MultiMedia Retrieval, MSc-course Computer Science, Utrecht University, NL
Egon L. van den Broek, Miroslav Živković, and Frans van der Sluis
Project description, version 2.0, course year 2017-2018, term 1

Adaptive Information Retrieval (AIR)

Change log

<table>
<thead>
<tr>
<th>date</th>
<th>version</th>
<th>changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>29-09-2017</td>
<td>2.0</td>
<td>• simplified data description, including references to the available data files and how they can linked to each other.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• general guidelines for the WPs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• specs on WP2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• included this change log</td>
</tr>
<tr>
<td>17-09-2017</td>
<td>1.2</td>
<td>• data description</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• a step-wise process for processing eye-tracking data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• some specs on the work plan</td>
</tr>
<tr>
<td>07-09-2017</td>
<td>1.1</td>
<td>• an extended data description</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• info on the R programming environment</td>
</tr>
<tr>
<td>06-09-2017</td>
<td>1.0</td>
<td>n.a.</td>
</tr>
</tbody>
</table>

Summary

Value of information (e.g., relevance and interestingness) changes both over time and per user. Continuous user feedback would enable a personalized prediction on multimedia’s information value. This project aims to explore the use of eye tracking to provide such feedback. Eye tracking features could unveil how a user experiences a text and, consequently, could provide a multimedia filter. This enables the foundation for a real-time closed-loop system that applies implicit feedback to continuously personalize user’s multimedia filter; in other words, using wearables (i.e., initially, an eye tracker) we achieve personalized Adaptive Text Mining (ATM).

Main goals

This project matches the subconscious signals users send all of the time, via the eyes, with the value of information by:

1. Analysing eye tracking data in relation to the complexity of a text and
2. Evaluating the explanatory power of these analyses for information value.

Description

Background: Early on, the invention of the printing press already inspired the idea of information overload: an ever-expanding amount of information that becomes unmanageable (Strother, 2012). Information Retrieval (IR) and text mining techniques have the potential to select Valued Information at the Right Time (VIRT) (Denning, 2006); but, seem to be plagued by a magic barrier: An unexplained limit to their potential to predict the value of information (Said et al., 2012). Because the value of information changes over time and per user (Saracevic, 2007), continuous feedback is needed to better predict whether information will be valuable.

Asking users to provide copious and continuous input about the information they want is not likely to succeed. Instead, implicit feedback that does not require any interaction from the user is a more viable option. Basic interaction data has already been successfully leveraged for text mining. Features from click-through data, browsing data, and query-text relations enhance the prediction of relevance by up to 31% (Agichtein et al., 2006). New data sources that offer an opportunity to further expand this success are
becoming available through the emergence of wearables and portables. Devices such as Google Glass, eye tracking fitted tablets, and the numerous digital watches now on the market not only create new platforms to deliver VIRT but also lead to a surge of available data for implicit feedback.

Current technology to analyze and leverage the surge of new data for better text mining is yet to catch up with the advent of new portable and wearable devices (Jacobs, 2009) and ubiquitous computing (Van den Broek, 2011). For example, eye tracking data is becoming increasingly available as tablets and other portable devices can now be equipped with a consumer-market eye tracker. Yet, this has not been explored for text mining purposes thus far.

**Research questions:** Building on two concepts of information theory (Shannon, 1948), we treat text as a signal (Van der Sluis et al., 2014):

1. Entropy of a signal, formalized through the complexity of a text.
2. Channel capacity of the receiver, derived from eye tracking data, as a window on cognitive activity (Hess and Polt, 1964; Ledoux et al., 2006).

