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

Deliverable 2.2 (M20)
Metrics-based Success Factors and Predictive Analytics, First Version

Version 1.5

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	2.2
Deliverable Title	Metrics, Success Factors and Predictive Analytics v1
Due Date	M20
Submission Date	August 31, 2019
Work Package	WP2
Partners	webLyzard technology, MODUL Technology, Genistat
Author(s)	Arno Scharl, webLyzard technology Lyndon Nixon, Jakob Steixner and Adrian Brasoveanu, MODUL Technology Krzysztof Ciesielski, Genistat AG
Reviewer(s)	Konstantinos Apostolidis, CERTH
Keywords	Temporal Annotation, Event Extraction, Event Modeling, Audience Metrics, Success Metrics, Prediction Models, Predictive Analytics
Dissemination Level	PU
Project Coordinator	MODUL Technology GmbH Am Kahlenberg 1, 1190 Vienna, Austria
Contact Details	Coordinator: Dr Lyndon Nixon (nixon@modultech.eu) R&D Manager: Prof Dr Arno Scharl (scharl@weblyzard.com) Innovation Manager: Bea Knecht (bea@zattoo.com)

Revisions

Version	Date	Author	Changes
0.1	24/6/19	L. Nixon	Created template and ToC
0.15	1/7/19	A. Scharl	Edits and structural changes
0.2	30/7/19	L. Nixon	First draft of Chapters 1 and 2
0.25	2/8/19	L. Nixon, J. Steixner	Finished text on events (Ch 2)
0.3	6/8/19	L. Nixon	First draft of Chapter 5 Prediction
0.4	13/8/19	J. Steixner, A. Brasoveanu	Contributions and revisions
0.5	14/8/19	L. Nixon K. Ciesielski	Finished draft of Chapter 5 Prediction; Initial inputs on audience metrics and prediction
0.6	15/8/19	A. Scharl	Major revision with a focus on Section 3
0.7	18/8/19	K. Ciesielski	Completed audience metrics and prediction (Sections 4 and 5)
0.8	21/8/19	L. Nixon	Wrote conclusion & added ethics self-assessment for audience metrics
0.9	22/8/19	A. Scharl	Completed Section 3.3
0.95	26/8/19	A. Scharl	Completed Section 3.1
1.0	26/8/19	K. Apostolidis	QA review (ReTV internal)
1.5	27/8/19	L. Nixon, A. Scharl	Post-QA updates

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This deliverable contains original unpublished work except where clearly indicated otherwise.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 the information contained therein.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This deliverable presents the first version of the success factors and predictive analytics based on the annotations and metrics derived from TV content which are stored and analysed by the Trans Vector Platform. This covers (T2.1) placement of content parallel to future events; (T2.2) sentiment detection and desired and undesired associations; (T2.3) trend detection from historical audience and viewer data and (T2.4) a generic prediction model for content independent of vector. The predictive capabilities of the TVP are used in the recommendation and personalisation services and surfaced to the professional users and TV consumers in our scenarios.

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.
JSON	JavaScript Object Notation: an open-standard file format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types.
NER	Named Entity Recognition: a subtask of information extraction that seeks to locate and classify named entity mentions in unstructured text into predefined categories such as Person, Organisation and Location.
NLP	Natural Language Processing: a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human languages, in particular how to process and analyze large amounts of natural language data.
POS	Part of Speech: in NLP, refers to the categorization of a word, a word-part or a phrase in a language as belonging to one or more grammatical forms, e.g. noun, verb, adjective, adverb.
RDF	Resource Description Framework: a method for conceptual description or modeling of information that is implemented in web resources.
RMSE	Root Mean Square Error: a frequently used measure of the differences between the values predicted by a model and the values observed.
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

1 Introduction

This deliverable presents a first version of a set of data-driven services to support the task of prediction in ReTV. One of the biggest challenges for any media owner is knowing the future, since if they could anticipate the future interests of their audience, they could publish the right content at the right time. Success is being measured on the TV channel vector in viewing audience; on online channels like social media there are the metrics of reach (how many people see the content) and engagement (how many people interact with the content - give a like, share or comment). Our goal in ReTV is to offer predictive services that can optimise those success metrics for the content owners. To be able to offer prediction, appropriate data has to be available. The content and the online discussion around it is collected by our data ingestion pipeline (cf. D1.2) and we annotate both the video of the content itself (from the TV channel or archive) as well as Web and social media about the content. To be able to predict into the future, we consider three different paths to collect and analyse data, updating on the work already presented in the deliverable D2.1:

(1) Data collection of past and future events (Chapter 2);
(2) Data analytics is applied to measure the past success of content by vector (Web and social) (Chapter 3);
(3) Data collection of past audiences by TV channel (Chapter 4).

In the final chapter (Chapter 5) we outline how the various data is used to provide a first version of a set of prediction services for ReTV.

2 Event Extraction and Temporal Annotation

Event extraction and temporal annotation has the intention to collect information about past and future events into a Knowledge Graph so that we can correlate past events to changes in audience or content-based success metrics and use those correlations in predicting future changes based on future events. We re-use our Semantic Knowledge Base (SKB) (cf. D1.2) for annotated Named Entities in documents (generally of types Person, Organisation and Location) to also store the events as instances of Event type named entities. As per our event description model, we capture event metadata such as location and (start/end) time with the event.

Our event collection has focused on three sources:

- WikiData: typically well-known events are represented in this knowledge graph as resources. We collect events based on the resource typing, using a predetermined list of specific types (related largely to sports or politics) in order to reduce query complexity and avoid largely irrelevant event instances.

- iCal files: less well-known events may have an effect on national or regional TV content, but are not occurring in global knowledge graphs like WikiData, for example, football league matches. To address this gap in event knowledge, we found national sport schedules (Germany, Austria and Switzerland) in iCal format on the Web and developed an iCal content ingestion to generate for each iCal entry a new event for our Knowledge Graph.

- Public holidays: as a special case of event, since dates may change from year to year or between countries and the holiday itself may only apply to certain locations, we could extract this data from the timeanddate.com Holidays API.

The WikiData event ingestion performs a daily SPARQL query for events of one of the defined set of types which start up to 50 days from the day of the query. A sample mapping from our semantic description model (see D2.1) to the internal document model for the metadata repository - an Elasticsearch (www.elastic.co) index hosted and managed by webLyzard - is given here, in the case of an event extracted from WikiData:

```
 WIKIDATA_MAPPING = {
    "label": "http://www.w3.org/2000/01/rdf-schema#label",
    "altLabel": "http://www.w3.org/2004/02/skos/core#altLabel",
    "description": "http://schema.org/description",
    "location": "http://www.wikidata.org/prop/direct/wdt:P276",
    "country": "http://www.wikidata.org/prop/direct/wdt:P17",
    "pointInTime": "http://www.wikidata.org/prop/direct/wdt:P585",
    "startDate": "http://www.wikidata.org/prop/direct/wdt:P580",
    "endDate": "http://www.wikidata.org/prop/direct/wdt:P582",
    "coord": "http://www.wikidata.org/prop/direct/wdt:P625",
    "frequency": "http://www.wikidata.org/prop/direct/wdt:P2257",
    "entityType": "EventEntity",
    "key": "$uri",
    "dmDate": "http://weblyzard.com/skb/property/dmDate",
    "year": "http://weblyzard.com/skb/property/year"
}
```

An example event instance extracted from WikiData and serialised according to the above mapping from our description model (common RDF-based representation of events independent of source) to the document model (using JSON serialisation) would look like this (for a 2018 FIFA World Cup soccer match):


```
"http://www.wikidata.org/entity/Q46720461": {
    "https://www.wikidata.org/wiki/Property:P585":"2018-07-02T00:00:00Z",
    "provenance":"https://query.wikidata.org/sparql#event_search",

"https://www.wikidata.org/wiki/Property:P1346":"http://www.wikidata.org/entity
/Q43122121",
    "https://www.wikidata.org/wiki/Property:P1923":[
        "http://www.wikidata.org/entity/Q43249900",
        "http://www.wikidata.org/entity/Q43122121"
    ],
    "entityType":"EventEntity",
    "https://www.wikidata.org/wiki/Property:P625":[
        "Point(39.741666666 47.208333333)"
    ],

"https://www.wikidata.org/wiki/Property:P361":"http://www.wikidata.org/entity/
Q43214603",
```

```
"https://www.wikidata.org/wiki/Property:P641":"http://www.wikidata.org/entity/
Q2736",
    "https://www.wikidata.org/wiki/Property:P276":[
        "http://www.wikidata.org/entity/Q4439101"
    ],
    "http://weblyzard.com/skb/property/mdDate":"07-02",
    "http://schema.org/description":[
        "2018 FIFA World Cup round of 16 match@en"
    ],
    "http://weblyzard.com/skb/property/year":2018,
    "https://www.wikidata.org/wiki/Property:P17":[
        "http://www.wikidata.org/entity/Q159"
    ],
    "http://www.w3.org/2000/01/rdf-schema#label":[
        "Belgium 3-2 Japan@en"
    ],
    "https://www.wikidata.org/wiki/Property:P664":[
        "http://www.wikidata.org/entity/Q253414"
    ],
    "https://www.wikidata.org/wiki/Property:P580":"2018-07-02T00:00:00Z"
}
```


Our public holiday Event collection needed only a one-time processing of public holidays (restricted to those which are observed in European countries) and includes a calculation of annual recurring dates from 2000 up to 2099. A Python script is used with Wikipedia and timeanddate.com as sources for holidays. Here is a sample output:


```
{
"EntityType":"EventEntity",
"mirror_date":"2019-01-07",
"http://weblyzard.com/skb/property/country":[
  "http://sws.geonames.org/3077311/",
  "http://sws.geonames.org/2510769/",
  "http://sws.geonames.org/2802361/",
  "http://sws.geonames.org/2750405/",
  "http://sws.geonames.org/3017382/"
],
"http://weblyzard.com/skb/property/mdDate":"5-16",
"http://purl.org/dc/terms/type":[
  "http://weblyzard.com/skb/events/holiday",
  "customary_observance_CZ",
  "customary_observance_ES",
  "national_holiday_NL",
  "customary_observance_FR",
  "religious holiday:Christian",
  "national_holiday_BE"
],
"http://www.w3.org/2000/01/rdf-schema#label":[
  "Whit Sunday@en"
],
```

```
"http://purl.org/dc/terms/date":"2027-05-16",
"http://weblyzard.com/skb/property/holidayIn":[
  "http://sws.geonames.org/2802361/",
  "http://sws.geonames.org/2750405/"
],
"http://weblyzard.com/skb/property/location":[
  "http://sws.geonames.org/3077311/",
  "http://sws.geonames.org/2510769/",
  "http://sws.geonames.org/2802361/",
  "http://sws.geonames.org/2750405/",
  "http://sws.geonames.org/3017382/"
],
"provenance":"internal:holiday_calculations",
"uri":"http://weblyzard.com/skb/holiday:Whit%20Sunday#2027-05-16",
"http://www.w3.org/2004/02/skos/core#altLabel":[

],
"http://weblyzard.com/skb/property/year":2027
}
```


