Deliverable D4.7 Evaluation and final results

Daniel Stein / Fraunhofer, Germany
Dorothea Tsatsou, Pantelis Ieronimakis, Vasileios Mezaris / CERTH, Greece
Tomas Kliegr, Jaroslav Kuchař / UEP, Czech Republic
Matei Mancas, Francois Rocca, Pierre-Henri Dedecken / UMons, Belgium

07/04/2015

Work Package 4: Contextualisation and Personalisation

LinkedTV

Television Linked To The Web

Integrated Project (IP)

FP7-ICT-2011-7. Information and Communication Technologies

Grant Agreement Number 287911

Dissemination level	PU
Contractual date of delivery	31/03/2015
Actual date of delivery	07/04/2015
Deliverable number	D4.7
Deliverable name	Evaluation and final results
File	LinkedTV_D4_7.tex
Nature	Report
Status & version	Final & V1.0
Number of pages	40
WP contributing to the deliverable	4
Task responsible	Fraunhofer
Other contributors	
Author(s)	Daniel Stein / Fraunhofer, Germany Dorothea Tsatsou, Pantelis Ieronimakis, Vasileios Mezaris / CERTH, Greece Tomas Kliegr, Jaroslav Kuchař / UEP, Czech Republic Matei Mancas, Francois Rocca, Pierre-Henri Dedecken / UMons, Belgium
Reviewer	Michiel Hildebrand / CWI
EC Project Officer	Thomas Küpper
Keywords	head pose estimation, kinect, user trials, implicit contextualized profiling, client-side recommendation
Abstract (for dissemination)	This deliverable covers all the aspects of evaluation of the overall LinkedTV personalization workflow, as well as re-evaluations of techniques where newer technology and / or algorithmic capacity offer new insight into the general performance. The implicit contextualized personalization workflow, the implicit uncontextualized workflow in the premises of the final LinkedTV application, the advances in context tracking given new technologies emerged and the outlook of video recommendation beyond LinkedTV is measured and analyzed in this document.

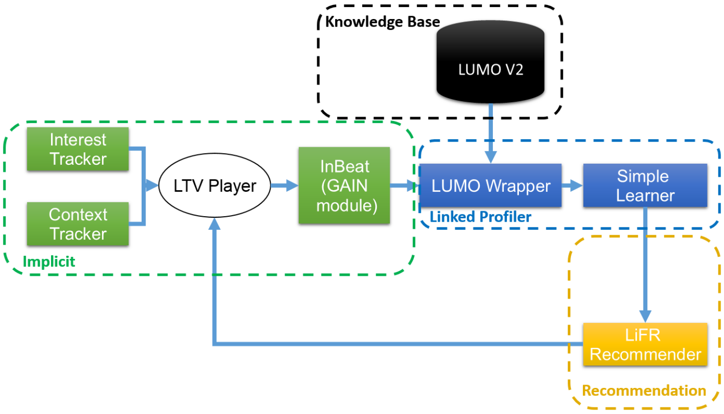


Figure 1: The WP4 core implicit personalization and contextualization workflow.

1 Introduction

Content annotation and enrichment within LinkedTV produces arbitrarily large amounts of quality links to the web. While this offers much content that is of general interest to a particular media item, the sheer amount can be overwhelming. The core motivation of this Workpackage 4 on personalization and contextualization is to filter and prioritize the content that is offered to a user, based on both his or her general preferences as well as his or her current context and situation.

The architecture used to achieve this has already been described extensively in ”Content and Concept Filter v2” (D 4.5) and ”Contextualisation solution and implementation” (D 4.6). In summary (cf. Figure 1), the Kinect-based behavioural Interest/Context Tracker sends events to the player. The player enriches the events with the video ID and time when the events occurred and passes them to the GAIN module, to enable retrieval of the specific media fragment for which an Interest/Context event was manifested. In addition to the behavioural Interest/Context events, the player also sends player interaction events (like pause, play, bookmark etc. . .) using the same channel to GAIN.

The GAIN module fuses this data and provides a singular measure of user interest for all the entities describing a given media fragment and in a given context (alone, with other people, etc. . .). In addition, the InBeat Preference Learning (PL) module (cf. D4.4, D4.6) detects associations between entities for a given user, which it formulates in association rules. This information is sent to the model building step, namely the Linked Profiler. This step comprises conveying entities into the ontology LUMO via the LUMO Wrapper utility and using this data to progressively learn user preferences over time based on the Simple Learner component. Finally, the user models created by the Linked Profiler are passed onto the LiFR-based recommender, along with the entities of candidate content and concepts to be recommended. Finally, the recommender matches user preferences to entities in the candidate dataset and as a result provides recommendations over this data.

The tools necessary to offer such functionality have already reached their maturity state over the course of the LinkedTV project. Several evaluation activities have been conducted in attendance to their development process and thus already reported in the aforementioned deliverables D 4.5 and D 4.6. Some techniques using the implicit personalization ontology LUMO, such as topic labelling, consist of content annotation services and are as such described in ”Final Linked Media Layer and Evaluation” (D 2.7).

This document covers all the aspects of evaluation that are based on the overall workflow as well as re-evaluations of techniques where newer technology and / or algorithmic capacity offer new insight into the general performance.

1.1 History

1.2 Document scope and structure

This document is organised as follows: in Section 2, we measure the performance gain for the contextualization feature ”user head gaze” with different algorithms and evaluate against marker-based ground truth. Further, we evaluate the main difference between Kinect v 1 and Kinect v 2 in our setting. The implicit contextualized profile learning and recommendation approach in terms of scripted user behaviour

Table 1: History of the document

Date	Version	Name	Comment
04/06/2014	V0.01	Stein, Fraunhofer	template
04/02/2015	V0.1	Stein, Fraunhofer	ToC, general introduction
11/03/2015	V0.2	Kliegr, UEP	Outlook subsection
19/03/2015	V0.3	Kuchař, UEP	GAIN evaluation
20/03/2015	V0.31	Kliegr, UEP	GAIN evaluation polishing
23/03/2015	V0.4	Tsatsou, CERTH	First input chapter 4
23/03/2015	V0.5	Mancas, UMons	Input Kinect evaluation, Sec. 2
25/03/2015	V0.51	Tsatsou, CERTH	First input Linked Profiler evaluation chapter 3
26/03/2015	V0.6	Stein, Fraunhofer	clean-up, proof-reading
01/04/2015	V0.7	Mancas, UMons	corrections to chapters 2 and 3
02/04/2015	V0.71	Stein, Fraunhofer	clean-up, proof-reading
03/04/2015	V0.8	Tsatsou, CERTH	QA comments, User Trials evaluation, Linked Profiler evaluation
06/04/2015	V0.9	Tsatsou, CERTH	Implicit profiling recommendations evaluation, proofreading, polishing

Authors	Group
Stein, Daniel	Fraunhofer IAIS, Germany
Kliegr, Tomas	UEP, Czech Republic
Kuchař, Jaroslav	UEP, Czech Republic
Mancas, Matei	UMons, Belgium
Rocca, Francois	UMons, Belgium
Dedecken, Pierre-Henri	UMons, Belgium
Tsatsou, Dorothea	CERTH, Greece
Ieronimakis, Pantelis	CERTH, Greece
Mezaris, Vasileios	CERTH, Greece

is evaluated against explicit user feedback in Section 3. User trials, where the complete (uncontextualized) LinkedTV workflow is implemented and recommendation accuracy is measured based on explicit user feedback is described in Section 4; in addition, in this Section, detailed interviews address added value, storage and privacy issues for the end users of personalization within the LinkedTV product. Section 5 then gives an outlook for future work outside the LinkedTV project, dealing with bag-of-entities text representation for video recommender systems. This deliverable is concluded in Section 6.

2 Kinect performance evaluation for head pose estimation

In this section, we show generic tests of the Kinect sensor in a real-life TV setup. The results show that face direction (which is closely related to eye gaze) can be reliably measured using the attention tracker developed during LinkedTV. The attention-tracker is compared to other state-of-the-art methods and we show that it achieves better results even when using the first version of the Kinect sensor.

Note that this section consists of attention-tracker assessment alone. In Section 3, real trials are done in a classical TV setup using news content and show how it is possible to map extent of gaze to the screen into user profiling instructions (more attention = more interest in concepts of that TV segment, less attention = less interest).

The Kinect sensor provides implicit information about viewer behavior. While explicit viewer behavior in terms of interaction with the player (play/pause, etc.) can already be tracked, only the use of a camera provides the possibility to go further. We mainly extract the number of viewers (for which the use of such a sensor is the only way to proceed) and how those users watch the main (or the second) screen by using the face direction detection.

While in deliverable D4.6 we already made a short assessment of the Kinect sensor alone, here we compare it to other state-of-the-art methods also extracting face direction.

The RGB-D Kinect sensor provides information about the head pose and the number of viewers which are present. While a feature like the number of users is considered to be stable as delivered by the Kinect sensor and does not need validation, the more complex head pose estimation needs such a validation. In the following sections, we validate the head pose estimation against 2 other state-of-the-art techniques and at several distances from the TV.

We show that the Kinect V1 is the best sensor in this case. While Kinect V1 is already the best compared to state-of-the-art systems, the second version of this sensor called here Kinect V2 provides an increased efficiency compared to the version V1 that we tested here (see section 2.9). In our trials we used this Kinect V2 which is the best tool in case of real-life TV setups.

2.1 Marker-based vs. Marker-less head pose estimation

Head pose estimation and head movements are captured commonly with physical sensors and optical analysis as we can see in the animation industry. Physical sensors such as accelerometers, gyroscopes and magnetometers are placed on the head to compute the head rotation. Another way is marker-based optical motion capture systems that are able to capture the subtlety of the motion. In these methods, markers are located on the head of the actor and they are tracked through multiple cameras. The markers are often colored dots or infrared reflective markers and the cameras depend on the markers type. Accurate tracking requires multiple cameras and specific software to compute head pose estimation. However, these systems are very expensive and complex, need for calibration, precise positioning of markers (e.g., Qualisys [qua]). We use the Qualisys marker-based motion capture system to evaluate the markerless methods.

Marker-less tracking is another approach to face motion capture and a wide range of methods exists. Some marker-less equipment uses infrared cameras to compute tracking of characteristic points. For example, FaceLAB [facb] gives the head orientation and the position of lips, eyes and eyebrows. For webcams, some algorithms exist as well. We can cite FaceAPI [faca] from SeeingMachines for example. Marker-less systems use RGB cameras or infrared cameras to compute tracking of characteristic points. We choose several freely accessible methods for a fair comparison in a real-world TV context.

2.2 Kinect versus state-of-the-art methods for face direction detection

The first method that we use is based on the Microsoft Kinect SDK. The Kinect SDK is free, easy to use and contains multiple tools for user tracking and behaviour modelling such as face tracking and head pose estimation. These tools combine 2D and 3D information obtained with the Kinect sensor.

Secondly, a head pose estimation solution based on 2D face tracking algorithm using the free library OpenCV. The face tracking part of this method was developed by Jason Saragih and it is known under the name of ”FaceTracker” [SLC11].

Finally, we use a fully 3D method for real time head pose estimation from depth images based on a free library called PCL (Point Cloud Library) [RC11].

2.3 The Kinect V 1 sensor

Microsoft provides a Face Tracking module with the SDK which works with the Kinect SDK since version 1.5. The first output contains the Euler rotation angles in degrees for the pitch, roll and yaw of the head calculated relatively to the sensor.

The head position is located using 3D skeleton automatically extracted from the depth map.

2.4 FaceTracker

This method is a combination of FaceTracker and a head pose estimation based on the features extracted from the face tracking part. FaceTracker allows the identification and localization of landmarks on a RGB image. These points can be assimilated to a facial mask allowing to track facial features like the edge of lips, facial contours, nose, eyes and eyebrows. Based on this, we apply the perspective-n-point (PNP) method to find the rotation matrix and 3D head pose estimation.

The advantage is that FaceTracker does not require specific calibration, and that it is compatible with any camera which could thus be a webcam. In our setup we use a 480×640 pixel webcam. The initialization of the algorithm is based on the Haar [VJ04] classifiers, thus the face tracking is optimal if the face is centred in front of the camera and straight. We can also observe significant perturbations when an object starts occluding some landmarks or when head rotation is rapidly done with a wide angle.

To find the Euler angles of the rotation of the head we use 2D Points from Facetracker, 3D points from a 3D head model and we compute the rotation matrix based on the perspective-n-point method.

A set of seven points are taken among the 66 points from FaceTracker. These points were chosen because they are far enough and stable regardless of the expressions and movements of the face. In parallel to this, we use a 3D head model from which we extract 3D points corresponding to 2D previous points.

Once the seven 2D and 3D coordinates are set, and the camera matrix found, we can calculate the matrix of rotation and translation of the 3D model by reporting the data from the face tracking. The pitch, roll and yaw can directly be extracted from the rotation matrix in real time (from 19 to 28 fps). The computing time per frame is about 50 ms by single thread on a Linux OS with Intel Core i7 2.3GHz and 8 GB of RAM. For the next steps of this analysis, this method is named "Facetracker".

2.5 3D Point Cloud face Analysis

The method used here is based on the approach developed in [RMG14] This solution relies on the use of random forest regression applied on a 3D cloud. This cloud is obtained with a RGB-D camera, such as Microsoft Kinect or Asus Xtion. Random forests are capable of handling large training sets, of generalization and fast computing time. In our case the random forests are extended by using a regression step. This allows us to simultaneously detect faces but also to estimate their orientations on the depth map.

The method consists of a training stage during which we build the random forest, and an on-line detection stage where the patches extracted from the current frame are classified using the trained forests. The training process is done once and it is not requested for any new user. The training stage is based on the BIWI dataset containing over 15000 images of 20 people (6 females and 14 males). This dataset covers a large set of head pose ($\pm$75 degrees yaw and $\pm$60 degrees pitch) and generalizes the detection step. A leaf of the trees composing the forest stores the ratio of face patches that arrived to it during training as well as two multi-variate Gaussian distributions voting for the location and orientation of the head. A second processing step can be applied, it consists in registering a generic face cloud over the region corresponding to the estimated position of the head. This refinement step can greatly increase the accuracy of the head tracker but requires more computing resources. A real-time mode is possible to use but it works at around 1 fps. That is why we decided to run the system off-line. This allows a full processing of data corresponding to a recording 20 fps with the refinement step.

The advantage of such a system is that it uses only geometric information from the 3D point cloud extracted by a Kinect, and is independent of the brightness. It can operate in the dark, which is rarely possible with face tracking systems working on color image which are highly dependent on the illumination. This approach was chosen because it fits well in the scenario of TV interaction. In addition, the use of 3D data will simplify the integration of future contextual information about the scene. For the analysis, this method is named "3DCloud".

