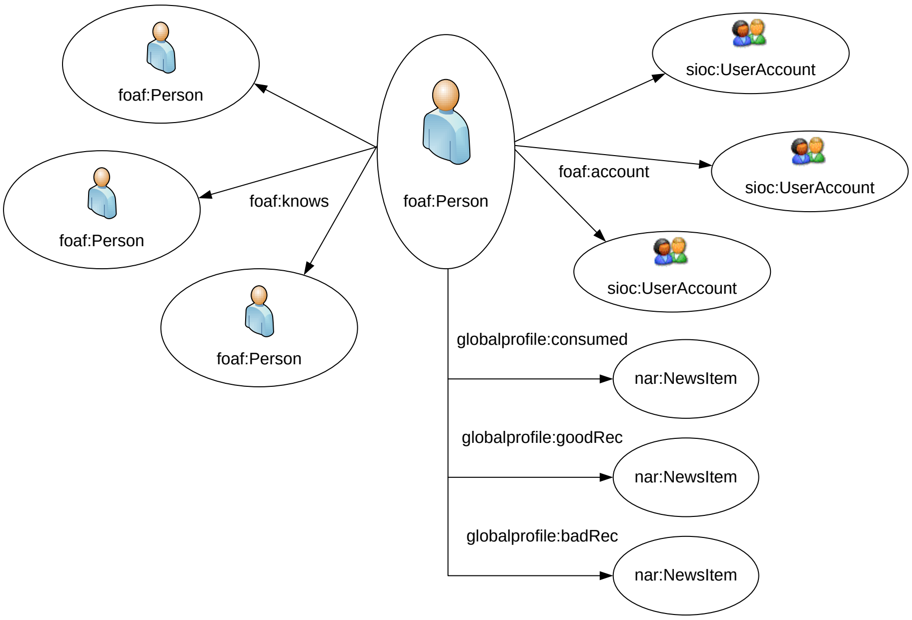

Fig. 2 The Global Profile.

This global profile (as can be seen in Figure 2) allows storing the necessary elements to yield a better profile, more useful to a recommendation system, because it combines information from different user communities. This information needs to be aggregated from the profiles the user has in several user communities. For this, we used an *OpenID*[25] identity provider service. It provides an identity, e.g., http://myname.newsfeed.be, together with a profile, i.e., the aggregated global profile. By letting users link this identity to the identities they already possess in different user communities, this identity service identifies uniquely the user with all his other identities. This *OpenID* identity service is used as authentication mechanism, for proving who you are and what your other identities are.

For populating the global profile, the identity service has connectors to *OpenID* identity providers, e.g., *Digg*, or *Facebook*. This way, the user can keep control over the global profile by selecting the identity providers he trusts. The global profile gets then aggregated from the identity services he has trusted. *OpenID* is a good authentication mechanism, but not a good authorisation mechanism. Indeed we need a mechanism

[25] http://openid.net/

for the authenticated user to explicitly permit the data/profile provider to use his *OpenID* credentials to connect to a profile provider and retrieve a particular piece of the user's private information. A combination of *OpenID* and *OAuth*[26] lets the user control his permissions for web services in a fine-grained manner. Our connectors use a combination of the *OpenID* protocol and the *OAuth* protocol for retrieving the profile information.

By providing these connectors, the user can also use this authentication information from the identity service to log on the platforms that support *OpenID* identification. At the same time any application, i.e., identity relying party, that consumes profile information can use this identity service as long as they have an *OpenID* connector as profile provider and authentication mechanism and an *OAuth* mechanism for authorisation.

6 News Recommendation System

In real life, we rely on recommendations from other people either by word of mouth, recommendation letters, surveys, or movie and book reviews printed in newspapers. Recommender systems assist and augment this natural social process [29]. According to Burke [3], recommender systems provide suggestions for items or content likely to be of use to a user, they act as 'personalized information agents'. Traditionally, recommender systems have been categorised into two main classes: *Content-Based* (CB) methods and *Collaborative Filtering* (CF) techniques.

Content-based or information filtering techniques generate recommendations by matching descriptive product information to the user's profile, or other user data [24]. This profile is often created and updated automatically in response to feedback on the desirability of items that have been presented to the user. Examples of CB techniques include decision trees, (linear) classifiers and probabilistic methods [27].