The iCal event ingestion uses calendars from football seasons and therefore need only processing once a year (we have currently collected the games for the 2018/9 seasons in England, Germany, Austria and Switzerland and will shortly include the new 2019/2020 season information). For iCal, the fields DESCRIPTION + SUMMARY are used as “extracted_content”. A key - unique identifier - for each event is generated from the md5 sum of the raw event's UID, an @google ID if given (used with Google Calendar) is preserved in an owl:sameAs property. An example serialisation of an iCal event (a German Bundesliga football match) looks like this:


```
{     'EntityType': 'EventEntity',
      'http://weblyzard.com/skb/property/location': u'WWK Arena, Augsburg',
      'provenance': 'ical',
      'uri':
'https://www.weblyzard.com/skb/events/ef11171e9ca46305abd6b093d28d0335',
      'http://weblyzard.com/skb/property/temporal_start':
'2018-12-23T13:30:00',
      'http://weblyzard.com/skb/property/mdDate': '12-23',
      'http://schema.org/description': u'1. Bundesliga, 17. matchday',
      'http://purl.org/dc/terms/source':
'https://www.google.com/calendar/ical/spielplan.1.bundesliga%40gmail.com/publi
c/basic.ics',
      'http://purl.org/dc/terms/type': [
          'https://www.wikidata.org/wiki/Q16466010',
          'https://www.wikidata.org/wiki/Q82595'],
      'http://weblyzard.com/skb/property/temporal_end': '2018-12-23T15:30:00',
      'http://www.w3.org/2000/01/rdf-schema#label': u'FC Augsburg - VfL
Wolfsburg (2:3)',
      'http://weblyzard.com/skb/property/year': 2018   }
```


As of 1 August 2019, we have approximately 30 000 events in the SKB. We store the events from the three different sources distinctly in the SKB so that a user could also choose to only draw from one or the other source when using Event data. There are (at the time of writing) 736 WikiData events, 11914 iCal events and 17438 holidays (NB. each year's occurrence of a holiday is a separate entity so there are ca. 175 distinct holidays in one calendar year). We have extended the event extraction query to look up to 180 days into the future & added new entity types as we discover them (e.g. we identified the Eurovision Song Content to be a significant event in audience trends and found that the WikiData entity representing each year's content is typed as 'wd:Q276' : 'Eurovision Song Contest'). Since event metadata may be completed or corrected as we come closer to the occurrence of the event, we implemented an update mechanism that checks if an event returned by the query is already in the SKB, if metadata has changed and if so, adds a new triple to the event entity with the changed information and a provenance marker. Typically, RDF triples (subject-property-value) are extended to quads or reified (referenced in another triple) to add provenance information. We extend triples with both provenance source and last_modified, so that an entity may have multiple subject-property values according to the provenance (e.g. differing values from two sources, or a value changed over time). As an example, consider the football league schedules, where some matches are provisionally scheduled (as teams reaching later stage cup games or international competitions need sufficient rest time before league games) and only confirmed later. The initial triple extracted from the iCal data gives the confirmation_status of the match as 'unconfirmed', stored in the SKB thus (using JSON serialisation format):

```
{
  "lang": null,
  "provenance": "ical",
  "name": "skbprop:confirmation_status",
  "value": "unconfirmed",
  "last_modified": "2019-07-04T11:50:55.187949"
}
```

Later, the scheduling of the game is confirmed by the league and the confirmation_status is updated to 'confirmed':

```
{
  "lang": null,
  "provenance": "ical",
  "name": "skbprop:confirmation_status",
  "value": "confirmed",
  "last_modified": "2019-07-06T11:47:59.637972"
}
```

Therefore older data is not deleted but can be disregarded in a search by defaulting to only triples with the most recent provenance (last_modified). So we continue each day to add new events and update existing events in the SKB.

The event description model has been assessed with respect to the use of events in audience prediction together with GENISTAT. We identified and corrected a number of issues:

- Event categorization. While event typing is very rich, the first prediction experiments focused on broader categories of TV content associated with audience trends (News, Sports, Weather etc.). To ease the direct correlation of TV audiences to events, we added a broader categorization property to our events (‘`skbprop:eventCategory`’).

- Event locations. Location properties of WikiData events (values of wd:P276, or wd:P1427 and wd:P1444 - start and destination points) vary greatly in granularity (e.g. referencing a stadium, village, city or region), whereas the more general country property (value of wd:P17) is not always present. To test if geographical coverage is relevant for an event to affect TV audiences, the country level may be sufficient and is simpler to train a prediction model with (cf. Section 4). Where the event's country is missing and the country may be inferred from a given location, we complete the event metadata with the additional country property.

- Event classification. Direct classes of WikiData events can be rather arbitrary in reality, with entities being typed by classes that are very specific to them and that class is first subclassed from a more general type that we query for such as ‘sports competition’. For example, a specific badminton event may be typed first as ‘badminton competition’ which itself is defined as a subclass of ‘sports competition’. To avoid an uncontrolled expansion of the types we must search for and given the difficulty of users creating new classes just to specifically type a newly added event, we use the transitive nature of the property wd:P31 (instance of) in our SPARQL query to collect events of the given types we search for even when they are not directly typed as such; we can capture events whose type is a subclass of any of our given types.

Since the Event data should be available to other services (to drive prediction), we have developed an Event API to abstract from formulating and performing SPARQL queries over Apache Fuseki to get data from the SKB. It supports queries for Event entities whose occurrence is within a given timespan and with additional, optional, filters to match on specific properties. The query format follows a basic form of ElasticSearch Query DSL where the queries are serialised in the lightweight JSON format and express conditions on entity properties in the form ‘`property_name : condition`’. For example, to query for all events between the 1st and 15th of March 2019:

```
{"from_date": "2019-03-01", "to_date": "2019-03-15"}
```

Additional filters can be expressed on the events returned according to the time span, such as they have a label which mentions ‘final’:

```
{"filters": [["rdfs:label", "final", "text", "contains"]]}
```

Date ranges and filters can be combined as desired to return the relevant events, e.g. all association football matches (the entity type) in March 2019 whose iCal description contains ‘premier league’ (i.e. English league games):

```
{"from_date":"2019-03-01", "to_date": "2019-03-31", "filters": [["dct:type",
"https://www.wikidata.org/wiki/Q16466010", null, "term"],
["schema:description", "premier league", "text", "contains"]]}
```

We improve the Event API according to user feedback, e.g. the response includes the total number of results at the beginning so that results can be more easily programmatically managed in a loop. A new version of the API will default to returning, for subject-property pairs which occur multiple times due to updates, the most recent in provenance. It will also include an 'anniversary' search which we have already implemented internally, which takes a date (day-month pair) and returns entities of all types in the SKB which celebrate an anniversary on that date (this covers anniversaries of persons births and deaths, organisations founding dates as well as recurrent events). We will return to the anniversary search in Section 5.

Event metadata can be inconsistent regarding precise detailing of the temporal scope of the event. For example, some WikiData events were found to be dated only to the year in which they occur (e.g. '2019'). Temporal annotation can be applied to unstructured text in documents in order to identify more specific temporal references in association with events. An experiment where we extracted from 100 news articles the (a) title, (b) subject, predicate and optionally object of the title sentence, (c) persons, organisations (both agents) and locations as detected by our NER tool *Recognyze* in the text, plus (d) temporal references mentioned in the text suggested this approach would not be accurate enough for an automated extraction of events to add to the SKB. However, we parsed the documents in which we could identify a specific date through the temporal annotation and found that by aggregating the keywords annotated to that set of documents, we could identify terms that would be more relevant on that date. For example, taking English language news media from 1 Jan to 15 May 2019, where mentions of the date 31-10-2019 were identified, we found that the list of associations (document keywords weighted by tf-idf) mention: EU, Britain, Brexit and extension as terms related to this date, as well as Halloween (see Fig. 1 below left). Sampling other, less obvious, dates, found that while not every date would show clear trends towards related terms on that date, there were other cases that indicated relevant events that we would not otherwise have found - e.g. from the same document set using 19-07-2019 as the seed date. We have uncovered, based on this approach, the date on which the live action remake of the Lion King movie would be launched in cinemas (see Fig. 1 below right). This also indicates that, besides specific Event (entities) known to occur in the future (and retrievable from our SKB through the Event API) and event anniversaries (which can also relate to Person and Organisation entities), we can also use top keyword associations for a future date as another feature for training a prediction model (see Section 5).

ASSOCIATIONS	
eu	749
britain	529
brexit	840
halloween	229
extension	474
mrs may	155
brexit party	129
parliament	522
theresa may	506
tusk	180
prime minister	585

trailer	42
lion king	49
lion	50
live-action	40
remake	26

Figure 1. Keyword associations for the dates of 31-10-2019 (left) and 19-07-2019 (right)

3 Content-Based Success Metrics

While we can still refer to audience size as a measure of success for any broadcast channel or Web stream (when content is still linear, e.g. the live Web stream of broadcast TV), the new non-linear channels need new metrics to capture the "success" of a publication. Considering Websites alongside social media, where the actual forms of measurement will vary (e.g. page visits, video views, engagements with a posting), and direct comparison is ineffective (1 000 tweet impressions is good for a channel with 1 500 subscribers, less so if they have 1 million suebscribers), we have developed an approach to normalize the value of content 'success' across the vectors by source (Section 3.1). Even given the same source, publishing on a certain vector at a certain time, we know that the publications' content and style affect its success in terms of having an impact on the public debate and engaging the target audience. We therefore measure success by topic (in our case, the combinations of keywords and entities annotated to every Web or social media document). Publication success for a topic is measured in terms of frequency of mentions, share of voice (compared to other topics and considering daily fluctuations in the total volume of postings), target sentiment (Section 3.2) and the desired and undesired associations captured by the WYSDOM metric (Section 3.3).

3.1 Reach Normalization Across Vectors

WLT developed an algorithm to compute and normalize per-source reach values for Web sites (based on ingested Alexa traffic statistics) and various social media platforms (based on the number of followers and likes derived from the platforms' APIs). Since the distribution of all audience sizes was negatively skewed (i.e., the mean of values was lower than the median, reflecting that there are many social media accounts with quite a small audience), logarithmic transformation was chosen. We obtained the number of channels whose audience numbers were more than six times higher than the mean to manually set the upper border of transformation. That gave us an opportunity to control distribution of reach metric despite the constantly changing outer conditions. The result distribution of reach metrics for three channels (Youtube, Twitter and news media) is shown on the following histogram (Fig. 2).

The *Source Table* and *Source Map* of the TVP Visual Dashboard were updated accordingly (Fig. 3), using impact as the primary sorting criterion. In a first version (March 2019), this was based on re-ranking the Top 100 results. The latest version (June 2019) is already able to compute global rankings in real time (minor deviations are theoretically possible due to the sharded indexing structure of Elasticsearch, but the actual impact on shown results is negligible).

While the logarithmic transformation described above is more robust vis-a-vis outliers, a potential disadvantage is the under-representation of the true impact of very influential vectors, for example large national broadcasters. Possible future extensions could give users control over the impact calculation, allowing them to specify the desired relative importance of the number of mentions versus the reach of a source.