2.6 Qualisys ground truth

In the experiment we compare the three techniques against a ground truth that is obtained with the highly accurate Qualisys motion capture system. The used setup consists of eight cameras, which emit infrared light and which track the position of reflective markers placed on the head. Qualisys Track Manager Software (QTM) provides the possibility to define a rigid body and to characterize the movement of this body with six degrees of freedom (6DOF: three Cartesian coordinates for its position and three Euler angles – roll, pitch and yaw – for its orientation). We used seven passive markers: Four markers were positioned on the TV screen and three markers were fixed to a rigid part of a hat (the three markers were placed with a distance of 72 mm, 77 mm and 86 mm between them). Both TV screen and hat were defined as rigid bodies in QTM. The framerate tracking is constant at 150 FPS, so it gives the values of the 6DOF each 0.007 seconds.

Before each recording session, a calibration procedure was made: the subject, who wears the hat on his head, sat in front of the screen and QTM nullified the 6DOF values for this head position. By this means, all the head movements were measured relatively to this initial starting position. To check the quality of the tracking data, QTM computes the different residuals of the 3D points compared to the rigid body definition. Over all the experiments, the software has calculated an average error of each head marker about 0.62 mm.

2.7 Experimental setup

Qualisys produces marker-based accurate data in real-time for object tracking at about 150 frames per second. The infrared light and marker do not interfere with RGB image and with infrared pattern from the Kinect. The choice of Qualisys as reference has been done especially in order to compare markerless methods without interferences. We perform the recording of the KinectSDK and the Facetracker in the same time under normal conditions and correct face lighting. And we have chosen to perform the 3DCloud method separately from the first record because interferences are observed between 2 running Kinects heading in the same direction. This positioning is shown on Figure 2. The angles computed from the different methods are the Euler angles.

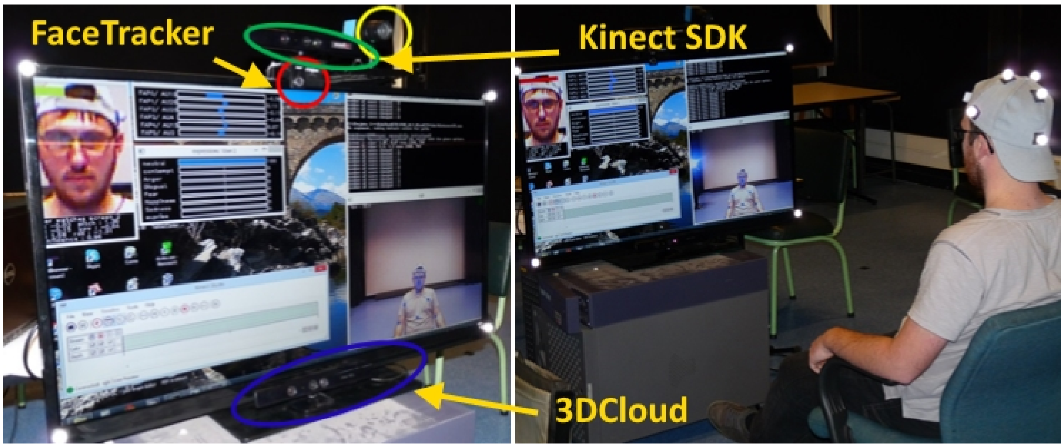


Figure 2: Setup for facial animation recording. Kinect for the Kinect SDK is in green, Webcam for the face tracking algorithm is red, Kinect for 3DCloud is blue and a 2D camera synchronized with the Qualisys is yellow.

We made several recordings with 10 candidates. Each one performs a head movement sequence at 5 different distances from the screen: 1.20 m, 1.50 m, 2 m, 2.5 m and 3 m. Movements performed are conventional rotations when we are facing a screen (pitch, roll, and yaw; combination of these movements; slow and fast rotation). Six of the viewers have light skin, others have dark skin. Three of them wear glasses and six of them wear beard or had a mustache.

A preliminary test showed that the optimal position of the camera for Facetracker and KinectSDK is on top of the TV screen, while for 3DCloud which uses the shape of the jaw, is at the bottom of the TV screen.

2.8 Experimental results

The correlation between the ground truth data from the Qualisys system and the data from the Kinect sensor, FaceTracker and 3DPointCloud respectively is computed. This correlation is a good indicator used to establish the link between a set of given values and its reference. It is interesting to analyse the correlation value obtained for each distance, with average for all candidates, to know which methods are better correlated with the reference data. If the correlation value is equal to 1, the two signals are totally correlated. If the correlation is between 0.5 and 1, we consider a strong dependence. The 0 value shows that the two signals are independent and a -1 value correspond to the opposite of the signal. Figure 3 shows the correlation for pitch, Figure 4 for yaw and Figure 5 for roll. The three curves from KinectSDK, Facetracker and 3DCloud are compared with the reference obtained with Qualisys.

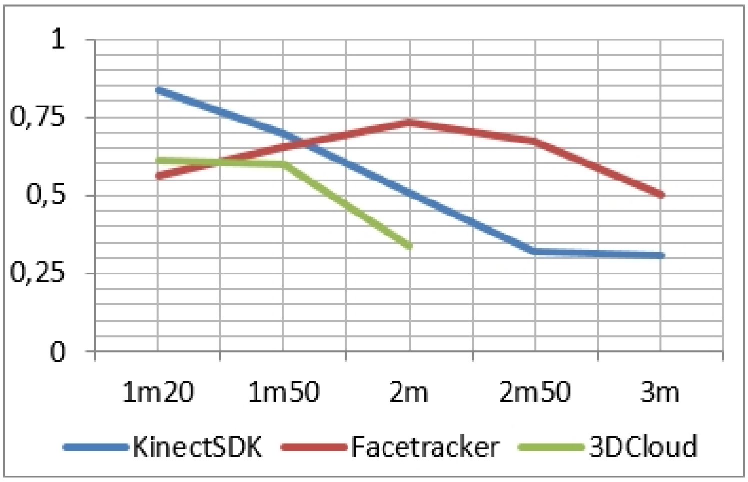


Figure 3: Mean correlation for the pitch depending on the distance.

On Figure 3, we observe that the pitch of the KinectSDK has a good correlation (0.84) at a distance of 1.20 m. The Facetracker and 3DCloud are lower with values about 0.6. We observe that the facetracker stays stable with the distance between 0.5 to 0.73. But KinectSDK and 3Dcloud decrease with the distance under the correlation value of 0.5 for KinectSDK at 2.50 m with 0.32, and for the 3DCloud at 2 m with 0.34.

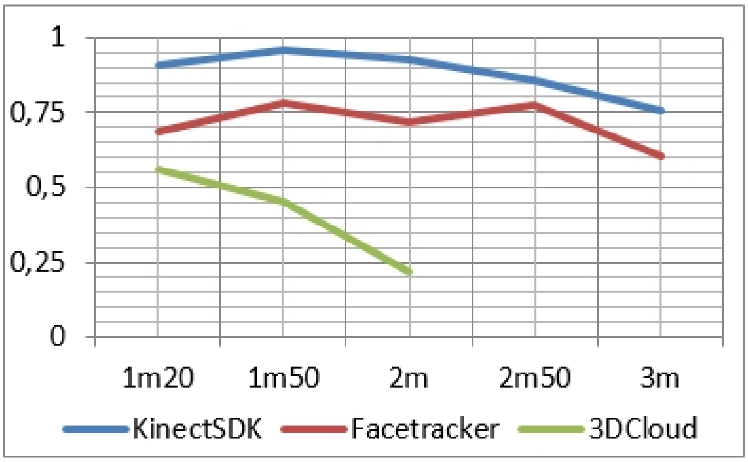


Figure 4: Mean correlation for the yaw depending on the distance.

For the second angle, the yaw, corresponding to a right-left movement, we can see good results for the KinectSDK with values larger than 0.9 for 1.20 m, 1.50 m and 2 m (Figure 4). The values decrease from 0.85 for 2.50 m to 0.76 for 3 m. The curve of the Facetracker is similar but worse with values at around 0.75. For the 3DCloud the values are worse with 0.61 at the beginning and less after. KinectSDK outperforms the other two methods.

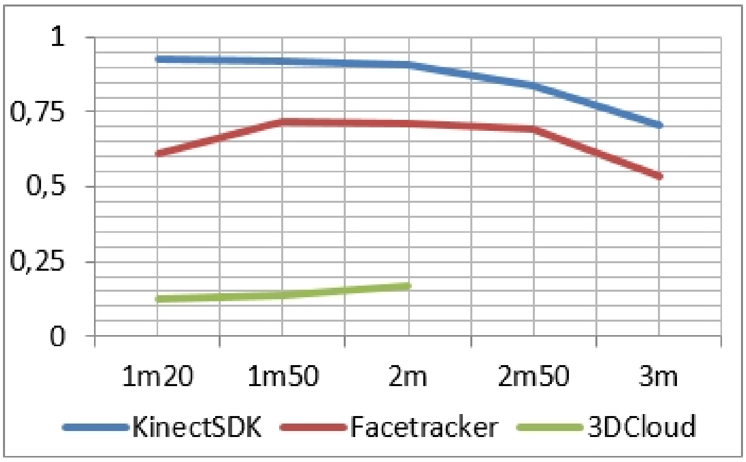


Figure 5: Mean correlation for the roll depending on the distance.

The 3DCloud give bad values for the roll. The KinectSDK have good correlation as the yaw curve (0.93 to 0.7). Facetracker correlation is also good with lower result than KinectSDK with about 0.65 (Figure 5).

After watching the correlation values, it is also interesting to look at the mean error made by each system. Indeed, a method with a big correlation and low RMSE is considered very well for head pose estimation.

We observe a RMSE similar for the pitch about 10 to 15 degrees for each method, whereas the KinectSDK is good at 1m20 with 5.9 degrees. The error grows with the distance.

On the yaw, we observe a slight increase of the error with the distance. But the KinectSDK is better with RMSE from 10 to 12 degrees, 15 to 18 degrees for Facetracker and around 20 for 3DCloud.

In the case of roll, the RMSE is similar for Facetracker and KinectSDK (around 10 degrees with a smaller error at 3m for KinectSDK). The error of 3DCloud is arround 13 degrees. This error can be put in perspective because the correlation for the roll was poor.

After watching the values of the root means square error and correlation according to the different distances, it is interesting to look at the average values of these two indicators for each individual. It will be interesting to link some observation to candidates facial features.

Individual observations:

– We first observe that the pitch correlation for each individual is about 0.6. All these values are similar. But a correlation about 0 is observed for the candidate number 5 for the 3DCloud method which means that the pitch did not work at all for this candidate.

– For yaw, the KinectSDK gives a correlation higher than 0.75 for each candidate followed by Facetracker with values higher than 0.5. 3DCloud method gives the worse correlation with values between 0.1 and 0.64. For this method, candidate number 5 gives also the worse correlation for the 3DCloud.

– For roll, we observe that the KinectSDK and the Factracker method give good values higher than 0.5, with a better correlation for KinectSDK. Results for the 3DCloud are worse.

– RMSE for pitch is about 10 for the KinectSDK and 3DCloud for all candidates. We observe that the error on the pitch for the Facetracker method is higher for candidates 5, 7, 8 and 9, these candidates have darker skin.

– For yaw RMSE, the 3DCloud gives worse results than KinectSDK and Facetracker. We also observe bigger error for darker skin for the Facetracker method.

– Concerning roll RMSE, the error is about 10 degrees for KinectSDK and Facetracker and greater for the 3DCloud method.

After analyzing all data obtained by the three different methods we are able to establish the advantage and the drawbacks for each method in a TV context.

These results show that the better correlation values are obtained with the KinectSDK. The Face-tracker based method also gives good results. We also have similar errors for these methods. Concerning the third method, 3DCloud, the RMSE and the correlation are worse than the two other methods and do not work at a distance of more than 2 m from the screen. The estimation of roll is also of poor quality.

For all these methods, errors are mainly due to face tracking errors and tracking losses. If we cut all sections with bad detection of the head and the characteristics point of the face, the RMSE will decline significantly and the correlation will increase. But in our context, we want to get results without post-processing corrections. We can also say that from a distance of 1.50 m, an error of 10 degrees generates an error on the screen of 26 cm (150sin(26)). This is quite acceptable for whether a person looks at a TV screen.

2.9 Kinect V2

The new Kinect sensor (called Kinect V2 for Windows or Kinect One) developed for the Xbox One is a low-cost depth and RGB camera. In Figure 6, we observe the improvements of this new sensor compared to its predecessor.

Feature	Kinect for Windows 1	Kinect for Windows 2
Color Camera	640 x 480 @30 fps	1920 x 1080 @30 fps
Depth Camera	320 x 240	512 x 424
Max Depth Distance	~4.5 M	~4.5 M
Min Depth Distance	40 cm in near mode	50 cm
Horizontal Field of View	57 degrees	70 degrees
Vertical Field of View	43 degrees	60 degrees
Tilt Motor	yes	no
Skeleton Joints Defined	20 joints	26 joints
Full Skeletons Tracked	2	6
USB Standard	2.0	3.0
Supported OS	Win 7, Win 8	Win 8

Figure 6: Comparison between Kinect V1 and Kinect V2.

The main drawback of the first sensor was the impossibility to measure thin objects. The technology behind the new sensor is called ”time of flight” (ToF) and it attempts to address this major roadblock. This novel image sensor indirectly measures the time it takes for pulses of laser light to travel from a laser projector, to a target surface, and then back to an image sensor.

Microsoft provides many modules with the Kinect’s SDK such as Face Tracking in low and high definition, Gesture Builder and Recognizer, Coordinate Mapping, etc. In this project, we use the high definition face tracking which detects more than 1,300 points on the face, but only 35 of them represents interesting points.

In addition to these data, the high definition face tracking module can give us the head pivot and the quaternion which represent the orientation of the face. In order to obtain all these information, we have to provide the Kinect color, depth, and IR images as input to the face tracking module.

2.10 Kinect V2 implementation: relation between interest and screen watching duration

Based on the gaze estimation, or in this case on the head direction, it is possible to measure the interest of a person to a specific element of its environment by calculating the intersection between the direction of the face and a 2D plane (or a 3D volume). In this case, the TV screen will be represented by a 2D plane and another 2D plane for the second screen. The head-pose estimation will give an indication on what the user is looking at, specifically what part of the object it looks. The enrichment of interest also requires a measure of duration that a user takes to watch something. In a visual attention to television context, a study showed [HPH+05] that there are four types of behavior depending on the fixing duration:

– Duration ≤ 1.5 sec. : monitoring

– Duration 1.5 sec. to 5.5 sec. : orienting

– Duration 5.5 sec. to 15 sec. : engaged

– Duration > 15 sec. : staring

These four measures of attention correspond to be firstly attracted by something with "monitoring behavior", and then intrigued, "orienting behavior", and more time passes more the user become interested, "engaged behavior", and beyond 15 seconds the user is captivated with a "staring behavior". These measures have been established for a TV watching and used to correctly describe the interaction with one or more screens.

