Enhancing and Re-Purposing TV Content for Trans-Vector Engagement (ReTV)
H2020 Research and Innovation Action - Grant Agreement No. 780656

ReTV

Enhancing and Re-Purposing TV Content for Trans-Vector Engagement

Deliverable 6.2 (M20)

First Validation of Personalization Prototype

Version 1.0

This document was produced in the context of the ReTV project supported by the European Commission under the H2020-ICT-2016-2017 Information & Communication Technologies Call
Grant Agreement No 780656

DOCUMENT INFORMATION

Delivery Type	Report
Deliverable Number	6.2
Deliverable Title	First validation of Personalisation Prototype
Due Date	M20
Submission Date	31 August 2019
Work Package	WP6
Partners	CERTH, GENISTAT, RBB, NISV
Author(s)	Basil Philipp, Krzysztof Ciesielski, Alexis Charveriat (GENISTAT), Miggi Zwicklbauer, Willy Lamm (RBB), Rasa Bocyte (NISV)
Reviewer(s)	Jonathan Blok (NISV) Eleni Tzialli (NISV)
Keywords	Consumer Use Cases, TV Applications, Website Recommendations, Video Recommendations, Video Summaries
Dissemination Level	PU
Project Coordinator	MODUL Technology, Am Kahlenberg 1, 1190 Vienna, Austria.
Contact Details	Coordinator: Dr Lyndon Nixon (nixon@modultech.eu) R&D Manager: Prof Arno Scharl (scharl@weblyzard.com) Innovation Manager: Bea Knecht (bea@zattoo.com)

Revisions

Version	Date	Author	Changes
0.1	27.06.2019	Basil Philipp (GENISTAT)	Initial table of content
0.2	14.08.2019	Basil Philipp (GENISTAT)	First version of content from technical partners.
0.3	15.08.2019	Miggi Zwicklbauer, Willy Lamm (RBB)	Add Methodology, Scenario Preparation
0.3	16.08.2019	Rasa Bocyte (NISV)	Scenario Preparation
0.4	26.08.2019	Miggi Zwicklbauer (RBB)	Evaluation chatbot
0.5	27.08.2019	Krzysztof Ciesielski (GENISTAT)	Evaluation Content sWitch, Executive Summary, Conclusions and Outlook
0.6	28.08.2019	Jonathan Blok (NISV), Eleni Tzialli (NISV)	QA review
0.7	28.08.2019	Alexis Charveriat (GENISTAT), Basil Philipp (GENISTAT), Krzysztof Ciesielski (GENISTAT) Miggi Zwicklbauer (RBB)	Address QA comments
0.8	29.08.2019	Rasa Bocyte (NISV) Miggi Zwicklbauer (RBB) Krzysztof Ciesielski (GENISTAT)	Final QA review
1.0	29.08.2019		Final version

Statement of Originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

This deliverable reflects only the authors' views and the European Union is not liable for any use that might be made of information contained therein.

TABLE OF CONTENTS

1 Executive Summary

This deliverable focuses on the preparation of the consumer use case scenarios - Content sWitch and 4u2 chatbot., We developed the first prototypes for the two scenarios and executed their evaluation with consumers. The deliverable presents detailed descriptions of the two user scenarios and reports on the results of the user tests.

2 INTRODUCTION

The goal of task 6.2 is to design, develop and test the first version of the trans-vector platform (TVP) use cases for content and ad personalisation. In task 6.2 we:

- developed a methodology to gather qualitative and quantitative feedback from media consumers (cf. section 3);
- designed and developed the required prototypes that use the TVP modules of WP1-3 (cf sections 4 and 5);
- customised the prototypes for use at RBB and NISV taking into account their consumers and their content (cf. section 6);
- ran the first small-scale tests to evaluate these prototypes (cf. section 7).

The role of this task is to deliver the first round of feedback on running software so that the rest of the project can focus on iteratively improving the TVP with long-term tests through task 6.3, on the basis of the knowledge gathered from small-scale tests and running prototypes.

Based on the results from the deliverable D6.1 Requirements for Consumer Use Case, we implemented two use cases - 4u2 chatbot and Content sWitch - which cover the requirements of media consumers.

The 4u2 chatbot targets the *social media user* and *messenger user* personas from D6.1. It is meant to be an additional publication vector for broadcasters and media archives through which consumers can easily and quickly gain access to personalised content using a chatbot interface.

The Content sWitch targets the *targeted ad/content user* persona from D6.1. The Content sWitch is a software system that runs in the streaming system of a video streaming provider like Zattoo and it inserts personalised content or ads at opportune moments in the TV stream.

This deliverable describes the results of Task 6.2. It starts by introducing the methodology we used to gather quantitative and qualitative feedback (section 3) and then shows how we decided on the design of the use cases (sections 4-5). The subsequent section describes the preparation of content for the user tests. Section 6 describes the technical development of the scenarios, focusing on the development of the user interfaces and workflows. The technical work for the backend modules provided by WP 1-3 is described in the corresponding deliverables. Section 7 describes the results of the actual user tests. We conclude the deliverable by summarising our findings and giving an outlook into the future work.

3. METHODOLOGY

In the ReTV project, the content owners and the technical partners collaborated closely to prepare the consumer scenarios, develop the first prototypes and gather qualitative and quantitative feedback from potential consumers. Additionally, partners collected feedback on the consumer services from professional users.

The following approach was followed:

1. **Wireframes for consumer scenarios.** Based upon feedback from editorial departments and the outcome of the first consumer interviews in September 2018, wireframes were built for the chatbot/4u2 scenario workflow.

2. **Validation of user scenarios with consumer and professional users.** During the first validation period, RBB and NISV collected feedback on the consumer scenarios from the broadcaster's editorial departments and media professionals at RBB and NISV. Wireframes were demonstrated to consumers along with a questionnaire aimed at improving wireframes and workflows.
3. **Prototypes tailored for evaluation with consumer needs.** After optimizing the workflow of the chatbot, two prototype versions were built, one for RBB and one for NISV, and tested by consumers and were demonstrated to professional users. We also built and demonstrated a prototype of the Content sWitch. This prototype was also presented to professional users to get their input with regards to business viability.
4. **User testing**. User testing took place at GENISTAT, NISV and RBB in the first period. To gain more qualitative feedback, services were also tested with a smaller group of consumers in face-to-face interviews and think-aloud sessions. All tests incorporated questionnaires, the outcomes of which provided input for technical partners.

During this first validation period, GENISTAT, NISV and RBB focused on qualitative feedback from a smaller consumer group, which tested fewer features in smaller iterations. The goal was to have a more agile implementation phase to improve and optimize the consumer services.

4. Scenario Preparation

Consumer requirements gathered during the initial interviews and questionnaires at the IFA 2018 (presented in Deliverable D6.1) helped to specify features for the video summarisation service, the Content sWitch scenario and the chatbot/4u2 scenario.

Both scenarios tested in the first validation period were based on the automatic generation of video summaries. RBB and NISV provided media sources for user testing. Before the consumer tests took place, the content partners tested the Video Summarization Service of CERTH in several successive rounds. A detailed description of the entire test process with regard to the requirements covered and the resulting developments can be found in deliverable D3.2, Section 4.3.

At the same time, RBB built an initial version of the wireframes for the chatbot/4u2 that focused on the video summarisation feature and the recommendation functionality. The wireframes were tested with six consumers and the outcomes were integrated into the second version, subsequently provided to GENISTAT developers.

4.1 Content sWitch

The idea for the Content sWitch was born from the user feedback from the deliverable D6.1 but also Genistats own experiences with Dynamic Ad Insertion at Zattoo. Dynamic Ad Insertion (DAI) is a technology where ads are dynamically replaced within a live TV stream. The beer ad in front of a football game might, for example, be replaced by something with more general appeal, like a vacation ad, if the usage behaviour indicates a shared family account. DAI is in limited use at Zattoo, since it needs the cooperation of each individual TV channel. Along with their TV stream, channels need to send information that indicates upcoming ads and their length, something that is often done by embedding SCTE-35[1] markers within. Including those markers requires special and expensive hardware, which means that only a few channels are

[1] https://www.scte.org/SCTEDocs/Standards/ANSI_SCTE%2035%202019r1.pdf

technically able to allow DAI. Furthermore, inserting an ad in a live TV stream requires a very performant video streaming engine as well as fast response time from ad-servers; in the case of Zattoo, the whole request to have an ad inserted into a stream needs to happen in less than 10 seconds. We decided to give those technical challenges a lower priority for the time being, as we believe they will solve themselves over time: AI techniques to detect ads in live streams are getting more and more accurate (cf. works on ad detection reported in D1.2), negating the requirement of expensive hardware to include SCTE-35 markers. We do, however, believe that the idea of replacing content in a TV stream on a personal basis is a 'best of both worlds' scenario: users still get a sense of belonging when watching television, as their stream is mostly identical to what others see, only where necessary we personalise the offering giving them the experience they know from Internet video.

User feedback from D6.1 however indicated relatively low interest in advertisements, the most highly rated features were to see fewer ads and to have a timer to see how much longer the ad-break would last. Given this feedback and the fact that DAI is already used in practice, we decided to focus on Dynamic Content Insertion (DCI) believing that it will not only benefit the viewer, but mostly the broadcaster as it allows them to A/B test different trailers (i.e. assign them to groups that see different content and observe if there is a statistically significant difference in user engagement). Netflix shows the importance of content personalisation, as they A/B test different artworks to increase engagement.[2]

We did implement the first version of the Content sWitch that can replace advertisements or program trailers and also adapt them by making them shorter or longer in order to fit into an available slot in the program. Figure 1 demonstrates how three out of four users see the same ad, but for one user we show a different ad. The current small-scale test (cf. section 7.1) was focused primarily on testing what kind of content may be interested to the end users, and if the explicit (declared) preferences match the implicit interests (ratings given to each trailer/topic). In the next phase, we plan to integrate user tests with actual recommendations given by a model (cf. D3.2, task T3.4). In the ideal scenario, trailers with high ratings should be also high in the model-created rankings.

[2] https://medium.com/netflix-techblog/artwork-personalization-c589f074ad76

Figure 1: Screenshot of the Content sWitch replacing an ad for a single user

We extend on the idea of Dynamic Content Insertion in a blog post[3]:

"The fact that everybody gets exactly the same TV program is a feature of television that allows for relaxing routines, like the Sunday evening crime show, or watching the evening news as a family after dinner. It offers common ground for water cooler discussions at work and other social situations. For younger viewers though, the linear program does not seem to offer much value. A recent study looked at how much time different age groups spend on various media offerings. Two years ago, the 14-29-year-olds spent 47% of their media-time on television. This number has now fallen to 29%. This age group now spends the majority of their time on streaming services like Netflix and YouTube.