Collaborative or social recommenders are applicable to various types of content and also look at what people with similar interests and preferences like or liked: new content is recommended, starting from the identification and preferences of 'like-minded users'. CF techniques are based on the assumption that a good method to discover interesting content is to search for other people who have similar interests, and then recommend items that those similar users like [2]. Early research about CF systems has been conducted by the GroupLens research lab [29]. More advanced solutions like clustering models [34] and dependency network models [12] have been studied to improve the accuracy of the personal suggestions. In this context, Sarwar et al. proposed Singular Value Decomposition (SVD) to improve scalability of collaborative filtering systems by dimensionality reduction [31].

Content-based techniques do not consider the community knowledge [28]. In contrast, collaborative filtering tend to fail if little information is available about the user or the item (cold start problem), or if the user has uncommon interests. Therefore, hybrid content-based and collaborative recommenders have been explored to smooth out the disadvantages of each. These hybrid combinations have been studied in various

[26] http://oauth.net/

domains like movie recommenders [11] and online newspapers [5].

Nowadays online shops, like *Amazon*, apply CF to personalise their online webpages according to the needs of each customer [19]. Purchasing and rating behaviour are valuable information channels for online retailers to investigate consumer's interests and generate personalised recommendations [18]. CF algorithms are the most commonly-used recommendation techniques because they generally provide better results than CB techniques [13]. Because of the success of CF techniques for a big variety of items (books, DVDs, TV programs), it sounds logically to use the same recommendation techniques for suggesting news event items. However, some problems arise due to the inherent nature of time-specific items (events) [8].

To generate recommendations for a target user, user-based CF algorithms (UBCF) start by finding a set of neighbouring users whose consumed or rated items overlap the target user's consumed or rated items. Users can be represented as an N-dimensional vector of items, where N is the number of distinct catalogue items. Consumed or rated items are recorded in the corresponding components of this vector. However, this profile vector may remain extremely sparse (i.e., contain a lot of missing values) for the majority of users who consumed or rated only a very small fraction of the available catalogue items. Next, neighbourhoods of like-minded users are composed based on user similarity values. The similarity of two users, j and k, symbolised by their consumption vectors, U_j and U_k, can be measured in various ways. The most common method is to measure the cosine of the angle between the two vectors [30].

$$Sim(\mathbf{U}_j, \mathbf{U}_k) = cos(\mathbf{U}_j, \mathbf{U}_k) = \frac{\mathbf{U}_j \cdot \mathbf{U}_k}{||\mathbf{U}_j||\ ||\mathbf{U}_k||} \tag{1}$$

Next, the CF algorithm aggregates the items consumed by these similar neighbours, eliminates items the target user has already consumed or rated, and recommends the remaining items to the target user [19]. An alternative to this user-based CF technique is item-based CF, a technique that matches each of the user's consumed or rated items to similar items and then combines those similar items into a recommendation list. Measuring the similarity of items is based on the consumption behavior of the community, using the same metrics as with the user-based CF. If 2 products are frequently consumed together, they are considered as similar. Because of scalability reasons, this item-based technique is often used to calculate recommendations for big online shops, like *Amazon*, where the number of users is magnitudes bigger than the number of items.

Despite its popularity, CF is not generally applicable due to the sparsity problem, which refers to the situation that the consumption data in the profile vectors are lacking or insufficient to calculate reliable recommendations. Especially news systems suffer from sparse data sets, since most users only consume/read a small fraction of all the available news events. A direct consequence of a sparse data matrix is that the number of neighbours for a user/item might be very limited in a user-based/item-based CF system. Indeed, the majority of the similarity metrics that are used in CF systems rely on the vector overlap to determine the similarity of two users/items. Sparse profile vectors induce a limited overlap, which obstructs the creation of accurate and extensive neighbourhoods of like-minded people or similar items. Furthermore, because of this sparsity, the majority of these neighbours may also have a small number of consumptions in their profile vectors. Because the prospective personal recommendations are limited to this set of consumptions of neighbours, the variety, quality, and quantity of the final recommendation list might be inadequate.

To provide highly accurate recommendations even if data profiles are sparse, some solutions are proposed in literature [26]. Most of these techniques use *trust inferences*, transitive associations between users that are based on an underlying social network, to deal with the cold-start and sparsity problems [35]. Nevertheless, these underlying social networks are in many cases insufficiently developed or even nonexistant for (new) web-based applications that like to offer personalised content recommendations. *Default voting* is an extension to the traditional CF algorithms which tries to solve this sparsity problem without exploiting a social network. A default value is assumed as "vote" for items without an explicit rating or consumptions. Although this technique enlarges the profile overlap, it cannot identify more significant neighbours than the traditional CF approach. Grouping people or items/events into clusters based on their similarity can be an other solution, but finding the optimal clusters is a tricky task [7].