For LinkedTV, these generic experiments show that we used and implemented the optimal technology (Kinect V2) on the Attention-tracker given the real-life TV setup constraints. In the following section we show how the Attention-tracker can send useful information to the profiling pipeline of WP 4.

3 Implicit contextualized personalization evaluation

In this section, we describe the trials held to validate the Attention-tracker as part of the implicit profiling but also the entire WP4 pipeline from the attention tracker to user profiling. Having shown in the previous sections that the attention tracker is stable and is the current best implementation option, we will now investigate in how far it can be used in the LinkedTV WP4 pipeline. The purpose of the trials is to get a maximum of information in a limited period of time.

Concerning the number of people, the use of a camera is the only way to know it in an implicit way. Concerning the attention, there are several ways to estimate the interest of a person on the related media segment (e.g. by analyzing viewers interaction with the player, such as playing/pausing/browsing enrichment, etc. ...).

That is why we focus in these trials to validate the attention tracker together with the entire WP4 pipeline and other implicit behavior analysis like browsing enrichment while watching TV which is the core of LinkedTV project.

3.1 Overall setup

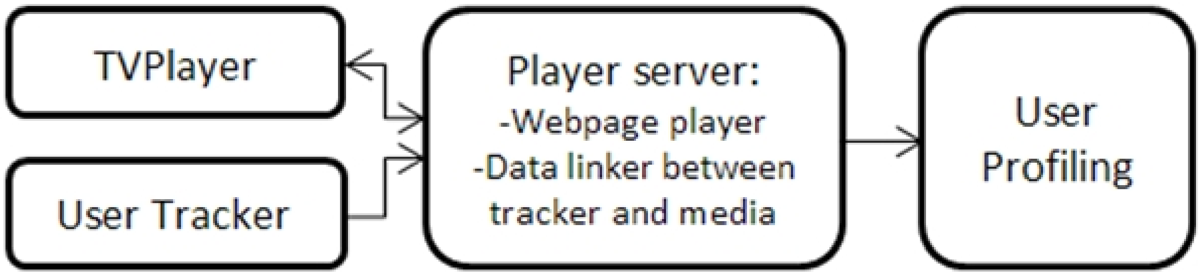


Figure 7: Main view of the experimental schema.

The overall system is shown in Figure 7. It is the WP4 pipeline where the LinkedTV player is replaced by the UEP test player. This player is already connected to the Attention tracker. It also get in the same time the different enrichment for each video segment by using the subtitles. The data comes from the attention tracker (Kinect-based) and is related to each displayed video segment through the player server (UEP). The test player of UEP which is shown on the main screen can be seen on Figure 8.

The User Profiling module receives measures of interest (or disinterest) for each video segment from the GAIN module. Based on these features, it is then able to establish a contextualized profile.

On the trials side, a sofa is installed 2.5 meters away from the TV which is equipped with a Kinect V2 (Figure 9, cf. also Section 2.9). At the same time, the head motion is recorded using a Qualisys MOCAP system. The RGB-D data is also recorded on a hard drive. Indeed, if new features are extracted later, they can be computed directly from the recording instead of doing trials again. The RGB-D recording is not mandatory for our trials, but it is interesting data to keep for further analysis. A consent was signed by the viewers which is better described at section 3.4.

All the steps achieved during the trials are summarized on Figure 10.

When the user comes into the field of view of the KinectV2, placed under the TV, his skeleton is tracked and the head orientation is estimated. The Tracker Computer performs the process and determines what the user watches with an accuracy of a few centimeters: Screen 1 (main screen), Screen 2 (a tablet) or elsewhere (no screen is viewed). These messages are sent to the Player Server in the following way:

- interest-changed = 0 : each time the user switches the screen
- interest-changed = 1 : user looking between 1.5 and 5 seconds to main screen
- interest-changed = 2 : user looking between 5 and 15 seconds to main screen
- interest-changed = 3 : user looking more than 15 seconds to main screen
- interest-changed = 4 : user looking between 1.5 and 5 seconds to second screen
- interest-changed = 5 : user looking between 5 and 15 seconds to second screen

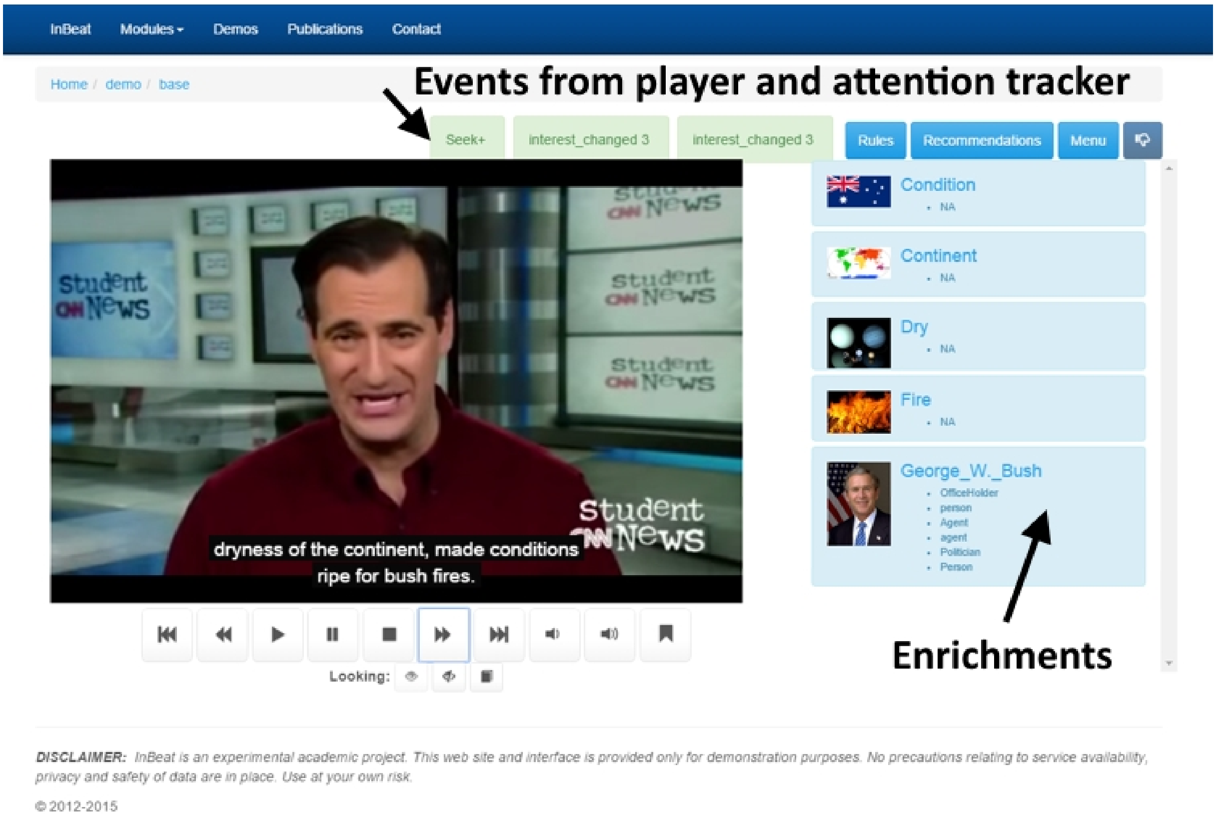


Figure 8: UEP test player. In addition to the video, it allows people to interact with the video and see the enrichment.

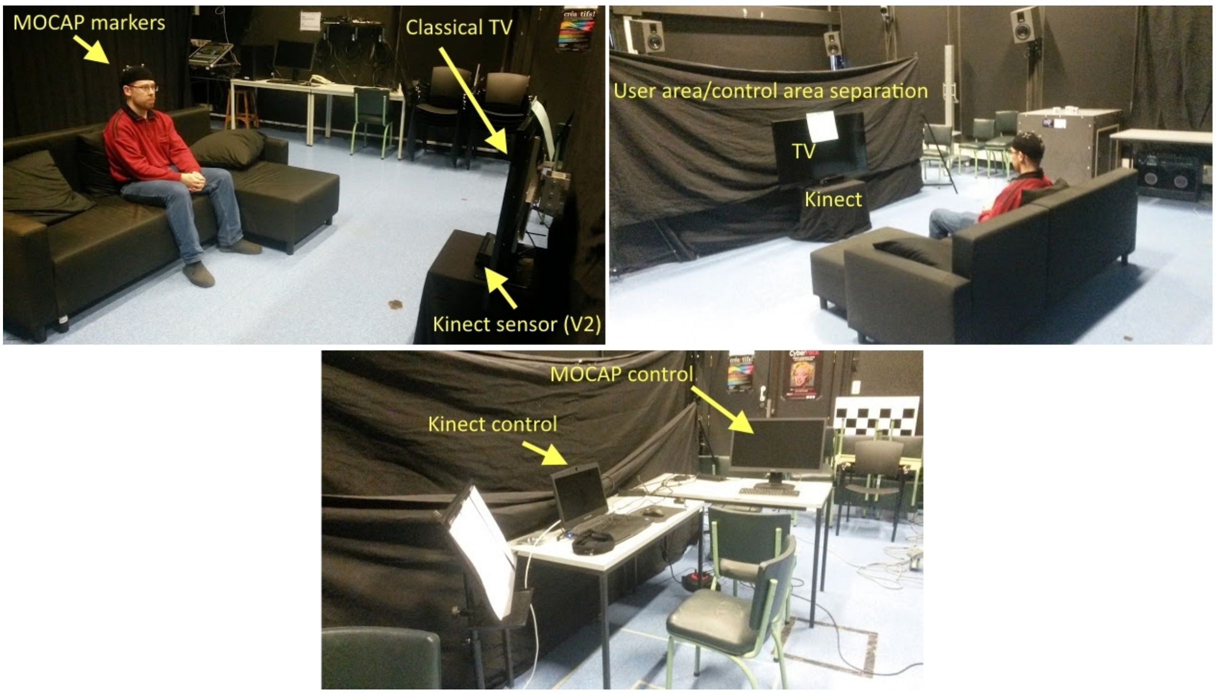


Figure 9: Main view of the experimental setup (top: viewer side, bottom: tester side)

– interest-changed = 6 : user looking more than 15 seconds to second screen

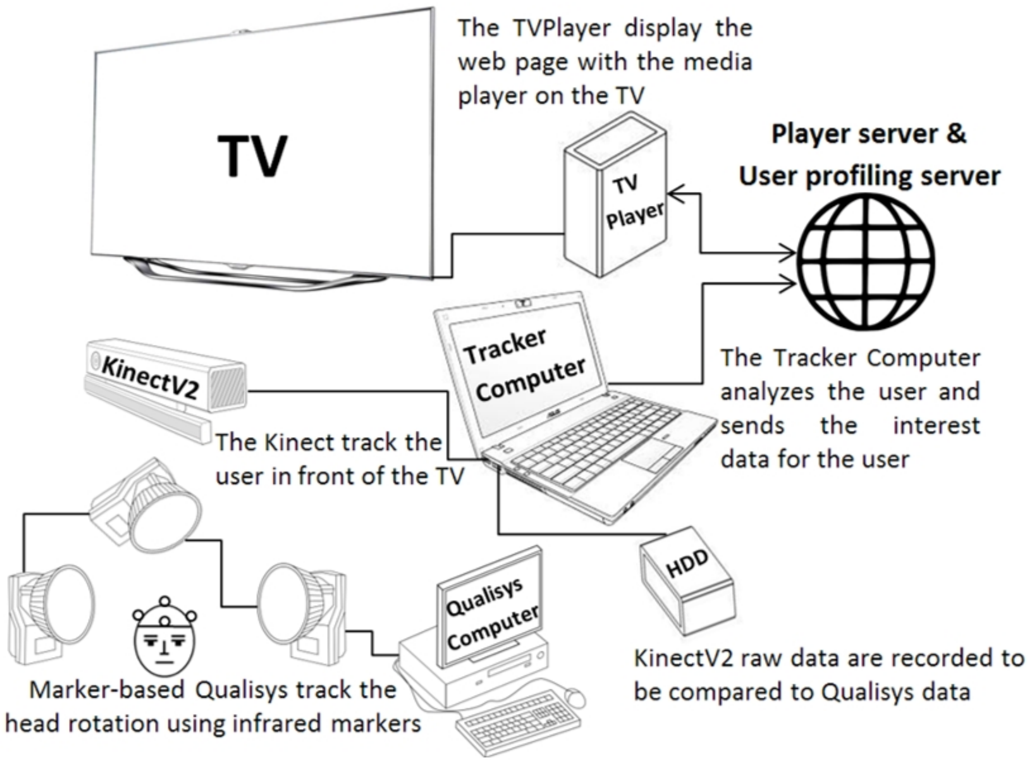


Figure 10: Main view of the experimental setup (top: viewer side, bottom: tester side)

3.2 Content and interaction

For the trials described in this section, the development player software developed by UEP was used (see Fig. 8). The selected YouTube video is played and entities that represent the content of each specific video pseudo-shot (i.e. a fragment of subtitles) are displayed alongside the video frame (Fig. 8, right-hand side). The Kinect interactions are propagated to GAIN by the demonstration player along with the new interactions raised by the player based on user actions.

The displayed content on the player consists of a selection of 7 small videos with a Creative Commons Licence of a US news show. The videos (and also the related subtitle tracks) are fused into a single 13 minutes video[1]. We chose this content because it is related to the news content that can be found in RBB partner, but it is in English which let people in Mons who do not speak German to take part to the trials.

The system allows to know when and how long the user watches the TV (screen 1) or the tablet (screen 2). When the user does not watch TV, the image is hidden but the media runs in order for the user to keep on hearing the sound. The viewer can use a keyboard to control the player and navigate into the video enrichment.

