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

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

Deliverable 2.1 (M10)

Temporal Annotation and Metrics Extraction

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	2.1
Deliverable Title	Temporal Annotation and Metrics Extraction
Due Date	M10
Submission Date	October 31, 2018
Work Package	WP2
Partners	MODUL Technology, webLyzard, Genistat
Author(s)	Lyndon Nixon, MODUL Technology
Reviewer(s)	Lizzy Komen, Sound and Vision
Keywords	Temporal Annotation, Event Extraction, Event Modeling, Audience Metrics, Success Metrics, Prediction Models, Predictive Analytics
Dissemination Level	PU
Project Coordinator	Vrije Universiteit Amsterdam De Boelelaan 1081 , 1081 HV, Amsterdam, The Netherlands
Contact Details	Coordinator: Prof Lora Aroyo (lora.aroyo@vu.nl) R&D Manager: Dr Lyndon Nixon (lyndon.nixon@modultech.eu) Innovation Manager: Bea Knecht (bea@zattoo.com)

Revisions

Version	Date	Author	Changes
0.1	24/8/18	L. Nixon	Created template and ToC
0.2	25/8/18	A. Scharl	First draft of content-based metrics section
0.3	18/9/18	L. Nixon	First draft of event extraction section
0.4	2/10/18	L. Nixon	Second draft of event extraction & first draf of prediction
0.5	4/10/18	M. Kappeler K. Ciesielski	First draft of audience metrics section
0.6	11/10/18	L. Nixon	Final draft of event extraction and predictior
0.7	12/10/18	K. Ciesielski	Final draft of audience metrics section
0.8	15/10/18	A. Scharl	Final draft of content-based metrics section, deliverable sent to QA
0.9	24/10/18	L. Komen	QA
1.0	25/10/18	L. Nixon	Resolved QA comments
1.1	27/10/18	A. Scharl	Final check by R&D Lead
1.2	29/10/18	L. Nixon	Final check by Project Lead

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This deliverable contains original unpublished work except where clearly indicated otherwise.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

EXECUTIVE SUMMARY

This deliverable presents the plans for implementing the ReTV event extraction capabilities and the temporal annotation of content items. It also identifies the content-based success and audience metrics to be measured and describes the intended approach to extract metrics across published vectors. Finally, it outlines how the extracted events and their temporal information, alongside temporal content-based success and audience metrics, may be combined to build a prediction model and enable cross-vector metrics-based prediction for digital TV content as part of the Trans Vector Platform (TVP).

ABBREVIATIONS LIST

Abbreviation	Description
API	Application Programming Interface: a set of functions and procedures that allow the creation of applications which access the features or data of an application or other service.
EPG	Electronic Program Guides: menu-based systems that provide users of television with continuously updated menus displaying broadcast programming or scheduling information for current and upcoming programming.
OTT	Over The Top: refers to content providers who distribute streaming media as a standalone product directly to viewers over the Internet
RDF	Resource Description Framework: a method for conceptual description or modeling of information that is implemented in web resources.
REST	Representational State Transfer: an architectural style that defines a set of constraints to be used for creating web services.
SKB	Semantic Knowledge Base: a knowledge base stores complex structured information in the form of a 'knowledge representation', when this representation is based on formal logics (e.g. in RDF) then it may be considered 'semantic'. The term is used in ReTV to refer specifically to an implementation of a semantic knowledge base by MODUL Technology.
SPARQL	SPARQL Protocol and RDF Query Language: a semantic query language for RDF-conform knowledge bases such as the SKB
URL	Uniform Resource Locator: a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it.

1 INTRODUCTION

This deliverable presents the plans within ReTV Workpackage 2 for implementing the event extraction capabilities and the temporal annotation of content items (T2.1, Chapter 2). It also identifies the content-based success and audience metrics to be measured and outlines the intended approach and implementation to achieve the metric extraction across published vectors (T2.2, Chapter 3 and T2.3, Chapter 4). Finally it looks forward to how the extracted events and their temporal information, alongside temporal content-based success and audience metrics, may be combined to build a prediction model and enable cross-vector metrics-based prediction for digital TV content as part of the Trans Vector Platform (T2.4, Chapter 5). As such, the collection, representation and provision of each of the data covered in this deliverable - events, audience figures & online success metrics - will become inputs to our prediction model, and first results of that will be the subject of the deliverable D2.2 (August 2019).

2 EVENT EXTRACTION AND TEMPORAL ANNOTATION

An **Event Knowledge Graph** will capture the existence and known characteristics of real world events, providing an unique and disambiguated reference (URI) which can be used in annotations and leveraged in data analytics, including classification (i.e. documents reference specific types of events), content recommendation (i.e. recommend a content item to a viewer because of the events it references match the types of events the viewer is interested in) and prediction (i.e. predict future impacts on a metric caused by already known future events based on the analysed impact measured during similar past events). **Events** are defined by us in this context as: *something occurring in the real world involving one or more agents creating some change during a finite temporal period at a bounded geographical location*. However, as we deal with TV programming, we may have to consider if we need to extend this at some point to fictional events, nevertheless for now we focus on prediction which requires knowledge of past and future real world events.

A Semantic Knowledge Base (SKB) (see Deliverable D1.1) has been set up which is triple-based (RDF) and captures entities of the form Named Entities (NEs), generally linked into their equivalents in public Knowledge Graphs like DBPedia and Wikidata, and of the form Non-entity Keywords (NEKs) - any concept which does not have an entity representation - expressed as lexical entries and linked to lexical ontologies based on data from OmegaWiki, modeled with the LEMON ontology. We will extend the SKB with a new **`Event`** entity class, which is linkable to event representations in other KBs and hence represented as a NE, extending the other supported NE classes of the SKB (Person, Organisation, Location and in due course Works[1]). Thus the ReTV Event Knowledge Graph will be part of the SKB which is also used in WP1 to provide resolution of other entity types such as Locations and Works.