One hypothesis is that the bad user experience of TV is behind this decline. Nobody likes to have their content interrupted with ads, especially if they are irrelevant to your interests and cannot be skipped. Television ad-breaks are long, often exceeding 5 minutes. On YouTube, however, viewers have the option to skip ads after a short wait of 5 seconds – a feature that linear television does not provide.

But even if TV channels wanted to change their user-experience, they do not have the tools to do so. Such changes require experimentation to figure out what works and what doesn't. YouTube and Netflix are known for running large-scale A/B tests, where hypotheses are tested against real user behaviour. Linear content delivery prevents TV channels from doing this. Everybody gets the same content at the same time. Television has no Internet-style programme planning and no Internet-style ad-targeting.

This is an alarming signal for broadcasters who want to maintain and connect to their younger viewers. A study based on the real-life case of a newspaper going fully digital showed that their print audience just evaporated. Television is going through a similar transition – young viewers who stop watching television now might stop for good.

We believe that TV as a source of content still has a place in the digital media landscape. We want to make it more appealing to younger generations who have different expectations for media consumption. The ReTV project aims to help broadcasters move their content online and offer personalised experiences to their viewers."

In the blog post, two demo videos of the Content sWitch are presented. Figure 2 shows a screenshot from one of the demo videos: The top left corner is the original trailer. In the top right, we tease the movie by showing the opening seconds. Later versions will display a button that allows the user to continue watching. In the bottom row, we show content for a completely different movie that we believe might appeal to a viewer with different interests. On the left, we see the trailer, and on the right, we start the actual movie again. This video is only for illustrative purposes, in practice each user would only see one of the four quadrants instead of all four at the same time.

[3] https://retv-project.eu/2019/05/21/content-switch-for-a-modern-content-experience/

Figure 2: Screenshot of the Content sWitch in action for trailers

As mentioned above, we did not focus on the problem of splicing trailers into a live video stream at larger scale because we know this problem to be already solved by specialised streaming companies (cf. e.g. MS Azure). Our prototype does splice the trailers into a video stream but the solution is not highly scalable. We did focus, however, on developing an interchange format, with which the GENISTAT Recommendation & Scheduling component can indicate to other modules what kind of trailer should be inserted and if it needs to be shortened or lengthened or have other adaptations applied to it. This format can be read by any specialised streaming service like those already used for DAI to splice the trailers in. We describe this format in more detail in deliverable D3.2.

4.2 4U2 CHATBOT

Interviews with TV viewers reported in D6.1 showed that a significant segment is interested in having content sent to them periodically via a chatbot. The chatbot uses the recommendation service previously defined for the 4u2 scenario as well as the personalised video summaries.

During the development of the scenario, it became clear that each content partner would need its own chatbot version. Firstly, users of the services of a public broadcaster (RBB) obviously expect different content than users of the channels of an archival institute (NISV). Secondly, this content must also be selected, prepared and presented in a different way.

We therefore decided from the start to offer two different 4u2 chatbot versions, each tailored to specific user as well as content owner requirements, but following the same workflow. In accordance with the TVP architecture, they both access the same modules provided by WP1-3 and thus allow us to support a multitude of users with minimal implementation effort. The difference between the two chatbot versions lies mainly in the user interface. In this section we describe the decisions that led to the creation of the user interfaces for RBB and for NISV.

Both chatbot versions are based on the wireframes and workflow that RBB built and tested with six consumers. The wireframes were built as a clickable prototype that was provided to

testers on a smartphone. For the wireframe tests, we used content from the RBB daily evening show 'zibb'.[4]

In the first version of the wireframes the user was able to communicate with the chatbot via free text and links in the text messages. The feedback from the think-aloud test indicated that communication by buttons would be more handy, simple and quicker for the defined scenario. Additional feedback indicated, however, that the amended chatbot should not lose the functionality of being able to understand written text input. In a second version of the wireframes the button interaction was realized and handed over to the technical partners.

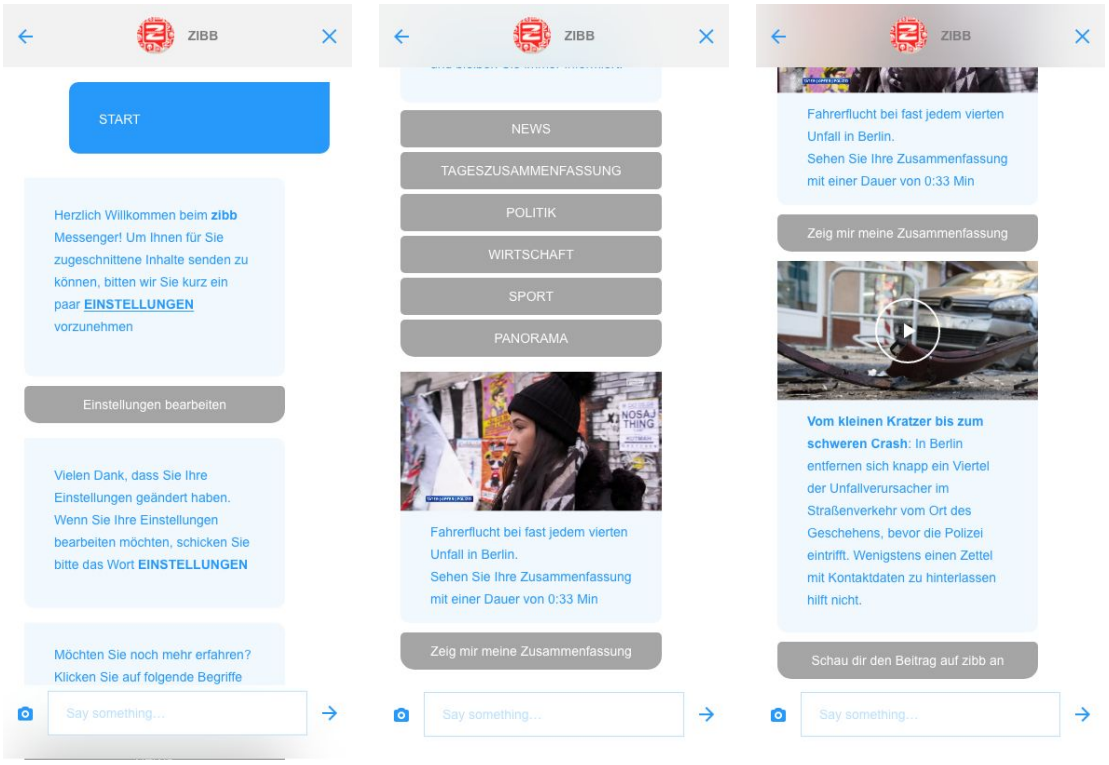


Figure 3. Steps of getting a video summary in the chatbot and interaction via buttons

Figure 3 displays one possible workflow of sending the user a video summarisation:

(1) The user sets up preferences in the settings
(2) The chatbot sends a GIF with recommended content at a predefined time
(3) The user is able to see the summary, the whole video or the article (if available)

83% of the users found the scenario logical, with one tester answering that he/she was neutral. Results clearly indicated that users did not want to leave the app - video summaries should be sent directly to them and not as links opening in another website. Also, 100% liked the intermediate step of first receive a preview GIF, and noted that this would motivate them to watch the video summary.

[4] https://www.rbb-online.de/zibb/

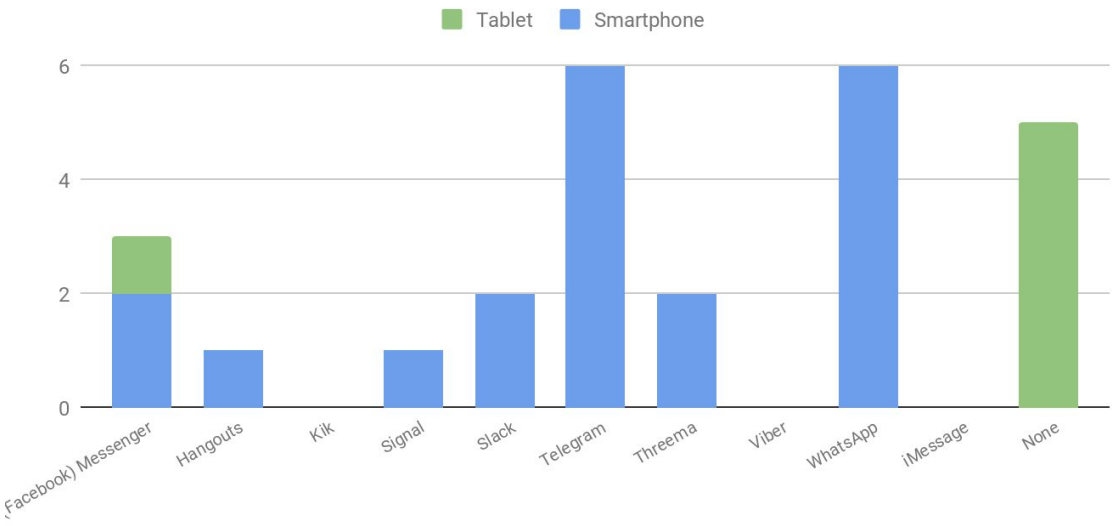


Figure 4. Messenger services by device

We also wanted to know which messenger service the testers use and on which mobile device. Telegram and WhatsApp were used on smartphones by all testers. Only one tester used the Facebook Messenger on a tablet. For ongoing development, we decided to focus first on Telegram, also because of various WhatsApp issues that emerged during testing. At this moment it is unclear how a chatbot could work in WhatsApp because, as of December 2019, the service will no longer allow sharing messages with many people at the same time.[5]

To gain a better understanding why a video summary would be useful for testers, we gave them five options to choose from: (1) ... to be informed about a certain topic; (2) ... to see only certain content of broadcast / on video (3) ... not having to read any articles.; (4) ... to get an overview of certain topics. (5) ... save time (e.g. two.minute-summary instead of a 30-minute-video), see figure 5. Interestingly, 66.7% of the testers still wanted to have a link to the original articles to gain further information. We therefore integrated the original articles into the final wireframes and the prototype. The key reason for watching video summaries via a chatbot was to save the users' time (66.7 % completely agreed, 33.3% generally agreed). A summary would provide a good overview of a topic, and users still have the possibility of watching the entire video or reading the original article.