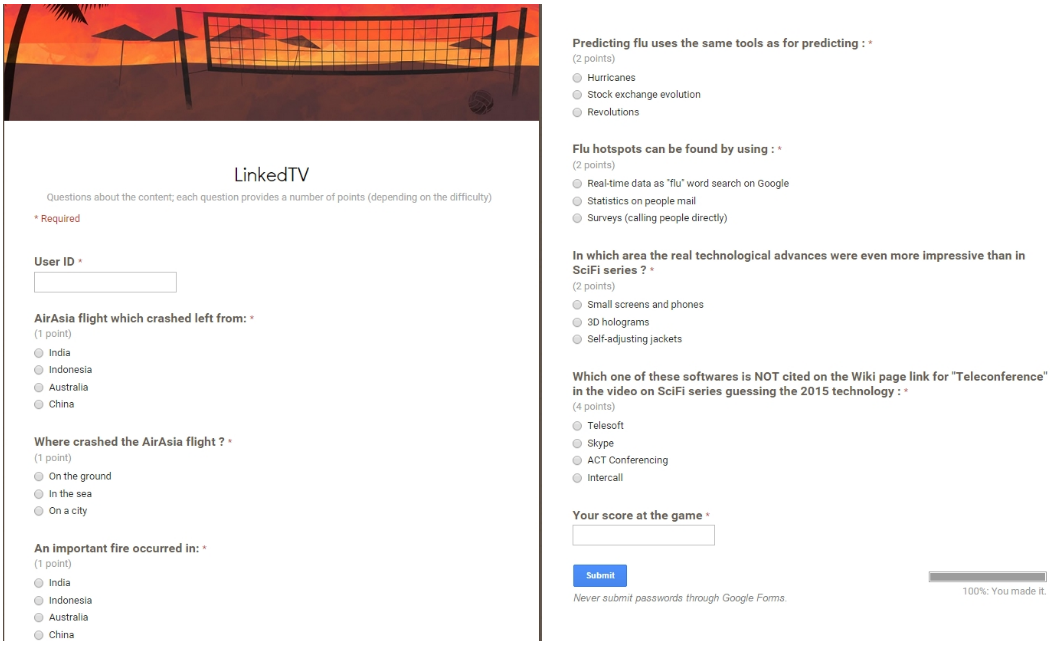
The video used for the tests (see footnote) is a mashup of CNN Student News for learning English. These TV shows are easy to understand even for non-native English speakers, and their subtitles are provided. The mashup covers seven different topics (North Korea, plane crash in Asia, Christmas tree recycling, bush fire in Australia, flu, American football, evolution of technologies in the last decades) for a total duration of about 13 minutes. It has been enriched with links to web pages that are displayed next to the video in the UEP player (8). Each enrichment is displayed for a short time, typically 4 seconds and is related to a video subtitle.

Users are invited to answer a questionnaire which focuses only on four of the seven topics. The users have simple control over the video player (play, pause, go forward for a few seconds, go backward), but they can also click on enrichment links.

[1] available here: `https://www.youtube.com/watch?v=GyhARZT7OqU`

3.3 Questionnaires

Three different questionnaires are submitted to the viewer. The first one concerns content-related questions linked to the viewer interest and it aims into simulating common interests between the different viewers. The participants are asked to answer a quiz of 7 questions, whose answers can be extracted either from the video or in the displayed enrichment. The questions concern a plane crash, a bush fire, the prediction of the flu and the current technologies compared to the previsions from the science fiction movies from the eighties. The questions are made in order to provide the viewer with a secondary interest (easy questions about plane crash and fire in Australia, see Figure 11, left-hand side) and with a main interest (complex questions on flu prediction and new technologies, see Figure 11, right-hand side).


LinkedTV
Questions about the content; each question provides a number of points (depending on the difficulty)
* Required

User ID *

AirAsia flight which crashed left from: *
(1 point)
India
Indonesia
Australia
China

Where crashed the AirAsia flight ? *
(1 point)
On the ground
In the sea
On a city

An important fire occurred in: *
(1 point)
India
Indonesia
Australia
China

Predicting flu uses the same tools as for predicting : *
(2 points)
Hurricanes
Stock exchange evolution
Revolutions

Flu hotspots can be found by using : *
(2 points)
Real-time data as "flu" word search on Google
Statistics on people mail
Surveys (calling people directly)

In which area the real technological advances were even more impressive than in SciFi series ? *
(2 points)
Small screens and phones
3D holograms
Self-adjusting jackets

Which one of these softwares is NOT cited on the Wiki page link for "Teleconference" in the video on SciFi series guessing the 2015 technology : *
(4 points)
Telesoft
Skype
ACT Conferencing
Intercall

Your score at the game *

Submit
100%: You made it.
Never submit passwords through Google Forms.

Figure 11: First questionnaire. Left: simple questions for secondary interest, Right: complex questions for main interest.

The seconds questionnaire focuses on the assessment of all the presented enrichments. The user needs to rate ”-1” if no interest, ”0” if secondary interest and ”1” if main interest each set of enrichments (s. Figure 12, left-hand side).

Finally in a third questionnaire the viewer is asked to rate the final user model found by the system (s. Figure 12, right-hand side). For user convenience, this questionnaire presented the user not the whole spectrum of his/her interests, but rather the top ten interests and top 10 disinterests.

3.4 Trials procedure

After being explained the experiment, the viewer signs a consent form. The consent stipulates that the subject acknowledged the project and that he agrees that some data such as video analysis, recordings, questionnaires, etc. will be stored with an ID for a total duration of 5 years. He or she also agrees that the same data can be distributed in an anonymous way for research purposes.

The user has 2 minutes to get used to the system with a different video content: s/he can browse on the player, look at it, look at the second screen (tablet). Once this is done s/he has time to read the content-related questionnaire which shows him or her the main and secondary interests. On the four topics of interest the first two are of secondary interest which imply simple questions in the questionnaire and the last two are main interest which imply complex questions on the questionnaire. For the 4th topic, the user must browse the enrichments to be able to answer one of the questions.

During the viewing the user also has a second screen (tablet) where s/he needs to play a game and get the maximum possible score. This game that s/he plays to is concurrent to the questions on the video content. The main idea behind this is that the user will mainly watch the main screen when

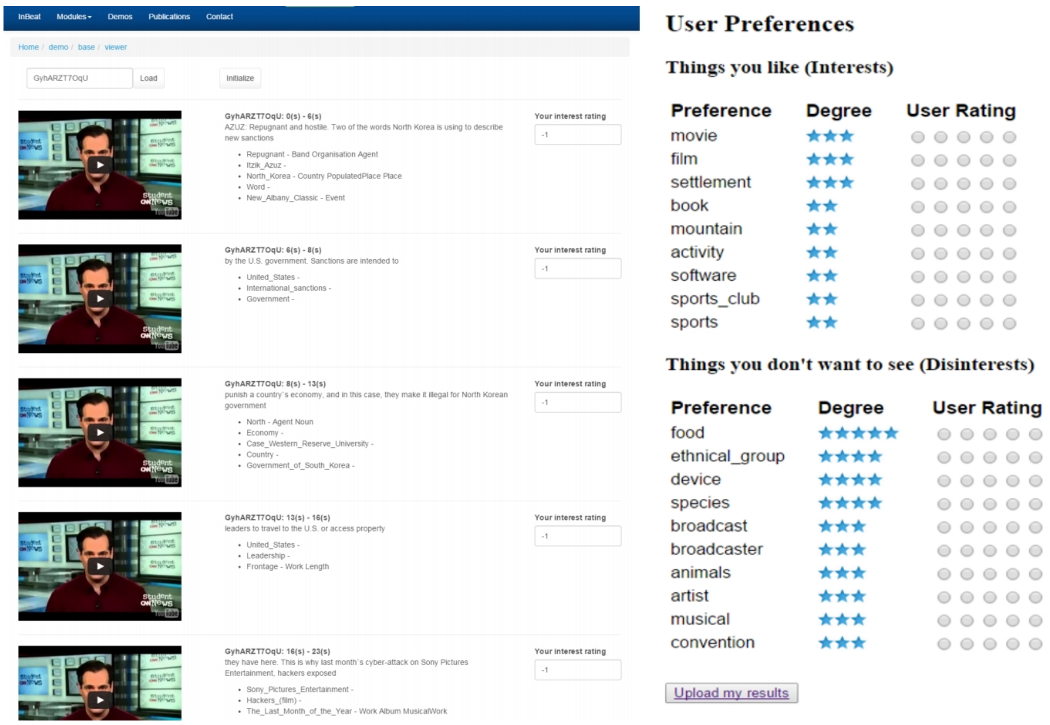


Figure 12: Left: second questionnaire on enrichment assessment, Right: third questionnaire on user profile assessment.

interested by the content and play the game when the video does not bring any information to answer to the quiz.

3.5 Evaluation

Evaluation of implicit contextualized profiling aims at the assessment of the interest computation workflow that is used to process collected implicit data and to generate a basis for creation of user profiles. The main component responsible for collecting and processing data is *InBeat – GAIN*.

3.5.1 InBeat-GAIN Setting

The player is connected with the Microsoft Kinect sensor as described in Subs. 3.1. The player interprets the events from the Kinect (e.g. user is looking or not) and combines them with the explicit user actions (e.g. user clicking the "stop" button). The information is then sent to *GAIN*. Since the sensor does not have any information about the video currently playing, the player extends the interaction received from the sensor with a shot identifier and details about the playing time. The fused data are sent to *GAIN* as an interaction. All interaction is interpreted in *GAIN* as interest clues and transformed to a real value in the interval $[-1,1]$ that represents the final level of interest for the specific pseudoshot.

To compute the final interest from the interactions we use an experimentally predefined set of rules that either increase or decrease the default value of the interest. The default interest value is 0, which is interpreted as the neutral level of interest. The set of rules used in the trials is given in Table 2.

If the computed value of interest exceeds 1 or is lower than -1, it is replaced by 1 or -1, respectively. The final value is extended with information describing the pseudoshot: identifier and a set of entities with corresponding DBpedia types. The final export of interest for each pseudoshot is compared with the explicit values provided by the participants of the trial.

3.5.2 InBeat-GAIN Evaluation metrics and results

For evaluation we used the following ground truth and baselines:

– *Trials ground-truth*: explicit annotations of interest from questionnaires filled in by participants. Participants annotated each pseudoshot of video with value that represents negative,neutral or positive interest $(-1,0,1)$.

Action	Value	Interest change	Interpretation
Play	NA	+0.01	Play video
Seek+	NA	-0.5	Go forward 10s
Seek-	NA	+0.5	Go backward 10s
Bookmark	NA	+1	Bookmark a shot
Detail	NA	+1	View details about entity
Volume+	NA	+0.5	Increase volume
Volume-	NA	-0.1	Decrease volume
Viewer looking	0	-1	Viewer not looking to screen
Viewer looking	2	-1	Viewer looking to second screen
Viewer looking	1	+1	Viewer looking to main screen
Interest changed	0	-0.2	Viewer switched the screen
Interest changed	1	+0.2	Looking between 1.5 and 5 seconds to main screen
Interest changed	2	+0.5	Looking between 5 and 15 seconds to main screen
Interest changed	3	+0.8	Looking more than 15 seconds to main screen
Interest changed	4	-0.3	Looking between 1.5 and 5 seconds to second screen
Interest changed	5	-0.7	Looking between 5 and 15 seconds to second screen
Interest changed	6	-1	Looking more than 15 seconds to second screen

Table 2: Predefined set of rules used in trials

– *Random baseline*: Baseline data computed as a random value from interval $[-1,1]$ per pseudoshot.

– *Most frequent baseline*: Baseline algorithm where all shots are labelled with the most frequent value filled by participant in a questionnaire.

The GAIN interest computation algorithm was evaluated used in two setups:

– *GAIN*: outputs from *GAIN* computed using a set of interactions per pseudoshot and a predefined set of rules to interpret importance of an interaction. *GAIN* provides outputs as real values from interval $[-1,1]$ for each pseudoshot.

– *GAIN-Window*: Sliding window approach – a mean value of the current, the previous and the following interest value is aggregated in order to decrease influence of transitions between subsequent pseudo-shots.

Basic trial ground truth statistics: 20 participants and 13,533 interactions (678 on average per participant). Table 3 presents the overview of the interactions collected during the trials.

Action	Ratio
Interest changed	69.50 %
Viewer looking	24.44 %
Seek+	4.35 %
Seek-	0.95 %
Pause	0.31 %
Play	0.30 %
Detail	0.12 %
Previous	0.01 %
Next	0.01 %
Stop	0.01 %

Table 3: Overview of collected interactions

As a metric for this evaluation we used *Mean Absolute Error* (MAE) computed as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|t_i - y_i| \tag{1}$$

where n is a number of shots in user trials, t_i is interest value for specific shot from *Trial* and y_i is value of *GAIN*, *GAIN-Window*, *Random* or *Most Frequent*.

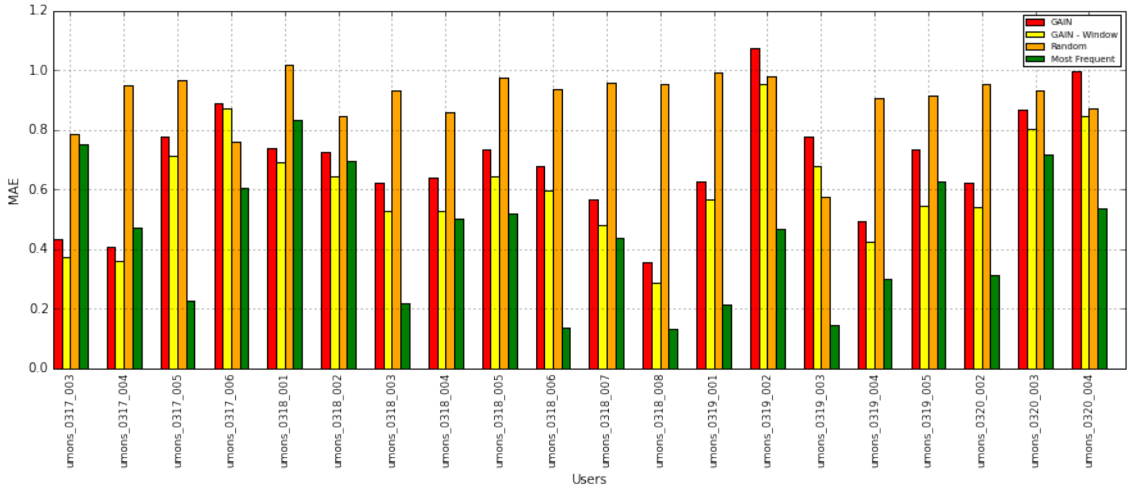


Figure 13: Evaluation results: MAE for each participant

Most Frequent	Random	GAIN	GAIN - Window
0.44	0.89	0.69	0.67

Table 4: Evaluation results: Macro-Average *MAE* for all participants

Figure 13 depicts the results of *MAE* for each user who participated in trials. The results of *MAE* averaged for all participants is in Table 4. Figure 14 depicts the average interest provided by *GAIN* and compares it to the average interest annotated by the trial participants. On this figure two plots have good correspondence and we can see 4 peaks. The 2 first peaks correspond to the two videos where people had to answer to simple questions in the content questionnaire (which means that they have a medium interest for those videos). The 2 last peaks correspond to the 2 videos where people had difficult questions (and even they needed to go into the enrichment for the 4th video). The data from the Kinect (Figure 15) and after GAIN processing (Figure 14, red curve) both also have higher values for those 4 peaks. The two first peaks are less well detected because simply the questions were easier, the user answered very quickly and then started to play the game on his tablet (which is interpreted as a disinterest to the related media segment). The two last peaks with difficult questions were much better spotted as the viewer needed to pay more attention to answer the related questions. For the last question the click on enrichment was quite complex and it logically brought a high interest to this video, that is why almost all the video has a high value of interest. Those results show that there is a coherence between the user ground truth interest and the one obtained by using the Kinect and the player interactions.