[1] As part of the work of ReTV WP1, see Deliverable D1.1, we will add entities for TV Series.

2.1 Event Description Model

An **event model** defines the permitted properties and values of an event, generally including, having assessed existing event models (see below):

- spatial coverage
- temporal coverage
- agents (persons and organisations with a role in the event)
- type (the class of event it belongs to)
- prefLabel/altLabel (how the event is officially or informally referred to)
- recurrence (indicates an event which is part of a repeated series of events)

We examined several published specifications for describing events online:

- http://schema.org/Event (is more specific to popular events like concerts and festivals)
- http://linkedevents.org/ontology/ (simpler, captures any type of event. Our property-value usage is proposed below)
  - Uses W3C OWL-Time https://www.w3.org/TR/owl-time
  - Uses W3C Basic Geo http://www.w3.org/2003/01/geo/wgs84_pos#
- DBPEDIA http://dbpedia.org/ontology/Event
  - Full list of dbpedia event attributes: http://mappings.dbpedia.org/server/ontology/classes/Event
  - Full dbpedia event category taxonomy: http://mappings.dbpedia.org/server/ontology/classes/#Event
- DUL (http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#Event)
- Wikidata (https://www.wikidata.org/wiki/Q1656682)
- Dublin CORE (purl.org/dc/dcmitype/)
- E5 Events (http://www.cidoc-crm.org/html/5.0.4/cidoc-crm.html#E5)
- motools (http://motools.sourceforge.net/event/event.html#Event)

We decided to model events in as generic as manner as possible, as we do not want to exclude any type of past or future event as the project proceeds (Table 1). We were guided by the Linked Open Description of Events (LODE) model[2], also because it supports LOD URIs to reference entities (Deliverable D1.1 has introduced our Knowledge Graph and policy to use URIs as identifiers for entities).

atPlace (URI)	A named entity of a location which encompasses where the event took place
atTime (TemporalEntity)	An instant in time or an interval of time, with a start, end and duration
circa (URI)	A named entity of a calendar date that generalises the temporal bounds of the event (e.g. 11th September 2001)

[2] http://linkedevents.org/ontology

illustrate (URL)	Any online media asset which provides a fair representation of the event
inSpace(SpatialThing)	A formally specified spatial region, e.g. bounding box with points (longitude/latitude)
involved (URI)	Anything which is related to the event. Could be e.g. lists of keywords derived from text about the event
involvedAgent (URI)	Named entities with a significant involvement in the event - normally of types Person and Organisation

Table 1. The LODE event model.

2.2 Event Knowledge Base

A Semantic Knowledge Base is maintained by MODUL Technology, running on Apache Jena and Fuseki which is a RDF triple store and data query/update server. So all entities in the SKB are represented in RDF, drawing as far as possible from existing published RDF vocabularies. Considering LODE as a conceptual model for events we want to capture as closely as possible in the entity representations used by the SKB, and aligning the properties of Event entities with the properties already being used for entities of other types in the SKB, we come to the below specification (Table 2). Properties used come either from the Dublin Core Metadata Set (dc:) or webLyzard document model (wl:). Since we begin event collection through queries against the public query endpoint of Wikidata (see next section) we add here the equivalence with Wikidata properties.

Property	Description	sameAs
dc:url	The entity key	WikiData URI
dc:type	Genre, event type	https://www.wikidata.org/wiki/Property:P31
dc:label	Textual label for the event	rdfs:label
dc:description	Textual description of the event	schema:description
dc:source	Publication source	“WikiData”
wl:temporal_start	Start datetime	https://www.wikidata.org/wiki/Property:P580
wl:temporal_end	End datetime	https://www.wikidata.org/wiki/Property:P582
wl:year	Just the year (for filters)	
wl:md_date	Just the month-day (for	

	filters)	
dc:publication_date	Date when published by the source	
wl:location	Geo location (lat/lon)	Various: "location": "https://www.wikidata.org/wiki/Property:P276", "administrative": "https://www.wikidata.org/wiki/Property:P131", "country": "https://www.wikidata.org/wiki/Property:P17"
wl:country	Country ISO code	https://www.wikidata.org/wiki/Property:P17
wl:coordinates	Coordinates (point or shape)	https://www.wikidata.org/wiki/Property:P625
wl:frequency	How often the event recurs	https://www.wikidata.org/wiki/Property:P2257
wl:previous_instance	Link to the previous recurrence of the event	https://www.wikidata.org/wiki/Property:P155
wl:next_instance	Link to the next recurrence of the event	https://www.wikidata.org/wiki/Property:P156
wl:participants	Any person/organization co-reference	Various: 'participatingTeam': https://www.wikidata.org/wiki/Property:P1923, "participant": "https://www.wikidata.org/wiki/Property:P710", "winner": "https://www.wikidata.org/wiki/Property:P1346", "speaker": "https://www.wikidata.org/wiki/Property:P823", "organizer": "https://www.wikidata.org/wiki/Property:P664, "openedBy": "https://www.wikidata.org/wiki/Property:P542", "guestOfHonor": "https://www.wikidata.org/

		wiki/Property:P967"

Table 2. Mapping of the event model into the RDF model of the Semantic Knowledge Base.

2.3 Event Extraction from Structured Data

We performed the initial seeding of the Semantic Knowledge Base from public knowledge graphs which contain entities of type Event. Following an examination of sample data, we found Wikidata to be generally cleaner and more expressive than DBPedia (and others) for events (e.g. details of recurrence of events are more present in Wikidata, also links to event locations and participants are more often and more complete). We constructed a (SPARQL) query which retrieves a set of entities of type Event within a restricted time frame (more general queries over the complete Wikidata KB would be too demanding) and map the response (a list of entities and their details) to our event model (create for each item in the response a new entity in the SKB which instantiates the event model). Table 3 shows the defined mapping between WikiData properties and our event model.