The issue of filtering specific topics of the program in a video summary received a more mixed response. On the other hand, testers liked to filter content by a defined category. The defined categories could be selected in the profile preferences of the user. This is one of the first steps the user will do by subscribing to the chatbot. Depending on the chatbot version (NISV or RBB), the topics could be different. In RBB case, they are related to the programme "zibb" and includes topics like "Angesagt", "Gäste, "Service", "Tipp" and "Vorleser". For NISV, these categories could relate to the most popular topics in the collection, e.g. "Dutch History", "Fashion", "Animals", etc.

[5] https://faq.whatsapp.com/en/general/26000259/

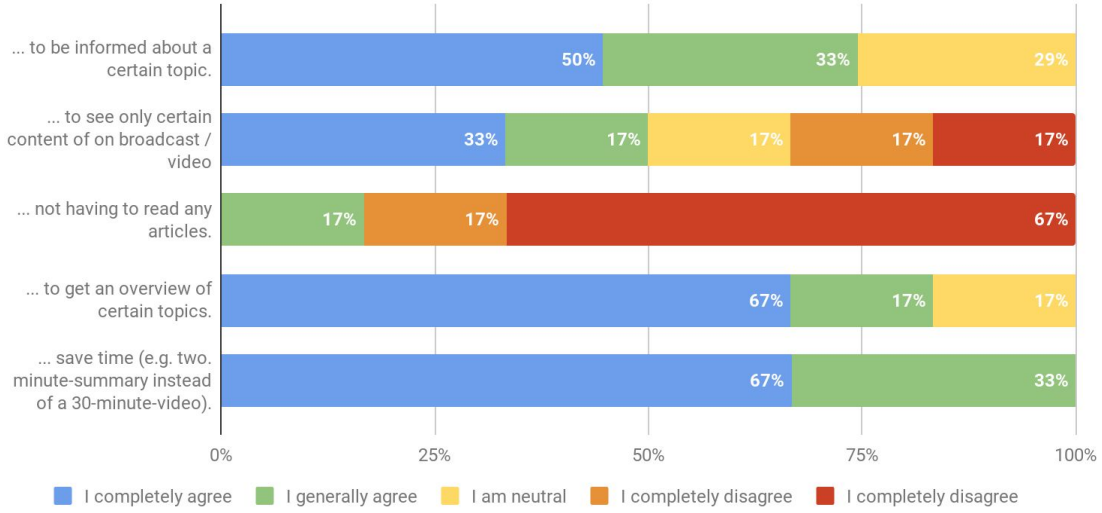


Figure 5. Outcome on why the user wants to receive video summaries via chatbot

To further develop the 4u2 scenario, NISV also tested the RBB chatbot wireframe with three professional users from their editorial team. The goal of these tests was to determine how a broadcaster archive could make use of the chatbot services and what could it offer to its audiences. The professionals at NISV seemed very interested in the idea of using a chatbot as another vector to communicate with the general public and promote the archival collection. The chatbot would present an opportunity to create longer-lasting engagement with one-time museum visitors at NISV and encourage them to explore the collection online, outside the physical boundaries of the museum.

During the tests, it became clear that an archival institute like NISV would not be able to use the exact same chatbot scenario as proposed in the RBB wireframe and would have to adapt it to their audiences and their content. The RBB chatbot would be able to offer videos about recent events, topical news, even content from yesterday's broadcast. Unlike a broadcaster, NISV can only republish content that is available under open licenses and in most cases, this is older archival material. In order to make this content more visible to the users, the chatbot could use a database of past events (e.g. historic occasions, holidays, birthdays) to find opportunities to republish this content as well as find content related to today's trending topics. Another feature specific to NISV collection is that often older videos are very short (around one minute long), therefore, one tester suggested that it would be useful to combine multiple videos which cover the same topic into one video summary.

The professional testers seconded the idea that it would be very useful to provide a link to the original video so that users could explore the collection further. Also, they suggested that summarised video should be provided in a format that makes it easy to share them on the chatbot application as well as other social media platforms. Furthermore, one tester also suggested that a user should be able to sign up to receive video summaries on a regular basis as proposed in the RBB wireframe but also to request them on the go by typing in or selecting a topic they are interested in.

Based on the results of these tests, the User Journey was created for implementing the first prototype of the chatbot scenario. RBB and NISV worked out a detailed workflow. First, six stories were identified:

- Welcome

- Settings
- Instructions
- Receiving personalised summary on the go
- Evaluation
- Receiving summaries on specified times of day (based on user preferences)

Then, for each category, actions were defined in detail, e.g. what happens if the user types in 'hello' or what options does the bot offer if the user wants to see a video summary right now. Together with GENISTAT, the project partner developing the chatbot, the feasibility of the proposed workflow and the current development status were regularly discussed. Due to the already mentioned necessity to create two chatbots, RBB and NISV each had to formulate their own responses and button labels in the language of their testers, i.e. in German for RBB testers and in English for NISV testers. This also led to minor deviations in the respective overall workflow. However, the changes always took place under the premise that they did not have a significant impact on the User Journey in general and thus ensured a joint evaluation of the chatbot scenario.

RBB, NISV and GENISTAT agreed on an agile[6] approach for the future development of the chatbot. The partners now expect the process, which is now set to three-week sprints, to provide an even more effective implementation of the features required in the future.

5. Technical Development

Following the architecture laid out in D4.1, all of the modules of the TVP providing data, content and analysis capabilities would be available through APIs, leaving the technical development for T6.2 mostly focused on the user-interfaces and user-experiences. The work on the user interfaces was mostly focused on the chatbot, as the Content sWitch does not need a new user interface but would rather integrate into the streaming system of an existing provider like Zattoo.

5.1 Content sWitch

For the Content sWitch, we developed a small demo frontend in task T6.2 which can be seen in figure 6. The demo frontend displays the program guide and lets a user watch any program aired within the last day. If a trailer is coming up, the frontend will send a request to the recommendation API and ask for a trailer that is suited for a particular user. In the demo we ask for personalised trailers for three users in parallel. The recommendation API returns the trailer that is most likely to fit the user's taste. This trailer is then played instead of the originally scheduled one. The extensive work concerning the recommendation engine is described in D3.2.

[6] https://www.agilealliance.org/agile101/

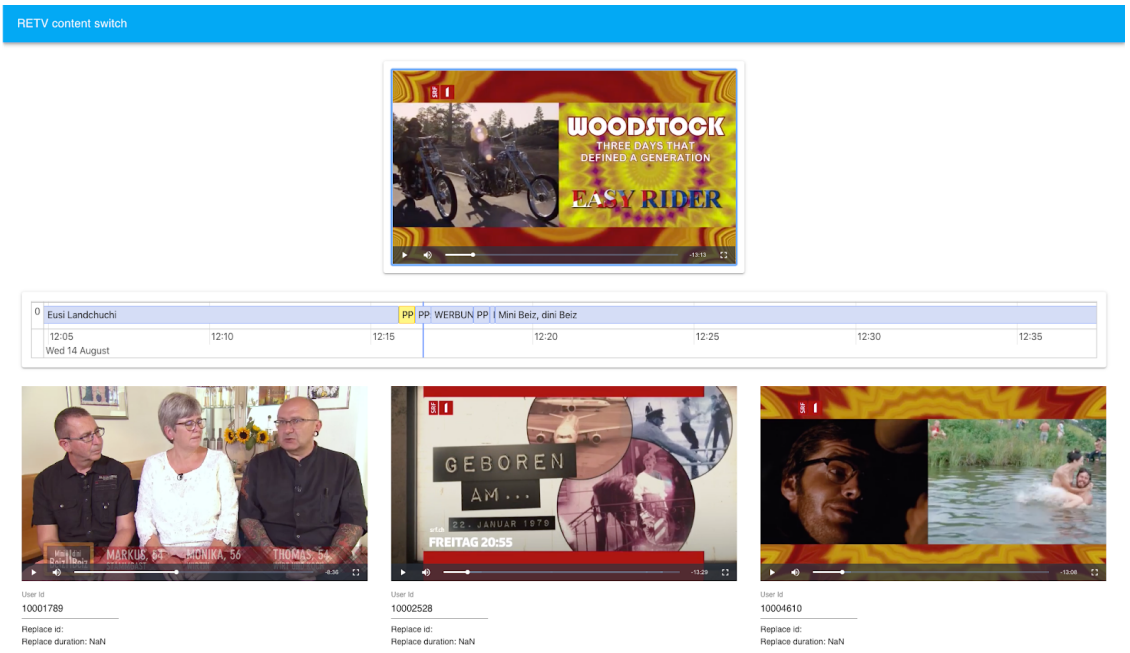


Figure 6. Demo frontend that replaces trailers in a personalised manner

The demo frontend is an interactive version of the videos mentioned in section 4.1. It is set up with the user id of four Zattoo users. When a trailer comes up as indicated by the program timeline in the middle of the screen, the demo frontend sends a request to the recommendation API for each of the users. The recommendation API builds an anonymised taste profile of the user, and selects a relevant trailer for this user. If the current trailer is fine, nothing happens and the user sees the same trailer as originally intended by the broadcaster. However, if the recommendation API finds a trailer that better matches the taste profile it will return this trailer to the demo frontend which will splice it into the video (for instance, if user profile suggest high interest in football, highly relevant trailer may be announcement of the match, or the biography documentary on the famous football player). As mentioned before, this splicing is not very scalable, as it happens in the browser. It is, however, enough to showcase how Dynamic Content Insertion works in practice.

5.2 4U2 CHATBOT

A messenger app provides the frontend for our chatbot (e.g. Telegram or Facebook Messenger). The chatbot middleware is the module that parses the inputs from the user and responds appropriately. In order to abstract from the intricacies of the different messenger platforms and to profit from community efforts, we decided to build our middleware using the Rasa[7] framework. We chose it as it allows us to run it on our own servers, protecting the privacy of the users. The main advantage of using a chatbot framework is that it abstracts from the intricacies of the different messenger platforms. We chose to use Telegram as the first messenger platform as it does not force users to log in with a Facebook account (as opposed to Facebook messenger) and it offers fully fledged chatbot support (as opposed to WhatsApp).

[7] Available at https://rasa.com/. Accessed August 2019.

In Rasa, we define so-called stories, which are dialogues the user navigates through. The registration process (see Figure 7) is the first of such stories, but there are many more. Other examples include a story to change user preferences or to give feedback to a video.