Figure 15 presents the data on ”looking at the main screen” averaged across all users participating in the the trial.

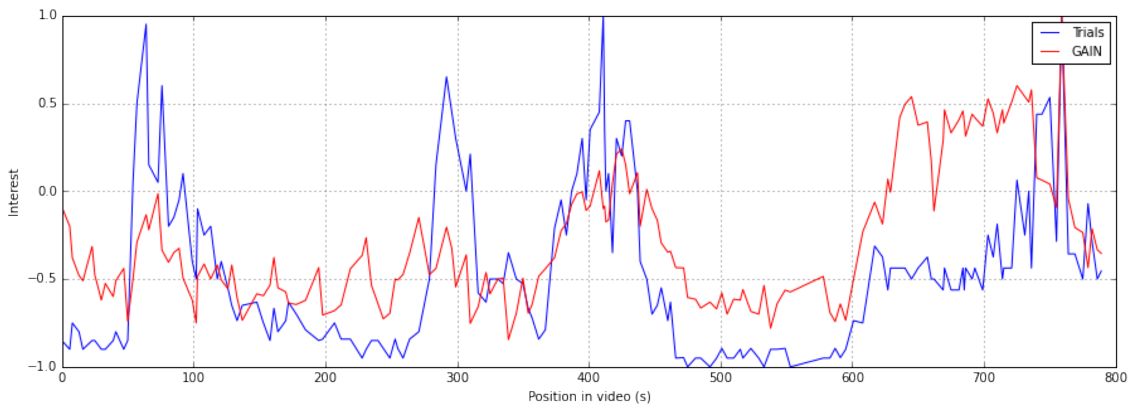


Figure 14: Timeline of the average interest from GAIN vs the ground truth from the questionnaires

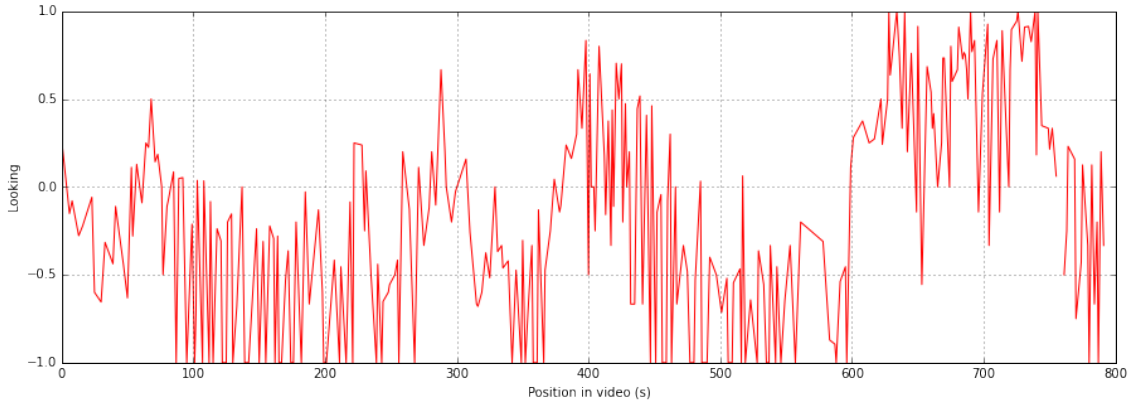


Figure 15: Timeline of the average viewer looking at the main screen (computed from the collected interactions)

The execution of the trial generated 173 pseudoshots (video fragments) for which user judgment is available. The results obtained by the evaluated GAIN workflow on this dataset indicate that the used feature set allows to estimate user interest in video content with significantly higher mean average error than a random guess. However, it should be noted that the default GAIN workflow is outperformed by the most frequent baseline.

The size of the groundtruth dataset (over 3000 instances)[2] that came out of the final trial, allows to to employ machine learning techniques instead of the hand-coded rule sets. As shown in the following subsection, the supervised approach provides an improvement over the most frequent baseline.

3.5.3 InBeat-GAIN Experiments with supervised classifiers

The groundtruth dataset that contains explicit interest levels per shot and a set of recorded interactions per shot allows to build a classifier that can "learn" relations between the interactions and the interest level. The model can provide improved classification results over the adhoc rules presented in Table 2 for which the MAE is reported in Table 4.

Our benchmark includes he following set of supervised algorithms: *Most Frequent*, *SVM with linear kernel*, *brCBA* [KKSV14], *k-Nearest Neighbour*(KNN) and *majority class voting*. The Most Frequent classifier simply predicts the most frequent class. The SVM was run with default parameters (C=0, $\varepsilon = 0.001$, shrinking). For kNN we used k=20 as empirically obtained value of the k parameter. The setting of the brCBA classifier was as follows: minConfidence = 0.01, minSupport= 0.025. The classifiers included into the majority class voting scheme include kNN, linear SVM and Most Frequent.

Input received by GAIN for each participant and pseudoshot was represented as a fixed-length vector. Three binary features were generated for each possible values of the actions listed in Table 3: one feature corresponding to the event value in the current pseudoshot, one feature for the preceding pseudoshot and one feature for the subsequent pseudoshot.

The matrix created for each used thus contains columns that represent interactions relating to the previous shot, current and following shot (a.k.a. *sliding window* approach). Example of the matrix is in Table 5. Column names are prefixed with *p_*, *c_*, *f_* for previous, current and following shot respectively, *pause* represents *Pause* interaction, *ic_1* is *Interest Changed* with value 1. The last column (*gt*) holds the value provided by the participant as the interest level ground-truth.

p_pause	p_ic_1	...	c_pause	c_ic_1	...	f_pause	f_ic_1	...	gt
0	0	...	1	1	...	0	1	...	1
1	1	...	0	1	...	1	0	...	0
0	1	...	1	0	...	1	1	...	-1
1	0	...	1	1	...	0	0	...	1
...	...	...	...	...	...	...	...	...	...

Table 5: Example of matrix for experiments with classifiers.

[2] 173 pseudoshots * 15 users, 5 users out of 20 were excluded for reasons given in Subs. 3.5.3

We performed 10-Fold stratified cross-validation for each dataset (there was one dataset for each trial participant). Only 15 participants are used for experiments: *umons_0318_008* was excluded because of a very small variance of the assigned interest value in the questionnaire and the first four testing participants (*umons_0317_003...umons_0317_006*) were excluded since they were used to verify the trial setup. The evaluation results are presented in Table 6. As the evaluation metrics, we use the Mean Absolute Error (MAE), which unlike accuracy reflects the different costs of misclassification (the predicted value of interest is one of the three values {-1,0,1}).

Participant	Most Frequent	SVM - linear	brCBA	KNN	voting
umons_0318_001	0.828	0.627	0.676	0.577	**0.564**
umons_0318_002	0.704	0.594	0.603	**0.569**	0.582
umons_0318_003	0.214	0.214	**0.212**	0.214	0.214
umons_0318_004	0.505	**0.475**	0.541	0.499	0.481
umons_0318_005	0.513	0.525	**0.511**	0.550	0.513
umons_0318_006	0.136	0.136	**0.133**	0.136	0.136
umons_0318_007	**0.440**	0.463	0.492	0.451	0.440
umons_0319_001	**0.107**	0.107	0.213	0.107	0.107
umons_0319_002	0.521	0.521	**0.471**	0.526	0.521
umons_0319_003	0.222	0.222	**0.140**	0.222	0.222
umons_0319_004	0.303	0.303	**0.300**	0.303	0.303
umons_0319_005	0.522	0.522	0.759	**0.509**	0.522
umons_0320_002	0.314	**0.255**	0.302	0.273	0.255
umons_0320_003	0.709	**0.701**	0.796	0.701	0.708
umons_0320_004	0.607	0.607	**0.555**	0.618	0.607
Average	0.443	0.418	0.447	0.417	**0.412**

Table 6: Classification results: *MAE* for all participants

The best performing algorithm with respect to the overall MAE is *voting*, which is a simple meta learning algorithm. However, from the perspective of the won-tie-loss record, the best performing algorithm is our *brCBA*.

Apart from the best won-tie-loss from the considered classifier, other advantage of brCBA is that the result of the algorithm is a rule set. Rules are in general one of the most easily understandable machine learning models. Within the personalization workflow, the fact that brCBA outputs rules, in theory, allows the model to be presented to the user, edited by the user (user discards some rules) and then deployed to GAIN instead of the predefined set of rules presented in Table 2.

We consider further improvements in the induction of interest classifiers as one of the most viable directions of further work, as the accuracy of interest estimation is one of the key inputs for the personalization workflow.

3.5.4 Linked Profiler Setting

In this section, the setting for the evaluation of the final user profiles produced is going to be described. Linked Profiler, the final step in the profiling pipeline of WP4, takes as input the preferences extracted by GAIN per pseudoshot (see previous subsections) for each user, along with association rules learned by the InBeat Preference Learning module (cf. D4.4). It aggregates these preferences under a single profile per user, in the following steps:

– It pairs named entities extracted by GAIN with their most specific DBPedia types. This means that generic types like ”Agent”, e.g. for a certain person recognized in GAIN interests, are omitted in favour of more specific types like ”Politician”, which a) reduces the profile size and b) provides more meaningful information about the user for the recommendation step to take into account (e.g. an interest in ”Politics” in general). The THD entity classification module (cf. D2.7) is used to extract types per entity and an embedded Linked Profiler module is used to filter out generic entities.

– It conveys the common entities (DBPedia classes) within the GAIN interests and the types retrieved in the previous step to types from the LUMO ontology (`http://data.linkedtv.eu/ontologies/lumo/`), based on the mappings within in the LUMO mappings ontology (`http://data.linkedtv.eu/ontologies/lumo_mappings/`). This step is necessary in order to bring the profile within the scalable and expressive LUMO concept space.

– It communicates with the InBeat Preference Learning (PL) module, sending all extracted LUMO-based preferences and receives corresponding LUMO-based association rules for a user.

– It aggregates the weight of preference and the frequency of appearance of a specific preference in all interactions of a user. This happens for atomic (one entity or entity-type pair) preferences.

– It stores the last timestamp by which an atomic or rule preference appeared in the user profile, based on which it applies a time decay factor to the final preference weight, in order to reduce importance of old preferences in the profile and bring to surface the most current ones.

– It produces a machine readable profile for each user in a JSON format, which includes the timestamp, preference (concept, entity-type pair or rule), aggregated weight, frequency, positive or negative status of the preference (interest or disinterest) and context attributes for all preferences retrieved for a given user, called ”plain profile”.

– Finally, it produces the end ontological user profile in the ontological KRSS syntax, from the ”plain profile”. The end profile incorporates the top-N interests and top-N disinterests (N to be established empirically for this experiment to 20). In this step, the final current preference weight is calculated, based on the aggregated weight, frequency and time decay as mentioned before [3]. This profile is the input for the recommendation step.

3.5.5 Linked Profiler Evaluation metrics and results

In the premises of the experiment, after users finished the session of interactions with the system, as described previously in this chapter, they were presented with an online survey which displayed their top 10 overall interests and top 10 overall disinterests as captured by Linked Profiler. Each interest/disinterest was coupled with an indication of the predicted weight of preference (simplified visually to 1-5 stars, corresponding to an [0.2,1] interval of weights). An example of the questionnaire presented to the users is displayed in Figure 12 (right).

The users were asked to rate the accuracy of each preference in accordance with its predicted weight on a scale of 0-5, according to their actual preferences, as they think those should have been captured based on the content they viewed. I.e. when presented with interest A with five stars, they were asked to evaluate whether they actually preferred interest A, as high as five stars.

It is worth mentioning that one user had no profile results, for reasons which remain to be investigated, although it is presumed that for whatever technical reasons this user was unable to complete a sufficient amount of interactions for the preference tracking and learning tools to yield results. It is also worth mentioning that one user performed solely negative interactions, thus yielding only disinterests and no interests.

The results of the questionnaires can be seen in Figure 16, which illustrates the average rating that each user gave to the presented predicted preferences, normalized in the [0,1] interval. The rating is grouped per interests (top 10), disinterests (top 10) and the overall combination of both. When the overall score coincides with the disinterest score for a user, that user had no positive interactions.

During the analysis of the results, it was observed that the preferences with the lower predicted weights received lower user ratings. This is to be expected, as low weighting scores correspond to low semantic significance, and thus can be considered as statistical errors. In a real-world setting, as more user interactions are accumulated over time and content consumption and preferences are consolidated, preferences with low semantic significance (i.e. with weights <0.3) are expected to be outliers and pruned out of the final user profiles. Therefore, we also analyzed the ratings of the users for preferences over an empirical threshold of semantic significance (>0.3), as seen in Figure 17. As expected, the rating population in this case had minimal deviations for each group (interests, disinterests, overall) and an overall better performance. Under this threshold, more users with insignificant positive preferences arose, thus having no interests but only disinterests. Again, when the overall average rating score coincides with the disinterest rating score for a user, that user had no interests.

Table 7 presents the average ratings for all the participants, normalized in the [0,1] interval, per case (interests, disinterests, overall), for both the full spectrum of preferences rated, and also for preferences that had a predicted weight >0.3. It also displays the standard deviation per case. User satisfaction regarding interests drops in the *no threshold* case, potentially because several users focused more on negative interactions. In contrast, in the *>0.3 threshold* scores, for the most semantically significant

[3]for more details on how the profile is produced within Linked Profiler and how the final weight is calculated, cf. deliverable D4.4

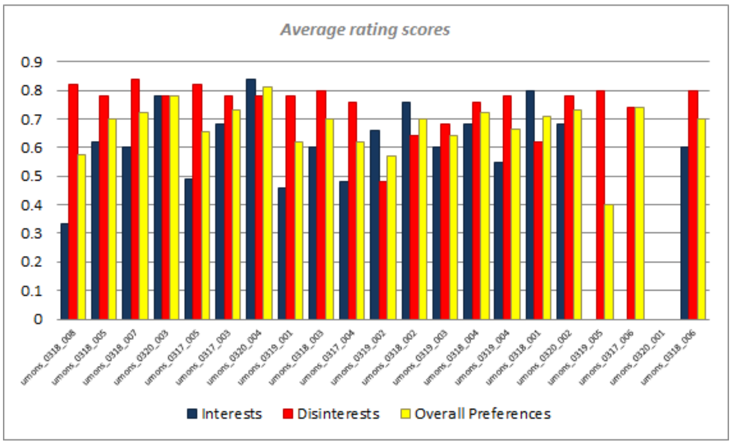


Figure 16: The average preference ratings per user, normalized in [0,1].