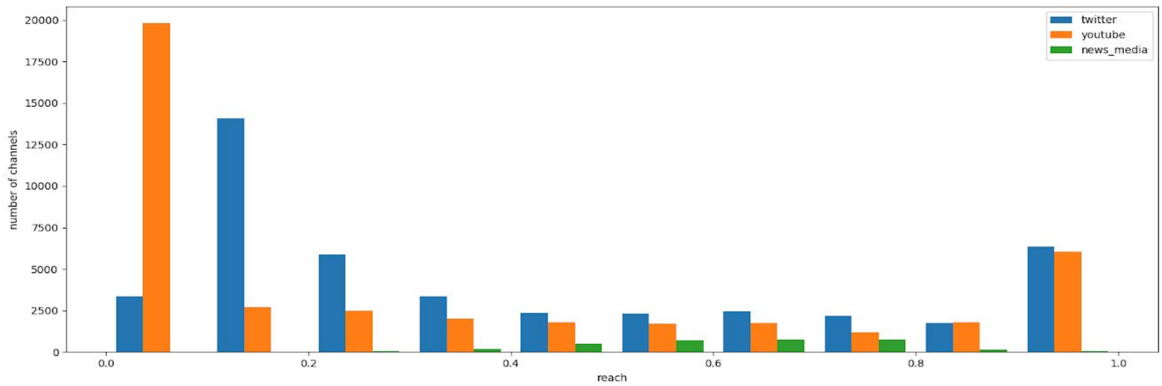


Figure 2: Histogram of cross-vector normalised reach

Description	Count	Reach	Impact ▾	Sentiment
rbb24.de aok · parks · andrea nahles	10	0.8	8	0
rheinpfalz.de mythology · german language · daenerys	11	0.6	7	+0.18
focus.de big · contender · commercial	7	0.9	6	−0.16
morgenpost.de al-quds day · conquest · commitee	7	0.8	6	−0.26
augsburger-allgemeine.de co-alition · andrea nahles · anti-semitism	5	0.8	4	+0.16

Figure 3: Source table sorted by global impact (number of mentions multiplied by the reach of a source), including per-source keywords and average sentiment

3.2 Sentiment Analysis

Multi-faceted sentiment analysis engines require a set of interlinked components on the syntactics (e.g. text normalization or POS tagging), semantics (e.g. concept and topic extraction, or named entities) and pragmatics (e.g. aspect extraction, or polarity detection) layers (Cambria et al., 2017). While the techniques covering the syntactics layer are widely available through a new generation of NLP frameworks (Young et al., 2018), the semantics and pragmatics layers are particularly challenging. The sentiment analysis engine adopted and extended as part of ReTV covers several of these layers, including aspect-based sentiment analysis (Weichselbraun et al., 2017), named entity linking (Weichselbraun et al., 2019) and an NLP annotation pipeline that includes topic and concept extraction (Scharl et al., 2017).

The development during the last year focused on integrating the engine with the Semantic Knowledge Base (SKB) (cf. ReTV D1.2), enabling advanced n-gram (contiguous sequence of n items in a sample of text, in this case multiple words of a compound term) and surface form processing and an improved negation detection. Additional work, as outlined in the following, focused on a revised n-gram processing pipeline, the analysis of unicode emojis as sentiment triggers and the consideration of a document's title in the sentiment aggregation. Arbitrary spans for the scope of a sentiment (or a negation) provide a more fine-grained analysis for several widgets of the ReTV dashboard.

The SKB has been developed by MODUL as an in-house knowledge graph to provide fast access to high-quality semantic information. This helps link and explain the various semantic terms that were spread through the visualizations. In order to build the SKB, lexemes were collected from multiple resources (e.g., OmegaWiki, Wikidata, etc). Duplicate resources were removed and properties were consolidated. This helped ground a number of our components, including sentiment analysis. The SKB integration will continue as part of WP4, as the various lexicons used in the process need to be regularly updated.

During the last development cycle, a special focus was given to the treatment of n-grams. Multi-word terms are now treated as single words for the purposes of calculating sentiment and negation (e.g., entity names will often span multiple terms, therefore now it is possible to consider all of these terms as a single entity). This was initially developed to block negation by non-negative constructions like ‘not’ (e.g., ‘not only’ has no negative connotation). Such constructions are extended to other cases. Words occurring as part of a surface of an annotated entity are now by default ignored during sentiment annotation.

Due to the emergence of short texts during the last decade, smileys and emojis have become an important part of conversation, especially on social media platforms such as Twitter (Novak et al, 2015). The latest version of the ReTV sentiment engine adds this feature and updates it to current emoji lexicons. Adding emoji sentiment allows us to compute sentiment for languages that are otherwise unsupported. While emojis might have different meaning across languages - e.g., see how irony is often used in English, a phenomena often called mock politeness (Taylor et al, 2015), we can consider this meaning to remain stable for languages from the same geographical region or continent (Novak et al, 2015).

The focus on multilingual content processing and ability to define cross-lingual queries using background translation services are an important aspect ReTV’s engine improvements. Manual inspections of sentiment values attached to the same senses in different languages yielded new insights and allowed us to optimize sentiment lexicon items stored in the SKB. Support for sentiment annotations in Dutch also has been added to the latest version. Since this is a new feature for a previously unsupported language, further optimization and an evaluation are planned as part of the integration work in WP4.

3.3 Trend Analysis and WYSDOM Metric

Visual depictions of time series data are important to convey content-based success metrics and their evolution over time. Within ReTV, this includes the trend charts (i.e., color-coded line charts for success metrics such as frequency of mention, share of voice and sentiment) as well as a stacked bar chart to represent the hybrid WYSDOM success metric, which computes the degree of association of a search term (brand name of a broadcaster, TV series, etc.) with sets of pre-defined desired and undesired topics.

In terms of temporal granularity, we developed a new line chart mode that calculates the temporal distribution based on hourly instead of daily data points (see Figure 4), including an adaptive recommendation of specific granularity settings based on the selected time interval for the analysis. This also included a more flexible way to define moving averages, for both the trend chart and the WYSDOM chart. Section 5 outlines how such time series data is used as a basis for predicting future content popularity.

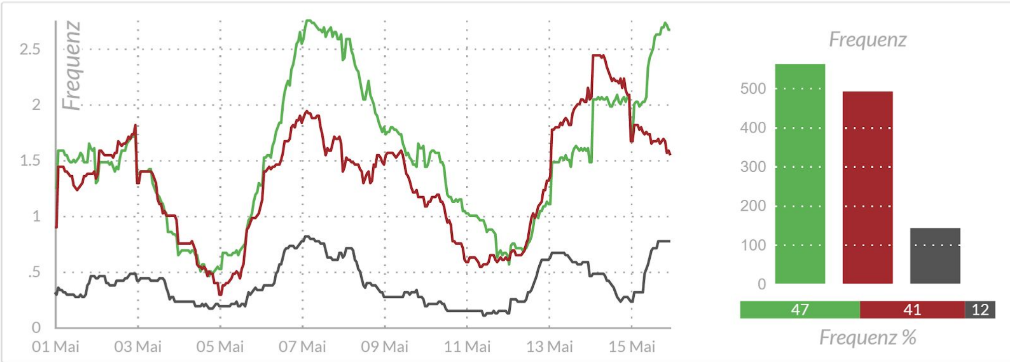


Figure 4: Line chart with hourly data points based on 1150 German *"Game of Thrones"* references in the first two weeks of May, distinguishing positive (green), negative (red) and neutral (grey) coverage

Just one global setting would not suffice to meet the flexibility requirements of ReTV's use cases. Given the number of programs and vectors, fine-grained customizability was another major goal of the development. In addition to the global setting, we therefore developed a new data model and update mechanism to store a specific WYSDOM configuration together with the bookmark definition. In conjunction with an extended advanced search module, the per-topic configuration of WYSDOM allows the tracking of user intentions (e.g. to watch a TV program or purchase a product, which would be a specific desired association). For the 06-2019 dashboard release, this new function was used to define desired and undesired terms for (a) broadcasters in general and (b) use case partner RBB in particular (cf. Figure 5).

The final step to be addressed in WP4 will be the integration of specific audience metric alignments into the per-topic WYSDOM configuration to enable users to explore correlations between the number of viewers and the public online debate.

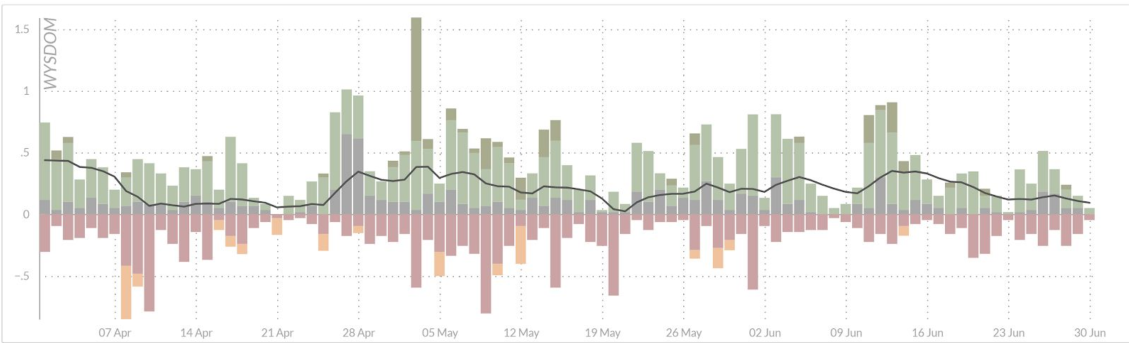


Figure 5: WYSDOM chart based on 3250 mentions of "RBB" in news and social media between April and June 2019, showing both sentiment (red, grey, green) as well as the association with desired and undesired terms (darker green, orange)

In terms of scalability, cross-lingual similarity computations and real-time negation processing within WYSDOM (as compared to the pre-computed sentiment annotations, which already benefit from the improved negation detection described in Section 3.2) there are computational challenges that remain and that will be addressed as part of the system integration in WP4.

4 Audience and Viewer Metrics

Details of the audience metrics provided via Genistat to the metadata repository were given in deliverable D2.1. We extended the audience metrics we push into the metadata repository with the unique ID of the program that was broadcast on that particular channel for that audience. This allows for easy aggregation of viewership numbers by program.

In the following figure we display a sample data point. This data is extracted from Zattoo (an OTT TV provider in Switzerland and other European countries) by Genistat and sent to the webLyzard platform via their statistical API. When we can associate the audience data point to a TV program in our EPG data, we add the identifier ("uri" field) which is also used in the EPG data feed to the metadata repository (cf. Fig. 6); as a next step webLyzard will be able to combine audience metrics and TV program information in the TVP Visual Dashboard.


```
{
        "uri": "http://api.bee.genistat.ch/program/zattoo/158",
        "added_date": "2018-09-10T15:01:48.623816",
        "date": "2019-05-22T13:30:00",
        "indicator_id": "retv_audience_live",
        "indicator_name": "ReTV ETV live audience data",
        "value": "12110",
        "source_country": "CH",
        "source_type": "country",
        "source_location": {
            "name": "Switzerland",
            "point": {
                "lat": 46.8182,
                "lon": 8.2275
            }
        },
        "meta_data": {
            "temporal_start": 153844530,
            "temporal_end": 153844500,
            "media_brand": "The Simpsons",
            "media_broadcaster": "ProSieben"
        }
}
```


Figure 6: JSON of one audience metric reading

We analyzed if the type of TV content being broadcast has an effect on audience numbers, we used two sources of EPG metadata: (a) The first source contains an enhanced categorization of the programs (in particular, including different sport disciplines) and the start and end times are more accurate; (b) the second source contains a basic categorization (News, Documentary, TV Series, Entertainment, Kids, Movies, Sport) of the programs and the start and end times are approximate to about 5 minutes. Comparing results with both content feature sources allows us to verify whether (a) the model is flexible enough to use different kinds of attributes (b) how the information granularity affects the model quality. We took the past 5 months of audience data and matched it to the corresponding EPG data. We categorized the EPG data into five categories: sports (green), news (yellow), movies/TV series (blue), ads/promos (red) and other (black). Audiences numbers (dashed line) were smoothed to medians aggregated over channel, hourly and weekly seasonal variations. A sample plot of audience by TV content category is shown in Fig. 7.