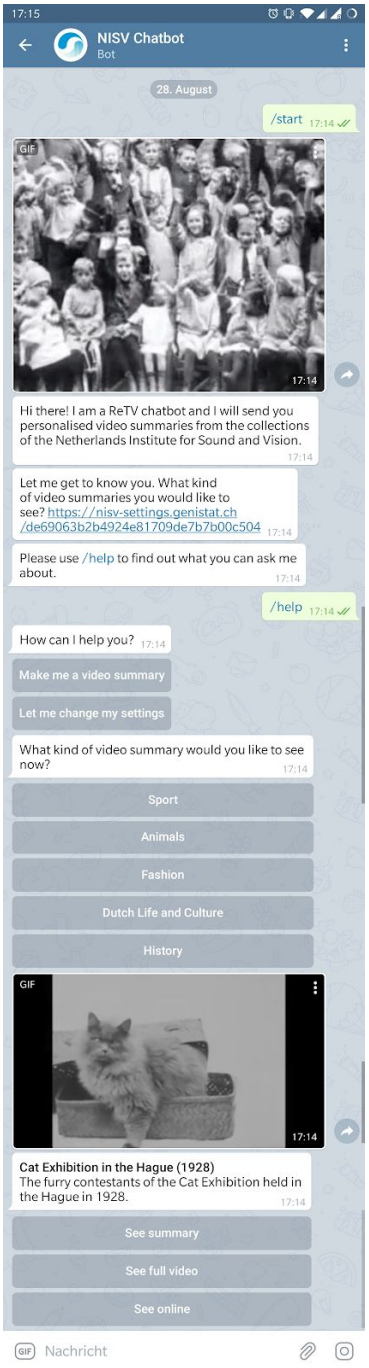


Figure 7: Registration screen of the chatbot

We solved the cold-start problem (i.e. the lack of relevant information on a first-time user preferences) by asking the user to choose from a set of predefined content categories during the registration story. This also allows us to respect the users privacy, as we need to store any personal information or detailed viewing history, just anonymized identifiers assigned to a set of preferred topics. We also ask users how often they would like to receive content. The selected preferences, as well as the desired schedule, are stored in a database. A second

component of the middleware regularly checks this database to see if a user's desired time of delivery is close. At the time of desired delivery, the middleware loads the user preferences and calls the content adaptation service, asking for a personalised video. Once the content adaptation service returns this video, the middleware sends it to the user as a video message.

From a technical perspective, what we call middleware actually consists of five different applications, interacting with each other, handling tasks as various as interpreting user input, to storing and serving summaries and videos according to user preferences. We rely on Rasa to handle natural language understanding (NLU) of the user messages, and its mapping to relevant actions.

The figure 8 below depicts a rough architecture model of the chatbots applications. By design, we split each component to be fully independent, in order to maximize reusability across contexts (for RBB and NISV).

All logic and language processing is handled by Rasa, we just need to make sure such logic is exposed to the chat platform (Telegram in our case). We achieve that through the endpoint exposure, and the configuration of a webhook (REST API over HTTP) accessible to the chat platform.

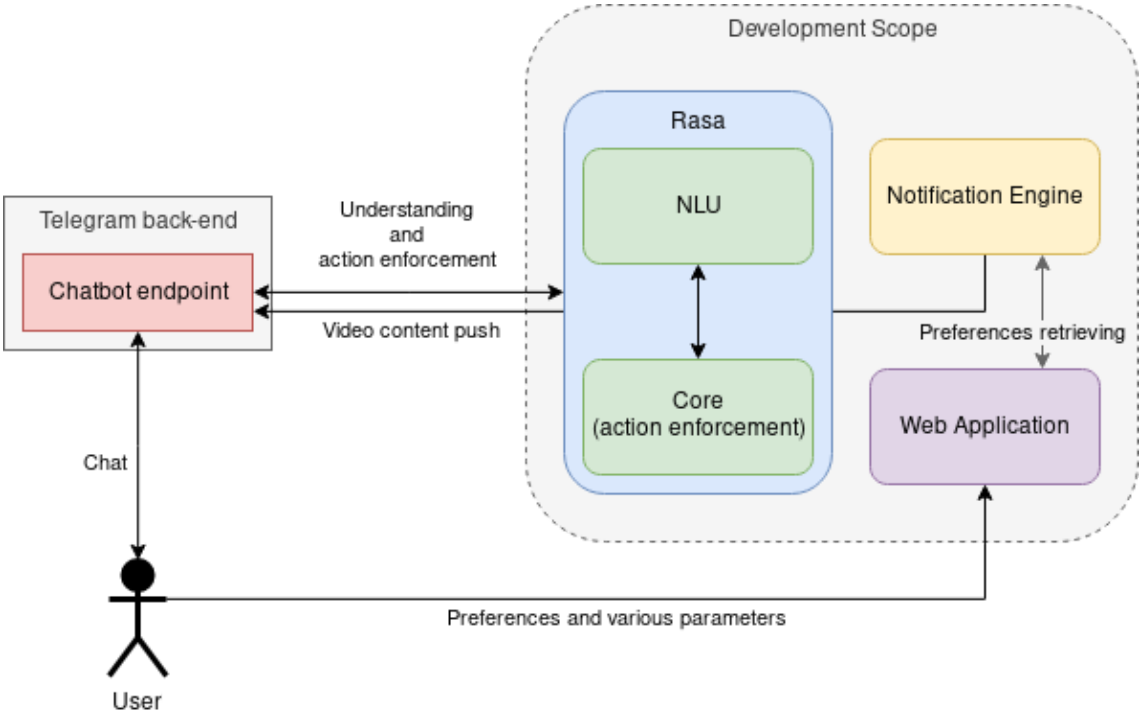


Figure 8: Architecture model of the chatbots applications

Since we are developing a new paradigm for chatbots, we had to overcome limitations inherent in the current messenger and chatbot platforms. The main challenge we faced during the technical implementation was to have two different components pushing messages to the Telegram back-end as both Rasa and the notification engine talk to the user (one through messages, the other through videos). This led to confusion on the NLU part, since the engine wouldn't be able to account for messages it did not instigate in the first place. We solved that problem by removing any correlation between the stored user input in the Core engine of Rasa and its choice of decisions. This means a simpler state machine, that relies on a standardized

API calls to recover the exact answer it should give to the user, which is itself context-aware. In other words, the chatbot was designed so that extended stories (back and forth conversations between the chatbot and the user) are not needed.

For example, this means the chatbot can understand user's reaction to a video that was pushed by the notification engine, which is something that is neither supported out of the box by Telegram, nor Rasa.

6. Content Preparation and Customisation

Both the 4u2 chatbot and the Content sWitch are inherently content-driven: the best possible usability and personalization is worthless if no relevant content is delivered. We therefore decided on a set of relevant content sources for both content partners RBB and NISV. A special case of content preparation was the testing of the video summarisation module, as we wanted to make sure that the results would be of acceptable quality for users.

6.1 Partner Content

6.1.1 RBB

For the tests of the Content sWitch scenario, RBB provided eight trailers from its media archive. All videos were previously broadcasted by RBB. These trailers were selected for two reasons. First, each represents a specific genre. Consequently, they can be clearly distinguished from each other in terms of content, which was essential for the two goals of the test: (1) to evaluate and improve the recommendation algorithm and (2) to verify whether the preferences explicitly declared by users are consistent with their implicit preferences. Secondly, the videos advertise shows and series that are familiar to RBB viewers, making the testing experience less artificial and more authentic for participants.

For the 4u2 chatbot tests, RBB also provided content that had already been broadcasted via its TV services. This content consisted of five videos each from the popular categories 'Angesagt', 'Gäste', 'Service', 'Tipp' and 'Vorleser' from the RBB show 'zibb'. This pool of 25 videos made it possible to offer the testers a varied yet familiar setting. In addition, this content guaranteed that all current features of the chatbot could be tested extensively.

6.1.2 NISV

For the chatbot tests, NISV provided archival content from the Open Images collection[8]. The majority of videos in this collection are published under an open license, namely Creative Commons or Public Domain Mark. This means that NISV can easily repurpose, remix and publish this content online. Videos from this collections are already regularly shared on the NISV social media channels and would be ideal for the chatbot scenario. Same as for the RBB chatbot, five video topic categories were defined for the chatbot, based on the most popular topics in the collection. These were: Animals, Fashion, Dutch Life and Culture, History and Sport. Five videos were manually selected for each category.

The Content sWitch scenario is not applicable to NISV therefore no content was provided there.

[8] https://openbeelden.nl/.en

7. Evaluation Results

7.1 Content sWitch

7.1.1 Test questionnaire setup

Content sWitch trailers were prepared by RBB and tested with 19 participants that were selected among active RBB watchers. For the evaluation, RBB prepared eight trailers of their programs. The programs were selected using the following criteria:

- A popular program gathering medium to big audiences
- The trailers should cover a varied set of topics (cf. below)
- Programs that are currently being broadcast and thus are known to RBB audiences

The topics and trailers that were included in the user test were the following:

1. Unterhaltung (entertainment) - trailer for the TV quiz show “Gefragt - Gejagt”
2. Kultur (cultural programs) - trailer for “RBB Kultur”
3. Information (news) - trailer for “Abendshow”
4. Heimat (local content) - trailer for “Heimatjournal”
5. Politik (politics) - trailer for “Talk aus Berlin”
6. Gesundheit (health and wellbeing) - trailer for “RBB Praxis”
7. Reisen (travel) - trailer for “Flussauf, flussab”
8. Krimi (crime series) - trailer for “Polizeiruf 110” series

To run the questionnaire, we used the Lamapoll platform (https://www.lamapoll.de). The questionnaire consisted of four parts.

In the first part, users were asked to declare their interests, by selecting one or more of the above-listed topics.

Next, users were asked to watch the eight trailers (duration between 30 and 90 seconds). Each trailer corresponded to one of the above-listed topics. Then, we asked users to give a star rating to each of the trailers. Ratings ranged from 1 (don’t like it) to 10 (like it very much). Figure 9 presents how it looked in the questionnaire.

Figure 9: Lamapoll questionnaire:
Please rate the trailers on a scale from 1 (= very bad) to 10 (= very good)

As the final question, we presented users with an interactive list of trailers, and asked them to rearrange the list, in the descending order of the perceived relevance, i.e the most relevant trailer at the top of the list, and the least relevant trailer at the bottom of the list (figure 10 shows how it looked like in the questionnaire).