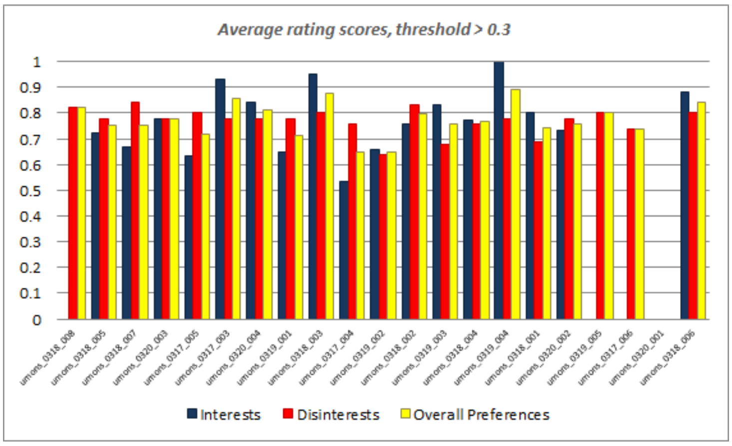


Figure 17: The average preference ratings per user, for preferences with predicted weight >0.3, normalized in [0,1].

preferences, the user satisfaction is high and consistent. The low standard deviation, inclining to 0, shows that there was were not significant deviations between the ratings that the users provided and the preference weights, and thus user satisfaction was high and consistent.

	Interests	***Disinterests***	***Overall***
		No Threshold	
Macro-average rating	0.590	0.751	0.674
Standard Deviation	0.186	0.083	0.087
		Threshold >0.3	
Macro-average rating	0.774	0.771	0.773
Standard Deviation	0.121	0.049	0.064

Table 7: The macro-average preference ratings for all participants, normalized in [0,1], and the standard deviation for each population.

3.5.6 LiFR-based recommendations setting

After the end of the user trials, the user profiles learned were used to produce recommendations for all the pseudoshots of the trial's video using WP4's recommendation service, which is based on the LiFR semantic reasoner. The LiFR-based recommender received as input the final semantic (KRSS) user profile learned and the semantic descriptions for all pseudoshots per user. Finally, the recommender produced a degree in the [-1,1] interval, denoting how much each pseudoshot matched (or not) the user profile per user.

The semantic descriptions of the pseudoshots consisted of the entities that annotated each pseudoshot along with their types, those types having been translated from their original DBPedia entries to LUMO, and formatted in the KRSS syntax. During the translation of DBPedia types to LUMO classes, again the most specific DBPedia types were considered, while the more generic types were filtered out, in order to minimize the size of the content descriptions and boost scalability, as inference of the more generic types takes place by default during reasoning.

3.5.7 LiFR-based recommendations results

The recommendation degrees for each pseudoshot per user were compared against the manual interest values the users provided for each pseudoshot. The metrics used to compare the recommendations to the user interest was the Mean Absolute Error (MAE), as described in Formula 1 (with y_i being the value of the recommendation degree), but also the Mean Squared Error (MSE), as described in Formula 2. MSE, in contrast to MAE, gives rise to distances that deviate more from the mean, i.e. can illustrate whether there are cases where large distances between interest values and recommendation degrees are observed (e.g. a recommendation degree of -1 as opposed to an interest of 1). MSE also relays bias, precision, and accuracy in statistical estimation of a predicted value (recommendation degrees) against the observed value (user interest).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(t_i - y_i)^2 \tag{2}$$

where n is the number of shots in user trials, t_i is the user interest value for a specific shot and y_i is the degree of recommendation for that shot produced by the *LiFR-based recommender*.

The MAE and MSE per user can be seen in Figure 18. A somewhat larger MSE than MAE can be observed for some users, however the small difference between the two measures reveals that there were seldomly large errors in the predictions/recommendations. And as can be observed by the average MAE and MSE for all users in Table 8, the low overall error, close to zero, relays a very good performance of the recommender system.

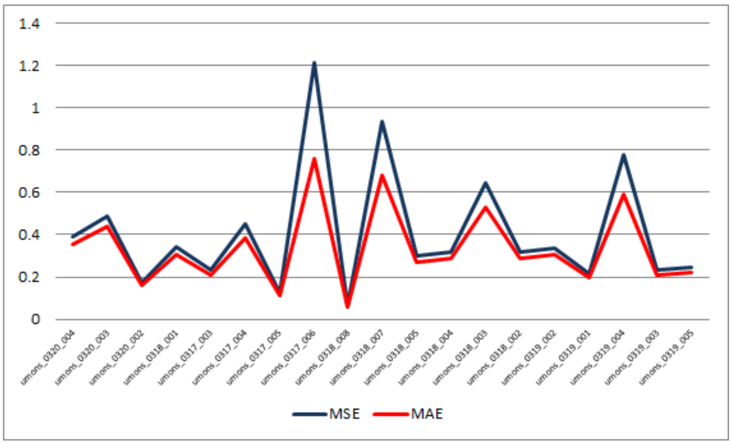


Figure 18: The MAE and MSE per user for the total of pseudoshots.

	MSE	**MAE**
Average	0.412	0.333

Table 8: Macro-average MAE and MSE for all users.

4 Personalization in the LinkedTV user trials

In conjunction with WP 6, two sets of user trials took place in the premises of partners RBB and MODUL. Their objective was to assess the overall quality of personalized recommendations of seed and enrichment content for the users that took part in the trials.

Indirectly, since the personalized recommendation service is the last point in the automatic content analysis and delivery pipeline, this experiment also reflects the collaborative performance of many tools in the LinkedTV pipeline, mainly the performance and cooperation of the entity extraction and classification services of WP 2 and of the entire implicit personalization pipeline of WP 4.[4]

The main goals of the trials can be broken down into two aspects:

- **Quantitative evaluation**
  - Automatically learn user profiles from training content
  - Evaluate automatically produced recommendations (from those profiles) against ground truth manual user ratings of test content
- **Qualitative evaluation**
  - Do LinkedTV personalisation options add value to the service?
  - Privacy and presentation of personalized services

4.1 Longitudinal tests at RBB

The first set of trials took place in Brandenburg, and partner RBB was responsible for conducting it. Six users took part in this trial. They were assigned pseudonyms (Annie, Lisa, Steve, Marvin, Harry, Martha) which comprised their user IDs in the trials' application.

The users were met with the RBB task leader, were instructed by the task at hand and were given tablets to take home and use the system on during a period of five days (02.03.2015 – 06.03.2015).

Each day they would view one news show, the show that aired on RBB the previous day, and go through the enrichment content. They spent roughly one hour per show. At the end of the five days, they were provided with and completed a questionnaire, assessing qualitative aspects of the personalization service.

4.2 End user trials at MODUL

A week later from the RBB trials (12.03.2015 and 13.03.2015), the second set of trials took place in MODUL, Vienna, with 5 users from MODUL. These users were assigned with generic user IDs (MODULUser01-05).

The setup was a little different than the previous trials, in the sense that the users were not handed tablets, but rather conducted the trials on their desktops/laptops. Also, the trials weren't week-long, but rather conducted at the same day, with around one hour allocated for the training phase and one more hour for the testing phase. Again, at the end of the testing phase, the users completed the qualitative assessment questionnaire.

4.3 Trials setup

The backbone of the setup for both trials was similar. It consisted of having users view and interact with RBB news shows on the Linked News application (for more information about the application, its webplayer and shows presented, cf D6.5). Users would also explore and interact with the offered enrichments for the shows. These interactions were tracked by GAIN, while all stored data was anonymized for all users in both sets of trials.

The users took part after signing consent forms (available in D6.5) and were instructed accordingly about how they can use the web player, what actions they are expected to perform on it and an overall picture of how these actions affect the purpose of the trials.

[4]Due to the trials being held in RBB and MODUL (Germany and Austria, respectively), so remotely of the context tracking partner UMONS (Belgium), it was not possible to integrate usage of Kinect for these trials. Therefore, they reflect only the non-contextualized personalization pipeline.

Each trial was split into two parts: a) the training phase, were each user's profile was captured and learned and b) the test phase, were users would rate content based on what they thought should be recommended/filtered for them.

The two sessions are different in purpose and the users were instructed to react accordingly. In the first part, they were told that they are freely showing preference to some subjects, topics, entities, based on their interactions with the content in webplayer. In the second part they were were instructed that they were not declaring interest anymore, but were asked to consider what they think the system should have learned about them in part 1 (in their opinion) and how this should have affected the content shown in part 2. In other words, in part 2, they were instructed to ask themselves: if this content was given to me by a recommendation system, how accurately do I think this recommendation system did? User instructions will be presented in more detail in the following sections.

4.3.1 Training phase

The training part consisted of having users view and interact with three RBB news shows on the Linked News application webplayer. They would view the news show and also explore and interact with offered enrichments for each chapter of the show.

The users were instructed about the actions they could perform on the player and how this would affect their profiles. The actions that could be performed and be interpreted by GAIN were: bookmarking/unbookmarking a chapter, giving a thumbs-up/thumbs-down to a chapter, giving a thumbs-up/thumbs-down to an enrichment. They could also choose to not perform any action on a chapter/enrichment. Snapshots of these actions can be seen in Figures 19 and 20.

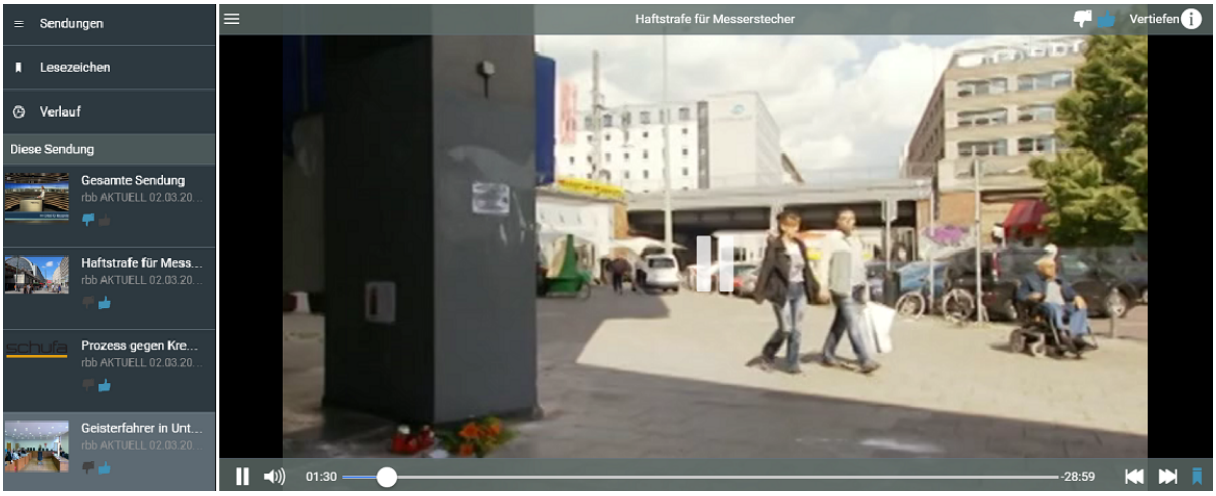


Figure 19: View of the interest tracking actions for a video chapter on the webplayer (left: chapters menu, right: main screen view)

Figure 21 shows the description of assigned tasks, as it was presented to the users.

During this phase, user interactions were captured in GAIN. GAIN also recorded the entities that annotated each chapter or enrichment that the user interacted with and assigned a preference degree to them, based on the type of interaction, as outlined in Table 9. Association rules among those preferences were learned in the InBeat Preference Learner (PL). A combination of GAIN-tracked preferences and association rules were learned over time and frequency by the Linked Profiler, which produced the final user profile, in the same manner as described in subsection 3.5.4.

Interaction	Preference
Bookmark	1.0
Unbookmark	-0.5
Thumbs Up	1.0
Thumbs Down	-1.0

Table 9: Overview of preference value per type of interaction for the training phase

Figure 20: View of the interest tracking actions for an enrichment on the webplayer

4.3.2 Testing phase

The testing part consisted of having users go through two RBB news shows, different than the previous ones, on the webplayer. They would view or browse quickly through all chapters of the news shows and rate them according to how they think these should be filtered for them. They also explored the offered enrichments for the shows and similarly rated them. They were instructed that the criterion for their ratings should be adapted to what they thought the system should have learned about them, in their opinion, during the training phase.

The users were again instructed what the significance of these ratings is. The actions that could be performed and be interpreted by the evaluation were: positively rating a chapter/enrichment, in a scale ranging from 1 to 5, or rejecting a chapter/enrichment. Snapshots of these actions can be seen in Figures 22 and 23.

Below follows the description of assigned tasks, as it was presented to the users.

The ratings were again tracked by GAIN as user interactions. The rating scores were translated in the $[-1,1]$ interval by GAIN, as seen in Table 10, and were used as reference to the automatic recommendation process.

Interaction	Rating
1 Star	0.2
2 Stars	0.4
3 Stars	0.6
4 Stars	0.8
5 Stars	1.0
Reject (X)	-1.0

Table 10: Overview of rating value per type of rating for the testing phase

4.3.3 Content annotations setup

As was mentioned before, the content is annotated with entities, which the WP4 profiling workflow uses to learn user preferences, in the same manner as described in Section 3.5.4, and WP4's LiFR-based recommendation service uses to match content to a given user profile (as described in the following section).

