Bachelor Project
Walking Wales : The Data Challenge

David Kolb

University Konstanz, Chair for Data Analysis and Visualization,
Konstanz, Germany
{david.kolb}@uni-konstanz.de

Table of Contents

1 Introduction

The handling and analysis of large amounts of data is no trivial task. Especially, if the given data comes from a numerous variety of different sources and was not yet preprocessed. However, this task gains importance as the number of everyday people, who record their daily routine, continuously grows. Nowadays, it is a simple matter for the most people to produce and store huge amounts of data which holds information about their lives, as nearly everyone in an industrialized country possesses a smartphone or other tracking devices. Yet, as easy as the generation of this information is, the processing is rather cumbersome. This leads to the problem that most of this data remains untouched as its not easy to analyse and to incorporate when there are diverse types of data sources. Furthermore, there are countless data formats which all treat the information differently and make it hard to find a uniform approach in dealing with the data. Another common problem is that the data can be of relatively poor quality as it is often recorded in a non-professional fashion and without a controlled environment. Built in sensors of smartphones often lack accuracy or produce noisy measures as they don't have the requirement to be used in the industry or for commercial purposes. However, they are widely available, easy to use and have low costs of operation.

In this project I deal with a very similar task. The problem is about understanding and making sense of a real world data set which was created in a semi-professional manner. The recording was done using common technology which is available for everyday people. The data set comprises very different forms of data which need to be brought together in order to add value to the data. All data is in it's raw form as the sensors produced it. For a quick overview of the background of the data see the following section. The main aspects of the project are the understanding, analysis and visualization of this data set. The first challenge was to find a way to deal with the different formats and to do preprocessing. Secondly, some properties of the data are analysed and assessed in terms of data quality. The last part is about the creation of a visualization which allows easy browsing and shows interesting aspects of the data in more detail.

1.1 Alan Walks Wales

The project revolves around the data set recorded by Alan Dix[1] as he was doing a walk around the complete perimeter of Wales. He walked approximately 1700 kilometers over a period of three and a half months. During his walk he used various sensors to record quantitative and qualitative data describing many aspects of his journey. The data includes vital recordings, GPS traces, texts, pictures and some other sources. His motivation was partly of personal and scientific nature as he wanted to explore the benefits and limitations of the usage of mobile technology while being on a long distance walk. Another goal

[1] `http://en.wikipedia.org/wiki/Alan_Dix`

was to collect a vast variety of data as well as to serve as a living-lab testing the recording devices regarding his roots in human computer interaction. The whole walk was documented on his blog 'Alan Walks Wales'[2]. He provides all of his recorded data on his blog free for everyone to use and analyse as well as a lot of additional information on his walk. There is also a presentation about his walk which is available on his Youtube Channel[3]. Figure 1 depicts the route of his walk and a picture of Alan Dix. Although, his walk was a remarkable achievement it is also a great opportunity to get hands on this data due to its uniqueness and its real world application character.

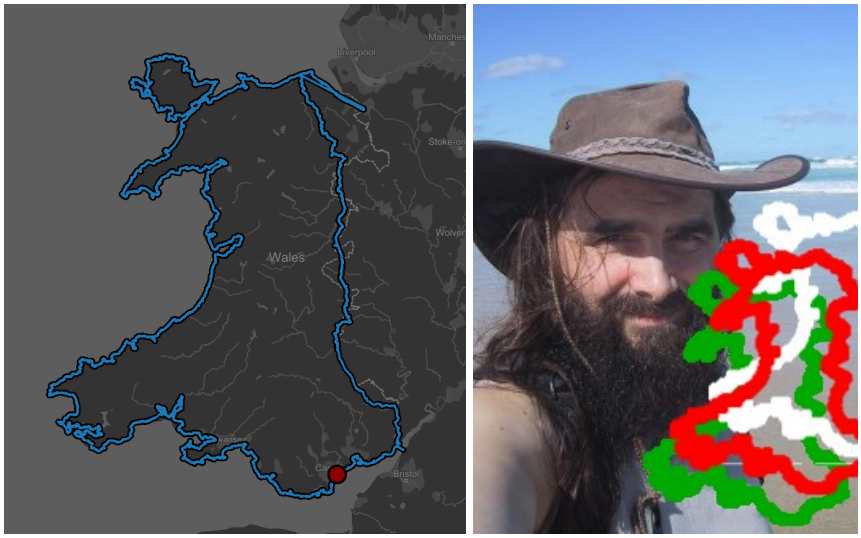

Fig. 1. Left image: A map showing the (preprocessed) GPS traces of his walk. The start and endpoint is marked (*red dot*) and he walked counter clockwise. Right image: A picture of Alan Dix.

2 Motivation

As briefly mentioned in the introduction the use of mobile technology in our everyday life has increased substantially. Almost every aspect of life gets infused with little computers in order to make it 'smarter'. Nowadays, even refrigerators can be connected to the internet and the vision is that someday it automatically tells you if you run out of groceries. Another field were this kind of technology is booming is the sport sector. There are countless fitness apps which allow the user to track every activity and to create statistics about your progress. Everything

[2] `http://alanwalks.wales`
[3] `https://www.youtube.com/channel/UCihmFcvs1rSLoSrPGQBUEaA`

is connected with social networks so the user can share his achievements with his friends or the world. Also these apps create a lot of data which holds much information about the user and his activities. Location data is usually recorded which provides information about the distance, speed, and route of the activity. And at the end of a year you can see an overview of you accomplishments.

This kind of development goes by the term of 'Internet of Things'. It means to create a network of physical objects (things) that contain electronics and sensors. The created data is shared between the entities in order to obtain benefits for the user or the manufacturer. One of the key problems of this progression is the amount of data that is created is often derived from diverse locations. This increases the need of efficient processing of such data. Furthermore, visualization is very important because it helps the user to draw valuable information from the data which may reveal unseen relationships. Keeping these facts in mind it is especially interesting to process and analyse a real world data set which is the case for this project because it was created in very similar conditions.

Another point is that the data is unique in many ways. Firstly, it is very interesting to look at this many data sources which all cover the same time period and activity. Also, the variety of data is a big benefit as is allows complex analysis in order to find relationships and correlations. Secondly, most streams are almost continuous for the whole time period. This fact makes the bio data recordings very special because they are one of the longest in the field.

3 Problem Definition

The main tasks of the Project can be divided into three parts. The first part is about the understanding of the data and its formats. As already mentioned before the data originates from very different sources and sensors which all present their output in different forms. So a main problem is to be able to read the data in order to do any analysis with it. This is especially difficult as every format has its own characteristics and organization of the data. Another part of the data understanding phase is to do a data quality assessment and some cleaning of the data. Due to the fact that it is not possible for a single person to produce perfect and noise free recordings while being on a long distance walk, it is important to investigate how expressive the data is and if the quality is good enough in order to produce a valid analysis. Another problem of the data quality may be that there are wrong or missing measures which need to be identified and handled. Some data streams also require further familiarization with the corresponding domain. For example the bio data recordings measure specific values of the human body which first have to be understood in order to create a correct analysis with them.