Both cognitive activity and textual complexity predict the same aspect of information value, namely the interestingness of information (Hess and Polt, 1964; Ledoux et al., 2006; Van der Sluis et al., 2014). This shows that both can work in tandem to optimize IR. For example, when a given text is found too difficult, subsequent texts can be selected that are less difficult. Or, when too easy, subsequent texts can be selected that are more informative. We aim to explore the possibility of using eye tracking data to provide implicit feedback about information value and, consequently, answer the following research question:

*Can eye tracking data unveil (unstructured) big data’s information value?*

By answering this question, this project would yield a substantial innovation in multimedia retrieval. A foundation would be paved for personalized, adaptive IR. Such, next-generation applications require closed-loop models, which will be discussed next.

**The closed-loop model**

We adopt closed-loop retrieval as MultiMedia Retrieval (MMR)’s ultimate goal. For over a century, closed loop models are known in science and engineering, in particular in control theory and electronics. More recently, however, a new class of closed-loop models was initialized, closed loops that take a human / a user into the loop. Their descriptions target other areas; but, are essentially the same, comprising: sensors, processing, modeling, and actuators. However, the target state is not one of the user; but, mostly one of the system: the user controlling the system instead of the system.
steering the user (in our case, to a certain information interest). Closed-loops models would allow IR systems to become personal.

The (general) human-machine closed loop model. The model's signal processing + pattern recognition component, denoted in gray, is the component on which this project will focus (for more detail, see Figure 2). Closed-loop MMR systems put relatively much emphasis on the measurement and signal processing and classification phase. In general, two phases in this processing pipeline can be distinguished:

1. signal processing
2. classification (e.g., in terms of matching the user's interests).

These two phases form the core of the closed-loop model, which can be considered as a signal processing + pattern processing pipeline, as is shown in Figure 2.

Therefore, we will now first describe this general processing pipeline, before going back to the domain of MMR.

MMR is essentially a signal processing + pattern recognition problem. The goal of pattern recognition is to develop an artificial system that is able to recognize (complex) patterns, in our case interests, through (statistical) learning techniques. It follows the classic pattern recognition processing pipeline (see also Figure 2 and Meisel, 1972): a signal is captured and, subsequently, processed by a physical system (e.g., a CCD sensor, PC's audio card, or biosensor). After physical processing the raw signals provided (e.g., an image, audio track, or biosignal) that form the measurement space.
The raw signals are preprocessed (e.g., filtering and artifact removal), which provides `clean' signals. These `clean' signals are synchronized with each other for reasons on which we will elaborate later. Subsequently, the signals can be segmented, based on events or stimuli, which facilitate their further analysis. Next, features need to be extracted from the signals and parameters of these signals need to be calculated. The set of features and their parameters provides the required pattern space.

The pattern space of calculated parameters from the recorded signals is defined for the pattern classification process. Next, feature selection / reduction is applied. This improves the prediction performance (or power) of the classifier, reduces the chances of overfitting, provides faster and more cost-effective classifiers, and aids our understanding of the underlying process that generated the signals. Consequently, the reduced parameter set reduces the curse of dimensionality, removes redundancy among the signal's features and their parameters, and, hence, becomes more generic. So, an optimal set feature vector (or more accurately: parameter vector) or reduced pattern space is generated, which can be fed to the classifier.

Figure 1. The (general) human-machine closed loop model. The model's signal processing + pattern recognition component, denoted in gray, is the component on which this article will focus (for more detail, see Figure 2).

The next phase in the signal processing + pattern recognition is the actual classification of the optimized feature vectors. Three classes of pattern recognition techniques can be distinguished: statistical pattern recognition (including artificial neural networks, ANN),
template matching, and syntactic or structural matching. Note that ANN are sometimes considered as a separate class of pattern recognition techniques. In MMR, template matching, and syntactic or structural matching are seldom used. In almost all cases statistical classification (including ANN) algorithms are applied.

Statistical pattern recognition can be employed through either unsupervised or supervised classification (including reinforcement learning), where the latter compared to the former requires an additional loop in the pipeline. In both cases, this classification process can either be the development of the classification system or its execution on a new set of data. Next, we will first denote the distinction between unsupervised and supervised classification. Subsequently, I will briefly depict the difference between the development of a new classification system and its employment on a new set of data.