Therefore, we propose an advanced hybrid recommendation algorithm (illustrated in Figure 3) which takes into account the sparsity and content-based features of news event systems. This robust algorithm (named UBExtended) extends the user profiles with additional consumption data, making the profile vectors less sparse. The items that have the highest probability to be consumed by the user in the near future are added as probable consumptions to the user's profile. These probabilities are estimated by calculating the temporary personal recommendations with a traditional CF algorithm based on the user's profile as a priori knowledge (Status A in Figure 3). The probability is inversely proportional to the index of the item in a regular top-N recommendation list, and can be estimated by the confidence value which is associated with the recommendation. After all, a top-N recommendation list is a prediction of the items which the user will like/consume in the near future. Based on these calculated probabilities, the profiles are extended until the minimum profile density is reached (Status B in Figure 3). To emphasize the uncertainty, the predicted consumptions are marked as "potential" in contrast to the initial assured consumptions. For a news event service, e.g., the real news item consumption correspond to a 1 in the consumption vector, which refers to a 100% guaranteed consumption, while the potential future consumptions are represented by a decimal value between 0 and 1, according to the confidence value, in the profile vector.

Based on these extended profile vectors, the similarities are recalculated with the chosen similarity metric, e.g., the cosine similarity (equation 1). Because of the added 'future consumptions', the profile overlap and accordingly the number of neighbours are increased compared to the traditional CF. This profile extension might be an iterative process consisting of calculating standard CF recommendations (Status A in Figure 3) and adding potential future consumptions to the profile (Status B in Figure 3). To produce personal suggestions, a recommendation vector is generated based on these extended profile vectors. The recommendation vector, R_j, for target user j can be calculated as:

$$\mathbf{R}_j = \frac{\sum_{k=1,k\neq j}^{M} \mathbf{U}_k \cdot Sim(\mathbf{U}_j, \mathbf{U}_k)}{\sum_{k=1,k\neq j}^{M} Sim(\mathbf{U}_j, \mathbf{U}_k)} \tag{2}$$

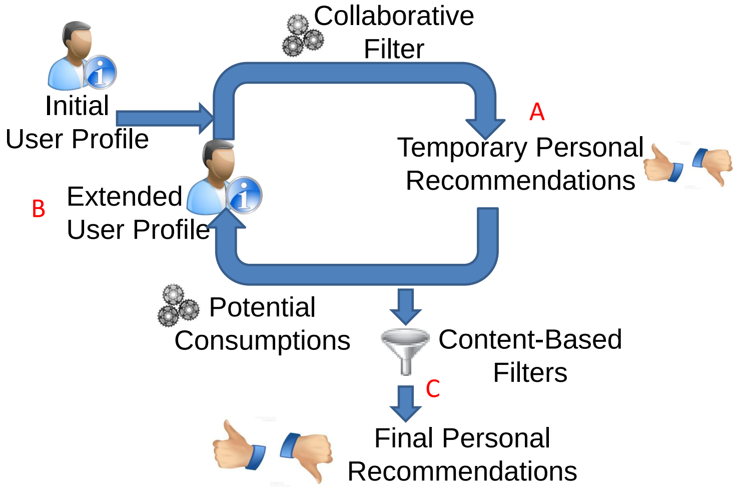

Fig. 3 The dataflow of the recommendation algorithm

where M stands for the number of users in the neighbourhood of the target user. U_j and U_k represent the consumption vectors of users j and k, which might contain real values. Subsequently, the top-N recommendations are obtained by taking the indices of the highest components of the recommendation vector, R_j, and eliminating the items which are already consumed by user j in the past.