The second part is about the analysis and correlation of the streams. I also focus on feature construction for this stage. Some data stream are still in a very raw form that does not allow to draw conclusion from it. So the goal is to analyse them in a way which results in a more meaningful output. This is especially the case with the bio data streams as the heart activity data only

gives the raw heart activity and not the heart rate per minute. The raw heart activity is very important for the analysis of health related issues but does not hold much information for the scope of this analysis goal. The next focus lies on the texts which were written during the walk. The goal is to extract a sentiment measure from the blog entries which should allow to make statements about how Alan felt during his traveling. One problem with this approach is to assess if a analysis of the texts is able to identify a correct sentiment which matches with the real sentiment of Alan. The last goal of the analysis phase is the correlation of some of the data streams. Sometimes there is more than one source for the same data stream so it would be interesting to see how the different sources match one another. Furthermore, some streams may be directly correlated. For example it is not hard to imagine that the elevation of the terrain correlates with the heart rate as a steep landscape definitely influence the demands on the fitness.

The last part of the project deals with the visualization of the data. The goal is to allow easy browsing and to show the results of the analysis stage. Moreover, the user should be able to identify relationships in the data such as the correlations between the data sources or if there are exceptional situations. The main focus of the visualization should be to incorporate all data sources into a easy comprehensible view so a user can review the walk or parts of it. One problem is to deal with the relatively large data amount as all sources were captured over a period of three and a half months. This could easily confuse the user as the mere amount of data may hide interesting relationships in the data.

4 The Data

This section is about the data set which was captured during the long distance walk of Alan Dix. All of the different data sources will be introduced in terms of their format, size, origin and specific characteristics. The data is divided into quantitative and qualitative data.

4.1 Quantitative Data

The quantitative data consists of three major sources:

- GPS (location)
- ECG (heart activity)
- EDA (skin conductance)

Each of the streams was captured using a distinct device. Some of this devices also record further information such as acceleration and skin temperature. Also there are more than one sources covering the same data for some of the sources.

GPS
The location data was captured using a Garmin eTrex 30 and simultaneously

with a smartphone software named ViewRanger. Both produce their output in the GPX format. This format is based on a XML-standard but defines some new tags which hold GPS specific information such as coordinates or the elevation.

Extract from one GPX file recorded by the Garmin sensor

```
<trk>
    <name>2013-04-18 09:27:00 Auto</name>
    <trkseg>
        <trkpt lat="51.4631031454" lon="-3.1627476308">
            <ele>16.48</ele>
            <time>2013-04-18T08:27:00Z</time>
        </trkpt>
        <trkpt lat="51.4631220046" lon="-3.1627062242">
            <ele>16.48</ele>
            <time>2013-04-18T08:27:01Z</time>
        </trkpt>
            .
            .
            .
```

The example shows how the data is stored in the GPX format. First a Track ($< trk >$) is defined which holds a list of Track-Segments ($< trkseg >$) as well as additional attributes of the Track itself like the name of the Track or additional meta information. Thereby, each Track-Segment stores a list of Track-Points ($< trkpt >$) which hold the actual location traces as value of the Track-Point tag. The location is stored as standard coordinates measured in longitude and latitude. Each Track-Point, again, contains further information about the location. In our case its the exact timestamp when the location was measured as well as the current elevation of the terrain for this point. The data can be easily accessed by using a simple XML parser.

Both GPS sources differ quite strongly in size as the ViewRanger has a much lower sampling rate than the Garmin device. Also the ViewRanger software creates one big GPX file containing all of the location information, whereas, the Garmin device create several files containing mostly one or two days. The Garmin recording consist of 215.500 recorded Track-Points and the ViewRanger of only 11.165 Track-Points. Figure 2 shows images of the uncleaned and unfiltered sources. Although, the ViewRanger GPS contains much less Track-Points than the Garmin GPS it correctly manages to display the route of the walk. The images also show that there are clearly some problems with the data quality as some points get connected very arbitrary. The quality of the GPS data is further discussed in Section 5.

EDA (Electro Dermal Activity)
One of the bio sensors Alan was carrying measured the conductance of the skin. The recording was done with a AffectivaQ sensor developed by the Affective

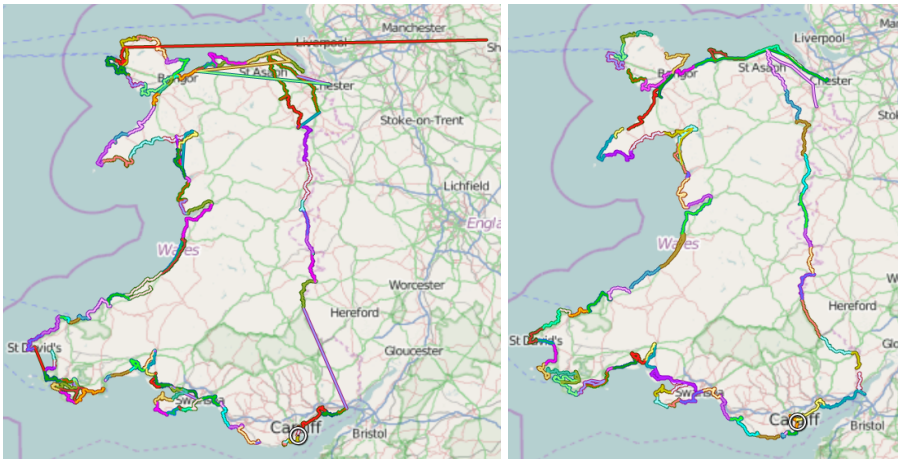

Fig. 2. Left image: The unprocessed Garmin location traces. The image shows some inconsistencies in the GPS data which are especially visible in the top of the map. Right image: The unprocessed ViewRanger location traces.

Computing group[4] in MIT. The sensor measures the 'galvanic skin response' which is basically a measure of how strong someone is sweating. The sensor has also a built in accelerometer which measures the acceleration in three axis. Furthermore, the temperature of the skin is recorded. The format of the data is simple CSV which allows easy processing.

Extract from one CSV file recorded by the AffectivaQ sensor

```
Log File Created by Q Live - (c) 2012 Affectiva Inc.
File Version: 1.01
Firmware Version: 1.71
UUID: AQL221200DG
Sampling Rate: 8
Start Time: 2013-05-08 09:16:30 Offset:+01
 Z-axis | Y-axis | X-axis | Battery | Celsius | EDA(uS)
-----------------------------------------------------------
-0.220,0.370,-0.900,-1,22.000,0.032
-0.180,-0.060,-0.850,-1,22.000,0.020
-0.170,-0.040,-0.900,-1,22.000,0.020
-0.160,0.150,-1.340,-1,22.000,0.030
-0.360,0.000,-1.080,-1,22.200,0.034
-0.450,0.530,-0.980,-1,22.000,0.037
```

The format of the data is pretty simple. The first three columns give the acceleration of all three axis which is measured in units of gravitational force (g).