If a set of predefined classes (or labels or categories) to which the measurement space belongs is available (e.g., user’s interests), the feature vector can be identified as a member of a predefined class and given the accompanying label. This approach is, therefore, baptized as supervised learning / classification (e.g., Fisher’s linear discriminant analysis, LDA or support vector machines, SVM). Such predefined classes are sometimes referred to as the ground truth. In contrast, with unsupervised classification (e.g., principal component analysis, PCA or hidden Markov models, HMM) the decision rule for the classifier is developed for hitherto unknown classes. The classification process is based on the similarity of patterns, determined by a distance/similarity measure and an algorithm to generate the clusters.

With the development of a new classifying system, one can choose for either an unsupervised or a supervised approach (as explained in the previous paragraph). Unsupervised classification does not need a priori knowledge and often only entails saving the pattern space in specified format. Supervised classification requires the training (or learning) of a new classifying system, before the actual classification can be conducted. Using labeled feature vectors for training, a discriminant function (or network function for artificial neural networks, ANN) is used to recognize the features and initial classification is realized. Classification errors can be determined using a certain error criterion and the classification process can be adapted. This training or learning phase is depicted by gray boxes in Figure 2. When this process is finished, the actual (supervised) classification can be conducted.

This machine learning pipeline can be employed for each data source (i.e., modality such as vision, speech, and biosignals) separately. Alternatively, after the features and their parameters from all signals are extracted, they can be merged into one pattern space. Both of these approaches are applied frequently.

**Challenges**

Features: To use eye tracking as feedback on textual complexity means that both need to be calculated. For the latter, state-of-the-art achieve a classification accuracy of up to 93.62% and a correlation with human ratings of up to $r = .703$ (Van der Sluis et al., 2014). For the former, this project aims to make a contribution by creating algorithms that describe eye tracking data. Currently, low-level features from eye tracking measurements have been shown to predict cognitive activity during learning with 68.8% accuracy (Conati and Merten, 2007; Cole et al., 2011). Continuing on this work, this project aims the use of gaze behavior and pupil size to predict how complex users find a text. Features that describe gaze behaviour (e.g., fixations and saccades). A saccade is
defined as a rapid change in gaze location, and a fixation is regarded as being bordered by two saccades. The resulting features will be tested on their reliability with consumer-market eye trackers, which are expected to soon proliferate across many systems and platforms.

**Validation:** Eye tracking’s value will be evaluated by comparing the following variables:

< textual complexity, eye tracking feedback, information value >

This evaluation is split in two phases:

- The eye tracking features will be compared to human ratings of how difficult it was to read a text (i.e., **textual complexity vs. eye tracking feedback**). The resulting model indicates the cognitive activity during information processing. In combination with the textual complexity analysis of Information eXperience (IX) this model forms the basis of a closed-loop solution (see the previous Section), which integrates eye tracking for Adaptive Text Mining (ATM).

- The resulting eye tracking model of cognitive activity will be compared to how interesting a text is (i.e., **eye tracking feedback vs. information value**). From previous research it is known that complexity, and the resulting cognitive activity, is a key determinant of information value. Amongst others, the right level of cognitive activity brings a text to the ‘sweet spot of interest’ (Silvia, 2008; Van der Sluis et al., 2014). To verify this, the current project aims to determine what level of cognitive activity leads to maximum interest. It provides the necessary understanding for ATM.

The data set to evaluate both phases will be supplied via the course website. It contains 504 ratings of complexity, comprehension, and interest (i.e., value) with accompanying eye tracking data of users reading a text. The resulting evaluation shows the explanatory power of eye tracking in combination with textual complexity for predicting information value.