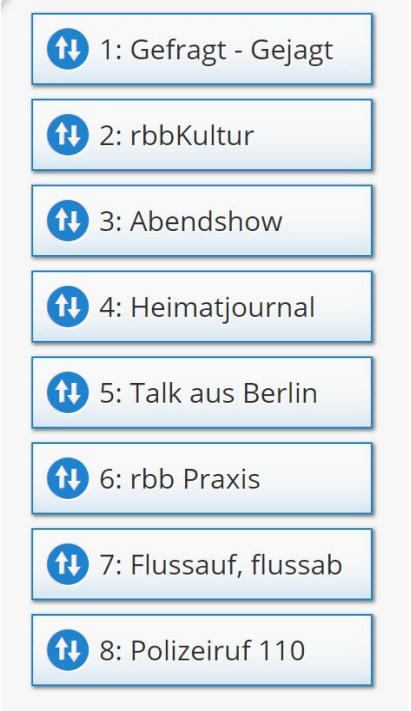


Figure 10: Lamapoll questionnaire: Please arrange the trailers as follows:
at the top = your preferred variant, at the very bottom = the least preferred variant.

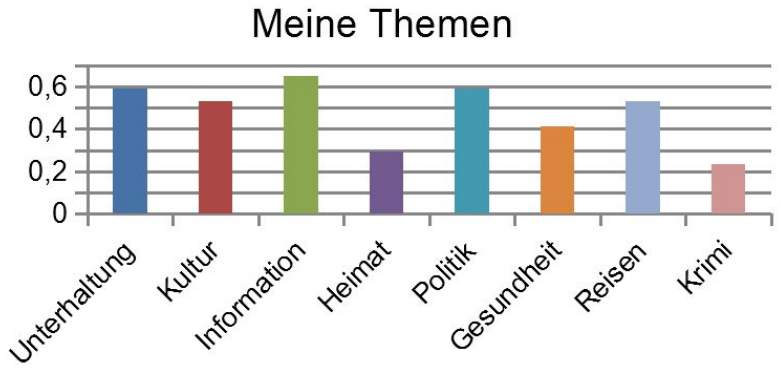
7.1.2 Results and discussion

Out of 19 participants, 13 participants have completed the questionnaire (i.e. selected preferred topics, watched all the trailers and answered all the questions).

7.1.2.1 User preferences

Figure 11 presents user preferences distribution. Each topic was selected by at least a few users. However, some topics were preferred: entertainment, information and politics, then travel and culture. On the other end of the popularity ranking, local programs and crime series were rarely selected.

That is particularly interesting for us, highlighting that our initial category subset would cover user interests, regardless of their background, or more specific liking. In essence, the categories are diverse and broad enough to have everyone interested in at least one.



Topic	Count	Frequency by answers
Unterhaltung	10	15.38%
Kultur	9	13.85%
Information	11	16.92%
Heimat	5	7.69%
Politik	10	15.38%
Gesundheit	7	10.77%
Reisen	9	13.85%
Krimi	4	6.15%

Figure 11: Lamapoll results: Users preferred topics distribution

7.1.2.2 User ratings

Figure 12 presents the distribution of the user ratings. Each row corresponds to one trailer. Each rectangle corresponds to given vote (rating 1 on the left, rating 10 on the right). The width of the rectangle represents the percentage of a given vote. For instance, “Gefragt - Gejagt” trailer (row 1) got rating 1 (the lowest rating) in 23% of the cases and rating 10 (the highest rating) in 8% of the cases.

User declared preferences were partially reflected in the ratings given to the trailers. In particular, RBB’s Kultur trailer got the highest average (8.0) and median (8) rating.

However, there were also significant differences between the average rating and the corresponding topic popularity. Especially for travel category (which was among the popular ones in the declared users interests), the corresponding trailer for “Flussauf, flussab” program got only 3.85 average rating (median rating = 3). It suggests that we should have asked more detailed questions in the second iteration of the evaluation poll, by separating the evaluation of the program and topic on one side, and the evaluation of the particular trailer. One idea could be also to prepare a set of personalized trailers per each of the topics.

Another example of a topic that was in general relevant to the users but where the trailer got low ratings was “health” (and the corresponding RBB Praxis program). Here again, similarly to “travel” topic, users might have evaluated the topic of a given RBB Praxis episode. If it was not relevant to them, they gave it a lower rating than their interest in the “health” topic might have indicated. Having multiple, personalized trailers per topic would help us to evaluate this.

Our results show that interest in a topic does not directly translate into interest for a trailer or a show. Therefore the Content sWitch will need to look deeper than just the broad categories.

It is worth to note that average ratings are not giving us the full insight about user preferences distribution. Some of the topics have more uniform distribution of ratings (again, travel and health, but also local programs [Heimat topic]), while some other topics obtained significantly higher fraction of the extreme ratings (1 - very low or 10 - very high rating). High number of high ratings were given to: RBB Kultur (86% of the ratings above 8), and crime series (45% of the ratings above 8). However, for the crime series, high ratings were balanced with a high fraction of very low ratings (31% of the ratings below 3), which resulted in the medium average rating (6.4). It shows that some of the ratings have more polarized audience.

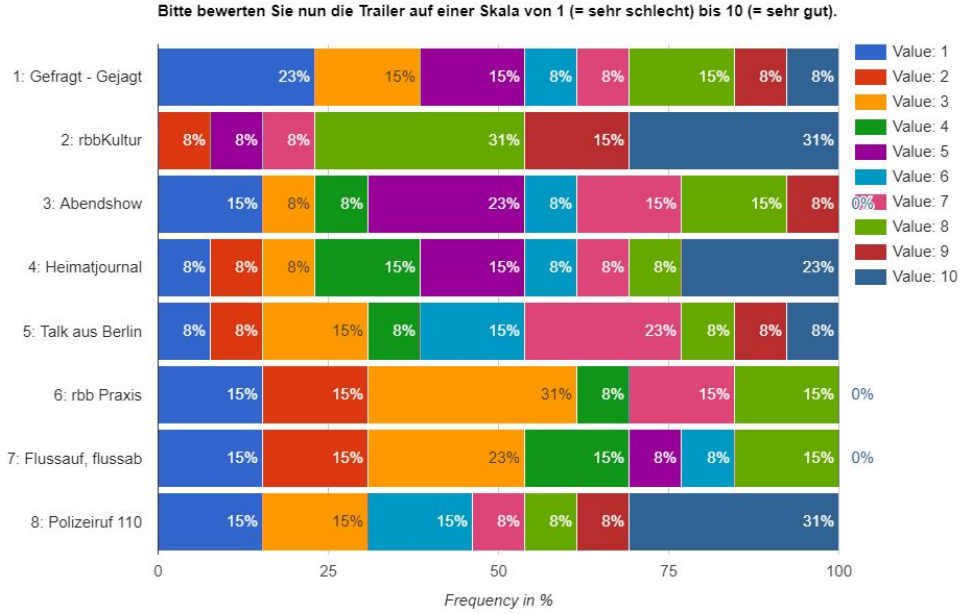


Figure 12: Lamapoll results: Users ratings for each trailer

7.1.2.3 Users preferences rankings

Figure 13 presents the distribution of the user rankings. Each row corresponds to a given rank position (first row: top rank, last row: lowest rank). Each rectangle corresponds to given trailer. The width of the rectangle represents the percentage of cases when a given trailer got a particular rank. For instance, in row 1 (top rank), “Gefragt - Gejagt” trailer got this position in 14% of the cases. “RBB Kultur” was the trailer that got top position in 43% of the cases.

Figure 14 presents comparison of ratings versus rankings. Our goal with this question was to verify the consistency of the user preferences. Ideally, if for each user we sort trailers in a descending order by star ratings, we should obtain the same ranking as the rankings directly created by users in the last question (please note that the order of ratings and rankings is different, i.e. the higher rating is, the lower is the rank). It appears to be the case, with the slight shifts between “Abendshow” and “Talk aus Berlin” at the fourth and fifth position respectively. However, both of these shows are focused on politics, and they may gather attention of similar audiences.

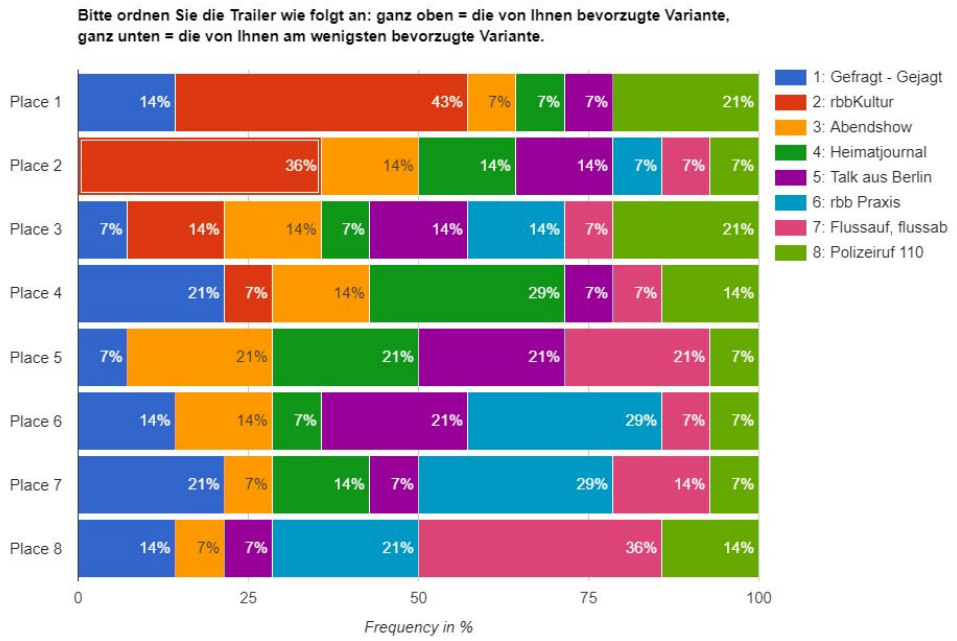


Figure 13: Lamapoll results: Users rankings distribution for each trailer

Trailer	Ratings (average)	Ratings (median)	Rankings (average)	Rank
1: Gefragt - Gejagt	5.15	5	5.07	6
2: rbbKultur	8.00	8	1.86	1
3: Abendshow	5.31	5	4.36	4
4: Heimatjournal	5.77	5	4.21	3
5: Talk aus Berlin	5.62	6	4.50	5
6: rbb Praxis	4.00	3	6.00	7
7: Flussauf, flussab	3.85	3	6.00	7
8: Polizeiruf 110	6.46	7	4.00	2

Table 14: Lamapoll results: Comparison of ratings (question no.1) and rankings (question no.2)

7.1.3 Conclusions

From the first iteration of the questionnaire, we found out that we need to (a) dig deeper into more detailed user interests and the topics relevant to them and (b) personalize trailers for each topic in order to separate user (dis)satisfaction with a particular trailer from the (dis)satisfaction with a topic itself.