[4] http://affect.media.mit.edu/

The Battery column can be ignored because its an artifact of the sensor. The next two columns measure the skin temperature and the skin conductance. The response of the skin is measured in micro Siemens (μS) which is the inverse of the electric resistance. It is derived by applying electric potential between two points of the body and measuring the resulting current. Generally, a resulting EDA signal comprises of two major components. One is the Skin Conductance Level (SCL) and the other is the Skin Conductance Response (SCR)[1]. The SCL component expresses a baseline of the signal. This baseline may be relatively high or low and differs between subjects as every person has different characteristics of the skin or a different state of arousal. The changes in SCL are slow and represent general changes in arousal as the overall baseline slowly increases or declines. The second major component expresses faster changes in the signal which may be related to a presented stimulus or certain events. They can be seen as peaks in the signal which originate back to the overall baseline. Regarding the physiology, EDA measures the level of stress or arousal someone is experiencing because these emotions are connected to the swat production of the skin. Figure 3 shows a typical signal from the dataset.

The data was sampled with a frequency of 8Hz which results in 2.36 million measurements for the whole walk. It is stored in 68 files with a overall size of 650MB and the readings cover about two thirds of the whole walk.

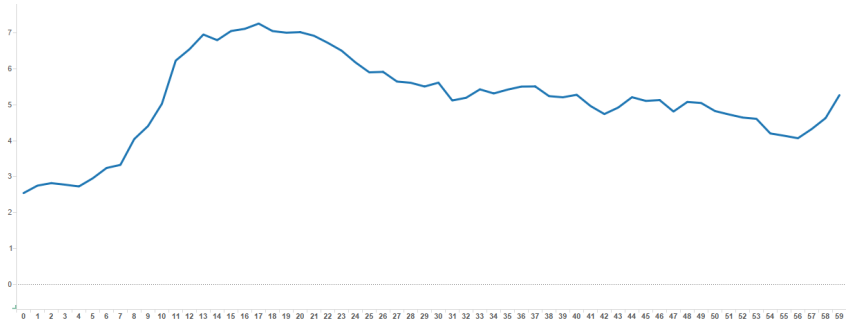

Fig. 3. Image showing an extract of the EDA signal over a period of one hour. The x-axis is showing the time in minutes and the y-axis shows the EDA amplitude in μS. Both changes in SCL (*rise starting at minute seven*) and SCR are clearly visible (*little peaks in the signal*).

ECG (Electrocardiography)

The last major numeric data source is the heart activity signal. It was captured using a Actiwave Cardio[5] sensor. It also has a built in accelerometer which is the second source of acceleration data. The data is stored in the EDF+[6] which

[5] `http://vivonoetics.com/products/sensors/actiwave-cardio/`
[6] `http://www.edfplus.info/`

is an industry standard binary format. Additional, the files were converted into CSV in order to make them easier to read.

Extract from one CSV file converted form EDF+ to CSV

```
Time,1,2,3,4
0.000000,-163.776719,-4.759033,8.499129,0.272715
0.015625,-94.756530,,,
0.031250,-42.991389,-4.759033,8.499129,0.086909
0.046875,8.481294,,,
0.062500,94.464072,-4.759033,8.499129,0.272715
0.078125,197.701897,,,
0.093750,283.392216,-4.759033,8.298338,0.272715
0.109375,283.099757,,,
```

The first column gives the time offset in seconds beginning from the start time of the recording. The start time was documented in a meta file which can be joined with the EDF+ files. The next column contains the ECG signal which is measured in micro Volts (μV). It measures the change of electricity on the skin which is produced by the heart muscle while pumping blood through the body. The last three columns are the acceleration readings which are measured in meter per second squared ($\frac{m}{s^2}$). Every second line of the acceleration readings is empty as they are only sampled at half of the frequency of the ECG readings. The ECG is sampled at 64 measures per second, hence, the acceleration data has a sampling rate of 32Hz. This leads to quite a lot of data values as the recording almost covers a period of three and a half months. There are 35 files which typically contain 40 hours of recording. Overall, there are 250 million ECG measures which leads to a file size of 10GB for the uncompressed CSV files. However, the EDF+ files are compressed and only add up to 1.9GB.

Like the EDA signal the ECG signal contains different components. One of the most important components is a QRS complex which determines if there is a heartbeat[2]. It is typically expressed as two negative and one positive peak in the signal which occurs when the heart is pumping the blood out of its ventricles. There are further components of the ECG signal which are used to study health issues of the heart. However, in order to measure the frequency of the heart beats the QRS complexes are decisive. Figure 4 shows one beat of a ECG signal with all of its components. The duration of the intervals between the components as well as the amplitude values may indicate certain heart related problems. An example of the ECG signal captured by Alan is shown in Figure 5.

4.2 Qualitative Data

There are several qualitative data sources. They consist of the following:

- Text (written diary for every day)
- Photos
- Audio Recordings

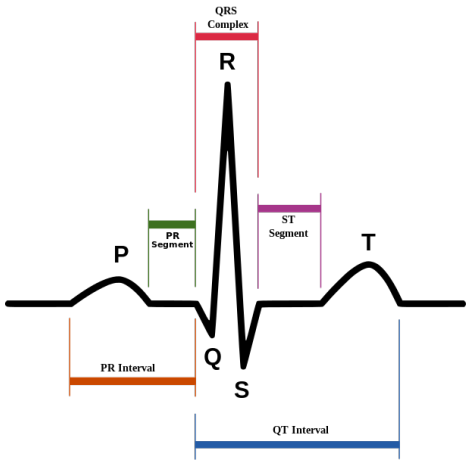

Fig. 4. Image showing the schematics of a ECG signal of a healthy heart. In order to identify the heart rate the frequency of the QRS complexes is counted per minute.

Fig. 5. Image showing an extract of the ECG data recorded by Alan. Comparing it to the ideal signal shown in Figure 4 one can clearly see the distinct beats of the heart.

Text
During his walk Alan wrote diary about every day of his walk. He wrote approximately 2000-3000 words each day which were published on his blog. His writings are mostly about the events that happened each day while he was walking but sometimes also about the people he met and their stories. He describes the places he visited and often gives some background information. Partially, he also writes about how he was feeling during his walk as there are texts which contain a lot about is well-being and others which contain mostly information on the walked day. Most of the days he embedded some pictures of special places into his texts and sometimes he even wrote little poems. Regarding processing, the text is stored in the JSON format.

Extract from one JSON file containing the blog entries


```
[{
    "id": 520,
    "title": "day 1  Cardiff to Newport",
    "permalink": "http://alanwalks.wales/2013/04/18/day-1-cardiff-to-newport/",
    "content": "[day 1]
The first day started at the {{National Assembly}} buildings in {{Cardiff Bay}}.
        .
        .
        .
    "excerpt": "The first day started at ...
    "date": "2013-04-18 23:55:15",
    "author": "alan",
    "categories": ["general"],
    "tags": {}
}]
```


Each file contains meta information about the text like the title or the timestamp when it was written. The most important part is the 'content' object as it contains the text itself. One problem is that it doesn't only contain the text but also references and the embedded images as HTML tags which need to be removed before beginning with the processing.