**Relevance**

ATM relieves the problem of information overload, as becoming increasingly problematic due to the big data push and the (remaining) lack of adequate filters (Strother et al., 2012). Par excellence, this is illustrated by IBM, who initiated the big data era. Recently, IBM acknowledged that solely big data processing could not solve the big data challenges. Therefore, they initiated the [cognitive computing](http://www.research.ibm.com/cognitive-computing/) lab, which aims to process big data in an “intelligent” manner. This exceeds the process of going from data to information. It initiates the process of going toward understanding information, going from information to knowledge. As IBM says it:

“Explore why cognitive computing ... is essential in the era of Big Data”

Par excellence, this project realizes this principle via combining, on the one hand, fundamental work from control theory (Shannon, 1948) and psychology (Wundt, 1896; Silvia, 2008; Van der Sluis et al., 2014). To verify this, the current project aims to determine what level of cognitive activity leads to maximum interest. It provides the necessary understanding for ATM.
Hess & Polt, 1964) and, on the other hand, IR (Strother et al., 2012) and big data analytics (Jacobs, 2009).

This project provides the tools to maximize the information throughput by bringing the textual complexity close to a user’s cognitive capacity, similarly to maximizing the data throughput with respect to a receiver’s channel capacity (Shannon, 1948). The possibility to optimize not only the information value but also the information throughput makes this a foundation for a disruptive technology that transforms how information-intensive industries communicate. Depending on the targeted information value (i.e., the application’s goal), this can be for:

- Education, by tracking how a student processes information and optimizing the selected texts accordingly;
- Marketing and communication, by using eye tracking feedback to learn what (type of) writing creates the biggest impact, setting a new standard for the industry; and
- Publishing and search, by leveraging eye tracking data to see whether a message, discourse, or answer comes across.

As the preceding examples show, this project offers new opportunities for innovation to the key information-intensive industries. With this project, you can set a step towards turning the information overload into an information opportunity and provide more efficient and effective communication.

**Data description**

**Data set**

A collection of 14,856 articles from The Guardian was used as the data set. The data set consisted of articles from the following news feeds: culture; environment; financial, market and economics; commentary; life and style; and science and technology.

To reduce variation that originates from differences in article length, we truncated all articles after 1,200 characters. The cutoff point was placed before the end of the word at position 1,200, and three dots were added to indicate the story normally would continue. Any layout was stripped from the articles, leaving only the title and textual content.

Subsequently, the articles are filtered in three steps:

1. Articles from the lower, middle, and upper part of the distribution of textual complexity were preselected.
2. A final selection consisting of 18 articles was performed based on suitability.
3. Selected news items differed in topic to ensure a variation in topical familiarity would be existent. The textual complexity of the resulting selection of 18 articles, grouped by three levels of complexity.

For more, detailed information on how the articles’ textual complexity was calculated, we refer to Van der Sluis et al. (2014).

All 18 Guardian texts the participants read are provided in the file 18_Guardian_texts.txt at the course website. Each text has a unique ID, which is also provided in the participant_ratings.xlsx; hence, these files can also be conveniently linked to each other.

**Participant ratings**

Data of 29 participants is available. Their average age was 28.60 (SD = 6.06). They voluntarily took part in the experiment. None of the participants was a native English speaker, but all graded their reading literacy as high (M = 4.63, SD = .62, range 1–5, 5
highest). All participants were well-educated; they either had a university degree or were enrolled as a student at a university.

The participants were asked to read a set of articles self-paced and judge them on:

- **novelty-complexity**: the assessment of whether information is sufficiently novel and complex or too predictable and not challenging enough to stimulate interest.
- **comprehensibility**: the participant's coping potential related to prior knowledge, available resources, and so forth. For example, if a text is too complex, the coping abilities most likely do not suffice, leading to a decline in interest.
- **interest**: an emotion characterized by its cognitive, subjective, physiological, and expressive response components, concretizing three aspects of the User eXperience (UX): the thoughts, feelings, and actions. This allows us to identify the causes of interest; that is, learn the relationships between sources of information and reader's responses.
- **Familiarity**: This is a control variable. Participants were asked to indicate to what extent they were familiar with the topic of each Guardian text.

using seven-point *semantic-differential scales* (Van der Sluis et al., 2014).