Wikidata Property Mappings

Event property	**Wikidata property**	**Wikidata property label (and/or comment)**
Type of event	P31	Instance of
Spatial entities (atPlace)	P17 P276 P1427 P1444	Country Location Start point (e.g. for a race) Destination point (e.g. for a race)
Spatial datatypes (inSpace)	P625	Coordinate location (get lat and long from the location entity)
Temporal entities (circa)		(extract Day-Month and Month-Year entities from the temporal range)
Temporal datatypes (atTime)	P585 P580 P582	Point in time Start time End time
Involved entities (involved, involvedAgent)	P541 P641 P1923 P664	(NER/NEL for entity extraction from descriptive text) Office contested (in elections) Sport Participating teams organizer

Preferred label	rdfs:label	Label
Alternative labels	skos:altLabel P1449	(also known as) nickname
Recurrence (previous, next)	P155 P156	Follows Followed by
Other	P361	Part of (could be followed to extract additional, super-events for the retrieved event)

Table 3. Mapping between our event model and WikiData properties

Wikidata Types

The top level Class for any type of event is **Q1190554** - occurrence. However tests showed that even a query restricted to this type also with shorter time frames could often return many entities of less relevance to ReTV (a single example: Q955075 “Max Headroom broadcast signal intrusion” which is a ‘Chicago television hijacking incident’ from November 22, 1987) as well as show poor response performance (as the query runs against the entire Wikidata knowledge graph).

We sampled events we knew about and felt would be relevant to ReTV and collected their direct types (which are more specific as the “occurrence” type which is modelled as a superclass) - considering events for prediction, we considered the most likely events to cause variations in TV viewing and focused on Sports (e.g. a World Cup) and Politics (e.g. a national election). We began as a result with the following political and sports event types in extracting Wikidata events for a 30 day period in our queries:

- 'wd:Q16466010': 'association football match',
- 'wd:Q47089371': 'rugby union match',
- 'wd:Q2618461': 'legislative election',
- 'wd:Q17195514': 'political conference',
- 'wd:Q2705092': 'Formula One racing',
- 'wd:Q13406554': 'sport competition',
- 'wd:Q18573266': 'ski race',
- 'wd:Q1076105': 'general election',
- 'wd:Q47459169': 'tennis match',
- 'wd:Q858439': 'presidential election',
- 'wd:Q2515494': 'constitutional referendum',
  # types of bike races
- 'wd:Q22231118': 'CC',
- 'wd:Q20680270': 'medium mountain stage'

As also noted later, we can extend this more specific type list in our query as well.

Extraction Workflow

Once a day, a query is sent per Wikidata entity type to the public SPARQL query endpoint[3], with date range [0d-50d] days into the future from presence. This yields about one to two events over all Wikidata types, per day.

A first extraction run led to 247 events for a window from 60 days in the past to 180 days in the future (current time range: 3 July 2018 to 26 February 2019). An analysis of these events (Table 4) found 200 unique events (45 repetitions, 2 incorrect), of the following categories:

Category of event	Number of unique instances
Sports	91 (45.5%)
Politics	85 (42.5%)
News	10 (5%)
Weather	13 (6.5%)
Culture	1 (0.5%)

Table 4. Extracted events

We also considered the coverage of the event extraction. We looked at https://en.wikipedia.org/wiki/Portal:Current_events/July_2018 and https://en.wikipedia.org/wiki/Portal:Current_events/August_2018 for stories related to events in the categories (Culture, Disasters, Politics, Sports). We identified 36 events of relevance, thus the Wikidata extraction seems more detailed, mentioning primarily more individual sports events (including World Cup games) but also covering more news/weather events too. However only 13 events actually were present in both sources: the remaining 23 sports, political and news events were only in Wikipedia (the main overlap was on weather events). We identified that 7 had no entity in Wikidata and for the other 16:

Had an event type we weren't querying for: 13
Had no event class in Wikidata: 2
Had no precise date: 1

As a result we could add 8 more event subtypes to our query, which would find the majority of the Wikidata events which were listed in Wikipedia Current Events but not extracted by our original query. Regardless, we could see that coverage from Wikidata is satisfactorily complete (considering Wikipedia as summarizing only more major events we would now have extracted 26 of the 36 events (72%) it references, and the remainder were effectively missing from the current Wikidata store).

[3] https://query.wikidata.org/

The missing events serve as a reminder that not every (significant world) event becomes an entity in a Knowledge Graph, reliant as these public graphs are on whether or not a Wikipedia article or Wikidata entity is specifically created for the event. Therefore some events may first appear in the event extraction some time after the event itself.

We also recognized that event metadata may be subsequently updated. As a first step we looked to remove repeated events from subsequent query responses. However a further step is to modify the logic of the event extraction such that when a previously extracted event is returned, to check first if the metadata has changed and adopt the updated/additional metadata into the original event entity in our KB.

2.4 Event Extraction from Unstructured Data

Especially if we want to regard local or regional events, it becomes less likely that they will be represented in a public, global knowledge graph such as Wikidata. Also, in the TV world, such events may be referred to soon after the event itself (thinking of local or regional news, for example). A solution to this would be to be able to extract ad hoc references to events from unstructured data sources, e.g. online social media postings or news articles on websites (also carried by broadcasters websites themselves). In an initial implementation, we are getting - for a tweet or a news article headline - the (a) title, (b) subject, predicate and optionally object of the title sentence, (c) persons, organisations (both agents) and locations as detected by our Named Entity Recognition tool RECOGYNZE [Brasoveanu, 2018] in the text plus (d) dates mentioned in the text (for now only absolute values). This allows us to test the extracted data quality for determining references to known and new events in unstructured data sources.