Photos & Audio Recordings
Alan took about 250 to 450 pictures each day which are available on his flickr page[7]. Altogether, he took about 19000 pictures of his walk using a standard digital camera. Some of the images are geo-tagged and contain further information on how they were taken.

The data also includes audio recordings he made additionally to his blog writings. The amount of recordings he created varies heavily from day to day. They range from five up to thirty each day. In his recordings he also described the things that happened on his walk as well as when he reached certain destinations.

[7] `https://www.flickr.com/photos/alandix/sets/`

Fig. 6. A picture showing Alan on the first day of his walk (left) and a picture showing him on the last (right).

5 Data Quality

Especially with this data set the data quality may be problematic. There are many problems that could arise since the data was created during a walk and these are not ideal conditions for some of the recorded streams. In order to produce error free data one needs a controlled environment which is only partly the case in the given data set. Alan often talks about the problems he had with the technology in his presentation or on his blog. One of his major struggles was to keep all of his devices fully charged and working properly, as there were quite a lot, and to not to forget anything. Each of his devices had a different workflow in terms of how to charge them, how long the battery lasts and how to synchronize the data with a storage device. This sometimes led to gaps in the data as he simply forgot to think of everything. Another problem was that he had connectivity issues, whether it was with the mobile signal or with a internet connection. Some parts of Wales have very poor signal coverage which also lead to errors in the data. Regarding the bio sensors one problem may be that sometimes the sensor was not perfectly operated, hence leading to wrong measurements and noise. For the data quality assessment I will focus on the GPS, ECG and Text data as these were my focus in the analysis part.

5.1 GPS Data Quality

The main goal of the GPS processing was to get one GPS track in one file that shows each route he walked for each day excluding any backtracking, times when he travelled by bus, wrong measures and times when he stayed at the same point for a longer time. As shown in Figure 2 there are some inconsistencies in the GPS data which need to be dealt with. The main problems are the following:

– Big jumps between two measures.

- Gaps in the data.
- Clusters when he slept or when he stayed in a city.
- Outliers.
- Accuracy differences of the both sources.

These problems are present in both of the data sources. Yet, for the ViewRanger source the accuracy is lower as it only records a Track-Point approximately every minute while the Garmin records it every ten seconds. Figure 7 shows a comparison of the different sampling rates. The Garmin device seems to be more appropriate in capturing the route than the ViewRanger app. However, this leads to a way higher amount of Track-Points and the high accuracy of the Garmin may not be necessary because the GPS traces do not need to be this accurate in order to show where he walked.

Fig. 7. Comparison between the accuracy of the Garmin GPS sensor (*orange line*) and the ViewRanger app using a smartphone sensor (*white line*). The difference in sampling rate is clearly visible as the ViewRanger only captures the rough direction and the Garmin the exact route.

The next problem is that there are some gaps in the data. The biggest gap is in the Garmin traces which can be seen in Figure 8. The gap extends to several days where there are no location measures from the Gramin device. Yet, they can be filled in with the ViewRanger data. One reason for this gaps may be the struggles with the technology which were mentioned before. In this case the problem was that he lost his Garmin device at the beginning of his walk and couldn't find it anymore. However, it was buried in his stuff somewhere and managed to record the first two days before the battery ran out.

Another problem are jumps in the data. This is the case when two successive Track-Points are very far away from each other and the location of the second Track-Point can't be associated with a probable location. The following Track-Points connect back to the correct route most of the time. An example of this

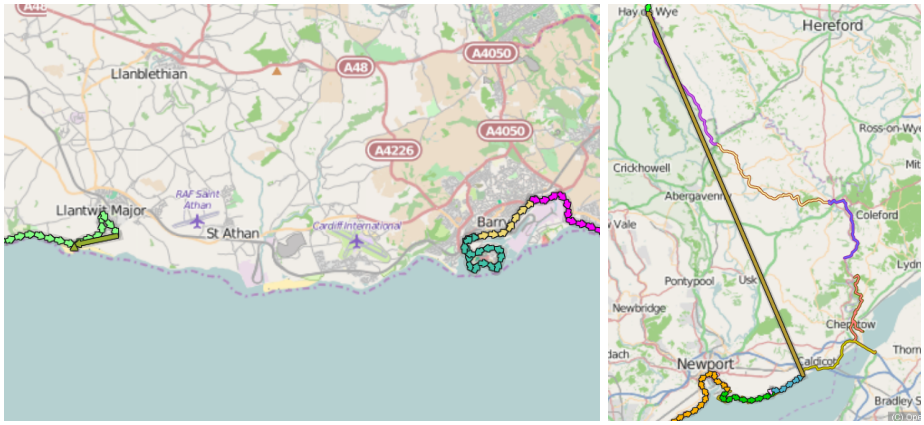

Fig. 8. Images showing the biggest gaps in the GPS data. Left image: One of the last days of the Garmin recordings is missing. Right image: The gap resulting from the lost Garmin device (*straight brown line*). The selected route is the Garmin source (visible at the arrows and thicker line). Fortunately, it can be filled in with the ViewRanger data.

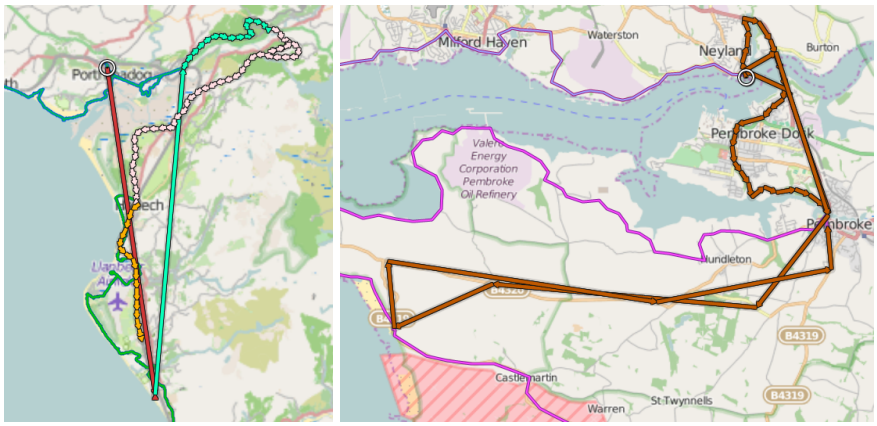

Fig. 9. Images showing jumps in the location traces most probably caused by a wrong measurement of the sensor. Left image: Garmin. Right image: ViewRanger.

behavior can be seen in Figure 9. This strongly indicates a wrong measurement of the GPS sensor but one has to keep in mind that there may be some correct outliers. One example of such a outlier can be seen in Figure 2 (left image, big orange line at the top of the image). This outlier was caused by a trip he took to England one time. He was flying with a plane in order to attend a wedding he was invited to and turned his Garmin device on for a short time. A reason for the wrong measures could be bad weather conditions or general connectivity issues with the satellites. Furthermore, there could be bugs in the software especially when using a smartphone app.

