Automatic News Recommendations via Profiling

Erik Mannens
Ghent University - IBBT
ELIS - Multimedia Lab
9050 Ghent, Belgium
erik.mannens@ugent.be

Sam Coppens
Ghent University - IBBT
ELIS - Multimedia Lab
9050 Ghent, Belgium
sam.coppens@ugent.be

Toon De Pessemier
Ghent University - IBBT
INTEC - WiCa
9050 Ghent, Belgium
tdpessem@intec.ugent.be

Hendrik Dacquin
VRT
VRT-medialab
1043 Brussels, Belgium
hendrik.dacquin@vrt.be

Davy Van Deursen
Ghent University - IBBT
ELIS - Multimedia Lab
9050 Ghent, Belgium
davy.vandeursen@ugent.be

Rik Van de Walle
Ghent University - IBBT
ELIS - Multimedia Lab
9050 Ghent, Belgium
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.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: :Miscellaneous; D.2.11 [**Software Engineering**]: :Software Architectures [Domain specific architectures]

General Terms

Design, Management, Standardization

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. In particular, we automated the production of multiple versions of news bulletins for different consumption platforms since news broadcasters generally aggregate and produce more material than is required for broadcast or online distribution. 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] by merging different news sources and using the NewsML-G2 specification[2]. While the impact of file-based production indeed mainly affected the work methods of the news production staff - journalists, anchors, editorial staff, etc - [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[3] 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] http://www.iptc.com/std/NewsML-G2/NewsML-G2_2.2.zip
[3] http://www.w3.org/TR/owl-features/

2. NEWS MODELLING

The IPCT News Architecture framework (NAR[4]) 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[5] 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 inline 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[6] to address these shortcomings and we have discussed the design decisions regarding its modeling from existing XML Schemas [10].

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 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*[7] (MXF) instance before the MAM can process it. Afterwards, the essence is transcoded into a consumer format, such as H.264/ AVC[8], 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[9] and the *OpenCalais* infrastructure[10] 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*[11] for the locations, or in *DBPedia*[12]/*FreeBase*[13] for the persons, organisations and events. The string 'Tom Barman' is therefore mapped to its URI in *DBPedia*[14] 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.

[4] http://www.iptc.org/NAR/
[5] http://iptc.cms.apa.at/std/EventsML-G2/
[6] http://www.ebu.ch/metadata/ontologies/NML2/
[7] MaterialExchangeFormat(MXF)--FileFormatSpecification,SMPTE377M
[8] AdvancedVideoCodingforGenericAudiovisualServices,ITU-TRec.H.264andISO/IEC14496-10AVC
[9] http://www.iknow.be/
[10] http://www.opencalais.com/
[11] http://www.geonames.org/
[12] http://dbpedia.org/
[13] http://www.freebase.com/
[14] http://dbpedia.org/resource/Tom_Barman