The participant ratings are provided in the file **participant_ratings.xlsx** at the course website. Each participant has a unique ID, which allows linking this data to their eye tracking data.

**Eye tracking data**

This is data as obtained while the 29 participants were reading a document on a screen. A remote eye tracking device was used to track the participants' gaze on a standard TFT monitor, with 1280 x 1024 resolution. Software has been used to process and record the eye movement signals from the eye tracking device. Viewing was binocular and each participant's eye movements were sampled at sampling rate indicated in the header of the file. The eye movement's raw data is a sequence of eye gazes, represented in the display's 2D coordinate system. These data are analyzed in order to segment eye activity into states (events), including being the most noteworthy:

- **gaze**: the point on which a person's eyes are focused;
- **conjugate gaze**: unison gaze with both eyes; that is, focus in the same direction at the same time;
- **saccade**: a quick, simultaneous movement of both eyes in gaze location; that is, between two or more fixations;
- **fixation**: maintaining visual gaze on a single location, in between two saccades;
- **blink**: a semi-autonomic, often unconscious rapid closing of the eyelid, also considered as a special case of fixation.

These states (events) can be considered as markers of the participants' search paths, in our case, while reading the document.

We have removed the file header from the eye tracking data files, as most information is not of interest. However, one type of information is important: the signal's sample rate, which was 60Hz in all cases.
Table 1. Raw Data Export Messages. The following fields are typical for one message, along with the actual message.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Timestamp of the sample.</td>
</tr>
<tr>
<td>Type</td>
<td>The type is MSG</td>
</tr>
<tr>
<td>Trial</td>
<td>The number of current trial</td>
</tr>
</tbody>
</table>

Table 2. Raw Data Export Samples. The following fields can be exported for one sample (when applicable).

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Timestamp of the sample.</td>
</tr>
<tr>
<td>Type</td>
<td>The type is SMP.</td>
</tr>
<tr>
<td>Trial</td>
<td>Number of current trial.</td>
</tr>
<tr>
<td>L Raw X [px]</td>
<td>Horizontal pupil position.</td>
</tr>
<tr>
<td>L Dia X [px]</td>
<td>Horizontal pupil diameter.</td>
</tr>
<tr>
<td>L Dia Y [px]</td>
<td>Vertical pupil diameter.</td>
</tr>
<tr>
<td>L CR1 X [px]</td>
<td>Horizontal corneal reflex (CR) position. One or two CRs can be present.</td>
</tr>
<tr>
<td>L CR1 Y [px]</td>
<td>Vertical corneal reflex position.</td>
</tr>
<tr>
<td>L POR X [px]</td>
<td>Horizontal gaze position.</td>
</tr>
<tr>
<td>Timing, Latency</td>
<td>Quality values</td>
</tr>
<tr>
<td>L Plane</td>
<td>Plane number</td>
</tr>
<tr>
<td>L Object Hit</td>
<td>Name of area of interest (AOI) that is hit by current sample</td>
</tr>
<tr>
<td>H POS X [mm]</td>
<td>Head position on X</td>
</tr>
<tr>
<td>H POS Y [mm]</td>
<td>Head position on Y</td>
</tr>
<tr>
<td>H POS Z [mm]</td>
<td>Head position on Z</td>
</tr>
<tr>
<td>H ROT X [°]</td>
<td>Head rotation on X</td>
</tr>
<tr>
<td>H ROT Y [°]</td>
<td>Head rotation on Y</td>
</tr>
<tr>
<td>H ROT Z [°]</td>
<td>Head rotation on Z</td>
</tr>
<tr>
<td>L EPOS X</td>
<td>Eye position on X</td>
</tr>
<tr>
<td>L EPOS Y</td>
<td>Eye position on Y</td>
</tr>
<tr>
<td>L EPOS Z</td>
<td>Eye position on Z</td>
</tr>
<tr>
<td>L GVEC X</td>
<td>Gaze vector on X</td>
</tr>
<tr>
<td>L GVEC Y</td>
<td>Gaze vector on Y</td>
</tr>
<tr>
<td>L GVEC Z</td>
<td>Gaze vector on Z</td>
</tr>
<tr>
<td>Frame</td>
<td>Frame counter</td>
</tr>
<tr>
<td>L Event Info</td>
<td>Event type detected for the interval containing this sample (fixation, saccade, blink)</td>
</tr>
<tr>
<td>Stimulus</td>
<td>Stimulus associated with this sample</td>
</tr>
</tbody>
</table>