The last of the problems with the GPS data are occasions where he recorded data when he didn't move at all or didn't move much. Such situations are caused when he stayed in a city for some time or when he slept and kept on recording. The problem is that these don't add much information to the data and unnecessarily increase data amount and could confuse a user while browsing the GPS data. Figure 10 shows a example of such a cluster. There are approximately 1000 Track-Points on the same spot. Surprisingly, the location of the points doesn't agree much and varies quite a lot. The points vary about 20 meters around the center of the cluster. A reason for this could be that he was in a hostel that time and there were a lot of interfering signals around the sensor.

Fig. 10. Image showing a cluster of GPS Track-Points on the rote. There are about 1000 points contained in the cluster and the diameter is about 40 meter.

Elevation

Additionally to the location, the GPS devices recorded the elevation of the terrain. In order to find out how accurate the elevation was recorded I compared the elevation from the final cleaned GPS track with data from OpenStreetMap[8].

[8] `http://www.openstreetmap.de/`

In order to do that I extracted all the elevation values from the GPS track using a simple Java XML parser. After the extraction of the original elevation data the elevation for each Track-Point was replaced with elevation from OpenStreetMap and both sources put together in a simple CSV file. A comparison of an extract of both sources can be seen in the line plot in Figure 11. The plot shows the days he walked in the first month as well as some days of the next. At the beginning the streams differ a bit. Especially, on the last days of April, but the difference gets smaller as the time progresses. On the next month both streams pretty much give the same elevation which continues for the rest of the months. Moreover, the correlation between both sources is very high ($r = 0.988$), hence, as the data from OpenStreetMap is a trustworthy source, the recorded elevation data from the GPS devices seems to be accurate most of the time. Yet, it can be noted that the GPS elevation gives some negative values which are very likely wrong measures as he never went underwater diving.

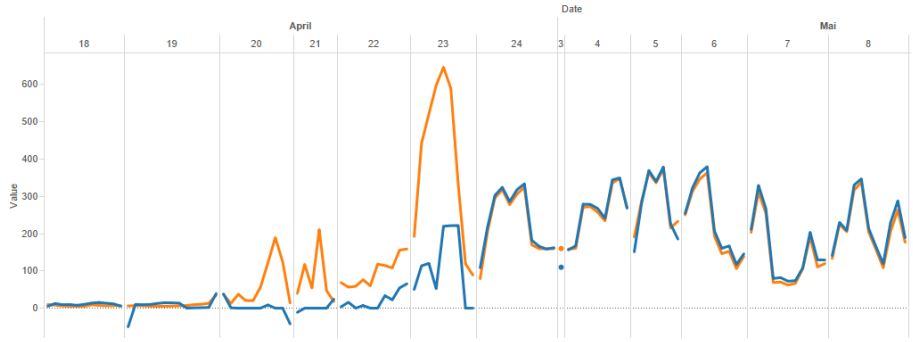

Fig. 11. Line plot comparing the elevation acquired from the GPS sensors (*blue line*) and the elevation acquired from OpenStreetMap (*orange line*).

5.2 ECG Data Quality

One of the major data sources is the ECG signal. The quality of this stream is very important as it determines how accurate the results of the analysis are (in this case heart rate extraction). On major problem can be seen in Figure 12. Some parts of the signal are very exaggerated as shown in the image. This probably leads to a too high heart rate for these parts of the signal. Unfortunately, big parts of the whole signal seem to show this extreme signal or something similar. Some reasons for this kind of signal could be the following:

- Partially detached sensor.
- Wrong skin preparation.
- Pad movement & unfitting pad type.

– Wrong positioning of the sensor.
– Interference caused by water.

For correct ECG results it is crucial that the sensors are correctly connected to the body[3]. There are many things which need to be considered when attaching the sensors which may sometimes did not work perfectly during the walk. It is very likely that the requirements for a good recording were not always met regarding the length of the recordings. One could easily imagine that maybe Alan wasn't as thorough in the end as in the beginning of his walk. Furthermore, it may be possible that he didn't recognize when the sensor detached.

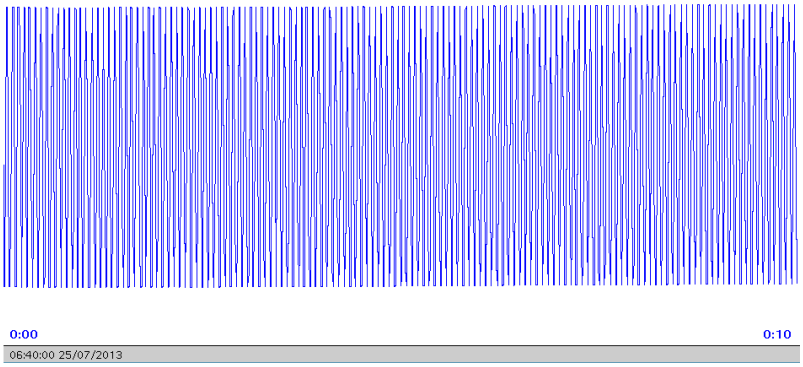

Fig. 12. Extract of the ECG signal which shows a very exaggerated signal. Probably caused by connection problems with the sensors and the skin.

Another issue is that there are hidden missing values when the signal dropped to zero for some time. The biggest gap is in the end of his walk and lasts for one day. The result of the heart rate extraction is a signal of beats per minute ($\frac{bp}{m}$), hence the previously mentioned gap accounts for about 1400 minutes where a heart rate can't be calculated due to the zero signal of the ECG data. Overall, there are 2450 hidden missing values but aside from the big gap they are more or less distributed over the whole signal.

5.3 Text Data Quality

The evaluation of the text data 'quality' will be with respect to the analysis goal. The aim was to extract a mood measure from the daily blog entries which should give information about how he felt during his walk and if the days were rather cumbersome or not. As the analysis approach is dictionary based the quality of the results is heavily depending on what words Alan used in his diary. Hence, there may be situations were the text seems to have a very bad sentiment but

doesn't reflect the real mood he was having at that moment. This also could be the other way around. Such behavior could result from situations where he used a lot of 'negative' or 'positive' sentiment words, depending on the used dictionary, in order to describe something which happened on his walk but isn't connected to his sentiment at all.

As the blog entries are a recall of every day he walked, Alan mostly writes about the events that happened and the things he saw during his walk. Of course, he wrote about when he was not feeling good or when something bad happened (like when he lost his Garmin device) but there are days which contain more description about his feelings and days which contain less. As he is very detailed with his overall descriptions he often enthuses about the beautiful landscape and the landmarks he encountered. This leads to a lot of positive sentiment words in the texts which have great influence on the analysis result. These kind of descriptions can be considered to correctly represent the sentiment of his walk as this is certainly connected to his his mood and the goodness of the day. In contrast to this he often recalls historic events which happened at certain places he visited. These descriptions sometimes contain a lot of negative sentiment words but can't be considered to display his sentiment on the walk. In order to evaluate the quality of the calculated sentiment measure I conducted a small study with Alan where I asked him how he recalled the sentiment of certain days. The results of this study will be further discussed in the sentiment analysis section.