Detecting references to time will require identification of both relative and absolute times at different levels of granularity (from minute to year). We are extending RECOGNYZE (which generally annotates textual documents with entities of different types such as Persons or Locations) to recognize temporal references and annotate (the main text of) documents with “Date” entities. Relative temporal references are resolved with respect to the known publication date of the document. At this point we may have any number of temporal references from the text of the document (a news article or a social media post). Using NLP, a detected reference to a time point associated with a statement about that temporal reference can be considered a potential event, with a NIL linking initially (i.e. label the statement with an unknown, unidentified Event entity), defined by its temporal bounds, the surface form (text associated with the temporal reference) and named entities detected within the surface form (Persons, Organisations, Locations). We expect with this next update to be able to associate temporal references with the event at that time, and based on extracted surface forms and participating events, determine if the reference is to an event already existing in the Semantic Knowledge Base or a new event which may be added.

2.5 Outlook (Events in Content Annotation and Prediction)

We have started with event extraction from structured data (WikiData). A prototype approach for event extraction from unstructured data has been implemented and will be evaluated using

a small set of documents to ascertain the usability of the data. The detection of temporal references in text and the annotation of documents with a resulting Date entity will be part of the next update to our RECOGNYZE service. Once we have Date entities (and the generic event extraction), we will test the identification of references to existing events and creation of new events from the unstructured data sources, complementing the current pipeline from structured data.

The (historical) events in the Semantic Knowledge Base will be correlated with selected statistical properties, i.e. numerical changes around the time of the occurrence of the event, in order to learn potential variations in some properties from the underlying trend due to that event. We will be able to combine the temporal range of the event with the time-based metrics described in the following chapters within the platform. This learning can be used in prediction, so that a similar future event can be used to predict similar changes in some properties at that time on the basis of including the learnt variations to the underlying extrapolated trend-based data.

3 Content-Based Metrics

ReTV will integrate metrics around content related to TV programs to unlock new ways to measure TV communication success (how a TV broadcast is being reacted to by the audience), far beyond classical lexical indicators such as document sentiment. In this context, the goal of our work is to provide success metrics within the TVP which are extracted from secondary content channels (Web documents, social media postings, accompanying content like snippets or trailers), related to a TV media asset, fully customized to the content owners' defined dissemination and positioning goals and going technically beyond the current metrics available to media organisations in generic Web and social media analytics platforms.

The current webLyzard platform provides document analytics which cover the classical Web and social media success metrics of sentiment as well as engagement (e.g. number of likes of a tweet, number of views of a video). WYSDOM, the *webLyzard Stakeholder Dialogue and Opinion Model* (Scharl et al., 2017), provides a dynamic assessment that includes sentiment, but also evaluates the degree of association with desired topics considered important. It also determines whether undesired media coverage was avoided successfully.

Building upon this previous work in this area, we will extend the WYSDOM success metric by (i) moving from positive or negative sentiment to multiple-dimensional emotional categories such as anticipation, surprise and joy, (ii) providing measures beyond awareness, for example the social perceptions of a specific program, and (iii) replacing daily data points by a more granular analysis to track the impact of short-term interventions, e.g. changes in an organization's online marketing at the level of an individual publication.

3.1 Multilingual Baseline Evaluation of Sentiment

To assess the accuracy of our existing affective knowledge extraction algorithms (to be deployed as part of the initial ReTV dashboard release), we generated a gold standard[4] dataset in close collaboration with HTW Chur in Switzerland for both English and German text. Students provided a single rating per article in the form of integer values (1 for negative, 2 for neutral, and 3 for positive, which we transformed into a -1, 0, and 1 annotation). The preliminary findings of this baseline evaluation include:

- At an *accuracy threshold* of 1.0, there are almost twice as many mismatches among the German than among the English articles in the sample.
- A majority of misjudged articles show a *positive bias*: Across all 500+ gold standard sentences, only 15 are rated more negative than the standard at a difference of 1.0 or more, while 109 are rated too positive. Overly positive ratings are thus responsible for the majority of outliers.
- While the average absolute divergence from the standard is similar for both languages, the bias towards too positive ratings is *more expressed for German* than for English:
- The *difference* between the average rating in the gold standard and the average baseline rating is + 0.23 points for English and +0.66 points for German.

[4] A gold standard refers to being a "best of" its kind and hence can be used as a benchmark to measure the comparative quality of other datasets or be used in training an algorithm

- Similarly, for *German*, almost all misrated sentences are too positive (77 vs. 4 that are rated too negative), while for *English* the errors too are more balanced (27 vs. 11 at a cutoff of 1.0 divergence from the Gold standard).

3.2 Planned sentiment Analysis Improvements

A follow-up review of the baseline evaluation revealed that negation processing errors (e.g., "not only") and sentence structures including words like "need" or "require" contributed to the observed patterns. To address these shortcomings, we will improve our existing affective knowledge extraction algorithm as follows:

1. revise the negation processing pipeline to consider specific n-gram phrases;
2. eliminate the positive bias of German sentiment analysis by improved context processing,
3. assess and adapt the sentiment lexicon terms stored in the knowledge graph to use them in the extended WYSDOM success metric.

Other conceptual improvements to WLT's sentiment analysis components will leverage the capabilities of the Knowledge Graph developed by MOD in T1.4 to capture the properties of lexical entities (in terms of sense disambiguation, query term expansion, and the classification into emotional categories such as anticipation, surprise and joy). Lexical entities are described by a lexical model based on lemon - the Lexicon Model for Ontologies (lemon-model.net). It has been populated initially based on the multi-lingual OmegaWiki (www.omegawiki.org), which resulted in 38,758 distinct German terms and 55,058 distinct English terms (Nixon, 2018).

The increased flexibility gained by refactoring the WYSDOM chart component and eliminating third-party library dependencies will help to represent those additional lexical categories in the extended WYSDOM success metric. In the next project phase, we plan to run additional experiments to determine whether (a) *negation detection* and (b) *distinguishing multiple emotional categories* beyond sentiment need to be precomputed, or could also be managed via on-the fly-computations.