Note. In case of binocular recordings, data from both channels (i.e., L and R) can be exported.

The eye tracking data is provided as 1 ZIP-file: eye-tracking_data.zip at the course website. This file contains 29 data (txt) files, each containing the complete eye tracking signal of one participant. The files name starts with the participant’s unique ID, which can be used to link it to the participant_ratings.xlsx.
Eye tracking signals: Some suggested processing steps

1. Detect events in the “raw” data and visualize both input data and events. Per sample, the raw data contains information about the horizontal/vertical pupil position, horizontal/vertical pupil diameter, etc.; see also the description in Table 2. You may need to perform data cleaning, and/or filtering of data. An event is initially either a fixation, a saccade or a blink. An extension of this triplet is encouraged. A good starting point for (some of) algorithms are Salvucci and Goldberg (2000) and Nyström and Holmqvist (2010). Alternatively, you may want to use pattern recognition or machine learning methods to identify events.

2. In this task, you will use the events data – this data is processed for you from the measurements (raw data) and contains information about blinks, fixations and saccades. You will have to:
   - Read the data (think about fast solutions to do this in R).
   - Clean it (i.e., remove missing values and possible anomalies)
   - Explore (Note. There are readily available packages in R for this)
   - Visualize
   - Make a model (e.g., machine learning) for prediction of users’ experienced interest /complexity comprehensibility.

3. Based on analysis performed in point 2, in this task you will take a single events data file, and, as you read the data, predict the interest and comprehensibility of the user. These will be compared to the actual user feedback(s). You will reason about discrepancies in results, and identify the issues for it.

4. Based on the model derived in 2 and 3, you will identify the document that will give the highest values for interest and comprehensibility. Once you do it, you will verify this by reading the events data and make predictions, which will be compared with actual feedback for a given document.

For more details, see the description of Workpackage 2 (WP2), which follows after the description of our programming language and our workplan.

Our programming language: R

R is an open source interpreted programming language and software environment, which can be conveniently accessed through a command-line interpreter. It is available for Linux distributions, versions of Windows and (Mac) OS X. R was primarily developed for statistical computing and graphics; but, over time, its application areas have been extended significantly. When installing R, a core set of packages is included. As of today, over 11,000 additional packages are available at CRAN (i.e., the Comprehensive R Archive Network), GitHub, Omegahat, Bioconductor (for bioinformatics), and other repositories. For more information on R, Wikipedia can serve as good starting point followed by the resources provided next.

R-resources

The main resource for all R-related material is the homepage of the R-project: https://cran.r-project.org/. This site includes dedicated packages for everything you will need for this course, including dedicated packages for signal processing, sensory data, and pattern recognition and machine learning. The last package is one of many that are available in R – for more information see an overview of machine and statistical learning packages in R. Among many other sides, R Tutorial: An R introduction to statistics is informative, including many pointers to other resources. Last, but not least, the R-package ISLR (Data for an Introduction to Statistical Learning with Applications in R) is available, which is accompanied with a freely available book.
Online many free books and tutorials are available, including:


In general, check CRC Press / Taylor & Francis Group, LLC's series *The R Series*, which includes the above mentioned *Wickham (2014)*. In case these resources do not suffice, check available online lists of books and tutorials, such as [http://www.statmethods.net/about/books.html](http://www.statmethods.net/about/books.html) and [https://www.r-project.org/doc/bib/R-books.html](https://www.r-project.org/doc/bib/R-books.html). Also, many video lectures are available at YouTube; for example, see [https://www.r-bloggers.com/in-depth-introduction-to-machine-learning-in-15-hours-of-expert-videos/](https://www.r-bloggers.com/in-depth-introduction-to-machine-learning-in-15-hours-of-expert-videos/).