By adopting the NewsML-G2 standard as event model, we provide a uniform event structure that can be utilised for semantic analysis. By enriching the news stories based on a set of knowledge sources, we can generate a memory of semantic competencies that can be exploited to reason about the content as well as to support the user profiling and recommendation process. These knowledge sources allow the system to make inferences based on the underlying reasons for which a user may or may not be interested in a particular event [23]. As a result, personal selection criteria regarding news events can be incorporated by content-based post-filters to complete the recommendation process (Status C in Figure 3). The current implementation of the filter is based on a text matching between the personal selection criteria of the user and the metadata of the news events as well as the metadata originating from the enrichment process. However the system can easily be extended with more advanced knowledge-based filters. These contextual filters operate after the main recommendation algorithm and remove or penalise the candidate recommendations which do not satisfy the personal selection criteria. These personal selection criteria, which can be specified by the end-user, are for example the location where the event happened, the language, the category, or the date of the news event. The contextual filters are provided as post-filters that affect the suggestion list after calculating the CF-based recommendations to enable a real-time filtering. Since the quadratic nature of CF-based algorithms, the recommendations can not be calculated at the time of request, but have to be calculated in advance (e.g. during an overnight process). Filtering the recommendations based on the personal selection criteria of the user is a process with a duration that is linear in the number of items. So, by employing the content-based filters in a post-process, a real-time filtering of the recommendations is possible. This allows the end-users to alter their selection criteria while checking out their recommendations.

To evaluate the proposed algorithm, it was benchmarked against the standard UBCF algorithm by means of a data set with historical consumption behaviour. At the present time, we are collecting rating and click behavior (i.e., event consumptions) on a Flemish event website, which will enable us to analyse the proposed recommendation algorithm based on user preferences regarding events. Unfortunately, the current amount of data is still rather limited and very sparse (291,751 consumptions (ratings), 33,348 events, and 147,957 users, or an average of 2 consumed events per user) to produce accurate user profiles and personal suggestions. Because of the extra uncertainty associated to click behavior, we opted to use only the ratings for this experiment. Rating an item is a strong expression of a user's preference for that item. Clicking on a web page of an item does not mean that the user really likes the item. As a result, click behavior introduces an extra uncertainty in the recommender process. Since the sparsity of the data set, a top-N recommendation list of (only) 10 recommendations per user is generated during this first experiment.

For evaluation purposes, we used 50% of the consumptions (the most recent ones) as the test set and the remaining 50% of the consumption records as potential input data. In order to study the performance of the algorithm under data of different sparsity levels, we created ten different training sets by selecting the first 10%, 20%, 30%, until 100% of the input data. The recommendation algorithm used these different training sets in successive iterations to generate personal suggestions which were compared to the test set. Afterwards, the recommendations are evaluated based on the commonly-used classification accuracy metrics: precision and recall[4]. In this simulation, the extended CF algorithm used 2 iterations to make the profile vectors denser. After 2 iterations, the number of similarities is approximately doubled for sparse data sets like the one used in our test, i.e. twice as much neighbors are discovered. Although more iterations will further decrease the sparsity of the profile vectors, thereby increasing the diversity and serendipity of the recommendations, extra iterations might have a negative impact on the accuracy of the recommendations (i.e. the recommendations might induce less user interaction). Each iteration introduces a cumulative uncertainty by considering the temporary recommendations as potential future consumptions. After 2 iterations the number of similarities is approximately doubled for sparse data set like this one, i.e. twice as much neighbors are discovered. As can be noticed from Figure 4, showing the results, the absolute values of the accuracy metrics are very low because of the limited data set. However, we clearly witness an improvement of the extended CF algorithm (UBExtended), compared with the traditional CF algorithm (UBCF). In order to be able to interpret the values, we included the results of the random recommender, a recommender that randomly suggests items (that are not yet consumed by the user who gets the recommendations). This way precision and recall can be compared against these random suggestions. The graphs show that the recommendations of UBCF and UBExtended are much more accurate than the recom-

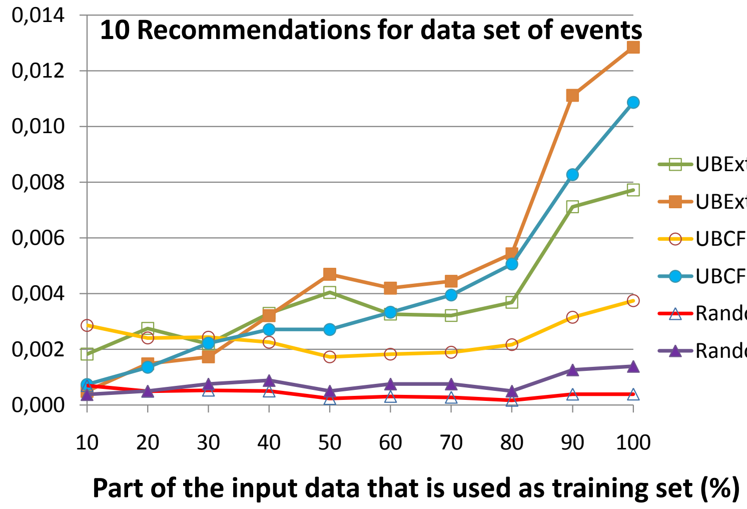

Fig. 4 The benchmarks of the recommendation algorithms (UBCF and UBExtended) on a data set of events

mendations of the random recommender.