6 Analysis and Preprocessing

This section is about the cleaning and analysis of the data sources. The focus lies on the cleaning of the GPS data and on feature construction. The main goals are the following:

- Filtering, Cleaning and Completion of the GPS data in order to get a smooth route without errors.
- Heart Rate extraction of the ECG data.
- Sentiment Analysis of the blog entries.

Furthermore, I will introduce all tools which I used to accomplish this goals as well as to give a rough guideline on how to deal with certain problems.

6.1 GPS Processing

GPS Tools
In order to process the GPS data I used the following tools:

- GpsPrune[9]
  Application for viewing, editing and converting GPS data. It has an option to import elevation data from OpenStreetMap which is used to get the elevation

[9] `http://activityworkshop.net/software/gpsprune/`

information for every Track-Point of the data. Then it can be compared with original elevation measured by the sensors.
- GPS Track Editor[10]
  Another application for viewing, editing and converting GPS data. Is is very fast and can deal with big amounts of data. The program also includes some useful filters which allow to remove a lot of wrong data.
- GPSBabel[11]
  Powerful command-line tool for GPS processing. It has a lot of functions to filter, merge and convert the data into different formats. It also allows to change the structure of the GPX format, to do time corrections and it supports almost every common GPS data format.

Since the Garmin sensor produced several files I merged all files into one using the merge option of the GPS Track editor. Also I used the Garmin data as 'primary' data source because it has a higher sampling rate and the sensor probably creates results which are more trustworthy than the ViewRanger data. As mentioned in Section 5.1 there are some problems which are inherent to the location traces. One goal was that the data only includes locations he went by foot. Sometimes he took the bus in order to get to a guesthouse when he hadn't enough time to walk there. Also, he had to do some backtracking because he didn't want to cover distance with a vehicle or he got lost. One easy way to remove such Track-Point is to filter them by speed. However, the data does not contain speed information but this can be easily calculated with the time and the location. After loading the data into GPS Track Editor the program automatically calculates the speed between every Track-Point and allows to remove points which exceed a threshold. Typically, the average speed of a vehicle is between 40 $\frac{km}{h}$ and 100 $\frac{km}{h}$ depending whether driving in a city or on a highway. The average speed of a person walking is lower than 10 $\frac{km}{h}$ especially while walking long distances for a longer period of time. Hence, all points with a speed higher than 15 $\frac{km}{h}$ can be safely removed without deleting any walking locations and to get rid of most of the vehicle traveling. Additionally, jumps in the data can be removed this way because the jumps lead to a very high covered distance but with a low amount of time between the points, hence, resulting in a high calculated speed. The filter removed about 5% of the Track-Points. The results of the filter were very good as it could be seen that a lot of double routes and jumps were removed.

The next step was to remove the clusters in the data. In order to address this problem a minimum-distance filter can be used. This filter deletes all points which are in a specified distance to the previous point. The higher this distance the more points get removed. The clusters mostly happened when he slept or when he stayed in a city for some time. Mostly, the points in this cluster are in very close range to each other as he was not moving at all. Ideally, the measurements would be at the exact same point but due to some disturbances the points vary around a location. These inaccuracies are typically rather small, yet,

[10] `http://www.gpstrackeditor.com/`
[11] `http://www.gpsbabel.org/`

in buildings they can be somewhat big as seen in Figure 10. Experiments show that a value of 10 meter performs well and removes most of the clusters and still preserves a good accuracy. Since all other points which do not belong to a cluster are affected by the filters as well the data is also simplified. As shown in Figure 13 the filter maintains enough information so the direction of the route is not altered. There are only minimal changes as a sampling rate of a Track-Point every 10 meters is sufficient to describe the path correctly. Moreover, potential jitter is removed because points with unnecessary detail are discarded. This is another advantage of the filter as the track is straightened and simplified. After the filter was applied the track contains about 125000 locations which is one third less Track-Points than before. This reduces storage costs and speeds up a visualization as less points need to be displayed. After the data was filtered most of the issues were resolved. Yet, there are still some inconsistencies as the filters do not work perfectly. There are some situations which were not covered by the filters but should be excluded from the data. In order to do this the remaining problems were resolved by browsing through the data manually and removing points by hand. This can be easily done with GPS Track Editor as the program is very fast and offers a nice user interface.

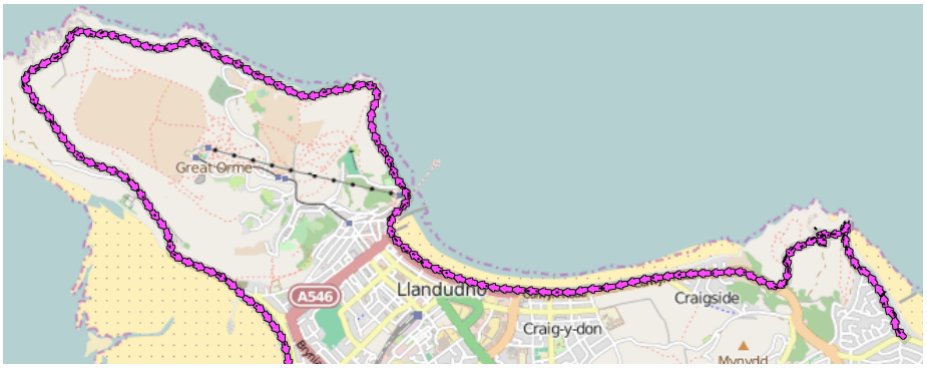

Fig. 13. Image showing a part of the Track before and after distance filtering. The *purple arrows* show the result of the filtering and the *dashed line* indicates which points were removed. It can be seen that the route of the path is almost not altered although more than a third of the points were removed by the filter.

Lastly, there were some gaps in the Garmin data which needed to be filled. Fortunately, most of the gaps are covered in the second GPS source so the ViewRanger data can be used to fill in the gaps. In order to do that both sources were merged using the GPSBabel tool. This function just puts all Track-Points from both sources together an sorts them by timestamp. As there is great overlap between the two sources not only the gaps are filled in but the rest of the routes is merges as well. Due to the different sampling rate and accuracy (shown in Figure 7) of both devices the merging adds some noise to the Garmin data.

This also results from a slight time difference between the devices. Hence, it can be reasoned that the merge for the overlap does not add much information to Garmin sources so only the gaps were filled with the ViewRanger data. Figure 14 shows images of the data after every filtering step.

6.2 Heart Rate Extraction

The second analysis task was about heart rate calculation from the ECG data. Since the goal is not to find out about any health related things in the ECG data, there is not much to gain from looking at the raw ECG stream only, due to the huge amount and the difficult interpretability. Therefore, with respect to my goals, the most interesting thing to do is to extract the heart rate from the data. After calculation it can be connected with other sources of the dataset and may reveal interesting correlations.