R also has dedicated packages for processing specific signals, including audio (incl. speech), images, and biosignals. Conveniently, also dedicated packages for eye tracking are available, including:

- **eye tracking-R**: The R package for analyzing eye tracking data. For this particular package, it may be useful to have a look at [GitHub repository](https://github.com). It mentions code dependency on R dplyr implementation and what to do with it.
- **ETRAN—R Extension Package for Eye Tracking Results Analysis** (note, article is freely available within the UU-domain).
- **Emov**: An R CRAN package for fixation and saccade detection in eye movement recordings.
- the Dutch package [gazepath](#).

### Work plan

**Table 4.** Gantt chart of the project’s proposed work plan. The dates are the deadlines for the Work Package (WP)s’ deliverables.

<table>
<thead>
<tr>
<th>WP1</th>
<th>WP2</th>
<th>WP3</th>
<th>WP4a</th>
<th>WP4b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intro,</td>
<td>Methods &amp;</td>
<td>Results &amp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rationale,</td>
<td>Analysis strategy</td>
<td>Discussion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Research question(s),</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22-09</td>
<td>06-10</td>
<td>27-10</td>
<td>31-10</td>
<td>03-11</td>
</tr>
</tbody>
</table>

**BONUS** Closed-loop demonstrator

*Notes. The specific Work Packages’ content will be explained below and during the lectures and practicals.*
General guidelines

- The size of WP1, WP2, and WP3 needs to be in the range 3-8 pages. For WP4, which unites WP1-3 and extends it, there are no size restrictions.
- WP reports can be delivered as pdf-file via email to vandenbroek@acm.org.
- Follow the author instructions of the International Journal of Multimedia Information Retrieval. Use the author tools to prepare a manuscript, including both a Word template and a LaTeX macro package.
- Your report should provide all necessary information to conduct an exact replication of the work, without any additional info.
- For all elements, you are expected to make well-argued decisions. For example, provide either references to a known method or a (sound) reasoning for a design choice.
- You are expected to respect other's work, whether this is code or text. So, if you found elements of your report on work of others, please make this explicit by including a reference. If you want to copy statements of others, please use quotation marks; otherwise, you conduct plagiarism.

Workpackage 1 (explained during lectures)
- Introduction
- Structured/systematic literature review
- Research rationale
- Research questions
- Requirements analysis

Workpackage 2
Full description of your signal processing methods, analysis strategy, and technical framework, including at least:

Signal processing
- Describe how the signals will be visualized (likely: in multiple ways!)
- Describe the filters you will use.
- Define (a) strategy(s) to determine anomalies.
- Describe the type of analysis you will execute (e.g., spectral, temporal, spatial, and combinations)
- Define the window function to be used (e.g., a sliding time window).

Analysis strategy
- Definition of the features and their parameters to be extracted.
- If applied, define the normalization and/or standardization applied.
- Define your strategy for feature-parameter selection/reduction.
- Define the model (e.g., a classifier such as a neural network or ANOVA) you will use to process the features’ parameters.
- Determine how you will evaluate you model and what is the threshold for success.

Technical specifications
- Describe how you have linked/parsed the different data sources to each other and secured it was synchronized and how you have segmented the data.
- Provide a description of your complete signal processing framework. Distinguish between the fully automated parts and the parts, where manual inspection was applied (of samples).
• Specify what platform(s) and libraries/modules will be used for the analysis.
• Describe how do the processing modules interface with each other, making an integral processing pipeline.
• Describe how you will calculate the theoretical and empirical computational complexity of your system.

Workpackage 3 (to be done)

Workpackage 4 (to be done)

References