To verify the accuracy improvements by extending the user profiles, we evaluated our algorithm by a second experiment, based on a bigger data set with similar characteristics as a data set on news events. Therefore, we used a data set of a user-generated content site (PianoFiles), which has a sparsity that is realistic for a news event recommendation system, in contrast to the sparsity of the data sets that are commonly used for benchmarking recommendation algorithms (like Movielens[27] or Netflix[28]). PianoFiles[29] is a user-generated content site that offers users the opportunity to exchange, browse and search for sheet music they like to play. The data set contains 401,593 items (music sheets), 80,683 active users and 1,553,586 consumptions (i.e., sheets added to the personal collections of the user) in chronological order. In this data set, users have on average 19 music sheets in their collection, providing much more information than the user profiles of the first experiment. Since this data set contains more information, we generated a top-N recommendation list of 20 personal suggestions for every user in the system.

In this second test, the same evaluation methodology is used as described above (for the first experiment). Figure 5 illustrates the evaluation metrics for the three recommendation algorithms (UBCF, UBExtended, and Random) and proves that the proposed algorithm (UBExtended) outperforms the standard UBCF and the random recommender for the two evaluation metrics (precision and recall) and for different amounts of training data. Due to the large content offer (401,593 items) and the sparsity of the data set, recommendation algorithms have still a hard job to suggest the most

[27] http://www.grouplens.org/node/73

[28] http://www.netflixprize.com/

[29] http://www.pianofiles.com/

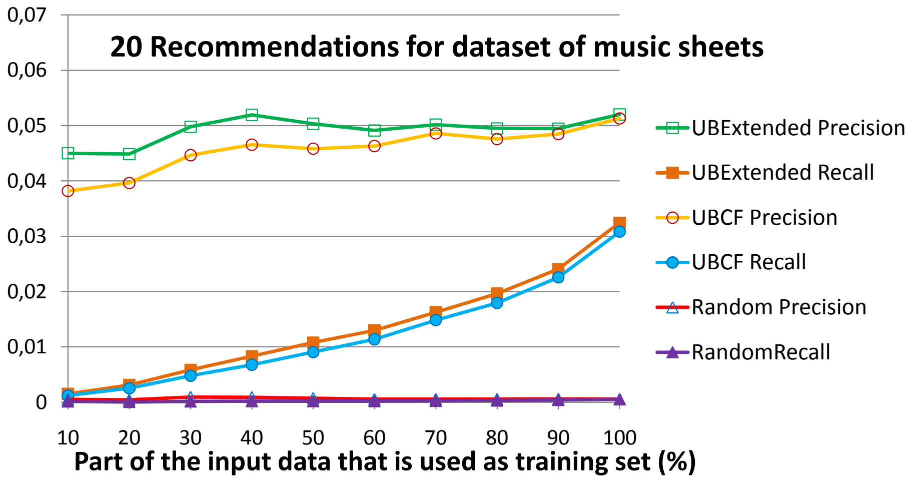

Fig. 5 The benchmarks of the recommendation algorithms (UBCF and UBExtended) on a data set of user-generated content (PianoFiles)

appropriate items for every user. Because of this, the absolute values of the evaluation metrics remain rather low. However, precision and recall values between 1 and 10% are very common in benchmarks of recommendation algorithms on sparse data sets [14].

7 News Distribution

As we have enriched our semantic news items (see Section 4), we can now start publishing them as Linked Open Data [1] (LOD) using a Jetty server[30]. These news items are distributed to the end user via VRT's portal site [footnote:deredactie] (beta of this platform with recommendation feature available by fall). This portal relies on the *OpenID* identity service for authentication (see Section 5), on the recommendation engine (see Section 6) for getting the rightly targeted news items, and on the LOD server for the effective, enriched information of these news items. Here, people can find the latest news items, search for particular new items, and view their personal recommended news items based on their global profile by exploring it using our faceted browser.