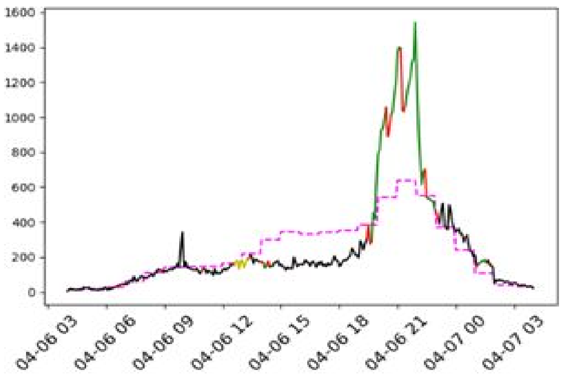


Figure 7: Plot of TV channel audience over 24 hours, colour coded by TV content category

Analysing the plots for all channels, we found that sport is related to most of the anomalies in audience figures. News is much less important. Longer ad breaks do lead to some audience erosion but it is also temporary - the viewer returns to the same channel. Channels that do not broadcast sports have very stable audience shapes for most of the time. Even the day-of-week (i.e. weekly) seasonality is not that important, just daily seasonality. The same holds for non-sport days on the other channels. This implies that the "typical" TV channel audience and its seasonality is enough to predict in many cases, without additional features. However, where a channel broadcasts a future content item which will cause an 'anomaly' in audience figures, as seen with live sports events, this would generate an out-of-trend variation. So we decided to consider TV content categories as a feature (categorizing EPG data for the next 24 hours of broadcast TV) to our prediction model to test if this improves prediction (Section 5).

MOD performed an initial analysis of whether external events such as those captured in our SKB as a result of the event extraction (Section 2) led to changes in TV audience trends. We took the audience data from Feb 16 to Oct 2, 2018 for several German and Swiss TV channels and chose several top channels from both countries: ARD, ZDF and PRO7 (in Germany) and SRF1 and SRF2 (in Switzerland). We used Anomaly Detection in SPSS. The initial threshold of three standard deviations from the mean (z-score = 3) was too discriminatory and we settled on z-score = 2 for extracting anomalies in the data. This returned 25 data points in ZDF audience data instead of 4, for example (Fig. 8).

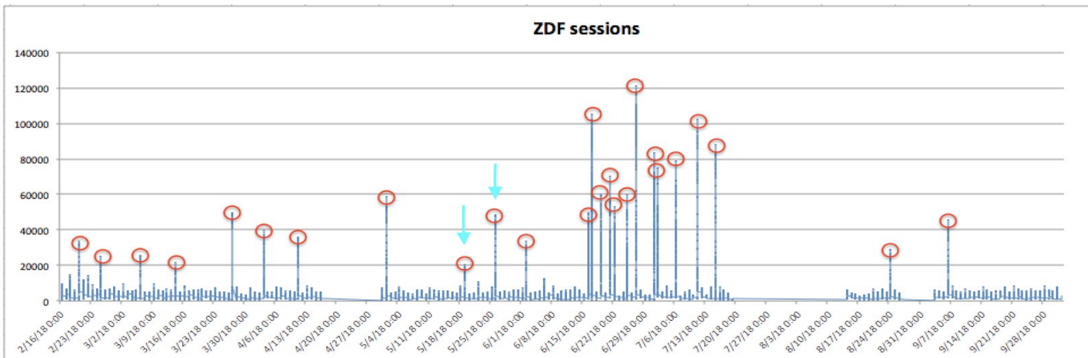


Figure 8. Outliers in the ZDF audience data. Highlighted by arrows as examples are the DFB Cup Final (19 May 2018) and the UEFA Champions League final (26 May 2018).

In Table 1, we summarize the results of looking at each anomaly for each channel and manually determining if they relate to (a) a TV specific event (like a series finale), (b) an external event broadcast on that channel (like live sports coverage), or (c) not explained. It can be seen that no anomaly was unexplainable. Only in the PRO7 case the anomalies occurred due to a TV-specific event, in fact they were the weekly broadcasts of “Germany’s Next Top Model” which attracted a much higher audience that any other programming on that channel. The weekly repetition of these outliers could be used to learn that this is related more to the schedule of TV programming than to external events (which do not occur as regularly). For all other channels, we could explain all of the anomalies by events that occurred at that time and were broadcast on that channel, indicating both that outliers in audience data can be meaningful for prediction and that they need identification with events for prediction model learning.

Channel	Total # Anomalies	TV-only event	External event	Not explained
ZDF	25	0	25	0
ARD	18	0	18	0
PRO7	12	12	0	0
SRF1	1	0	1	0
SRF2	6	0	6	0

Table 1: Identification of relationship between Events and “Anomalies” in TV audience data

We also looked at the types of events associated with the anomalies. The vast majority were sports (most obviously, many FIFA World Cup games). In Germany only the Royal Wedding (Prince Harry and Meghan Merkle) and Eurovision Song Contest were able to generate a similar spike in audience. In Switzerland, the SRF1 anomaly related to a Spring celebration parade in Zürich being broadcast, whereas all SRF2 anomalies were sports-related. Geographical location of the channel is also determinant of which events may cause anomalies, since all SRF anomalies (except one) related to events specifically involving Switzerland. We did not observe significant drops in audience on other channels at the same time, nor did we observe overall increases or decreases in audience across all channels that could be related to an event (e.g. a public holiday). We also checked the event coverage in the SKB of the events found to cause the anomalies. The SKB, via the WikiData and iCal sources, did contain the Champions League, DFB Cup and Bundesliga games but missed international games, including friendlies and the Nations League. WikiData tends to cover well sports finals, the FIFA World Cup games (after the group stage) and the few other non-sports events which were relevant (Eurovision Song Contest, royal wedding). So our main focus in the event-based prediction will be on learning about past events’ effects on TV audience and using this to predict TV audiences during future events (Section 5).

5 A Predictive Model For Content Publication

5.1 Introduction to Prediction

TV channels and other digital content distributors want to optimise the success of the content they publish, especially as content distribution shifts to non-linear (away from broadcast and into IP based) and those non-linear channels offer an incomprehensibly large choice of content to consumers. This is a significant difference from the linear TV audience: as we could observe in our Zattoo audience analysis (Section 4), there is a strong underlying trend related to channel and time which is more strongly associated with 'core' viewing activity (e.g. that broadcast TV is shown as background at home or in public buildings and the viewers tend to switch it on and off at the same times regardless of the TV schedule) than with the actual features of the content currently being shown (with only a rather small subset of content being significant enough to generate an outlier in the audience data). This new world of non-linear content consumption that is always-there and on-demand, and means that audiences are much more selective in what they view, when and on what channel. This requires greater understanding of the content offer on these channels (both from them and their competitors), the comparative success of that content and the factors which determine those differences.

In ReTV, the collection of past data about events, content success metrics and TV audience is the basis for projecting this data into the future, making predictions about the success of TV related content on different vectors at a future time based on the contribution of the content-based features to its comparative success. Features in this case refer to the annotation of the content (by keywords or entities) and its relationship to any external events.

Prediction has long been a task done by statistical analysis, starting with basic extrapolation of a historical trend into the future. As applied to time-series data, it is often called forecasting. ARIMA is a standard method for time-series forecasting, combining autoregression ("AR" in ARIMA), moving averages ("MA" in ARIMA) and integration (the "I" in ARIMA) of time-adjacent values by differencing each value by the previous value. The goal of a forecasting model is to decompose the time-series data into different parts, the three most usual being 'trend' (a regular change in the values at some small or medium size granularity), 'seasonality' (regular changes over larger granularities of the time component, e.g. annually) and 'residual' (values which are irregular to the detected trends and seasonality). When learning for the future, it has been recognized that there can exist other variables which also affect the future values of the data apart from those expressed by an initial forecasting model, explained without means to consider them explicitly within the 'residual' component of the observed data. An extension to the ARIMA model considers one or more of these external, irregular variables, known as "exogeneous variables" and referenced by the X in the ARIMA<u>X</u> model as it is now called. In ReTV, our primary focus is on these exogeneous variables, since they are what is still missing in classical TV audience and online content success prediction models. For example, we could show that most outliers in audience data could be explained by (previously knowable) events.

In recent times, implementations of predictive analytics have started using Machine Learning (ML) approaches, part of a general trend towards solution building by Artificial Intelligence (AI). ML / AI tend to work better with larger data sets and can result in being more computationally intensive, however the reported significant improvements in accuracy using such techniques has made them the de facto default choices for prediction experiments today. While the algorithms work fundamentally on numerical data, the use of semantic knowledge and other categorical data types in machine learning has been supported by recent research on how to integrate such non-numerical data (such as events or content descriptions) into an ML model. In our context, we consider, for any time-based content which could be the subject of a prediction task (i.e. "what is the best content to publish at what time on which vector?"), the following features :

- events occurring at some time related to this content
- keywords and entities annotated to this content

In line with the content publication scenarios (WP5), we also consider that there are several different prediction tasks to consider for media organisations in reTV:

- for a TV program, optimise audience by publication time on a linear vector
- for a Web or social media publication, optimise reach by publication time on a non-linear vector (in the short term, best for statistical prediction)
- for a Web/social media publication, optimise reach by publication time on a non-linear vector (in the medium to long term, needs a non-statistical prediction method)

We will describe below the individual experiments for each of these cases, present initial results and outline the next steps for a hybrid prediction model with content features.

5.2 Forecasting TV Audiences Based on TV Content and Events

Forecasting of TV audiences by GENISTAT was based on the random forest ensemble model. We also tested gradient boosting (xgboost library; xgboost.readthedocs.io), that usually produces more accurate results when compared in the scientific literature to other machine learning-based forecasting methods. In our case it was the opposite - boosting is well-known to be prone to overfitting to outliers in the data, and TV audience metrics have significant unexplained noise in the time-series data. On the other end, the random forest model (since it is based on averaging of the simple decision trees) is agnostic to outliers (at the cost of biasing the predictions towards the average values, i.e. underestimating high values and overestimating low values, in the case of the audience numbers that takes only positive values).

The target of our forecasting was the predicted total audience for each program. We do this by predicting the number of users watching a given TV channel at a given time, aligned to the TV channel schedule information (EPG). We consider how additional features in the model related to the TV content and events could affect the accuracy of our predictions.

Models are fitted separately for each channel. For the moment, we focused only on the 7 channels of Swiss public TV (analogously to experiments with events features in the context of recommendation, reported in D3.2). Some of these channels broadcast sports, some others not. Event-based features were created analogously to the recommendation experiments. In particular, we have features such as:

- event stage
- players countries
- players names
- sport discipline / content category
- derived features such as the number of Swiss players (for Tennis)

It should be noted that event-based features currently revolve around sports content only. This follows the observation discussed in Chapter 4, that sport is responsible for the majority of anomalies in the audience numbers seasonalities.

Another important class of features are the content-based features. The possibilities here are numerous (including the use of visual features extraction), we started with two basic features. First one is the general content category (e.g. movie, TV series, kids program, ad or promo, cultural programming, entertainment, sport). The second one is the detailed category (for sports, it is a sport discipline, for movies and series - genre etc.).

In addition to the sport event-related features and the content-based features, we have general "time-series" features, that are used in the full model (extended with additional features), as well as in the baseline model (no additional features):

- average and median audience for each hour of day (daily seasonality)
- average and median audience for each day of the week (weekly seasonality)
- preceding program audience (time-series lag) - more on this below

One significant change that we introduced in comparison to the standard forecasting approach is the following. In the standard forecasting task, one usually trains the model on the regular time-series (e.g. TV audience numbers modeled in K-minute intervals). Below, we modeled TV program audiences directly. Each observation in our training data represents a single TV program. So precisely speaking, it is not a time-series model, but still the seasonalities and autocorrelations are present in the data (as the future importance rankings, presented below, prove). Such an approach is better suited for the ReTV use cases, where our entity of interest is the TV content, rather than a time slot.