Following these initial findings, in the next iteration of user preferences tests, we would like to investigate more detailed questions:

- Can we improve the user experience by having more personalized trailers per each topic? In particular, by showing trailers selected by a recommendation model.
- Is the user satisfied/dissatisfied with content? If so, is it about the program, or rather about the particular trailer?

- Is the user satisfied/dissatisfied by user interface and the presentation layer? This should be separated from the personalized recommendation of the content.

Last but not least, the next iteration of users tests should include more users with more varied profiles. This would allow to achieve statistical significance and analyse results not only qualitatively, but also quantitatively.

7.2 4U2 CHATBOT

10 participants at RBB and NISV tested the chatbot, using their own smartphones. The users were firstly instructed to install the App Telegram[9] on their devices and then to subscribe to either the RBB chatbot (https://t.me/rbb_chatbot) or the NISV chatbot (https://t.me/nisv_chatbot). Users were given two tasks: (i) to set up a profile and (ii) to request a video summary. No further instructions or help was given. Users were then asked to complete a questionnaire on the topics of video summaries, recommendation systems and the general user experience. Figure 15 below shows how a user can interact with the chatbot to setup a subscription, or create personalized video summaries for a number of predefined content category.

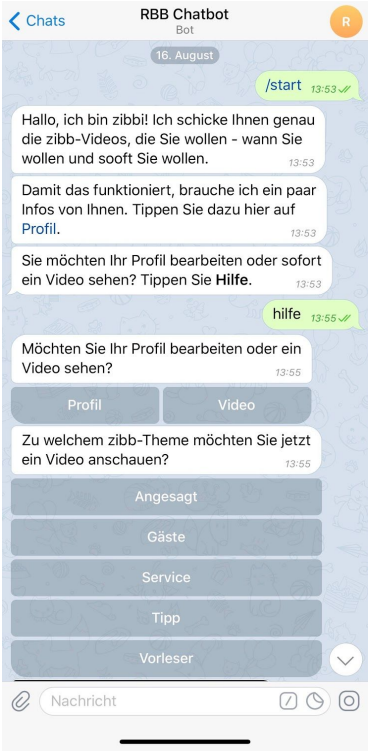


Figure 15: RBB chatbot subscription

7.2.1 Settings

After subscribing to the chatbot, users are welcomed with a request to set up a profile. A link opens the in-app-browser with profile settings (see Figure 17). Users are able to set up various preferences, such as interest in specific topics, the frequency of video summaries and their duration.

[9] https://telegram.org/

A multiple choice question asked users how they wanted to define their interests in the settings section. 53.85% preferred to select from a predefined list, 38.46% wanted to manually add their interests. One participant also mentioned that it would be helpful if the user was able to add interests immediately via the chat function of the chatbot (see Figure 16).

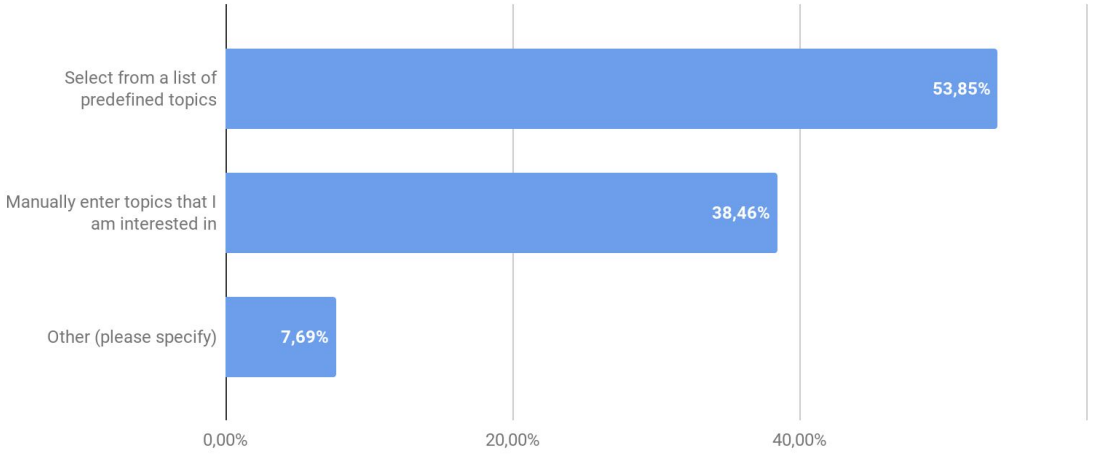


Figure 16: How would you like to define your interests in the settings section?

After the user defined their preferences, they were able to click on the “SAVE / SPEICHERN” button. Feedback indicated that some users were disappointed that the chatbot did not behave the way they expected it to after they saved their preferences. They expected the dialogue to automatically close after clicking the ‘save’ button.

At the time of testing, settings were not connected to video summary and recommendation services.

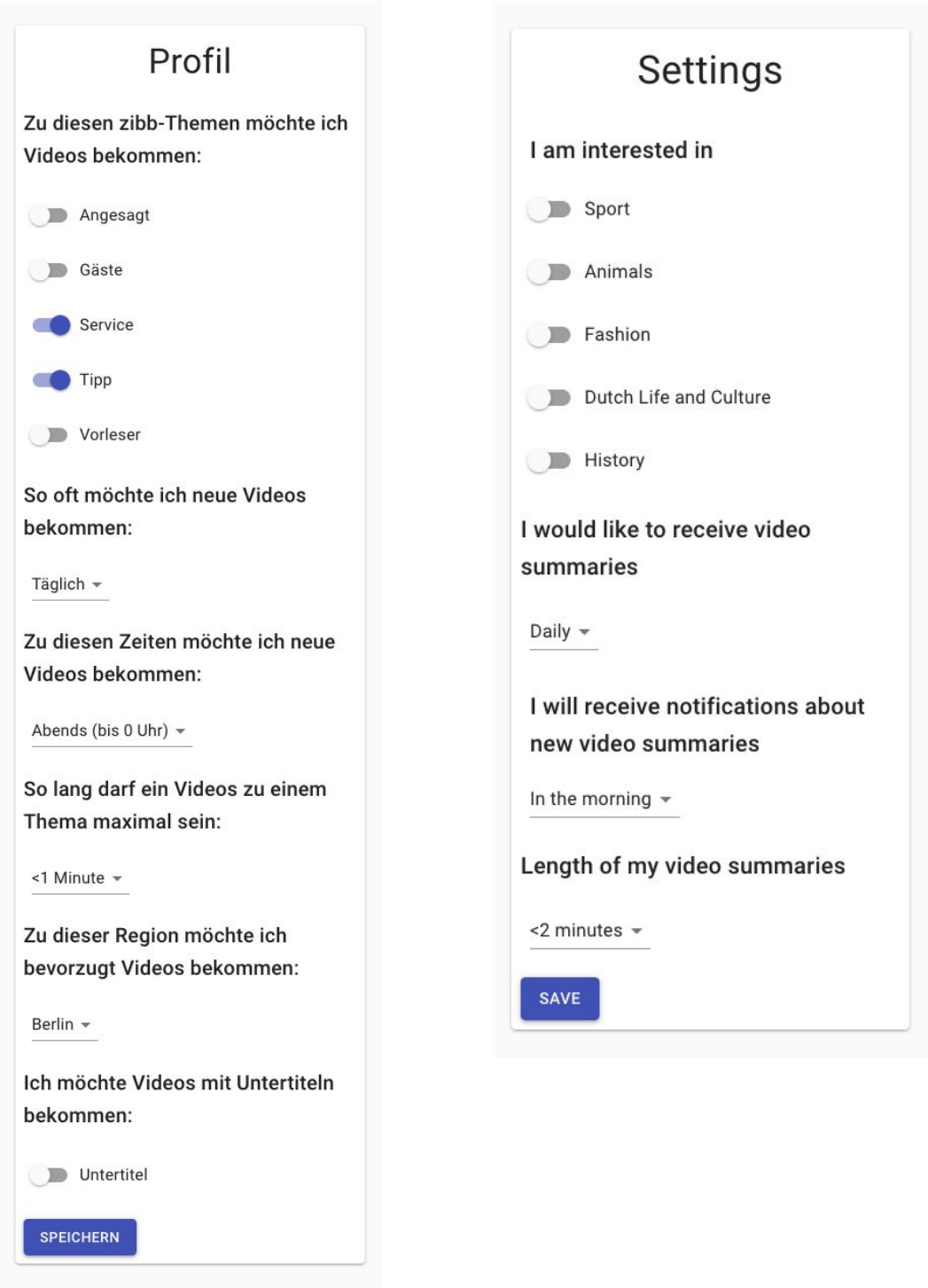


Figure 17: NISV chatbot (left) and RBB chatbot (right) user settings

7.2.2 Video Summarisation

Following profile creation, users were asked to request a video summary. To achieve this, they had to type “help/hilfe” into the chat. Users were able to select a specific topic by clicking on the topics button. A video summary preview GIF was then sent.

The next section of the questionnaire dealt with two types of video summaries (see Figure 18). A GIF was intended to help as a preview and as a possible decision-making aid. Participants were asked how such a GIF would encourage them to behave. 70% said that the GIF motivated them to watch the video summary and the same proportion said that they wanted to watch the full video. However, users indicated that the provision of a link with further information was unnecessary. This contradicts statements made during wireframe tests. Since the results of the current survey are inconclusive, this feature will continue to be part of the chatbot.