Because news is very volatile and we want the user constantly updated on new/developing news items, we offer them a *personal* RSS feed - containing a unique URI for each individual registered OpenId -. This personal RSS feed contains updates on the top 20 recommended news items for that user. The recommendation engine only takes news items of no more then five days old into account and for performance reasons all newly recommended (developing) news items are aggregated and pushed to the end-users only twice a day. These feed items a.o. things contain a link to the Linked Open Data published news items, a description of these news items, their date and

[30] http://jetty.codehaus.org/jetty/

their location. By providing such a *dynamic* personal RSS feed, which is updates every day, the users stay on top of the latest news items, they like.

8 Conclusions

In this paper, we have presented a semantic version of the NewsML-G2 standard as a unifying (meta)datamodel dealing with dynamic distributed news event information. Using that ontology as a data communication interface within VRT's end-to-end news distribution architecture, several services (aggregation, categorisation, enrichment, profiling, recommendation, and distribution) were hooked in the workflow engine giving our Flemish broadcaster a tool to automatically recommend (developing) news stories 1-to-1 to the targeted customer for the first time.

At the same time, we provided the (inter)national (news) community with mechanisms to describe and exchange news event and profile information in a standardised way. We demonstrated the concepts of generic data portability of user profiles, and how to generate recommendations based on such a global profile – within which we integrated information fields from all the different social networks the user wanted to share –. Our ideas were implemented with open standards like OpenID, OAuth, and OpenLike, thus keeping the architecture open for other news event providers and profile providers.

Acknowledgements The research activities as described in this paper were funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT), the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research-Flanders (FWO-Flanders), and the European Union.

References

1. Bizer, C., et al.: Linked Data – The Story So Far. International Journal on Semantic Web and Information Systems **5**(3), 1–22 (2009)
2. Breese, J., Heckerman, D., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 43–52. Madison, USA (1998)
3. Burke, R.: The adaptive web. chap. Hybrid web recommender systems, pp. 377–408. Springer-Verlag, Berlin, Heidelberg (2007). URL `http://portal.acm.org/citation.cfm?id=1768197.1768211`
4. Campochiaro, E., et al.: Do metrics make recommender algorithms? Advanced Information Networking and Applications Workshops, International Conference on **0**, 648–653 (2009). DOI http://doi.ieeecomputersociety.org/10.1109/WAINA.2009.127
5. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M.: Combining content-based and collaborative filters in an online newspaper. In: Proceedings of ACM SIGIR Workshop on Recommender Systems (1999)
6. Corcoran, S.: Using Social Applications in Ad Campaigns (2009). Available at `http://www.forrester.com/Research/Document/Excerpt/0,7211,54050,00.html`
7. Cornelis, C., et al.: Clustering Methods for Collaborative Filtering. In: Proceedings of the 15th National Conference on Artificial Intelligence – Workshop on Recommendation Systems, pp. 114–129. Madison, USA (1998)
8. Cornelis, C., et al.: A Fuzzy Relational Approach to Event Recommendation. In: Proceedings of the 1st Indian International Conference on Artificial Intelligence, pp. 2231–2242. Pune, India (2005)
9. De Geyter, M., et al.: File-based Broadcast Workflows: on MAM Systems and their Integration Demands. SMPTE Motion Imaging Journal (11–12), 38–46 (2008)