Moreover, if we remove the "lag"/autocorrelation features such as "preceding program audience" (which is important, but not the most important feature in the current models), we can use such a prediction model to find an optimal publication time for some TV content (on the linear channel, e.g. Web stream). In order to achieve such an optimization, we score a given piece of content (and its corresponding content- and event-based features) for all publication time candidates and take the one that maximizes the audience (other targets are also possible, e.g. we can optimize not for the total/general audience, but for some segment of audience, e.g. people from a given location).

In the following subsection we present and discuss major results comparing the baseline forecasting models (not exploiting event-based information) with the proposed new model.

5.2.1 Results: Models Accuracy for Audience Prediction

For the accuracy evaluation, we created four types of models per TV channel:

- **lag / events** - the model that includes both event-based features and the "lag" feature (prev_program_users), reflecting the autocorrelation aspect of time series data
- **lag / no_events** - the model that includes the "lag" feature, but no event features
- **no_lag / events** - model without lag feature but with event-based features
- **no_lag / no_events** - model without lag feature and without event-based features

As the accuracy metrics, we calculated:

- **MAE** - mean absolute error, i.e. the average of the differences between the forecasted and the observed audience numbers for each program
- **Spearman** correlation - the rank correlation coefficient between the observed programs ranking (sorted by the audience numbers) and the forecasted programs ranking. This metric is important in the context of model applications such as optimization of the publication time, where we don't care about the actual predicted audience numbers but rather about finding the time slot that maximizes the expected audience.
- **Pearson** correlation - the linear correlation coefficient between the observed programs ranking (sorted by the audience numbers) and the forecasted programs ranking

In Tables 2 and 3 below, we present results for the two channels, SRF-info - that regularly broadcasts sport programs, and SRF-1 - that never broadcasts sports. Accuracy metrics are calculated for all of the programs ("ALL" row), as well as per content category. For each category, we give the total number of programs in this category in the "cnt" column.

SRF-info		**lag / events**			**lag / no_events**			**no_lag / events**			**no_lag / no_events**		
category	**cnt**	**MAE**	**Spearman**	**Pearson**	**MAE**	**Spearman**	**Pearson**	**MAE**	**Spearman**	**Pearson**	**MAE**	**Spearman**	**Pearson**
info	996	24.33	0.99	0.71	17.83	0.99	0.8	12.99	0.99	0.94	26.37	0.99	0.67
film	0	-	-	-	-	-	-	-	-	-	-	-	-
sport	**86**	**6.07**	**0.99**	**1**	**7.45**	**0.99**	**0.99**	**6.45**	**0.99**	**0.99**	**27.58**	**0.98**	**0.86**
ads	7134	15.13	0.99	0.87	15.02	0.99	0.87	17.77	0.99	0.8	16.77	0.99	0.87
other	3359	13.95	0.99	0.88	16.49	0.99	0.81	16.53	0.99	0.89	13.99	0.99	0.92
ALL	11576	15.51	0.99	0.84	15.63	0.99	0.84	16.91	0.99	0.83	16.87	0.99	0.84

Table 2. Prediction accuracy results for SRF-info

SRF-1		lag / events			lag / no_events			no_lag / events			no_lag / no_events		
category	cnt	MAE	Spearman	Pearson	MAE	Spearman	Pearson	MAE	Spearman	Pearson	MAE	Spearman	Pearson
info	805	19.17	0.96	0.98	21.98	0.97	0.98	24.3	0.96	0.99	34.98	0.96	0.94
film	1738	18.82	0.96	0.99	21.93	0.96	0.98	24.78	0.94	0.98	25.93	0.95	0.95
sport	0	-	-	-	-	-	-	-	-	-	-	-	-
ads	15926	25.57	0.98	0.98	25.36	0.97	0.98	31.52	0.97	0.97	30.9	0.97	0.98
other	2997	23.51	0.97	0.98	26.85	0.97	0.97	33.38	0.97	0.97	32.08	0.97	0.97
ALL	21466	24.5	0.97	0.98	25.16	0.97	0.98	30.96	0.97	0.97	30.81	0.97	0.97

Table 3. Prediction accuracy results for SRF-1

Major observations:

- for the channel that broadcast sports (SRF-info), event-based features improves the model accuracy, both in terms of the error (MAE) and the popularity ranking (Spearman)
- the positive effect is even stronger if we don't include the lag feature. Such a model is especially important in the context of the publication time optimization, where we don't want to forecast just a couple of steps into the future, but we want to find an optimal timeslot, e.g. for the upcoming week and a given TV program (with known attributes such as category, or related event features).
- on average, the lag feature inclusion improves the model accuracy. It is consistent with the common intuition about audience flows from one program to the following one. However again, in some of the applications that requires long-term forecasting (publication time optimization) we don't want to include lag feature
- for the channel that does not broadcast sports (SRF-1), inclusion or exclusion of event features has a neutral effect on the model
- for some of the categories that are not sport, inclusion of the event-based features have a neutral or sometimes a negative impact (e.g. for ads or info). Still, knowing in advance if the program for which we are forecasting is a sport program or not, we can have two models trained per each channel, and use the events or no_events model correspondingly.

5.2.2 Results: Feature Importance in Audience Prediction

We calculated a number of feature importance metrics:

- **weight**: the number of times a feature is used to split the data across all trees. It is the least indicative among the importance features, since it highly depends on the number of distinct values of a given feature (or their distribution, in the case of continuous features). Also, categorical features will usually get lower weights than continuous features.

- **gain**: the average gain across all splits the feature is used in. Gain measures the relative contribution of the corresponding feature to the model calculated by taking each feature's contribution for each tree in the model. A higher value of this metric when compared to another feature implies it is more important for generating a prediction. More precisely, it reflects the improvement in accuracy brought by a feature to the branches it is on. The idea is that before adding a new split on a feature X to the branch there were some wrongly classified elements, after adding the split on this feature, there are two new branches, and each of these branches is more accurate (in terms of the optimized criterion).
- **cover**: the average coverage across all splits the feature is used in. Cover measures the relative quantity of observations concerned by a given feature. For example, if you have 100 observations, 4 features and 3 trees, and suppose feature X is used to decide the leaf node for 10, 5, and 2 observations in tree1, tree2 and tree3 respectively; then the metric will count cover for this feature as 10+5+2 = 17 observations.
- **total_gain**: the total gain across all splits (and all trees) the feature is used in. Equal to the product of weight and gain values. In consequence, this metric is also affected by the number of distinct values of a given feature.
- **total_cover**: the total coverage across all splits (and all trees) the feature is used in. Equal to the product of weight and cover values. In consequence, this metric is also affected by the number of distinct values of a given feature.

In Tables 4 and 5 below, we present exemplary results for the two models (however the feature importance rankings for the remaining models follow the same structure). The first table presents model that incorporates event-based features, the second table presents the model without event-based features. Feature rankings are sorted descendingly by gain column.

model with event features	weight	gain	cover	total_gain	total_cover
content_epg_main_cats_9	3	57,627,356	5,855	172,882,068	17,566
event_people_Rafael Nadal	2	26,638,854	7,328	53,277,708	14,656
event_stage_Finals	5	24,150,707.90	6,038	120,753,539	30,191
event_people_Roger Federer	1	16,545,830	7,333	16,545,830	7,333
event_countries_SUI	4	13,572,314	2,222	54,289,258	8,889
event_countries_SRB	3	13,345,356	6,756	40,036,068	20,268
event_countries_GER	1	9,317,020	6	9,317,020	6
event_people_Stan_Wawrinka	4	8,681,276	1,833	34,725,105	7,333
event_stage_Viertelfinal	2	8,278,286	17	16,556,572	34
content_duration	84	8,178,525	1,949	686,996,112	163,749

Table 4. Event and content feature rankings

model w/o event features	weight	gain	cover	total_gain	total_cover
content_epg_main_cats_9	27	11,466,775	2,857	309,602,930	77,140
prev_program_users	209	10,785,052	1,335	2,254,075,910	279,190
content_duration	129	6,221,502	2,035	802,573,843	262,550
avg_start_hour_users	97	2,737,444	1,215	265,532,150	117,910
median_start_hour_users	124	1,906,699	1,041	236,430,691	129,120
content_epg_main_cats_B	54	1,580,932	846	85,370,349	45,710
start_hour	206	1,163,997	842	239,783,510	173,640
avg_start_dow_users	148	1,160,986	738	171,825,931	109,280
start_dow	52	842,717	1,434	43,821,321	74,610
median_start_dow_user	99	836,377	545	82,801,371	53,970

Table 5. Model without events feature rankings

Major observations:

- for the models with event-based features, such features are among the most important ones in terms of model accuracy gain. However, some of them (e.g. event_stage_Viertelfinal, event_country_GER) are scored lower in terms of coverage (since they affect only a small fraction of the TV programs)
- the other class of features that is important both for the models with event-based features, and the models without such features, are the content categories. In particular, content category "9" - sport - is the top ranked category for both types of models. As we noticed earlier, sport broadcasts are the most anomalous (with regards to the audience metrics seasonalities) in the forecasted time-series data.
- when it comes to event-based features, sport event localness is important. Since these particular models were built on the data for Swiss public TV, Swiss players presence positively affect the predicted audience numbers
- another important feature is the "lag" feature, reflecting the autocorrelation in the time-series data (prev_program_users)
- remaining features reflect the typical seasonal patterns of the forecasted time-series, such as daily seasonality (start_hour, avg_start_hour_users) and weekly seasonality (start_dow, avg_start_dow_users, median_start_dow_users)

5.2.3 Forecasting - Final Remarks

TV channels benefit from being able to anticipate future viewer numbers. Private channels set advertising slot pricing according to the expected number of viewers of the programming into which the advertising is inserted. Public channels need to show they can fulfil the remit for which they are publicly funded, which typically includes maximizing the audience for programming which has a social or regional purpose. Public as much as private channels would value audience forecasts when making scheduling decisions or content purchasing/production decisions, by simulating the potential audience for different choices of which content is to be broadcast at which time. Regarding prediction of TV audiences, forecasting methods are applicable since TV programming can be both seasonal (e.g. summer vs winter schedules) and viewership follows identifiable trends (e.g. weekday 'prime time' in the early evenings) [Danaher, 2011]. Our baseline audience prediction used random forest models on viewing numbers per TV channel. For training, we use data from Zattoo (an OTT TV provider in Switzerland and other European countries) that gives us the information about who watched which program on which channel and at what time (user IDs were anonymized prior to analysis and only aggregations of viewers were used in the forecasting). Real-time data points (audience at every 5 minute time point in the last hour) were used to adjust predictions to most recent trends. We found that for the Sports category - the most significant for irregularities in audience metrics - adding event-related features to our prediction model improved the accuracy. For the scenario when we don't use any lag/autocorrelation information - which is the relevant one for the publication time optimization - mean absolute error was reduced from 27.58 to 6.45 and the correlation was increased from 0.86 to 0.99. For the scenario wit lag/autocorrelation features, MAE was reduced from 7.45 to 6.07.

5.3 Forecasting of Communication Success of TV Content across Vectors

5.3.1 Approach to Communication Success Forecasting

While TV audience forecasting builds on a longer history of implementation and use, where the adoption of ML/AI and the incorporation of exogenous variables still represents a modern innovation, ReTV is interested in supporting optimal publication of content across all vectors. This means including non-linear channels such as Web and social media postings, which can not be considered to have a singular publication event with a start and an end and thus also a singularly measurable metric of success (the audience). The question becomes: (i) for this piece of content, what is the optimal time for publication on which vector?, or (ii) for this vector at this time, what is the optimal piece of content to publish?