ECG Tools
All tools I used for the analysis of the ECG signal are from PhysioNet[4]. It offers a big database of physiologic signals and a a lot of related open-source tools. The programs are provided via the PhysioToolkit[12], which is a large library of software for a big variety of analysis and processing tasks of physiological signals including software which is able to identify heartbeats and calculate a heart rate signal. All software is available via the WFDB (WaveForm DataBase) Software Package. The programs can be used on every platform and there are detailed tutorials on how to install and run the tools on your machine[13]. Alongside the analysis tools there is also a nice program for viewing WFDB compatible signals called WAVE [5]. It is very useful for viewing and verifying the results of the analysis algorithms as well as browsing through the data.

The first thing to do was to make the data compatible to the WFDB Software Package. They support various formats but usually it includes a header file containing meta information, a signal file containing the data and a annotation file containing analysis results. Fortunately, the data was already in the EDF+ format so it could be easily converted into WFDB compatible records using the tool 'edf2mit' [6]. The converter creates two files for every EDF+ file. One header file and one signal file.

After the data is readable and in the correct format we can start with the heart rate extraction. Generally, it consist of two steps. First we have to know where the heart beats occurred in the signal. This is done by annotating the original signal with markings of each heart beat so we can later calculate the heart rate because we know how many beats occurred in a specific time period. There are several algorithms in the PhysioToolkit which can be used to annotate a signal file. The output of each of this programs is a annotation file which

[12] `http://www.physionet.org/physiotools/`
[13] `http://www.physionet.org/physiotools/wfdb.shtml`

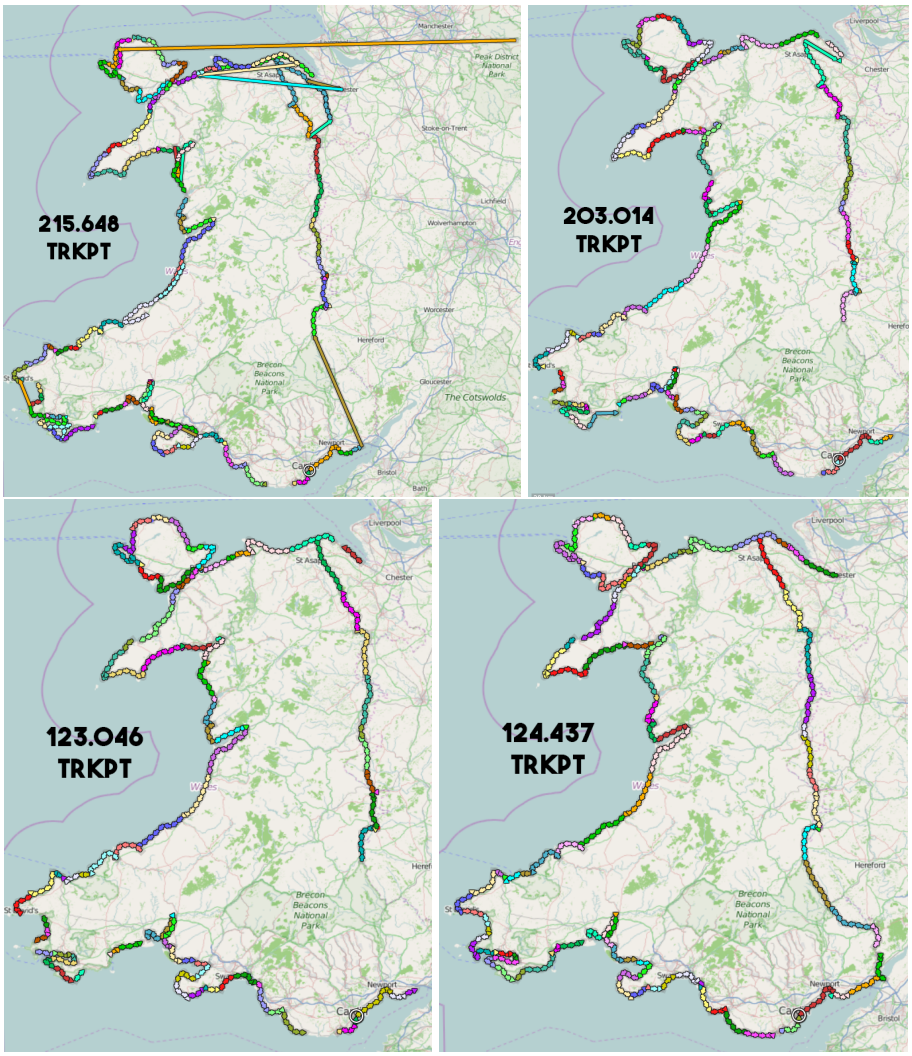

Fig. 14. Images showing the GPS data after every filtering step. The *black number* gives the amount of Track-Points contained in each image. Top row: The left image shows the raw Garmin data. The right image shows the data after the speed filter and some manual filtering. Bottom row: The Track after the distance filter was applied is shown in the left image. The right image shows the final Track after the View Ranger data was used to fill in the gaps. Note that the number of points doesn't increase much in the last step because of the low sampling of the ViewRanger data.

contains the time information when a beat was detected by the algorithm. There are three major programs which i used for my experiments:

- sqrs [7]
  Simple and fast QRS detector.
- wqrs [8]
  Also simple and fast QRS detector but generally should work a bit better than sqrs.
- gqrs [9]
  Reasonably fast detector that is optimized for human ECG signals. The program can also be adapted to children and animals.

All of the detectors have a threshold parameter which can be adjusted if the results are incorrect. They all follow the same behavior, so if too many beats were recognized one can increase the threshold if there are too many false detections and decrease it if too many beats are missed. I tried all three detectors and used the best for the final detection. The first detector 'sqrs' was not able to detect the beats and skipped most of them even if I used different threshold levels. The second one 'wqrs' detected too many beats and had the same problem with the threshold level. The best results could be achieved with 'gqrs'. However, I had to try a lot of different thresholds in order to produce good results. The best were achieved with a value of 500000. The results can be viewed using WAVE. Figure 15 shows an extract of the signal with marked heart beats found by 'gqrs'. For this part of the signal the algorithm found every beat and marked them correctly. However, the signal is often not as clean as in this example which mostly leads to too many detected beats. In the case of the exaggerated signal (Figure 12) it is difficult to find out what the real signal was regarding the extreme nature of the data. This problem is dealt with in the next step.

After all beats were detected we can start with the heart rate calculation which is the second part of the whole process. Now, all records have a additional file containing the annotations for the bets. This file is used in further programs of the WFDB software package in order to calculate a heart rate signal. There are two programs which provide this functionality:

- ihr [10]
  This program reads an annotation file and produces an instantaneous heart rate signal. The output does not have equal time intervals between the samples.
- tach [11]
  Works very similar to 'ihr' but with difference that the time interval between the output is equal and can be specified. It also has some means for outlier rejection and smoothing.