10. De Sutter, R., et al.: Automatic News Production. In: Proceedings of the International Broadcasting Conference, pp. 158–165. Amsterdam, The Netherlands (2008)
11. Good, N., Schafer, J.B., Konstan, J.A., Borchers, A., Sarwar, B., Herlocker, J., Riedl, J.: Combining collaborative filtering with personal agents for better recommendations. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence, pp. 439–446 (1999)
12. Heckerman, D., Chickering, D.M., Meek, C., Rounthwaite, R., Kadie, C.: Dependency networks for inference, collaborative filtering, and data visualization. J. Mach. Learn. Res. **1**, 49–75 (2001). DOI http://dx.doi.org/10.1162/153244301753344614. URL `http://dx.doi.org/10.1162/153244301753344614`
13. Herlocker, J., et al.: An Algorithmic Framework for Performing Collaborative Filtering. In: Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 230–237. Berkeley, USA (1999)
14. Huang, Z., et al.: A comparison of collaborative-filtering recommendation algorithms for e-commerce. IEEE Intelligent Systems **22**(5), 68–78 (2007). DOI http://dx.doi.org/10.1109/MIS.2007.80
15. International Press Telecommunications Council: NewsML-G2 Specification – Version 2.2 (2009). Available at `http://www.iptc.com/std/NewsML-G2/NewsML-G2\_2.2.zip`
16. Iskold, A.: The Art, Science and Business of Recommendation Engines (2007). Available at `http://www.readwriteweb.com/archives/recommendation_engines.php`
17. ITU-T and ISO/IEC: Advanced Video Coding for Generic Audiovisual Services (2003). ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC
18. Karypis, G.: Evaluation of Item-Based Top-N Recommendation Algorithms. In: Proceedings of the 10th International Conference on Information and Knowledge Management, pp. 247–254. Atlanta, USA (2001)
19. Linden, G., et al.: Amazon.com Recommendations: Item-to-item Collaborative Filtering. IEEE Internet Computing **7**(1), 76–80 (2003)
20. Mannens, E., et al.: Production and Multi-channel Distribution of News. Multimedia Systems – Special Issue on Canonical Processes of Media Production **14**(6), 359–368 (2008)
21. Mannens, E., et al.: Automatic Information Enrichment in News Production. In: Proceedings of the 10th International Workshop on Image Analysis for Multimedia Interactive Services, pp. 61–64. London, United Kingdom (2009)
22. McGuinness, D., et al. (eds.): OWL Web Ontology Language: Overview. W3C Recommendation. World Wide Web Consortium (2004). Available at `http://www.w3.org/TR/owl-features/`
23. Mobasher, B., Jin, X., Zhou, Y.: Semantically enhanced collaborative filtering on the web. In: Proceedings of the 1st European Web Mining Forum (EWMF2003), pp. 57–76 (2003). URL `http://www.springerlink.com/content/y8bd5n544j91wc8w`
24. Mooney, R.J., Roy, L.: Content-based book recommending using learning for text categorization. In: Proceedings of the fifth ACM conference on Digital libraries, DL '00, pp. 195–204. ACM, New York, NY, USA (2000). DOI http://doi.acm.org/10.1145/336597.336662. URL `http://doi.acm.org/10.1145/336597.336662`
25. O'Reilly, T.: What is Web 2.0 – Design Patterns and Business Models for the Next Generation of Software (2005). Available at `http://oreilly.com/pub/a/web2/archive/what-is-web-20.html?page=1`
26. Papagelis, M., et al.: Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences. Lecture Notes in Computer Science – Trust Management **3477**, 224–239 (2005)
27. Pazzani, M.J., Billsus, D.: The adaptive web. chap. Content-based recommendation systems, pp. 325–341. Springer-Verlag, Berlin, Heidelberg (2007). URL `http://portal.acm.org/citation.cfm?id=1768197.1768209`
28. Popescul, A., Ungar, L.H., Pennock, D.M., Lawrence, S.: Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, UAI '01, pp. 437–444. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001). URL `http://portal.acm.org/citation.cfm?id=647235.720088`
29. Resnick, P., Varian, H.R.: Recommender systems. Commun. ACM **40**, 56–58 (1997). DOI http://doi.acm.org/10.1145/245108.245121. URL `http://doi.acm.org/10.1145/245108.245121`
30. Sarwar, B., et al.: Analysis of Recommendation Algorithms for E-commerce. In: Proceedings of the 2nd ACM Conference on Electronic Commerce, pp. 158–167. Minneapolis, USA (2000)

31. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.T.: Application of dimensionality reduction in recommender system - a case study (2000). URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.38.744`
32. SMPTE: Material Exchange Format (MXF) – File Format Specification (2004). SMPTE 377M
33. Troncy, R.: Bringing the IPTC News Architecture into the Semantic Web. In: 7^{th} International Semantic Web Conference (ISWC'08), pp. 483–498. Karlsruhe, Germany (2008)
34. Ungar, L., Foster, D.: Clustering Methods For Collaborative Filtering. In: Proceedings of the Workshop on Recommendation Systems, pp. 114–129. AAAI Press, Menlo Park California (1998). URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.4026`
35. Weng, J., et al.: Trust-based agent community for collaborative recommendation. In: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1260–1262. Hakodate, Japan (2006)