We need to determine the metric to use to measure this optimum, considering that generally to predict possible future values of that metric we will need to train a model with the past values of the same metric as applies to the content in question. As outlined in Section 3, we can make use of the following success metrics for content as time-series data: frequency of mentions, share of voice, sentiment, WYSDOM.

Note that already in the audience prediction, we learnt not to use the specific piece of content (the TV program being broadcast) as the feature for model learning but the content's features. This is because one can not fairly assume next week's episode of The Simpsons to be the same as last week's (maybe next week there is a guest voice from a celebrity that causes more people than usual to watch), but as a feature both content items could fairly be said to share a keyword or entity 'The Simpsons'. So, as a common feature, the prediction for next weeks' episode can learn from past audiences of content with the keyword 'The Simpsons', but we can capture additional features on this future content which may also prove significant (or not) for the audience size. The same principle is applied to content being published on the Web or social media, and we can make use of the keyword and entity extraction done in the annotation of TV related content by ReTV. Therefore we consider any content under consideration in this prediction task as a set of keyword/entities, which we can extract from any concrete piece of online content or from which we can suggest a new piece of online content. We term a set of keyword/entities which can represent a piece of content a “topic”.

Our working hypothesis is that the frequency of mentions of a topic on a vector positively correlates with the reach of content about that topic (on the same vector at the same time)

Therefore, we want to predict the frequency of topics by vector and time, so that we could suggest the optimal time for publication of content about that topic on that vector. By comparing a set of topics on the same vector for a certain time, we can also suggest the optimal topic for which to choose or create content if publishing at that time.

We will test this prediction using our scenarios (WP5) as a basis, providing results for the prediction to our use case partners in their user tests (August 6-9, 2019). They have already created some working examples for their testers, where they can already inspect the trends around certain topics in the past using the Topics Compass and will be presented with the foreseen future trend of the same topics as a basis for deciding what to publish when. Those topics are: RBB - the Berlin Finals sport event, RBB - the Brandenburg local elections, NISV - the Tour de France, and NISV - Popular Topics in their video archive.

For vectors, we will consider the Web and social media as an aggregate. For time, we train with data up to and including August 5, 2019 and predict for the following four days (the days of the tests). The dependent variable is the frequency of mentions (no moving average). For each topic, we will use the metrics for the top keyword associations with that topic for the last week (so using the date range 29 July - 5 August). For baseline prediction, we will use autoregression. This machine learning approach uses time series models with observations from past time steps used to make predictions for the next time step.

To test the extent of past data necessary to make predictions, we also use 3 different training sets - one with a month plus a week of past data (38 data points), one with 13 weeks of past data (91 data points) and one with a year and a week of past data (372 data points, corresponding to 30 July 2018 to 5 August 2019). The validation set is the actual values for the last 4 days (August 2-5, 2019) with training using the remaining data (up to August 1, 2019) .

So we have the following datasets:

Name	Keywords tracked	Vectors
RBB_finals	deutschen meisterschaften, sportarten, athleten, florian wellbrock	News-DE, Social Media-DE
RBB_election	karl lauterbach, einkommen, höcke, millionen euro	News-DE, Social Media-DE
NISV_tour	bernal, ineos, froome, brailsford	News-EN+NL, Social Media-EN+NL
NISV_popular	tour de france, NOS, baan, EU	News-EN+NL, Social Media-EN+NL

We started with the RBB_election dataset as the keywords show distinct frequency behaviour: ‘karl lauterbach’ (a local politician) rarely occurred until recent days and often had zero mentions in a day, ‘höcke’ (another politician) varies between most days having a few mentions and fewer days with significant frequency (15-50) i.e. a high standard deviation, ‘einkommen’ (income) is also infrequent but regular (most days have at least one mention) and ‘millionen euro’ (million euros) has the higher average frequency but lower standard deviation.

We assess for each dataset the linear correlation (the p-value) between the variables to form a basis for using linear regression techniques. Significance of the input feature for the predictor variable is given if $p<0.05$:

input feature:	1 day earlier	2 days earlier	3 days earlier	4 days earlier
karl lauterbach	**.011**	.601	.117	.284
einkommen	.484	**.028**	.89	.201
höcke	**0**	.521	.816	.705
millionen euro	**.004**	.305	.929	**.032**

It can be seen that most often the past frequency of keywords is not a strong determinant of future frequency, i.e. the occurrence of keywords in the online documents is strongly non-deterministic and subject to strong irregularities that can not be known in advance for prediction (as is to be expected with the public discourse, which is driven by current events, trends, memes etc.). However, some past values have demonstrated sufficiently a significant effect on the future value, especially the value from exactly one day earlier. This suggests that most recent values of the success metric may be usable in predicting a coming value (without too high expectations of constant accuracy) but that this prediction can only be considered for very short periods into the future (as in reverse with the past values, we’d expect the prediction to work best for the next day’s value and then accuracy dropping sharply as we predict 2, 3 or 4 days into the future).

To have an overview of the performance of the models (on the test data once trained) we use the R^2 measure and compare model fit for the 3 historical time periods to determine which is the most reasonable training choice (the higher the R^2 value, the better the fit).

Dataset	Keyword	Training data size	Model fit (R^2)
RBB_election	karl lauterbach	372 91 38	0.164 0.133 -0.037
	einkommen	372 91 38	-0.069 -0.037 0.08

	höcke	372 91 38	0.13 0.19 0.05
	millionen euro	372 91 38	-0.12 0.01 -0.05

This produced the interesting finding that the best model fit for these keywords, given their different frequency behaviour, appears to be the intermediate training data size (91 data points) rather than a much longer time period (maybe because it is introducing more irregularity) or the shorter time period (possibly insufficient for the model to find a best fit). As already mentioned previously regarding the correlation strength, it is not surprising to find that the model fit is usually close to zero as we are seeking a baseline for predicted values and can not expect a better fit in most cases.

5.3.2 Evaluation of Communication Success Forecasting

We validate the prediction model (measured using RMSE - Root Mean Squared Error) using the next 4 days values, so for the autoregression we use a 4 day moving window in training (i.e. we train that the output value for day 5 is a factor of the input values from days 1-4). We include the model fit which generally shows that a worse model fit did not necessarily lead to less accurate predictions in our tests:

Dataset	Keyword	Model fit	RMSE
RBB_election	karl lauterbach	.133	6.80
	einkommen	-.037	5.33
	höcke	.19	6.11
	millionen euro	.01	3.11

We also evaluate the other datasets in the same way:

Dataset	Keyword	Model fit	RMSE
RBB_finals	deutschen meisterschaften	.68	39.89
	sportarten	.525	21.53
	athleten	.33	17.98
	florian wellbrock	.342	16.36

This dataset was particularly interesting as it related to a sports event taking place in Berlin for the first time, so the keywords were very sparse in earlier time periods and only began to become much more frequent very close to the event itself, so while frequencies were still increasing in the prediction period (as the event then took place precisely in the dates we were using for validation), our algorithm was already predicting that the peaks subside because the previous data has only shown very brief peaks. This led to widely varying predictions and hence the comparatively high RMSE. The results indicated the potential benefit of including event features into the prediction model since autoregression could not take into account that the event took place in the prediction period.

Turning to the NISV datasets:

Dataset	Keyword	Model fit	RMSE
NISV_tour	bernal	.654	4.21
	ineos	.72	6.73
	froome	.327	5.89
	brailsford	.451	2.04

We were expecting a higher RMSE here as the dataset is very similar to RBB_finals (Die Finals Berlin 2019, Aug 3-4 2019) in that the keywords are highly dependent on a specific sports event (Tour de France, Jul 6-28 2019). However in this case the event took place in the period covered by the training data and the frequencies peaked within the period and were already dropping to pre-event levels at the end, just before the dates we used for validation. See the screenshot below (Fig. 9):

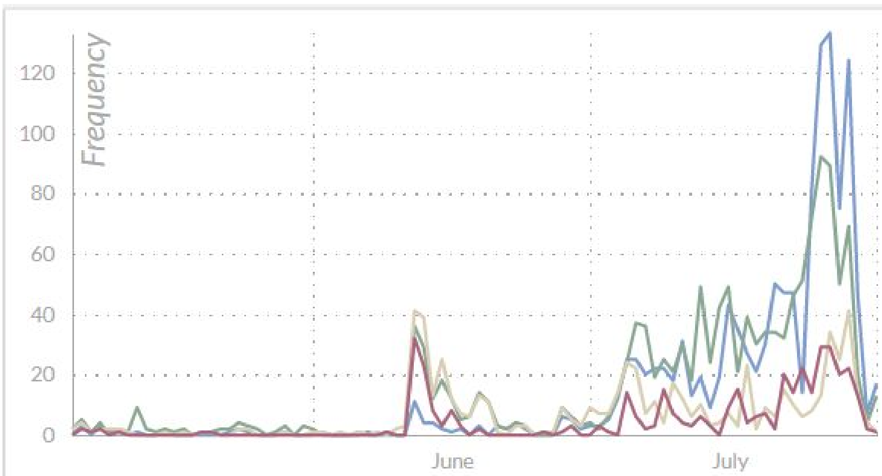


Figure 9: Keyword frequency metrics for NISV_tour (each line is one of the keywords)

So the algorithm could anticipate a continued drop in frequency for the keywords despite their recent peaks, and the prediction results appear better. The key to this was that the event was no longer taking place in the prediction period.

Dataset	Keyword	Model fit	RMSE
NISV_popular	tour de france	.79	12.44
	NOS (broadcaster)	.204	4.64
	baan (track)	.2	145.85
	EU	.391	15.74

This dataset offers again some variation in the patterns of the keyword frequencies. 'tourdefrance' is peaking in the training data and the algorithm expects to see that drop in the prediction period (as with the NISV_tour predictions). However it happens that there is another spike in frequency in the real data whereas our prediction expects the continued drop (see Fig. 10), causing the higher RMSE. The horizontal axis of Fig. 10 is days from current, so we are predicting from t to t-3 days here, i.e. chronologically we are moving from right to left. The gray line is the real values, where there is a drop but then on the last day another peak. The red line is the predicted values, where there is a sharp drop predicted in mentions. So while the t-2 and t-3 predictions are very close, we varied widely at t and t-1. Such irregularities can not be predicted in advance, and reflect the limitations of any prediction model of online success metrics.

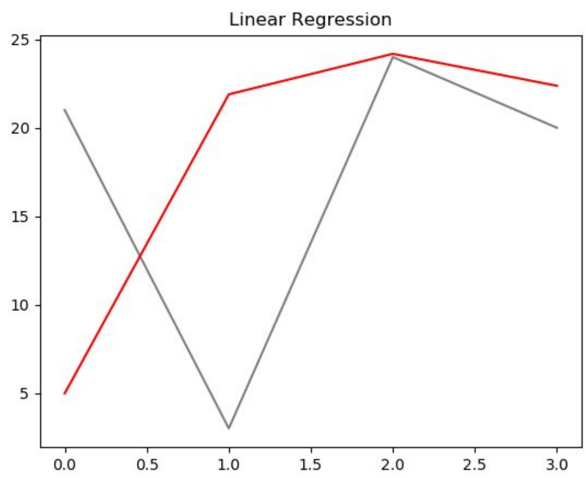


Fig. 10: Regression prediction results for "tourdefrance" keyword

The keyword 'NOS' is not highly frequent but occurs regularly over the period, and we see here better prediction results. For 'baan' we included English 'track' as keywords so the daily frequency was much higher than other keywords in the test (average frequency over the 91 data points: 472). As such, the RMSE of 145.85 suggests we vary by about +/-30% in our predictions, which may not be so bad as a baseline. Keyword 'eu' is another more frequent keyword (average frequency: 46) so the RMSE result again suggests predictions within a third +/- from the real values.