3.3 Reach Metrics Ingestion and Normalization across Vectors

Another major focus of the current work is on the ingestion of reach metrics from multiple vectors (i.e., social media channels but also Websites), and especially their normalization so that they can be compared to each other as part of joint visualizations (T4.2). Currently we have only absolute reach estimates from each platform e.g. video views on YouTube or the number of engagements with a social media posting plus the number of followers of the channel where it was posted. These are also not aligned to the overall number of users of each platform, e.g. having the same number of followers on Instagram as on Facebook is not the same as Facebook has maybe five times as many unique monthly users. This required significant changes to the underlying data model, which were implemented together with the migration from Elasticsearch 1.x to the latest 6.4 release in T4.4.

Individual metrics, depending on platform and API availability, are collected per platform and normalised onto a range of 0 to 1. This includes *Alexa rank* (by domain) for regular Websites, the *number of views* for Vimeo and Dailymotion, the *number of views, likes and dislikes* for YouTube, the *number of followers* and *accounts followed* for Twitter. For social platforms (= vector), the calculated number is capped (eg. for YouTube, values above 500.000 are mapped

to 1.0), the remaining interval of 0 to 500.000 is then divided into „bins“ - e.g., 250.000-500.000 is assigned a reach of 0.9. Ongoing work will further optimize both the per-vector normalisation as well as the normalisation across vectors.

4 Audience Metrics

Audience metrics are based on aggregations of raw data available from the Zattoo OTT TV platform. The way we think of audience data is as a graph or network. The nodes are the TV shows. Between each TV show we have flows of users. Those flows can be split up by dimensions: age, gender, geographic region, app used etc. The graph grows over time. As new shows air, they are added as new nodes, and users start flowing to them. Fig. 1 below shows an exemplary flow of users between shows on German private channel Pro7. In this example the flows are not split up by age, gender or location.

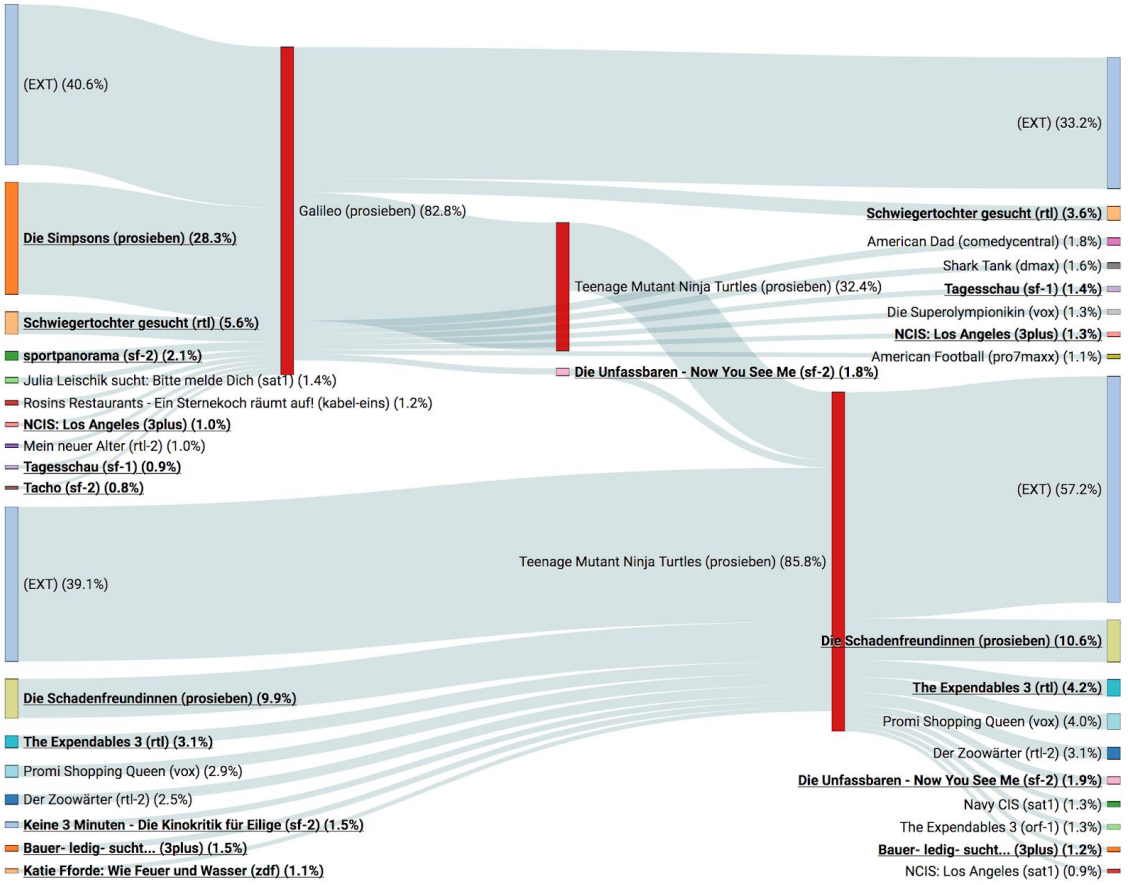

Figure 1. Visualisation of audience flow between TV programs

In the following sections we describe the raw data we need to build this graph.

4.1. Program Data

We need the program data to know which nodes are in the graph.

Source	BEE API
Export frequency	Every 12 hours

Example:

```
{
    "pid": 135738677,
    "dc_name": "sf-1.AxelSpringer",
    "epg_source": "AxelSpringer",
    "description": "Als \"Schweizer Taschenmesser\" bezeichnet Rolf Dobelli sein
neuestes Buch: 52 Werkzeuge, um sich ein gutes Leben zusammenzuzimmern...",
    "id": 156,
    "channel_id": 1,
    "title": "Sternstunde Philosophie",
    "subtitle": null,
    "start_broadcast_stream_time_s": 1520215500,
    "end_broadcast_stream_time_s": 1520218800,
   }
```


This EPG data is being added to the platform as part of the data collection work, see Deliverable D1.1.

4.2. Session Data

The session data is used to build the flows between the programs and is therefore the starting point for any clustering or prediction model.