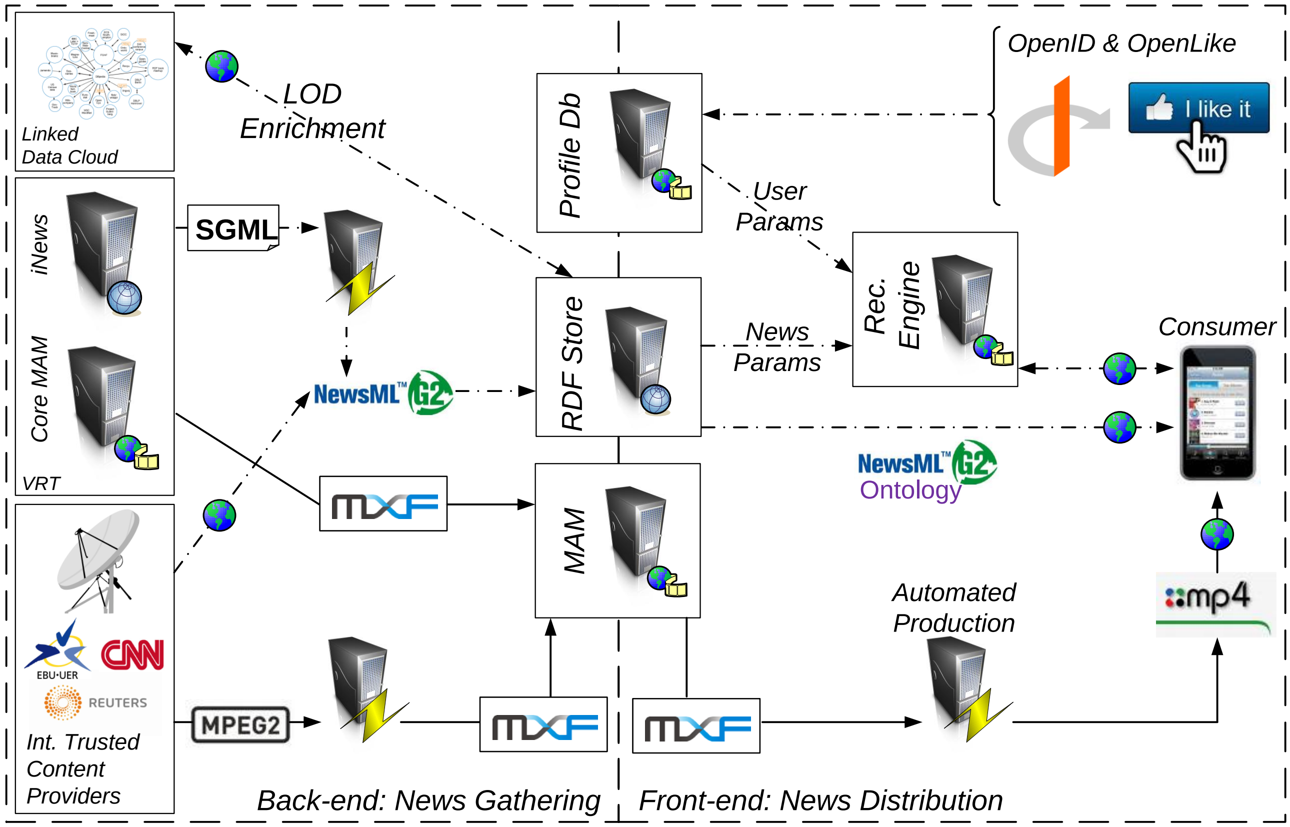

Figure 1: End-to-end News Distribution Architecture

5. USER PROFILING

Recent years technologies have appeared that empower the user to have more control over his experience. E.g., RSS[15] 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[16]. 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*[17] 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[18]. *Pandora*[19] 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[20]. The giant in the space remains *Facebook*[21], which gets 87.7 million unique viewers per month, according to *ComScore*[22]. 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*[23], *Twitter*[24], *LinkedIn*[25] and *Netlog*[26] are among the more popular ones. There are in fact already also a number of very popular social news websites, a.o. *Reddit*[27], *Digg*[28], and *Propeller*[29]. 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[30] (see Figure 1); 3) *the social graph*: this contains all the user's connections to other users, e.g., a friendlist.

[15] RSS:ReallySimpleSyndication,alsoseehttp://www.rss-specifications.com/
[16] http://oreilly.com/pub/a/web2/archive/what-is-web-20.html?page=1
[17] http://www.amazon.com/
[18] http://www.readwriteweb.com/archives/recommendation_engines.php
[19] http://www.pandora.com/
[20] http://www.forrester.com/Research/Document/Excerpt/0,7211,54050,00.html
[21] http://www.facebook.com/
[22] http://www.comscore.com/
[23] http://www.myspace.com/
[24] http://www.twitter.com/
[25] http://www.linkedin.com/
[26] http://www.netlog.com/
[27] http://www.reddit.com/
[28] http://digg.com/
[29] http://www.propeller.com/
[30] http://openlike.org/

Current recommendations are mostly offered within the closed context of a single community: e.g. *Facebook* recommends events based on RSVP event invitations[31] 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, 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.

This global profile 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*[32] 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 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*[33] 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

The overabundance of (news) content and the related difficulty to discover interesting (news) content items have already been addressed in several contexts. Online shops, like *Amazon*, apply *Collaborative Filtering* (CF) to personalize the online store according to the needs of each customer [8]. Purchasing and rating behaviour are valuable information channels for online retailers to investigate consumer's interests and generate personalized recommendations [7]. Because of the success of recommendation 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 events [4].

CF techniques are the most commonly-used recommendation algorithms because they generally provide better results than *content-based* (CB)techniques [5]. Most user-based CF algorithms (UBCF) start by finding a set of neighbours whose consumed or rated items overlap the 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. For the big majority of users who consumed or rated only a very small fraction of the catalogue items, this vector is extremely sparse. The similarity of two users, j and k, symbolized 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 [12].

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

Next, the algorithm aggregates the consumed items from these similar neighbours, eliminates items the user has already consumed or rated, and recommends the remaining items to the user [8]. 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. For measuring the similarity of items, the same metrics can be employed as with the user-based CF. Because of scalability reasons, this 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 the popularity of CF, its applicability is limited 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 datasets, 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 lead to 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 will 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.

In an attempt to provide high-quality recommendations even when data profiles are sparse, some solutions are pro-

[31] http://wiki.developers.facebook.com/index.php/Events.rsvp
[32] http://openid.net/
[33] http://oauth.net/

posed in literature [11]. Most of these techniques use trust inferences, transitive associations between users that are based on an underlying social network, to deal with the sparsity and the cold-start problems [13]. Nevertheless, these underlying social networks are in many cases insufficiently developed or even nonexisting for (new) web-based applications that desire to offer personalized 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 purchase. 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 [3].

Therefore, our developed algorithm (named UBExtended) extends the user profiles with additional consumption data. The items that have the highest probability to be consumed by the user in the future will be added as probable consumptions to the user profile. These probabilities are calculated based on the user's profile as a priori knowledge. The probability will be inverse proportional to the index of the item in a regular top-N recommendation list, and can be estimated by the confidence value which is calculated by a traditional CF system. After all, this 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 will be completed until the minimum profile threshold is reached. However, these predicted consumptions will be marked as uncertain 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 probability value, in the profile vector.