Given the variation in fit between past and future values for the keywords, it is unsurprising to find broad differences in the validation results (RMSE values). We observed that prediction worked best (lower RMSE) for keywords which were more regularly occurring (less 'null' values) but had lower standard deviations, i.e. were more constantly present over time in the data. Higher frequency keywords naturally returned higher RMSE as the unavoidable variations were greater in quantity but we find that for consistently frequent cases our prediction tended towards an error of +/- 30% which meant we would still largely predict the frequency within some lower and higher bounds accurately. The one time this would not happen is keywords whose frequency has been affected significantly by some external factor during the validation period. Our algorithm can only assume that this factor is not active when predicting, so that e.g. prediction for "tour de france" still proved effective as the factor for the increased frequency was no longer active and the algorithm (in this case, correctly) assumed the continued inactivity in the prediction period. However, for the keywords related to the event Die Finals Berlin 2019, the algorithm can not take into account that this event occurs in the validation period (causing further spikes in keyword frequency). We will return to this shortly regarding the incorporation of event features into our prediction model.

Again, as can be expected from the previous results, we find the RMSE increases as the prediction is validated against more days into the future. Where we found significant correlations between the predictor variable and the first feature (previous day's value), next day prediction tended toward RMSE=0. However, we do not find this a sufficient test of our prediction because it does not mean our prediction is perfect, it means that more times than not the following day's value is predictable from today's, e.g. because it is the same. As soon as we have a different trend, the prediction will not be accurate. So we find it more reasonable to train on 4 days into the past and predict 4 days into the future, which can allow for short term trends in the data to be learnt and built into future predictions.

5.3.3 Longer-Term Forecasting: Use of Keyword and Event Predictions

Statistical models as presented above become less accurate in prediction after a short period of time in the absence of significant trends or seasonality. While trends and seasonality can be found in TV channel audience data, it is lacking from our content-based success metrics, as we have seen in the initial tests for linear correlation. Therefore we do not consider this method to be useful beyond several days into the future, since it relies on the most recent trend in the data to make the prediction: With each day into the future we can consider the probability of the trend to continue to be reduced while more irregularity from the trend will be introduced. However, a publication event may be planned for a medium to long term future date, so how should we support this via prediction? As described at the end of Section 2, we found that we could use both the entities in our SKB as well as keywords aggregated via temporal annotation of documents to associate topics with future dates, where the topic is entities with an "anniversary" or associated with an event on that date or keywords predicted as strongly associated with that date. Having prepared this form of prediction for the professional user tests in August 2019, we predicted the following keywords and events for that coming week (see Fig. 11), using keyword predictions from German language news sources in the period Jan 1 - Jul 31, 2019):

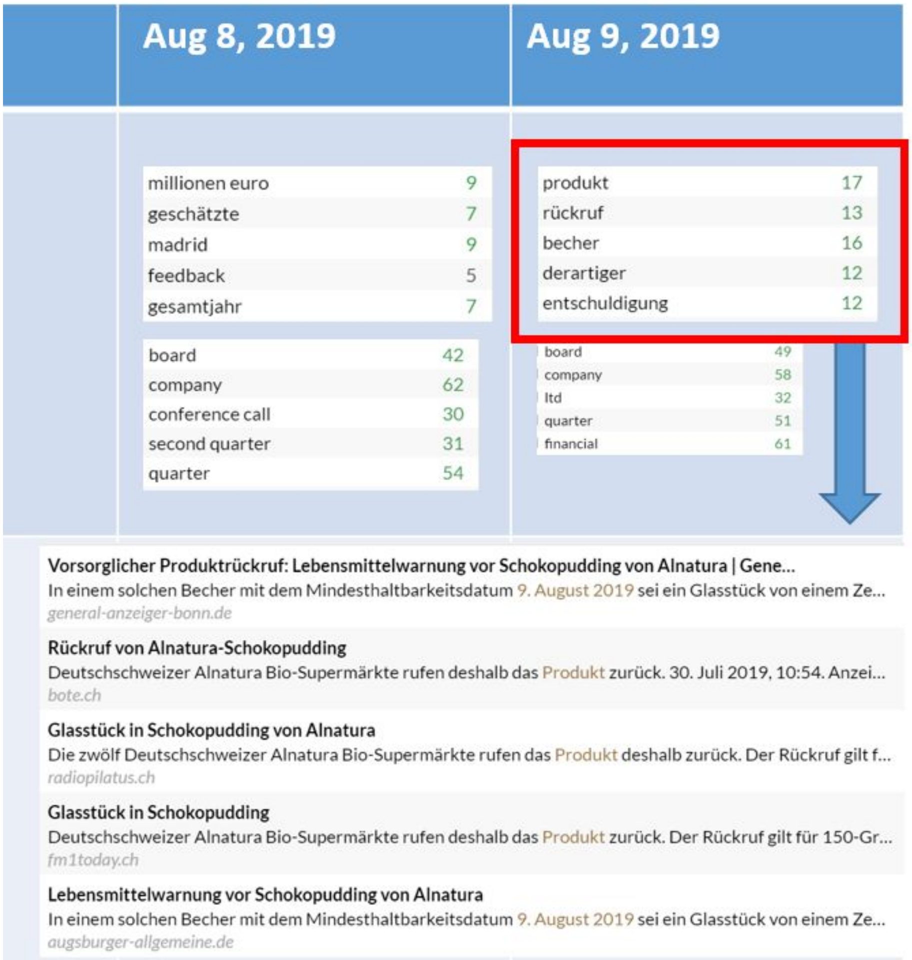


Fig. 11: Keyword-based prediction example

Keywords are in German - while there are some interesting references in the other dates, we found that the Aug 9, 2019 predictions were the most useful. Note that the number of keyword occurrences in the extracted documents are higher on that day than the other days - we found, reasonably, that the prediction relevance improved on dates which had more keyword occurrences in the document set. Here, we have clear references to “product, recall, cup, of this kind, apology”. We find in the news documents that this date is the deadline for a recall of chocolate pudding cups from Alnatura, as glass shards were found in an example. There is therefore the expectation that on that day this topic would return in the news and therefore communication around those keywords can expect an increased interest from the audience.

Event-based predictions can be configured according to a user’s expectations of their audience’s interests. For example, RBB’s own internal calendar (manually produced and not available in a machine-processable form) focuses often on local people, companies and events (i.e. related to Berlin and Brandenburg). NISV would focus on historical events that might have representations in their media archive. Fig. 12 shows an example of person-related anniversaries in our Knowledge Base for the week of the user testing:

Aug 6, 2019	Aug 7, 2019	Aug 8, 2019	Aug 9, 2019
Birth date: Danny Lee (HK actor and filmmaker) M. Revis (Dutch writer, d. 1973) Robert Mitchum (US actor, d. 1997) Death date: Feodor Lynen (German biochemist, d. 1979) Edsger Dijkstra (Dutch computer scientist, d. 2002)	Birth date: Ted Moore (S African cinematographer, d. 1987) Robert Mueller Thomas F Meyer (German biologist) Charlize Theron Jimmy Wales Death date: Voytek (Polish director, d. 2014) Michael Benjamin (German politician, d. 2000)	Birth date: Louis von Gaal Mohammed Morsi	Birth date: Roman Prodi Whitney Houston

Fig. 12: Event-based prediction example

For any given date, we can determine topics which are expected to be of greater interest to audiences on that date. If we add for each past date in a training data set these topics as additional features, we can learn the correlations between the existence of certain features and changes in the metric to be predicted. This could improve our prediction model when we choose a future date for prediction which shares features of the past data.

We performed a number of experiments to test this hypothesis. Experiment #1 asked if the correlation of keyword metrics to past events can improve the prediction of future keyword metrics, assuming a similar future event. Experiment #2 asked if the keyword prediction based on the temporal annotation could improve the prediction of future keyword metrics.

Experiment #1: the correlation of keyword metrics to past events can improve the prediction of future keyword metrics, assuming a similar future event?

We extend our autoregression to be multivariate, i.e. besides the past values of the same variable to be predicted, we add several features related to keywords or events associated to the date in the time-series. Keywords or events would be categorical, i.e. arbitrary strings drawn from a near infinite value space. However, ML algorithms typically work with numerical values (since their models are essentially mathematical in nature) and it is important to have the features occurring in the validation data also appearing in the training data, as otherwise the model can not learn from their effects.

To adopt a baseline for event-based prediction, we consider the Eurovision Song Contest (ESC), which is an annual event and has measurable effects on online discussion around the time of its occurrences (some research has even suggested sentiment and frequency of mentions over Eurovision tweets can predict the contest winner[1]).

We have taken the frequency of the keyword 'eurovision' in English & German language news from 1 March 2018 to 12 May 2019 (438 data points). From WikiData we have the metadata for ESC 2018 that the semi-finals took place on 8 and 10 May and the final on 12 May 2018. ESC 2019 dates are semi-finals on 14 and 16 May and the final on 18 May 2019. Our event

[1] An annual Twitter analysis has had some good results, unfortunately for 2019 they were far off! http://astrodataiscool.com/2019/05/predicting-eurovision-2019-from-twitter-data-draft (19/8/19)

features are threefolds: we have a binary variable for 'event effect' (to capture also the time before and after the event where the keyword is more frequent - we test with 1 day before and 1 day after) and for 'event duration' (strictly the period from start to end of the event), as well as an integer value for 'event level' (ESC has 2 levels: 1 = semi-final, 2 = final; however this gives us flexibility to capture the differing importance of sub-events, e.g. within the FIFA Soccer World Cup, there are group stage games, Group of 16 games, quarterfinals, semifinals and a final). So following the training, our validation will be by doing prediction for the following week (13-19 May 2019) which happens to be the week the ESC 2019 takes place. Fig. 13 shows the prediction results:

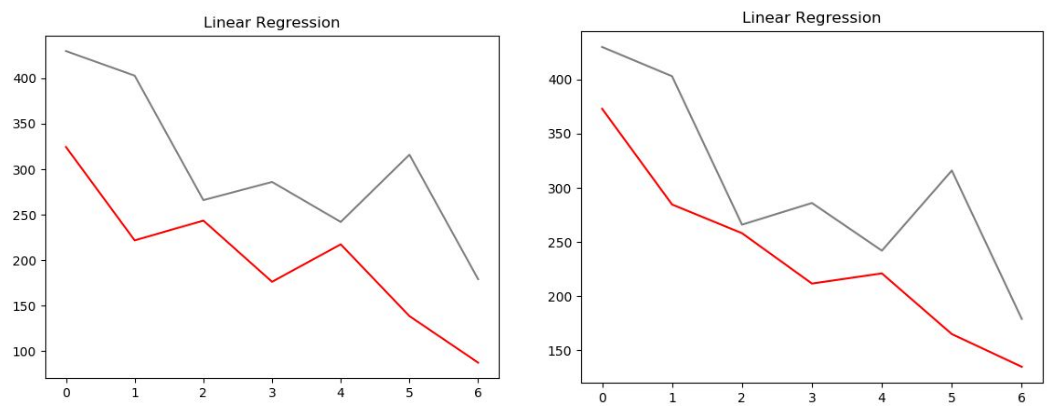


Fig. 13: (left) predictions without event features, (right) prediction with event features

Evaluation using only autoregression: Model fit: 0.45 Mean Absolute Percentage Error: 32.06 Root Mean Squared Error: 113.01	Evaluation with event features: Model fit: 0.54 Mean Absolute Percentage Error: 21.82 Root Mean Squared Error: 82.85

The results show that the event features are effective in making the predictions more accurate. The significance test also showed that all three event features were significant ($p<0.05$). In this case, we had an annually recurring event and learning from the previous years event was sufficient to improve prediction for the coming years event.