Source	CSV files or access to Vertica DWH (SQL)
Export frequency	daily

Field	**Description**
`id`	Unique ID for each session.
`public_id`	Hash of the zuid. If the user has user_type=anon the uid is generated using a cookie. An anonymous user might therefore be present with multiple public_ids.
`channel`	Name of the channel that the viewer is watching.
`start_time`	Wallclock time at which the user started watching the channel.
`duration`	Duration in seconds for which the user watched the channel.
`pvr_time`	Time at which the channel was broadcasted. Due to timeshifted viewing the difference between start_time and pvr_time might be very large.
`device`	Type of device used: [‘iphone’, ‘android’, ‘ipad’ etc.]
`country`	Country where the viewer generated the session.
`location`	Name of the location where the session was started.

	Determined by IP-lookup using MaxMind.
`zip`	Name of the location where the session was started. Determined by IP-lookup using MaxMind.
`latitude`	GPS latitude of the location. The GPS coordinates are of the location, not the user itself.
`longitude`	GPS longitude of the location. The GPS coordinates are of the location, not the user itself.
`connectivity`	Connectivity type: [‘mobile’, ‘cable’]
`asset_type`	*Live:* Live TV. *Recall:* Users watching content time-shifted. *Selective_recall:* Generally time-shifted viewing is not possible in Germany. Some selected channels do allow it though. If such a channel is watched time-shifted, this asset_type is used. *Pvr:* “Personal Video Recording”. Time shifted viewing when a user added a show to their recordings. Personal recordings are stored forever. Recall viewing of content that was not added to the personal video recordings is only possible for seven days. Technically there is no difference between PVR and Recall.
`seek`	If a user seeks more than 60 seconds in the player, a new program session is generated. This field is set to true, if a program session was generated out of such a seek.
`csid`	Sometimes a session is split into multiple parts. This can happen for different reasons. For example when a user switches from WIFI to mobile data it can be that a new program session is generated. Such split program sessions can be merged by the csid (channel switch id) which will be identical.

4.3. Aggregated Real-Time Data

The session data is not real-time. To display a live flow of the audience (something broadcasters are interested in), we need real-time data. This data basically tells us: at time x, there were n users on channel y in region z.

Source	Elasticsearch channel-users index or SQL database.
Export frequency	real-time

Field	**Description**

region_id	Geographical region
user_space	Type of Zattoo users (B2B vs. B2C for example)
channel	Name of the TV channel
timestamp	Point in time when the measurement was made.
users	Number of users that are watching the channel at this point in time.

Example:


```
{
                "_index": "channel-users-20180226",
                "_type": "channelstats",
                "_id": "AWHTPCvCw1iAHFPoLFCt",
                "_score": null,
                "_source": {
                  "region_id": 10204,
                  "user_space": 1,
                  "timestamp": 1519667385,
                  "users": 150,
                  "channel": "telezueri"
                }
}
```


4.4. User Data

We need the user data, in order to be able to split the flows between the programs by age, gender, country etc. This is data available within the Zattoo ecosystem and Genistat has signed an agreement for privacy-preserving access to the data in order to create audience metrics at the level of viewing cohorts (i.e. anonymising viewing preferences to a group of similar user types which can be used in ReTV re-purposing and recommendation components without sharing user's personal data).

Source	CSV files or access to Vertica DWH (SQL)
Export frequency	daily

Field	**Description**
public_id	Hash of the zuid. Can be used to join the user data to the session data.
birthyear	Birth Year of the user. Based on data reported by the user. We could break this down into age groups (15-25, 25-35 etc.)

gender	Gender of the user. Based on data reported by the user.
language	Language code of the user interface language that the user set.
reg_datetime	Time of registration.
reg_app_id	App the user registered with.
reg_country	Country the user registered in.
user_space	The type of user space the user is part of (Zattoo vs. B2B for example).
user_type	Free: Free user that uses the ad-supported service. Partner: User that accesses Zattoo over a B2B partner. Anon: Users that watch Zattoo without logging in. Possible when a newspaper includes a Zattoo player window. Zattoo_hiq: Users paying for a premium account. Premium users are the only ones that can watch recall content. Zattoo: Zattoo employees. Salt_hiq: Users that got a Zattoo subscription over Salt.ch Pay: Users paying for a premium account. Premium users are the only ones that can watch time-shifted content.

4.5 Implementation

Genistat has access to the Zattoo audience metrics. Genistat extracts and aggregates the data from Zattoo via an Elasticsearch interface. The data is stored on Genistat servers and then pushed to webLyzard via the Statistical Data API. The webLyzard API is documented here: https://api.weblyzard.com/doc/ui/#/Statistical_Data_API. See the pipeline below (Fig 2.) for the data flow between ReTV partners.

Figure 2. Audience data flow between partners
(from individual through aggregated to anonymised)

For the data push, webLyzard has rewritten and extended the data ingestion mechanism of the webLyzard platform to ensure compatibility with the Statistical Data API (originally this API

assumed a direct interface to Google Analytics and was not directly suited for the integration with the audience metrics data stream, see Chapter 4). In addition, safeguards to guarantee temporal data consistency via granularity constraints on the temporal data descriptors have been added. This ensures that for any given time frame (i.e. five minutes for the live audience data), only one observation can be submitted to the platform per indicator. At the time of this writing, the adaptation of the Statistical Data API towards the new ReTV use cases has been completed, and the ingestion of Genistat audience metrics into the platform is in operation.