For these trials, the entities that annotated the news shows' chapters were manually provided by RBB editors, using LinkedTV's Editor Tool. The entities in the enrichments were three-fold, since there were three dimensions of enrichments available in the webplayer:

1. Please watch the news of days 01.03.2015 03.03.2015
    a. rbb AKTUELL 01.03.2015 21:45
    b. rbb AKTUELL 02.03.2015 21:45
    c. rbb AKTUELL 03.03.2015 21:45
2. Check out all chapters; you can skip through the chapters, watch at least a part to get an idea of what they are about or watch them completely.
    a. You can bookmark chapters to see later. Be informed that bookmark also signals a preference for this content item to the personalization system.
3. Please click thumbs up/down (at the top right of the main screen or below each chapter at the left-hand side chapter menu) if you are interested/not interested in the chapter or, if you are indifferent, simply do nothing and skip the chapter
    a. **thumbs up** means that you ”interested in more information” on this subject (or combination of subjects), not necessarily that you like or approve of what is being said!
    b. **thumbs down** means that you never want to see content about this subject (or combination of subjects) – i.e., a filtering service should filter out this kind of content for you.
    c. **neglecting/skipping** means you are not interested but neither have a strong objection about the subject (or combination of subjects).
4. Explore enrichments for interesting chapters while the video plays (it will pause for the time you take for exploring) or later. You can explore enrichments for a chapter by clicking the 'Explore' button on the right of the top menu bar.
5. Click thumbs up/down (at the top-most right corner of the screen) to indicate preference for an enrichment, or do nothing (neglect) if you are indifferent about it. The thumbs up/down buttons mean the same as before (see 3a-3c).

Figure 21: Users Assigned tasks – Training phase

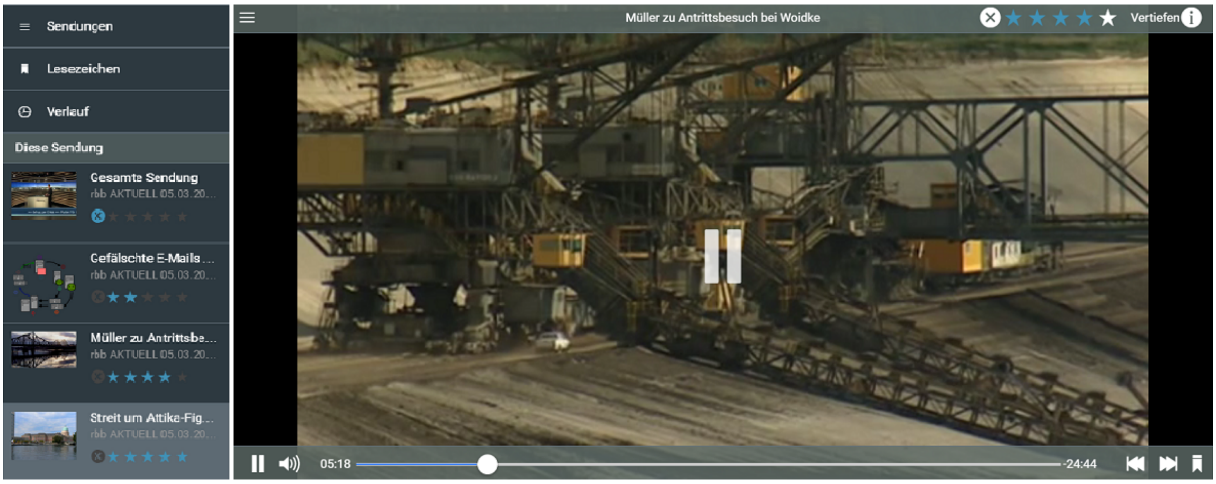


Figure 22: View of the rating actions for a video chapter on the webplayer (left: chapters menu, right: main screen view)

Figure 23: View of the rating actions for an enrichment on the webplayer

– The *related entities* dimension, consisted of a list of the entities that the chapter at hand was annotated with. When the user interacted with an item under this dimension, the input for the personalization tools (profiling or recommendation) was that sole entity.

– The *related articles* dimension, consisted of a list of web-based articles that the RBB editors found relevant to the chapter at hand. When the user interacted with an item under this dimension, the input for the personalization tools was extracted from the HTML of that web article, in a process that followed a series of steps:

a. The webplayer's embedded content proxy retrieved the HTML content of the web page, stripped it of irrelevant text (e.g. menus, sidebars), and stored it locally.

b. The body text of the stripped HTML was run thought the THD Entity Extraction (WP2) service, in order to extract DBPedia entities (resources) from the text. The salience score of extracted entities a) was employed in order to prune the less significant entities and minimize the input volume and b) for the remaining entities, it was conveyed across the next steps to take into account in the final step of the profiling and recommendation.

c. The THD Entity Classification (WP2) service was employed in order to retrieve DBPedia types (from DBPedia Ontology and/or the Linked Hypernyms Dataset) for the extracted entities.

On the second part of the trials, you will see a small change in the interface: Instead of Thumbs Up/Down you will see 5 Stars and also next to them on the left a Reject (X) button: at the top right of the screen of the enrichments view and the main screen view, and at the left-hand chapter menu, below each chapter. Stars can be clicked on, denoting a scale of 1 to 5 stars from left to right. The reject button can also be clicked.

1. Please watch the news of days 04.03.2015 05.03.2015
   a. rbb AKTUELL 04.03.2015 21:45
   b. rbb AKTUELL 05.03.2015 21:45
2. Check all chapters; either skip through the chapters or watch them completely.
3. For each chapter, please rate it using the stars/reject buttons (stars: positive rate in a scale of 1–5, reject: negative rating), or if you are indifferent, simply skip the chapter.
4. Please explore all the enrichments of all the chapters.
5. For each enrichment, please rate it.
6. Rating denotes how much you feel this chapter should have been recommended/not shown to you given your interactions (thumbs up/down, playing, skipping) during part 1.
   a. 1 Star means that this enrichment/chapter has little to no relevance to your interests as you think those should have been captured in part 1.
   b. 5 Stars means that this enrichment/chapter has nearly perfect relevance to your interests as you think those should have been captured in part 1.
   c. Reject (X) means that this enrichment/chapter should be rejected (not have been displayed) for you based on your disinterests as you think those should have been captured in part 1.

Figure 24: Users Assigned tasks – Testing phase

d. For the DBPedia ontology types of the last step, a custom mechanism was built to filter out the more generic types per entity (e.g. "Agent" for every person that appears in the texts) that would appear more frequently and misdirect personalization results and keep only the most specific and characteristic types for an entity (e.g. 'Politician' that gives an outlook of the general topic of the text).

e. Finally, these entity-type pairs were the input for the personalization tools.

– The *related chapters* dimension, consisted of a list of chapters from RBB video content that the editors found relevant to the chapter at hand. The entities that (should have) annotated these chapters would also rely on manual editor input. Unfortunately, not many – if any – related chapters were already pre-annotated, so this dimension was not taken into account for the end results.

Due to a technical problem and unforeseen requirements, the initial content annotations sent to GAIN by the Linked News application had to be revised. The technical problem occurred in day 1, when due to a minor detail of the Editor Tool, the entities assigned to the chapters of the day 1 video by the RBB editors, were not conveyed to the Linked News application with their proper DBPedia manifestation. This was quickly amended in the Editor tool, but the user interactions of day 1 were recorded with this error, rendering user interactions with chapters and related entities on day 1 unusable. Also, although originally the annotations for related articles came directly from the IRAPI enrichment service (cf. D2.7), it was observed during the trials that the volume of entities coming from IRAPI per article was substantially large, since all extracted entities, even ones with really low relevance to the text, were passed on to the Linked News application, which put a significant load in the communication of all involved services. Also, the relevance of each entity needed to be recorded for the recommendation service (refer to the following section) to take into account, while IRAPI did not provide the THD relevance (salience) score. Due to time limitations, we did not convey this relevance-score requirement (i.e. prune entities by relevance and transmit relevance score) to IRAPI, but rather reproduced related articles' annotations again directly from THD, as described in step (2c) of the list above.

Therefore, after the end of the trials, all recorded user interactions were re-sent to GAIN per user, amending the lost chapter annotations of day 1 and improving the related articles' annotation with relevance scores, while minimizing their volume to the most significant entities, by applying a salience threshold of $>$0.3 for accepted entities. Consequently, user profiles were re-learned. Due to this amendment, the opportunity arose to evaluate the personalization workflow with two different settings of received content annotation.

The editors in the Editor Tool do not provide types for entities, by default. In contrast, the entities provided by THD (either directly by THD or via IRAPI) for related articles are always accompanied by their types, were available. This prompted us to examine the performance of the WP4 workflow for content which is annotated with entities but not their types (hereafter referred to as the *"No types"* case) and for content in which extracted entities are coupled with their types (hereafter referred to as the *"With types"* case). The existence or not of types in the input content annotation was expected to impact user profiling, as more general preferences (e.g. "sports", "weather") would be captured by GAIN and the most persistent ones could surface in the Linked Profiler, and as a consequence impact also recommendation. So in the process of resending user interactions to GAIN, we created two aspects of the users, one with "No types" and one "With types" - thus two profiles per user.

4.3.4 Questionnaire

As part of evaluating the qualitative goal of the trials, after the users were finished with the testing part, they were handed a questionnaire. In it, their opinions about the added value of personalization to the LinkedTV product, as well as their preferences in using a personalized service and questions regarding privacy, were asked (Figure 25).

4.4 LiFR-based recommendation evaluation setup and metrics

After the trials were concluded and the entities in user interactions were amended, automatic recommendations were produced for all the chapters and enrichments of the news shows viewed by users in the test phase (i.e. for videos 'rbb AKTUELL 04.03.2015' and 'rbb AKTUELL 05.03.2015') by the LiFR-based recommender (cf. D4.5), based on the (amended) user profiles captured on the training phase, for both the *"No types"* and the *"With types"* cases.

1. Would recommendation of content bring added value to the service you've just used?
   (a) Yes
   (b) No
   (c) I am not sure
2. Would you allow your interactions to be tracked so that the service can provide you content tailored to your preferences?
   (a) Yes
   (b) Yes, but only if I was aware about it.
   (c) Yes, but only if I was aware about it and I am able to see, edit and delete the information I send.
   (d) No, not at all, I don't like being tracked.
3. If you would allow interaction tracking, where would you prefer the information tracked to be stored?
   (a) Anywhere, I don't have a preference
   (b) On a server to be accessible always
   (c) On my own tablet/mobile/pc, I don't want my information to be available to a third party
   (d) As I stated before, I don't want to be tracked
4. Would you like to see indication of recommended chapters in a news show, so you can visit them first/go through them in more detail/skip chapters that might not be interesting?
   (a) Yes
   (b) No
   (c) I am not sure
5. Would like to see recommendations of related content to a chapter?
   (a) Yes, with an indicator that this content is of interest to me (e.g. "Recommended for you")
   (b) Yes, with an indicator that also shows how interesting the content is for me (e.g. 1-5 stars)
   (c) Yes, not with an indicator, but content should be re-arranged (ranked), so the most interesting content would come first.
   (d) No
6. Would you prefer if content that the system learned that you do not like be filtered out for you?
   (a) Yes, I don't want to see it at all
   (b) Yes in the case of related content, no in the case of news show chapters
   (c) Kind of, I don't want to miss any content, but if an indicator pointed out that I probably don't like it, then I would go quickly through it or skip it
   (d) No, I want everything to be on my plate

Figure 25: Questionnaire on Personalisation

The LiFR-based recommender received as input the final semantic (KRSS) user profile, learned from user interactions in the training phase, and the semantic descriptions for all content (chapters, related entities, related articles) of the testing phase per user. Finally, the recommender produced a degree in the [-1,1] interval, denoting how much each content item matched (or not) the user profile per user.

The semantic descriptions of the testing phase content consisted of the entities that described each content item, along with their types, those types having been translated from their original DBPedia entries to LUMO, and formatted in the KRSS syntax. During the translation of DBPedia types to LUMO classes, again the most specific DBPedia types were considered, similarly to the setting of Section 3.5.6.

The recommendations generated by LiFR were compared against each user's explicit ratings for this content, measuring the MAE and MSE (Formula 2) between the predicted recommendations degrees and the explicit user feedback. In this case however, due to the fact that explicit user ratings could be no less than 0.2 apart, in contrast to the MAE measured in Formula 1, a window of error of $\pm$ 0.1 was allowed, modifying the MAE formula to the following:

$$MAE(window) = \frac{1}{n}\sum_{i=1}^{n}|t_i - (y_i \pm 0.1)| \tag{3}$$

where n is the number of content items (chapters, related entities, related articles) in the two test phase videos, t_i is the explicit user rating for a specific content item and y_i is the produced recommendation degree for the item.

4.5 Results and outlook

The results of the MAE*(window)* and MSE per participant can be observed in Figures 26 and 27 respectively. In contrast to the results of Section 3.5.6, the MSE is lower than the MAE here. This is because while MSE penalizes more the greater distances between the predicted value (recommendation degree) and the observed value (explicit user rating), it also favours small distances (errors). Therefore it provides an important indication towards the accuracy of the recommendation results, which have a significant convergence to their explicit ratings counterparts.

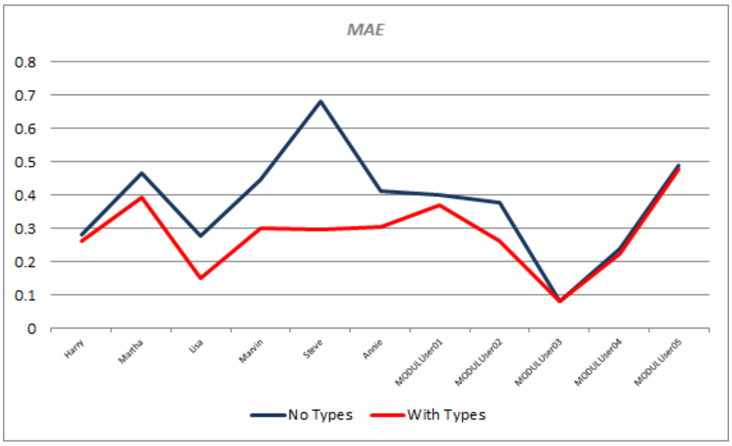


Figure 26: The MAE*(window)* for the total of content items per user.

The good overall performance of the recommendation, based on the low MA and MS errors, can be observed in Table 11. The substantial benefit of passing types along with entities in the annotation of content items for personalization is also apparent when comparing the scores of the *"With Types"* case as opposed to the *"No Types"* case.

	MAE	**MSE**
No Types	0.378	0.356
With Types	0.284	0.249

Table 11: Macro-Average MAE and MSE for all participants.