We test again by incorporating the features into the prediction for the RBB_finals dataset. Here, there is no past exact event to learn from, so similar to the audience prediction features we can learn from past events which are similar, e.g. share the same features as the new event. Considering the event features from the audience prediction (we remove top level category Sports as we are sure it is too general, and remove Organizer as too specific for this case) and using binary representations (either 0 or 1) for:

1. Sub-category: athletics
2. Location: is_in_Germany
3. Stage: is_finals

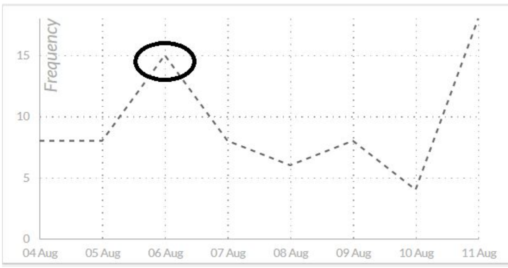
In our SKB with events extracted from WikiData, we find 12 athletics events during the dataset period (29 April - 5 August 2019). As a result we have 23 days in the training data (91 data points) and 2 days (4 data points) in the validation data on which an athletics event occurs. Of these, both days in the validation data are in Germany but this feature does not occur in the training data. Both days in the validation data are finals, as are 17 of the 23 days in training (a day on which medals could be won counted as a finals day). With the event features in place, we repeat the RBB_finals prediction experiments, and find little change in our prediction:

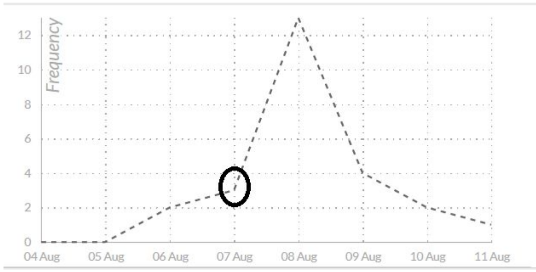
Dataset	Keyword	RMSE (without event features)	RMSE (with event features)
RBB_finals	deutschen meisterschaften	39.89	44.22
	sportarten	21.53	25.01
	athleten	17.98	18.41
	florian wellbrock	16.36	15.8

Probably the 3 months training data period is too short to provide enough events (features) for a good model learning: we have no date with the feature ‘located_in_Germany’ activated in the training data for example, so the feature is not relevant and could be removed. The feature ‘athletics’ is more significant than ‘is_finals’ - athletics finals do not seem to be as distinct as other sports such as football (soccer) - although neither demonstrate sufficient significance ($p<=0.05$) for prediction. Our general conclusion is that while exact recurrences of the same event can be used in prediction (for frequency of keywords fundamentally related to the event itself: so for the Eurovision Song Contest ‘eurovision’ is relevant but the name of a contestant isn’t, as they probably don’t occur again in the following year), prediction through associated events is challenging: for the future event, there is no guarantee that past associated events will have any correlation with changes in the keyword’s success metrics. Here, not even the frequency of ‘athleten’ correlates significantly with the occurrence of athletics events in order to be useful for prediction. Examining the data, we do see peaks in keyword frequency around some of our athletics events, but then for more of the other events there is no change and low frequency, and the combination means the model can not draw any meaningful correlations with this feature. Determining which events are more significant to a keyword for the prediction continues to be a research goal for the next period, e.g. a relationship to the country of interest to the audience we predict for.

Experiment #2: can the keyword prediction based on the temporal annotation improve the prediction of future keyword metrics?

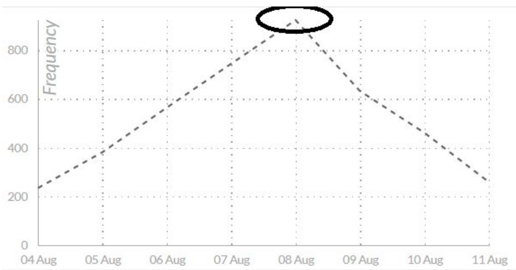
Our autoregression method can only make short term predictions, i.e. we focus on predictions up to 4 days into the future. Could we use the temporal annotation-based prediction of keywords for dates at periods further into the future as a basis for assuming there would be increased communication success on that date with the keyword (here assessed with the success metric of keyword frequency)? We looked at the top keywords predicted for the ReTV testing period (Aug 6-9) and the actual trend for that keyword's frequency in that time period (Aug 4-11).

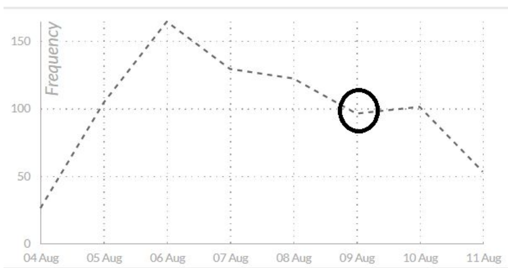




Above left: August 6, 2019. Top keyword prediction: 'abschiebungen': there is a clear peak in keyword frequency in the data (double so many mentions as the 2 days before or after).

Above right: August 7, 2019. Top keyword prediction: 'zweiten quartal': this keywords frequency peaks one day later significantly, so the prediction in this case did anticipate a communication burst in the period in question (the 4 and 11 August values indicate the keyword usually is not frequent at all).





Above left: August 8, 2019. Top keyword prediction: 'millionen euro': there is a clear peak in keyword frequency on this day.

Above right: August 9, 2019. Top keyword prediction: 'produkt': it has peaked 3 days earlier although mentions are still higher than usual in this time period. Since the product recall deadline is on this day (and hence the association in the keyword prediction) it seems that in this case the peak occurs a few days earlier (before the deadline). It is a reminder that no form of prediction can be considered perfect, since every discovered association has a different reason, and therefore different effects on the online discussion around that date.

Generally, we could show that the keyword predictions could be used in a content publication strategy as the keywords strongly associated with a future date do then tend to occur more often around that date.

5.4 ReTV Prediction Outlook

ReTV has produced several prediction approaches in this period. For future audiences on linear content vectors (broadcast TV and Web streams) we have extended our time-series forecasting with content and event-based features and experienced improved prediction accuracy. We found out that both types of features are very relevant. Our next steps will focus on extending it even further with more fine-grained content-related features (derived from video annotation tools, cf. WP1) and a more extensive set of event-based features (using events metadata which has additional property-values for events, cf. Section 2). Also, we will focus on constructing a forecasting model that aims to predict an optimal publication time by optimizing for the future audience numbers (cf. section 5.2). For future communication success on non-linear vectors (Web pages and social media posts) we have experimented with time-series forecasting using success metrics, with keyword aggregations using temporal annotations, and with events using an extension of our Events API (for “anniversaries”). For different cases, each of these approaches can be useful. Our next steps will be to integrate the predictive capabilities into the ReTV scenarios (Topics Compass and Content Wizard), so that professional users can directly access predictions when deciding what content to publish when on which vector.

6 Summary and Conclusion

This deliverable has presented the results of the project work in event extraction, temporal annotation, audience metrics, content-based success metrics and predictive analytics - where prediction uses these data points for greater accuracy in results. We successfully extract and use events in both audience and content success prediction, and will expose APIs for both event queries and anniversary queries (allowing for recurring events). We use our temporal annotation to identify future dates in documents and discover relevant keywords for that date. This capability will be integrated into the TVP Visual Dashboard (WP4) to allow keyword suggestions for future dates. Both this and future event knowledge can be provided to the professional user scenarios as a first predictive capability (WP5). Audience metrics are being generated and fed into the TVP; their visualisation will also be implemented as part of the TVP Visual Dashboard and we will be able to correlate TV audiences with other content-based success metrics. This will contribute further to the task of audience prediction, which has already been improved significantly by the inclusion of content and event features. Audience is complemented by content-based metrics such as frequency of mentions of a topic, share of voice, sentiment and the WYSDOM metric. Ex-post metrics are already available via the TVP Visual Dashboard. We already extrapolate these values into the short term future in order to support prediction of the communication success of a piece of content. This is complemented by the keyword-based and event-based predictions. We have seen that generally we can identify that certain topics will be of more relevance (popularity) in future time periods, despite the difficult challenge of the essential unpredictability of online discussion. This can guide content re-purposing and recommendation for Web or social media publishing. In the next experiments, we will continue to test models based on training from past data (e.g. optimise which event or content features work best for a prediction model) and develop a hybrid prediction solution which combines the insights from the various training inputs.

Ethics Self-Assessment

(a) We believe that the below given task(s) may raise one or more ethical issues:

WP Task	***Potential ethical issue***	***Steps identified to manage the issue***
T2.3 Audience and Viewer Metrics (GENISTAT)	Captured data could be associated back to an individual Zattoo user, thus contravening personal data conventions	• User information passed to ReTV should be obfuscated (e.g. hashing) so that it can not be reversed to the actual Zattoo user ID • Focus on aggregated views on the audience metrics, where individuals can be classified into audience groups which were defined in ReTV

(b) Below steps were taken to address the ethical issue(s):

WP Task	***Steps taken to manage an ethical issue***	***Responsible partner and statement on outcome***
T2.3	Only aggregated user data was used when training models, thus making it impossible to identify users.	Genistat AG. Their internal legal officer has confirmed that this does not infringe the GDPR. As there is no ethical issue with this approach, we do not need to contact the Ethics Board.

Based on the presentation of the solution to the Data Manager and their acceptance of its adequacy to avoid any ethical issues, the ethical self assessment is accepted as confirming that no ethical issues result from the project work.

Basil Philipp, 21.08.2019, Zürich

____

Signature, Data Manager

References

Cambria, E., Poria, S., Gelbukh, A., & Thelwall, M. (2017). Sentiment analysis is a big suitcase. *IEEE Intelligent Systems*, *32*(6), 74-80.

Danaher, P. J., Dagger, T. S., Smith, M. S. (2011). Forecasting Television Ratings. International Journal of Forecasting, vol. 27, pp. 1215–1240.

Meyer, D., & Hyndman, R. J. (2006). The accuracy of television network rating forecasts: The effects of data aggregation and alternative models. Model Assisted Statistics and Applications, vol. 1(3), pp. 147–155.

Novak, P. K., Smailović, J., Sluban, B., & Mozetič, I. (2015). Sentiment of Emojis. *PloS one*, *10*(12), e0144296.

Odoni, F., Kuntschik, P., Brașoveanu, A. M., & Weichselbraun, A. (2018). On the Importance of Drill-Down Analysis for Assessing Gold Standards and Named Entity Linking Performance. *Procedia Computer Science*, *137*, 33-42.

Weber, R. (2002). Methods to Forecast Television Viewing Patterns for Target Audiences. Communication Research in Europe and Abroad – Challenges of the First Decade. Berlin: DeGruyter.

Weichselbraun, A., Gindl, S., Fischer, F., Vakulenko, S., & Scharl, A. (2017). Aspect-based extraction and analysis of affective knowledge from social media streams. *IEEE Intelligent Systems*, *32*(3), 80-88.

Weichselbraun, A., Kuntschik, P., & Brasoveanu, A. M. (2019). Name Variants for Improving Entity Discovery and Linking. In *2nd Conference on Language, Data and Knowledge (LDK 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, *13*(3), 55-75.