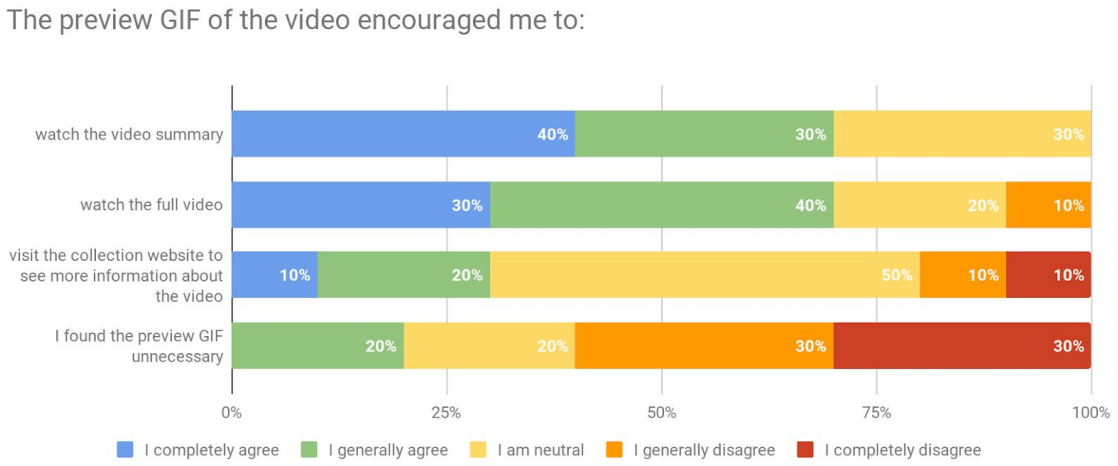


Figure 18: Preview GIF

Users were positive about having a preview of the video as a GIF. Feedback indicated that the GIFs were too fast, and users requested a stop functionality.

Directly after the GIF was sent to the user, users were requested to select a course of action by clicking on one of the following buttons (see Figure 19):

1. Watch video summary
2. Watch full video
3. Read article

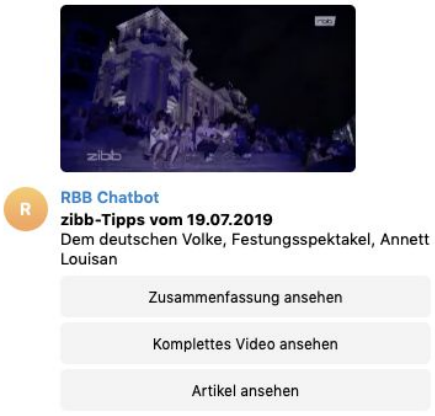


Figure 19: chatbot choices: video summary, full video or read article

By clicking on the video summary button, a 20-30 second video summary was played, with the actual length depending on the content. The overview presented by the summarised video was clear for 60% of users, and prompted 90% to watch the full video item (see Figure 20). Some users mentioned that they would have preferred to have audio included in the summarised video. 80% felt that the length of the video summary was suitable.

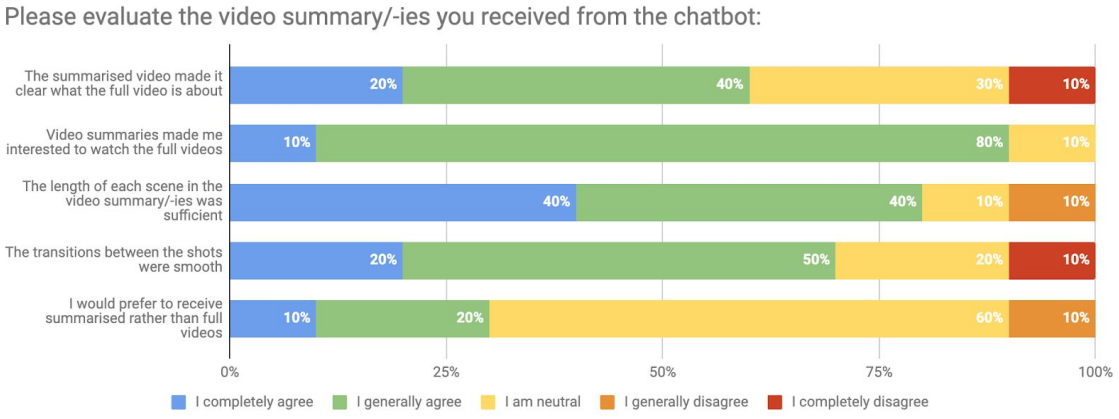


Figure 20: Video summaries

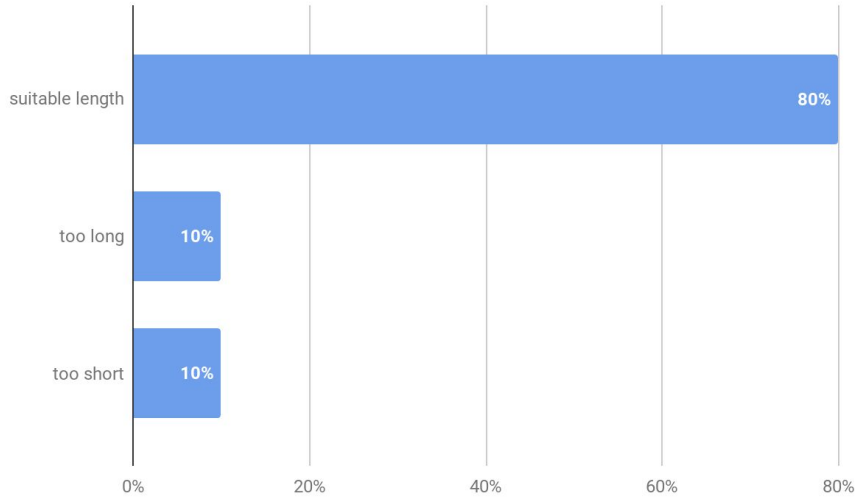


Figure 21: Video summary length

In user tests, we also wanted to discover what type of video summaries users wanted. 70% of respondents wanted personalised video summaries consisting of a single video (60%). At the beginning of the test, the video summarization service was temporarily not working, and two participants had problems with getting a video summary. This was why two of the participants answered with “Other” and described their difficulties.

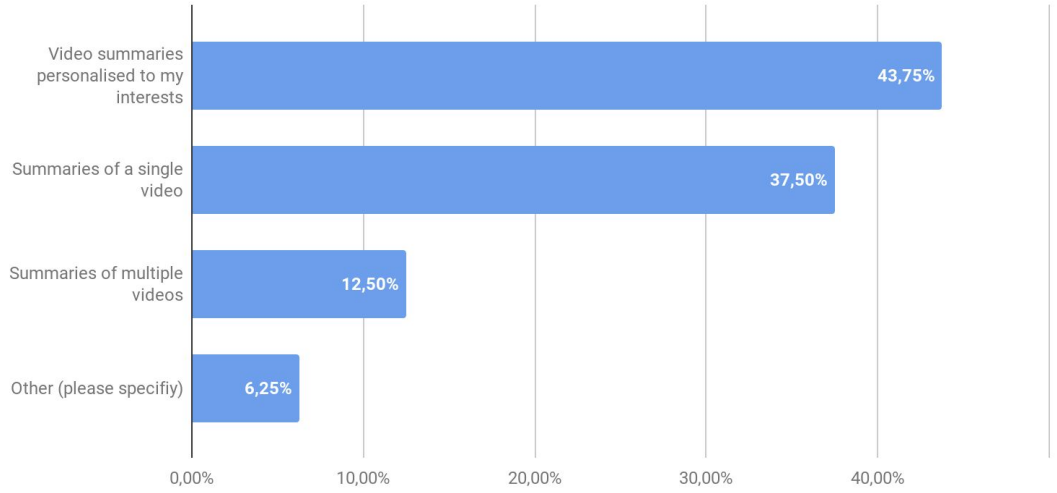


Figure 22: What kind of video summaries would you be interested in seeing?

The next question dealt with the push frequency of new video summaries. 90% answered that they want to be able to ask the chatbot to generate a summary by request. This means not having to wait for a specific date or time to have access to the content they are looking for. 60% wanted to receive new video content at regular intervals, as defined by the user in their profile. 20% of the participants selected typical push notifications when the chatbot found content of relevance to the user. This represents the fact that users want to interact via buttons whilst also having the possibility of communicating directly with the chatbot via text.

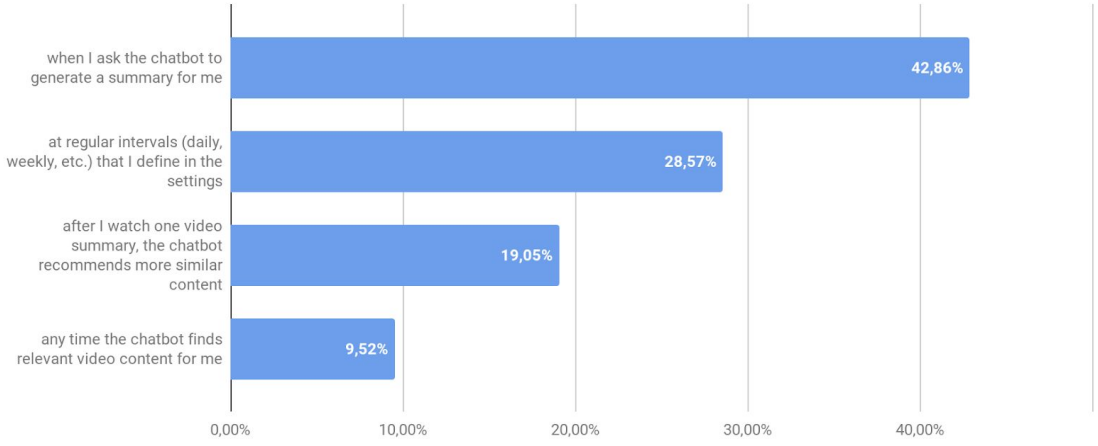


Figure 23: When would you like to receive new video summaries?

We also wanted to know what type of content the participants would like to see in videos. In this multiple choice question, 100% of users answered that content should be related to currently trending news stories. 90% were also interested in content related to their location. User location is defined in the user profile, enabling a fixed location reference rather than dynamically using the device's tracking services. 40% were also interested in content about historic events - such content could easily be automatically generated with archival material.