At the moment the following fields are accepted by the API.

```
{
  "_id": "1",
  "uri": "http://example.com/test-uri-01",
  "added_date": "2014-09-10T15:01:48.623816",
  "date": "2004-01-01T00:00:00",
  "indicator_id": "esairtrans2",
  "indicator_name": "ES Air Trans 2",
  "value": "1000",
  "year": "2004",
  "month": "string",
  "day": "string",
  "hour": "string",
  "location_id": "string",
  "target_type": "country",
  "target_poi_type": "string",
  "target_country": "CZ",
  "target_location": [
    {
      "name": "Czech Republic",
      "point": {
        "lat": 49.75,
        "long": 15
      }
    }
  ],
  "source_type": "string",
  "source_poi_type": "string",
  "source_country": "string",
  "source_location": "string",
  "producer": "Eurostat",
  "frequency": "year",
  "description": "Air transport of passengers",
  "unit_of_measurement": "string",
  "type": "observation"
}
```

We have agreed that we collect data for two time spans.

1. ***Live:*** For 5 min units
2. ***Daily:*** Daily units

We aggregate the data for the top 100 watched channels for Germany and Switzerland. The data is *non cumulative.* Adding the results will lead to users being counted multiple times. For example if user A is watching a channel in the first 5 min and then in the next 5 min as well, the user will be counted in each time span.

4.6 Models to Interpolate Missing Information and Extrapolate Future Metrics

4.6.1 Future audience forecasting (time-series)

The goal of audience forecasts is to predict the future value of several base metrics (no. sessions, no. users, avg. session duration), that can be used to calculate derived audience metrics such as TV rating or market share. Forecasts can be run in real-time mode (e.g. predicting values for the next hour in a 5 minutes time slices) or batch mode (e.g. predicting audience for the next week in a daily time slices). Input data to the models include:

- long-term aggregates of the predicted metric (e.g. the number of user sessions) per channel. Such aggregates describe long-term trends as well as patterns (e.g. how many users watch main news show on SRF-1 on Sunday evening)
- short-term aggregates of the metric, calculated in real-time and providing the current context that can modify the long-term pattern (e.g. current audience is lower than expected due to factors that are not in the data, such as current weather)

Input data are calculated separately per each TV channel based on Zattoo data. We have also built a separate extrapolation model that is able to re-scale these values to expected size of the OTV market.

The model is based on time-series forecasting and gradient boosting regression models (the latter allowing to include additional predictive features of the audience, such as interests or socio-demographic features).

It should be stressed that the current version of the model is not able to predict anomalies such as special sport events. However, it is possible to include additional data (such as EPG information or event-classification models) that would allow to automatically tune-up forecasts in such cases, making use of the event extraction work described in Chapter 2.

Structure of the predicted and observed data points:

Field	Description
`creation_time`	Time when the entry (prediction or data aggregation) was created
`slice_start`	Start of the prediction/aggregation window. The length of the window is defined by slice_unit column
`channel`	Name of the TV channel
`sessions`	The number of sessions (predicted or observed in the past)
`data_source`	‘forecast’ for predictions, ‘els_real_time’ for observed real-time data (recent past), ‘vertica_aggregates` for long-time aggregations of the past data (e.g.

	daily or hourly)
`slice_unit`	length of the time window (relative to slice_start) in seconds. Currently either 300 seconds or 86400 seconds (1 day)

4.6.2 Socio-demographic data predictions (gender and age)

The majority of Zattoo users have no associated socio-demographic information (gender and age). Also, the user reported sociodemographic values seem to be false in many cases (cf. clearly visible peaks in age distribution plot below). We built several socio-demographic models to predict age and gender of users, based on the partially available information. We estimate our gender models to be accurate in 80% of cases and our age model to be accurate to within 7.65 years on average.

- We built separate models for the gender and age prediction. Gender prediction models were based on a set of random forests classifiers, which indicated if the user is female. Age prediction models were based on a set of regression forests models, trying to approximate user age (in years).
- Both models were based on a same set of predictive features, however the structure of each model, as well as the predictive importance of individual features in each models, varied significantly. The features we used, reflected user behavioral patterns and were built upon the aggregated session data, as well as aggregated EPG data that Genistat receives from Zattoo.
- As we briefly mentioned earlier, target values for each model (age and gender) are not verified in any way. They are just values entered by the user during the registration process. It results in low reliability of the target variable, that initially prevented us from building high quality models. We decided to apply additional anomaly detection techniques in order to identify observations (i.e. users), whose behavioral patterns were significantly different from typical behavior for her/his group. An example are users that declared to be young men but had the viewing behavior of middle-age women. Such observations were excluded from the training set. It is worth to mention that many of the anomalous observations, were located in peaks visible on the plots (see Fig. 3) (i.e. users that declared round birth years: 1970, 1980, 1990). On the other hand, it is also worth to stress that we experimented with “naive” approaches. Like for instance to just remove suspicious years (“peaks”) from the training set and such approach led to much worse predictive power of resulting models.