Based on these extended profile vectors, the similarities will be 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 will be increased compared to the traditional CF. To produce personal suggestions, a recommendation vector will be generated based on these extended profile vectors. The recommendation vector, R_j, for user j can be calculated as:

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

where 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. Finally, in order to take into account personal news event selection criteria, we added some contextual post-filters to the recommendation system. These filters will 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

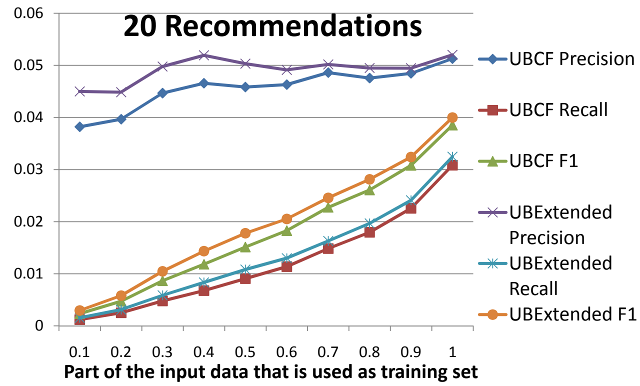

Figure 2: The benchmarks of the recommendation algorithms: UBCF and UBExtended

the end-user, are for example the location where the event happened, the language, the category, or the date of the news event.

To evaluate the proposed algorithm, it was benchmarked against the standard UBCF algorithm by means of a data set with consumption behaviour. Since we do not possess a data set with consumption behavior based on news events, we evaluated our algorithm based on a data set of PianoFiles[34]. PianoFiles 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. Since the sparsity of this data set 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[35] or Netflix[36]), we preferred the PianoFiles data set for the evaluations.

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. This way, a top-N recommendation list of 20 personal suggestions was produced for every user in the system. Afterwards, the recommendations are evaluated based on the commonly-used classification accuracy metrics: precision, recall and F1 [2].

Figure 2 illustrates these evaluation metrics for the two recommendation algorithms (UBCF and UBExtended) and proves that the proposed algorithm (UBExtended) outperforms the standard UBCF in all three evaluation metrics (precision, recall and F1) and for different amounts of train-

[34] http://www.pianofiles.com/
[35] http://www.grouplens.org/node/73
[36] http://www.netflixprize.com/

ing data. Due to the large content offer (401,593 items) and the sparsity of the data set, recommendation algorithms have a hard job to suggest the most appropriate content items for every user. Because of this, the absolute values of the evaluation metrics seem rather low. However, precision and recall values between 1 and 10% are very common in benchmarks of recommendation algorithms on sparse data sets [6].

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[37]. 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 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.

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

9. ACKNOWLEDGMENTS

The research activities that have been described in this paper were funded by Ghent University, VRT-medialab, IBBT-Flanders, IWT-Flanders, FWO-Flanders, and the EU.

References

[1] C. Bizer and et al. Linked Data – The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, July 2009.

[2] E. Campochiaro and et al. Do metrics make recommender algorithms? *Advanced Information Networking and Applications Workshops, International Conference on*, 0:648–653, 2009.

[3] C. Cornelis and et al. Clustering Methods for Collaborative Filtering. In *Proceedings of the 15th National Conference on Artificial Intelligence – Workshop on Recommendation Systems*, pages 114–129, Madison, USA, July 1998.

[4] C. Cornelis and et al. A Fuzzy Relational Approach to Event Recommendation. In *Proceedings of the 1st Indian International Conference on Artificial Intelligence*, pages 2231–2242, Pune, India, December 2005.

[5] J. Herlocker and 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*, pages 230–237, Berkeley, USA, August 1999.

[6] Z. Huang and et al. A comparison of collaborative-filtering recommendation algorithms for e-commerce. *IEEE Intelligent Systems*, 22(5):68–78, 2007.

[7] G. Karypis. Evaluation of Item-Based Top-N Recommendation Algorithms. In *Proceedings of the 10th International Conference on Information and Knowledge Management*, pages 247–254, Atlanta, USA, October 2001.

[8] G. Linden and et al. Amazon.com Recommendations: Item-to-item Collaborative Filtering. *IEEE Internet Computing*, 7(1):76–80, January 2003.

[9] E. Mannens and et al. Production and Multi-channel Distribution of News. *Multimedia Systems – Special Issue on Canonical Processes of Media Production*, 14(6):359–368, December 2008.

[10] E. Mannens and et al. Automatic Information Enrichment in News Production. In *Proceedings of the 10th International Workshop on Image Analysis for Multimedia Interactive Services*, pages 61–64, London, United Kingdom, May 2009.

[11] M. Papagelis and et al. Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences. *Lecture Notes in Computer Science – Trust Management*, 3477:224–239, May 2005.

[12] B. Sarwar and et al. Analysis of Recommendation Algorithms for E-commerce. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, pages 158–167, Minneapolis, USA, October 2000.

[13] J. Weng and et al. Trust-based agent community for collaborative recommendation. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1260–1262, Hakodate, Japan, May 2006.