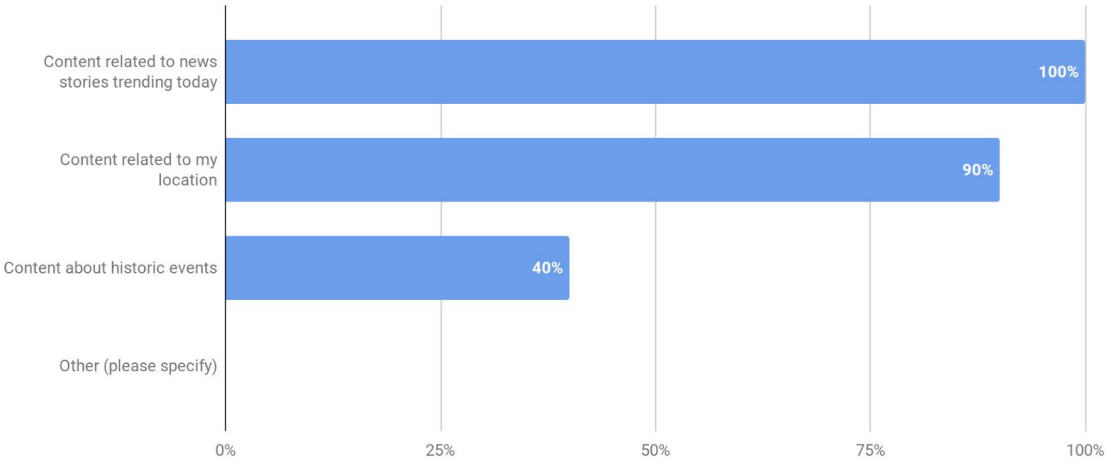


Figure 24: Please select what type of content you would like to see in the video summaries

7.2.3 Recommendations

The chatbot will also be able to send recommendations to the user. In the questionnaire we wanted to discover what kind of recommendations a user might need. In the multiple choice question "How would you like the chatbot to recommend content for you?", 40% of participants answered that the recommendations should be based on the preferences in the settings section. 35% of the user want to give feedback on summaries to improve the recommendation service.

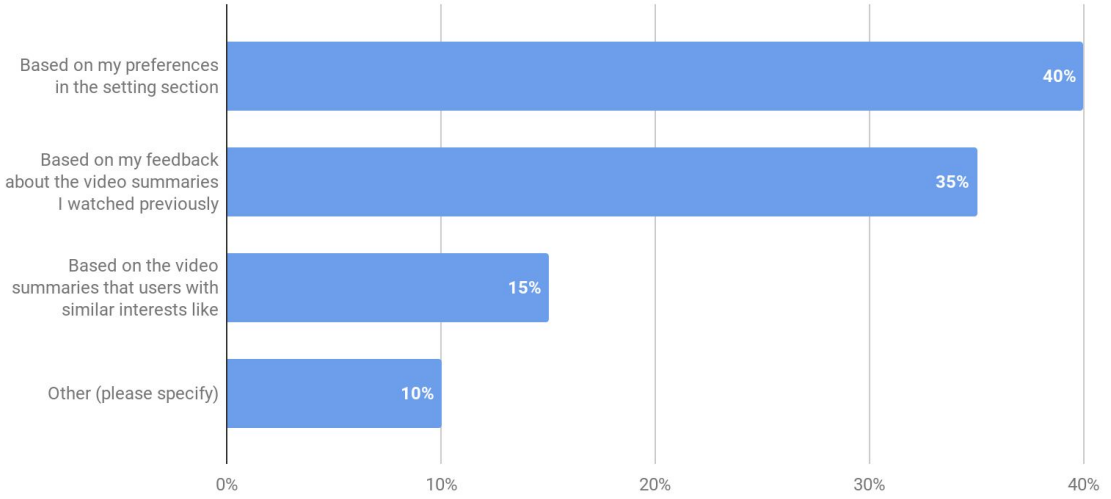


Figure 25: How would you like the chatbot to recommend content for you?

30% of the participants said that they completely agree with the statement "I would like to receive recommendations for more related content after I watch a video summary", and 70% indicated their general agreement with this statement. Participants were also willing to rate a video summary after viewing it, in order to receive better recommendations in the future (see Figure 26).

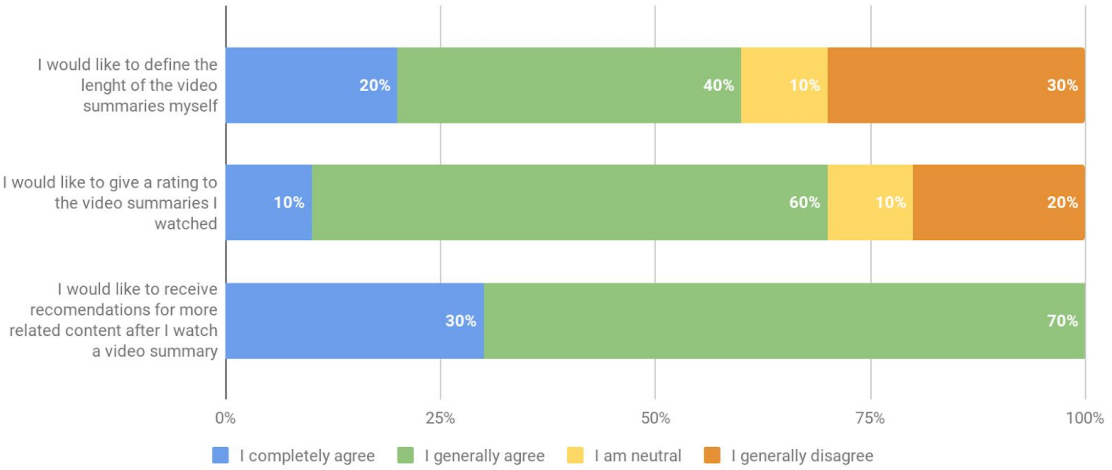


Figure 26: Rating and recommendations

7.2.4 User Experience

The user experience is one of the most important aspects of a chatbot. The current version is built on top of the Telegram app. This means that we are limited in interaction between chatbot and user. The user can either interact via predefined buttons or type free text into the chat section.

When creating wireframes, we received feedback that users wanted to have buttons. We integrated this into the workflow and tested it with real users who were able to use the buttons to interact with the chatbot on their devices. 50% wanted to have both possibilities of communicating, via buttons provided by the chatbot and also by manually typing text. 40% of users were satisfied by having only the button option (see Figure 27).

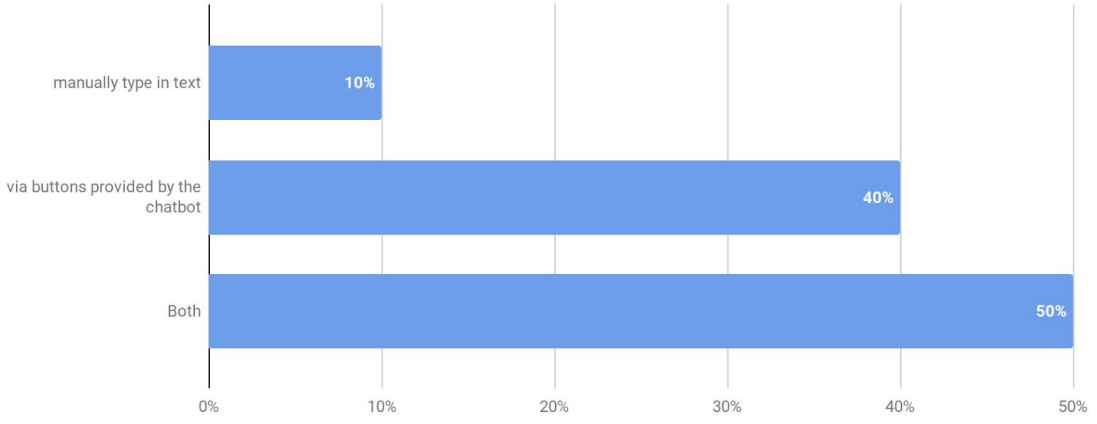


Figure 27: How would you prefer to interact with the chatbot?

The users also provided some additional suggestions for future improvements of the chatbot:

- integrating a link between settings and chatbot to send video summaries to the user at an interval defined in the users profile

- chatbot should be able to understand better the natural language
- adding audio to the video summaries that the user receives
- video summaries should be send as an mp4, not as a GIF
- GIFs should be played at moderate playback speed

8. Conclusion and Outlook

Creating quality content alone is not a success factor in the new paradigm for TV content consumption. One indeed has to trigger engagement from the users, ensuring low drop-out rate in the audience, high watch frequency and reach to a larger part of the audience. People spend most of their time nowadays between a chat window and a social media feed. It is only natural that we try to strengthen market penetration on these new vectors. Last but not least, new video consumption platforms emphasize the personalization of their suggested content, based both on user preferences, and wider understanding of the market trends (in large measure thanks to their massive user-base, and sharp expertise in its consumption habits).

This deliverable presented our first attempts to address this migration of TV viewers online and their changing media consumption habits through two user scenarios: 4u2 chatbot and Content sWitch.

The evaluation for the Content sWitch showed that broad categories are not enough to predict user interest in particular trailers and shows. The focus of our future work lies on better recommendation algorithms exploiting richer metadata (i.e. more detailed predictive features on the content and events, cf. D2.2 and D3.2) in order to deliver personalised TV content beyond broad categories.

The evaluation of the 4u2 chatbot confirmed that audiences are keen to watch broadcaster content that is tailored for their media consumption habits online. Video summarisation and personalisation services deliver content that motives users to watch videos at regular intervals and helps them find TV content that is tailored to their interests. User feedback highlighted were the chatbot workflow and video content needs to be improved (e.g. allow users to interact with the chatbot via free text, provide audio for video summaries) and our future efforts will focus on enhancing user experience and implementing scenarios with more varied content that can help broadcasters and media archives reach diverse audiences.

In detail, we would like to:

- Allow for adaptations that are semantically richer. An example is to have faster cuts for younger viewers. The video summarisation API already allows setting a so-called 'snappiness' parameter. Future versions of our content adaptation service will learn what the optimal value for the user is. When shortening trailers to fit into a particular slot, future versions will use a semantic and structural analysis of the video content to ensure no essential message is lost.
- Provide richer metadata on TV content. Examples include the detection of politicians and other persons of interest but also putting TV content in context with relevant events like the Football World Cup. This metadata is not only useful for recommendations as additional features but also content discovery in general (e.g. improved search or the possibility to subscribe to topics).
- Improve the frontends of both the 4u2 chatbot and the Content sWitch to be more user-friendly.
- Improve the model retraining by collecting more detailed user feedback. For the Content sWitch for example, we would like to track if a viewer watched the show for which we inserted a trailer.

- Connect prediction and analytic results from the Topics Compass into the two scenarios.
- Monitor WhatsApp development, which is expected to prevent the functionality of a chatbot (December 2019)
- Setup a focus group for longitudinal tests of the consumer use cases.

In general, we would like to expose our products to a wider and more diverse user-base, to gather broader results. We plan on continuing with the current test pool, while possibly expanding to new languages to reach an additional audience (we are currently discussing the possibility of translating the chatbot into Dutch). An alpha release of each of the products, along with recurring usage and feedback loop from our test users would allow us to continuously improve them, as well as the underlying predictive models.

The final prototypes for the 4u2 chatbot and the Content sWitch scenarios, developed on the basis of larger scale user tests, will be reported in deliverable D6.3.