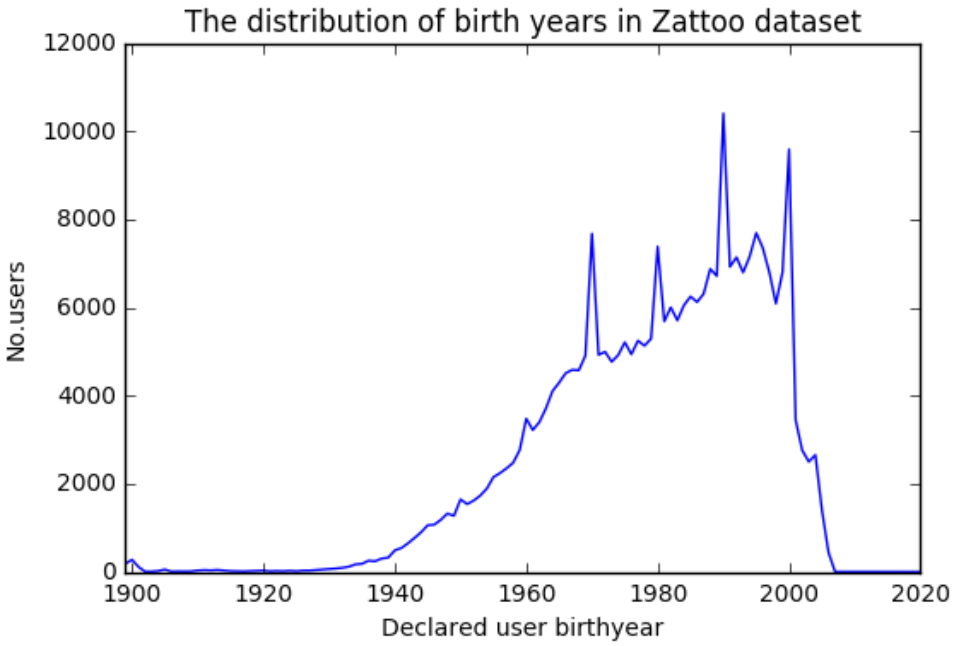

Figure 3. Sample plot from Zattoo user data

5 Conclusion and Outlook

While the events and various metrics will be useful in other contexts (events as annotation targets for content in WP1 for example, or the visualisation of metrics for professional user decision making in WP5), we consider here the potential for combining the extracted data into a prediction model. Prediction of TV audience has been long an important aspect for broadcasters, not only for those which run advertising against their content (since the price they can set for an ad slot is determined by the predicted "rating" for the TV program the ad is run against). Also for a public broadcaster, audience prediction can help them schedule the right content at the right time, or justify the expected ROI on future content production or licensing for broadcasts. TV "rating" is traditionally used, which is the percentage of all TV households which will tune into that channel at that time. Historic numbers are used in the prediction, which are determined by taking the rating of a broadcast among participating households and assuming that for the entire population. This has been the method since TV ratings began, despite the participating households typically being a very small percentage of all households, e.g. Nielsen uses circa 5000 households to represent the US TV ratings, where the US has around 116.3 million TV households. The advent of IP broadcast (OTT) provides a means to now measure more accurately how many households are tuned to a particular channel at a particular time, based on number of unique devices (includes Smart TVs and Set Top Boxes). As noted in the previous chapter, Zattoo has viewing data at the level of individual users.

Classical predictive analytics in TV is based on time series data analysis, which makes use of statistical extrapolation from historical figures into the future. The analysis models (i) the cyclical component of the numerical series, (ii) combined with detection of trends, and (iii) including a seasonal adjustment. Normally there will be irregularities in the data even after these three models are calculated and typically they are smoothed out for the prediction, meaning that future predicted data also does not anticipate the irregularities that will occur.

For the prediction model of ReTV (T2.4), our goal is a more accurate hybrid model, combining models for the time series analysis of the audience metrics (from T2.3) with anticipation of variations in the future data based on (a) foreseen trending topics based on past topic trends (using the content-based success metrics of ReTV from T2.2) and (b) known future events based on variations around similar past events using the event extraction in T2.1). This combination of models combined with learning about the optimal combination to achieve the most accurate results is known as a ***model ensemble*** (see Fig. 4 below).

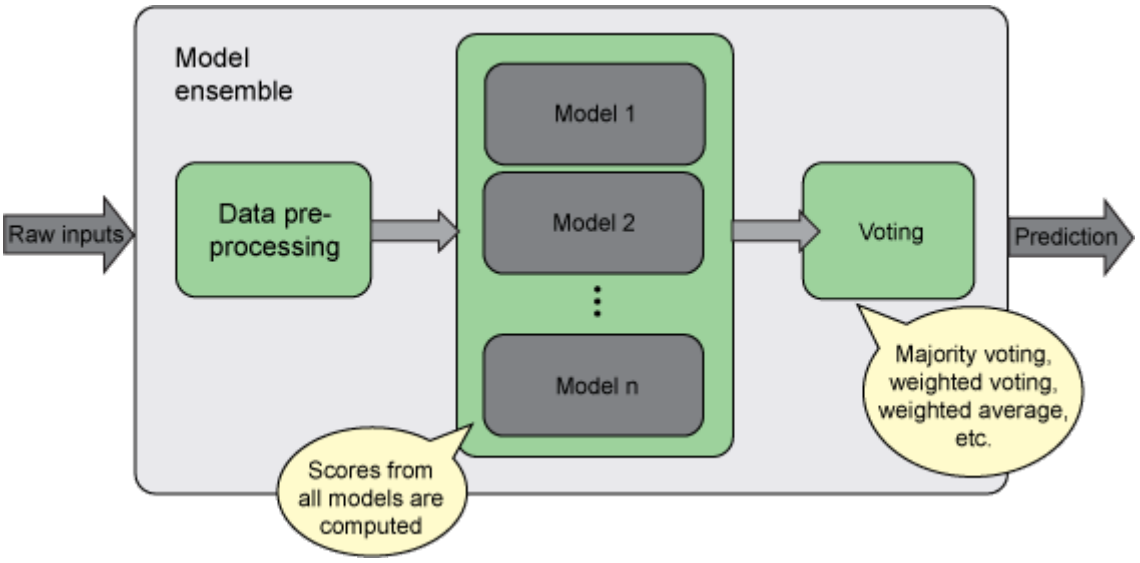

Figure 4. Illustration of a model ensemble for prediction

The follow-up deliverable D2.2 be reported in August 2019 will present a first model ensemble and results, as we work towards the achievement of more accurate prediction of TV program audiences and TV-related content popularity (on social media), to the benefit of the broadcasters and media organisations that deliver their programming.

REFERENCES

Brasoveanu, A., Nixon, L. and Weichselbraun, A. (2018). "StoryLens: A Multiple Views Corpus for Location and Event Detection". 8th International Conference on Web Intelligence, Mining and Semantics (WIMS2018). Novi Sad, Serbia, June 2018.

Nixon, L. et al. (2018) "Social media filtering and extraction, pre-processing and annotation, final version", InVID Project Deliverable D2.3, to be published at https://invid-project.eu, June 2018.

Scharl, A. et al. (2017). "Semantic Systems and Visual Tools to Support Environmental Communication", IEEE Systems Journal, 11(2): 762-771.