In retrospect, it can be observed that the recommendation accuracy in these trials is higher than the recommendation accuracy of the implicit contextualised profiling trials of Section 3.5.6, although not to

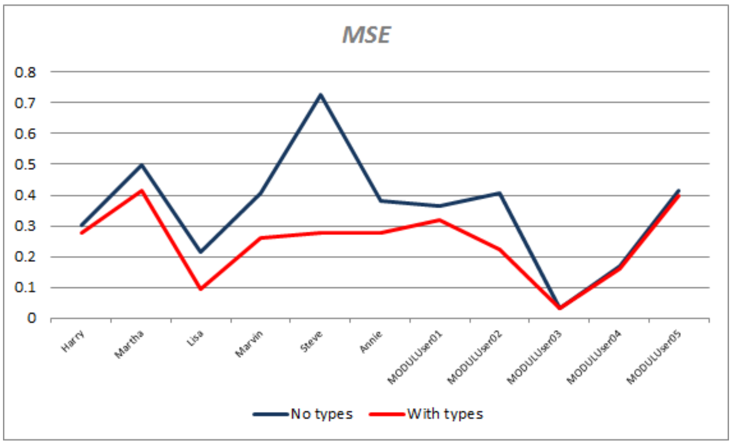


Figure 27: The MSE for the total of content items per user.

a great extent. This is a subject worth further scrutiny, but early assumptions can point towards either a) the better quality of content annotations, produced via the entire LinkedTV workflow in these trials as opposed to the annotations in the trials of Section 3, or b) the benefit of the more explicit user interactions (thumbs up/down), which should be substantial for such a limited amount of user transactions with the system, or c) both. However, the high accuracy of recommendations and small distance in error in both trials, verifies the good performance of the overall implicit personalization workflow in LinkedTV, with or without explicit interest indicators.

4.5.1 Questionnaire responses

The user responses in the questionnaire of Section 4.3.4 are displayed in Tables 12 and 13. They revealed interesting differences between the RBB and the MODUL users, which might be tracked back to their professional background: in the RBB case, the users were professionals of the broadcasting industry, while in the MODUL case, the users were science and technology professionals, thus arguably having an inclination towards new technologies.

Q1: added value		*Q2: tracking*		*Q3: storage*		*Q4: chapter recom*		*Q5: enrichment recommendation*		*Q6: filtering out*	
Yes	1	Yes	0	Anywhere	0	Yes	1	Yes, boolean indicator	**4**	Yes	0
No	1	Only if aware	0	Server	1	No	**3**	Yes, stars indicator	0	Yes, only for enrichments	0
Not sure	**4**	See, edit & delete	**3**	Own device	2	Not sure	2	Yes, ranked	0	Kind of	1
		No	**3**	No tracking	**3**			No	2	No	**5**

Table 12: Questionnaire responses for RBB users.

Although not having seen recommendations presented to them, many SciTech MODUL users can see the benefit of the service to their LinkedTV experience, whereas the RBB users remain unsure.

Unexpectedly, in terms of data privacy, MODUL users do not opt for the more ”safe” client-side storage, but are comfortable with having their data on a central server, opting for access everywhere, anytime, from any device. In contrast, RBB users prefer not to be tracked at all, and if tracked, storage in their own device is important.

Although chapter recommendation is prominent for MODUL users, RBB users find it of little to no use. Filtering out disinteresting content is also unwanted by the RBB users, while the MODUL users mostly value an indication if a content item is uninteresting, while some opt for rejection of irrelevant enrichments.

Both worlds meet in the control over personal profiles and in enrichment recommendations. In terms of profile management, they prefer a system that enables them to see, edit and delete their profiles. In

Q1: added value		*Q2: tracking*		*Q3: storage*		*Q4: chapter recom*		*Q5: enrichment recommendation*		*Q6: filtering out*	
Yes	**3**	Yes	1	Anywhere	0	Yes	**4**	Yes, boolean indicator	**5**	Yes	0
No	0	Only if aware	0	Server	**3**	No	0	Yes, stars indicator	0	Yes, only for enrichments	2
Not sure	2	See, edit & delete	**4**	Own device	1	Not sure	1	Yes, ranked	0	Kind of	**3**
		No	0	No tracking	1			No	0	No	0

Table 13: Questionnaire responses for MODUL users.

terms of recommending enrichments, most prefer a boolean indicator ("Recommended for you") to point out interesting enrichments, while degree of interest or rank is not valuable to them.

It is hard to make a generalized observation from the qualitative study for all types of users, as it seems that many attributes depend on the users' specific background. However, the two converging observations of questions 2 and 5 have provided a very good lesson on profile management and recommendation display for the general audience and will be taken into consideration for the application of WP4's personalized services beyond LinkedTV.

5 Outlook: Feature sets for client-side recommendation

With increasing regulation and demands on user data privacy running a recommender system client-side is becoming a viable option. Client-side recommendation requires enrichment of the videos delivered to the user with a list of related videos (or related content in general) selected using a content-based similarity measure.

Interest Beat (InBeat) is a generic recommender system that has been adapted to perform recommendation of related content to users of online TV. In this section, we present a work towards adapting InBeat to perform client-side recommendation: re-ranking the list of related content according to the user's profile. This approach is *privacy preserving*: the model is built solely from relevance feedback stored locally and the model building as well as execution is performed on client's hardware.

A possible bottleneck for client-side recommendation is the data volume entailed by transferring the feature set describing each video (both requested and a list of related ones) to the client, and the computational resources needed to process the feature set. We investigate whether the representation of videos with Bag of Entities (BoE) which is used in InBeat is more compact than the standard Bag of Words (BoW) approach.

In this section, we present the evaluation of the Bag of Entities representation used in InBeat. We cast the problem as *text categorization* task in order to be able to leverage large existing evaluation resources. Essentially, there are three types of documents: those for which the user interest is known to be positive, negative and neutral.

The experimental setup aims at comparing the performance of the BoW representation with the BoE representation. The comparison is performed on two versions of the classifier: brCBA and termAssoc, which are described in [KK14]. The brCBA is a simplified version of the seminal Classification by Association Rules (CBA) algorithm [LHM98]. The termAssoc is a modified version of the ARC-BC algorithm for text categorization by term association proposed in [AZ02].

5.1 Dataset

We use the ModApte version of the Reuters-21578 Text Categorization Test Collection, which is one of the standard datasets for the text categorization task. The Reuters-21578 collection contains 21,578 documents, which are assigned to 135 different categories (topics). Example topics are "earn" or "wheat". One document belongs on average to 1.3 categories. We use only a subset consisting of the documents which are assigned to ten most frequently populated categories as, e.g., in [AZ02]. Our dataset thus consists of 6,399 training documents and 2,545 test documents.

5.2 Preprocessing

The preprocessing is performed in two stages. First, the BoW or BoE feature sets are created from the underlying dataset. Then, depending on the classifier used, the term (concept) vectors are pruned.

5.2.1 BOW

The input documents contain 58,714 of distinct terms. To decrease the dimensionality, we performed the following operations: all terms were converted to lower case, numbers were removed, punctuation was removed, stop words were removed,[5] whitespace was stripped and the documents were stemmed. The final document-term matrix contained 25,604 terms.

5.2.2 BOE

The THD entity classifier available at `http://entityclassifier.eu` [DK13] was used to wikify the documents. The web service returned a list of entities (identified as DBpedia resources) and for each entity a list of the types (DBpedia Ontology concepts).

The result of the preprocessing is a document-term matrix containing 12,878 unique concepts (entities and types).

[5] A list of occurring 700 English stop words was used.

5.2.3 Term (concept) pruning

The rule pruning is performed differently for brCBA and termAssoc algorithms. For brCBA, top N ($tvSize$) terms are selected according to TF-IDF. For termAssoc, term pruning is performed separately for each category using a TF score, selecting top N ($tvSize$) terms. Using TF-IDF scores with termAssoc degrades results in our observation, since terms with low IDF value (computed on terms within a given category) often discriminate well documents in this category with respect to documents in other categories.

We also tried combining the BoW and BoE representations (denoted as BoW+BoE). For a given value $tvSize$ parameter, 50% were top-ranked terms from BOW and 50% top-ranked concepts from BoE.

5.3 Rule learning setup

To perform the experiments, we used the *minConf*=0.001 threshold, *minSupp*=0.001 for brCBA, and *minSupp*=0.2 for TermAssoc. The maximum rule (frequent itemset) length was unrestricted.

5.4 Results

The results are reported in terms of micro-average and macro-average F-measure (refer to [BEYTW01] for details).

The results, depicted on Fig. 28–29, indicate that for the smallest term vector size, BoE representation yields better overall F-Measure than the BoW representation. Also, the best overall results are provided by the termAssoc algorithm.

Surprisingly, the fusion of BoW and BoE into one term vector is dominated by the performance of the BoW/BoE alone.

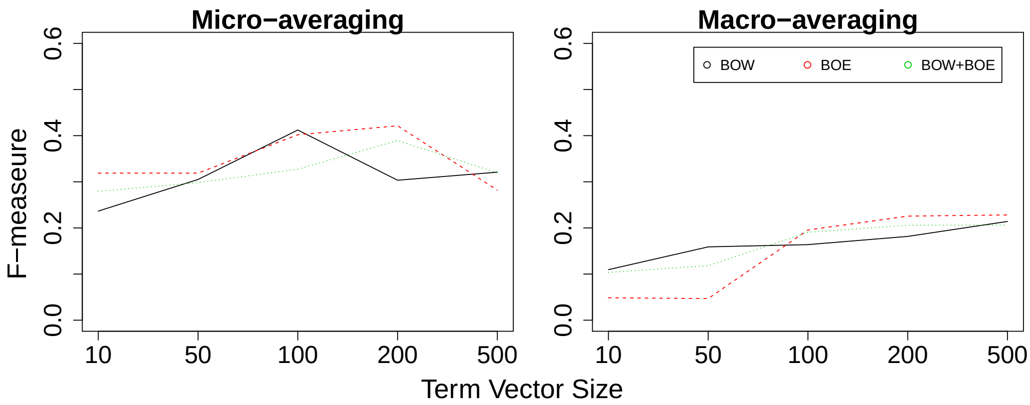


Figure 28: Results – brCBA

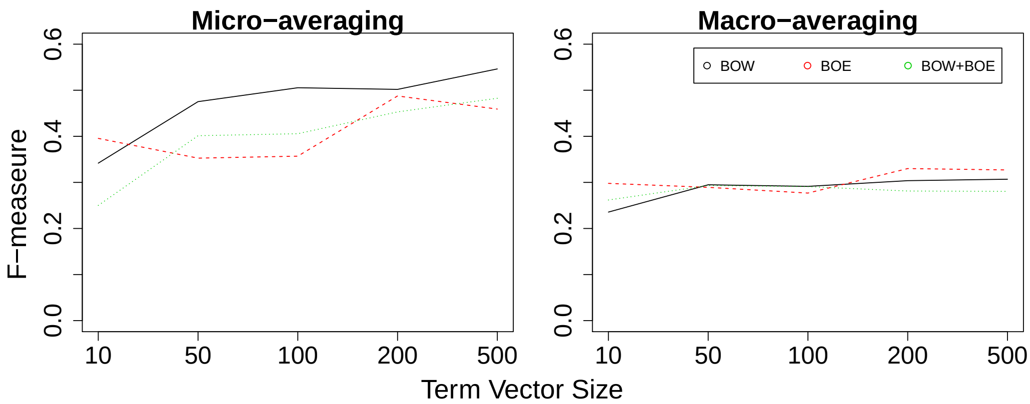


Figure 29: Results – termAssoc

It should be noted that significantly better results than we have achieved on Reuters-21578 are reported, e.g., in [BEYTW01] (although the relative improvement provided by the BoE representation is only marginal). We hypothesize that additional improvement can be obtained if the feature set reflects the entity salience information, which is made available by the latest release of the `http://entityclassifier.eu` service (refer also to D 2.7).

6 Conclusion

This deliverable has presented the final integrated implicit personalization workflow of LinkedTV. While individual tools have been previously presented in detail in previous WP4 deliverables, this document focused on presenting recent evaluations, which were performed either directly within the LinkedTV workflow, or on third-party datasets, or compared against the state of the art.

To this end, the evaluation results have consolidated the success in the performance of the personalization workflow, both within the general LinkedTV workflow as well as beyond it, thus validating its application to any given scenario and interface.

In addition, the standardization of the personalization tools and workflow is actively pursued well beyond the scope of LinkedTV. The LinkedTV personalization workflow prominently employs rule-based algorithms and in order to help advancing the state of the art, the LinkedTV partner UEP is participating on the organization of *Rule-based Recommender Systems for the Web of Data challenge*[6]. The dataset and the task is already available on-line, the results of the Challenge will be announced as part of the RuleML 2015 conference in Berlin.

Most of the individual components evaluated in this deliverable have been made available under an open source license, or open source release is in progress. This ensures that the software will be available well beyond the end of the project, and provides possibilities for future extensions and improvements. Also, exploitation strategies for most tools have been studied and presented in deliverable D 8.8.

[6] http://www.csw.inf.fu-berlin.de/ruleml2015/recsysrules-2015.html

7 References

[AZ02] Maria-Luiza Antonie and Osmar R. Zaïane. Text document categorization by term association. In *ICDM '02*, Washington, DC, USA, 2002. IEEE Computer Society.

[BEYTW01] Ron Bekkerman, Ran El-Yaniv, Naftali Tishby, and Yoad Winter. On feature distributional clustering for text categorization. In *SIGIR '01*, pages 146–153, New York, NY, USA, 2001. ACM.

[DK13] Milan Dojchinovski and Tomáš Kliegr. Entityclassifier.eu: real-time classification of entities in text with Wikipedia. In *ECML'13*, pages 654–658. Springer, 2013.

[faca] Faceapi. markerless face tracking application. `http://www.seeingmachines.com/product/faceapi/`.

[facb] Facelab 5. face and eye tracking application. `http://www.seeingmachines.com/`.

[HPH+05] Robert P Hawkins, Suzanne Pingree, Jacqueline Hitchon, Barry Radler, Bradley W Gorham, Leeann Kahlor, Eilleen Gilligan, Ronald C Serlin, Toni Schmidt, Prathana Kannaovakun, et al. What produces television attention and attention style? *Human Communication Research*, 31(1):162–187, 2005.

[KK14] Jaroslav Kuchař and Tomáš Kliegr. Bag-of-entities text representation for client-side (video) recommender systems. In *Proceedings of the RecSysTV'14 workshop*, 2014.

[KKSV14] Tom Kliegr, Jaroslav Kucha, Davide Sottara, and Stanislav Voj. Learning business rules with association rule classifiers. In Antonis Bikakis, Paul Fodor, and Dumitru Roman, editors, *Rules on the Web. From Theory to Applications*, volume 8620 of *Lecture Notes in Computer Science*, pages 236–250. Springer International Publishing, 2014.

[LHM98] Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In *KDD'98*, pages 80–86, 1998.

[qua] Qualisys. products and services based on optical motion capture. `http://www.qualisys.com/`.

[RC11] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE, 2011.

[RMG14] François Rocca, Matei Mancas, and Bernard Gosselin. Head pose estimation by perspective-n-point solution based on 2d markerless face tracking. In *Intelligent Technologies for Interactive Entertainment*, pages 67–76. Springer, 2014.

[SLC11] Jason M Saragih, Simon Lucey, and Jeffrey F Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2):200–215, 2011.

[VJ04] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.