First, I used 'ihr' in order to calculate the heart rate. Yet, it has some disadvantages over 'tach'. The goal is to create a signal containing a heart rate for every minute so 'tach' would be better suited for this task because there is a equal amount of values for every minute. Also it has the previously mentioned outlier

rejection and smoothing. For this reasons I settled with 'tach' for the heart rate calculations. I configured 'tach' to sample the output with 1 Hz so there is a heart rate value for every second. It outputs simple text and can be configured to to also show the sample time in various time formats. The last thing to do was to calculate a heart rate for every minute of the signal. In order to do this I calculated the mean over every 60 second of the previously generated heart rate signal. However, there are some very high values which result from extreme parts of the ECG signal. Therefore, all values lower than 40 $\frac{bp}{m}$ and higher than 230 $\frac{bp}{m}$ were excluded from the mean calculation because they can be considered to be wrong measurements. The resting heart beat of an trained athlete is 40 $\frac{bp}{m}$ (70$\frac{bp}{m}$ for most adults) so I chose this value for a lower bound. Also, a frequency higher than 230 is very unlikely to be a correct measure so I excluded this values. The resulting table for all of the data contains 64000 values for all minutes which are covered with the signal.

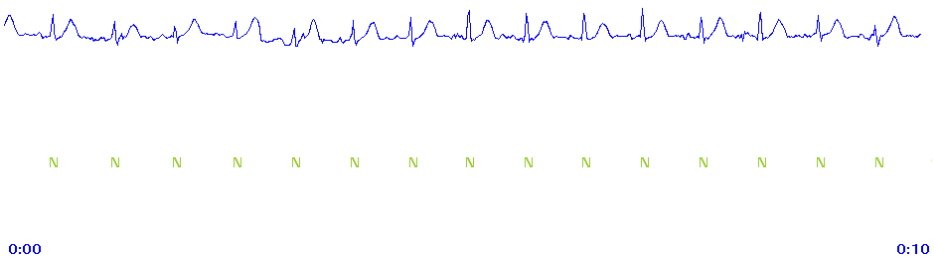

Fig. 15. Image showing an extract (10 seconds) of the ECG signal and found QRS complexes (heart beats) by the 'gqrs' program. The beats are marked by the capital '*N*' under the signal waveform.

I followed a very nice step-by-step tutorial[14] on PhysioNet in order to extract the heart rate. The tutorial covers how to create compatible records and explains most aspects of how heart rate extraction with PhysioNet software works. Furthermore, there is also information on how to do further analysis of the results.

7 Sentiment Analysis

The last task was about sentiment analysis of the text written by Alan. The goal was to extract a sentiment measure for each of the blog entries indicting how he might have felt on a particular day.

Sentiment Analysis Tools

The analysis was done using the KNIME Analytics Platform[15]. It's a free soft-

[14] `http://www.physionet.org/tutorials/hrv/`
[15] `https://www.knime.org/knime`

ware developed at the University of Konstanz and provides a lot of functionality in terms of data preprocessing & transformation, data mining and visualization for a wide variety of applications including text processing. The workflow of KNIME is node based and very intuitive and allows the user to easily implement rather complicated workflows. Moreover, they provide a lot of tutorials (also video tutorials[16]) on their homepage on how to use KNIME or on how to write own nodes if someone needs further functionality. The standard KNIME version does not contain text processing tools but all extensions can be easily added using the corresponding option in the user interface.

Since, the format of the text was JSON the first thing to do was to bring it into a KNIME readable from. KNIME supports all sorts of formats but the easiest are simple text files like for example CSV. Hence, the JSON files had to be converted to CSV. There are many little online tools and scripts but none of them managed to convert it into a CSV format so it could be correctly read by KNIME. Therefore, I used a simple Java toolkit for JSON called json-simple[17]. The toolkit provides functions to parse the JSON files and read the values of the objects. So, I just had to iterate over all files and write the content of the 'content' object into a text file with proper row and column delimiters. The row delimiters are important because mostly the texts are very long so a linebreak can't be used as indicator when a new row starts. However, another option would have been to remove all linebreaks. After the CSV file was created we now can start to read it in KNIME. There are two nodes in KNIME which can read in text data. One is the File Reader and one is the CSV Reader. In most cases the File Reader node is better suited but it has no option to configure the row delimiter, therefore, I used the CSV Reader which worked fine. The read in data consists of two columns, one containing the title of the day and one containing the text as Strings. As mentioned in Section 4.2 the 'content' object does not only contain the text itself but also HTML tags of embedded images, hyperlinks, book references and some other characters which do not belong to the text itself. In order to remove these I used String Replacer nodes which can be configured to search for a regular expression and replace the found substring with a specified String, in this case the empty String. After everything not belonging to text was removed I converted the content Strings to Documents which is the basic data type needed for text processing.

The sentiment analysis consists of four major steps, which are Enrichment, Transformation, Preprocessing and Analysis. First we need to enrich the text with so called tags. This means that we assign a tag to some words of the text on the basis of a certain context. These tags provide further information about the tagged word which can be later used for analysis. On of the most common tagging applications is part of speech (POS) tagging. It assigns each word a tag according to its part of speech in the sentence, so for example a verb in its base from is tagged with 'VB' or a noun in plural form is tagged with 'NNS'.

[16] `https://www.youtube.com/channel/UCRbKmV_XYB7C12SPBokLVHQ`
[17] `https://code.google.com/p/json-simple/`

The tagging is done by an algorithm that both uses the definition of a word as well as the context the word stands in for the assignment of a tag. Another method for tagging can be done using dictionaries which was used in this case. The Dictionary Tagger node in KNIME takes a list of words and tags each word in the text which is also in the dictionary with a specified tag. For sentiment analysis the idea is to tag the words according to their sentiment. Hence, we need two dictionaries containing negative and positive words that specify which words are associated with a negative of positive sentiment. The used dictionaries are further discussed in Section 7. There are also sentiment specific tags (e.g. 'NEGATIVE', 'VERY_NEGATIVE', 'IRONY') which I used to tag the words.

After the enrichment is done the next step is to do some transformation. The most common technique is to create a Bag of Words (BoW) which is also a input requirement in KNIME for some of the nodes and for the following steps. The Bag of Words Creator generates a list of distinct words which are contained in a document. So the output of the node are two columns, one containing a row for every word (called terms in KNIME) and the other containing the corresponding document. Words which are the same only appear once in the list. Originally, a BoW is used as an numeric representation of a text where a position in a number vector represents a distinct word and the number itself is the frequency of the word. This is commonly used for text classification.

The last but one step is to do some further preprocessing of the before created word list. This means to remove things which do not hold information for the analysis like punctuation, stop words, numbers and small words. Also we can filter by, in the first step assigned, tags and stem the words with various algorithms. The Text Processing extension of KNIME offers a lot of preprocessing nodes and I used the following:

- Number Filter:
  Removes all terms containing only numbers.
- Punctuation Erasure:
  Removes all punctuation characters from the terms.
- Stop Word Filter:
  Removes words which do not contain important significance for analysis like 'the', 'and', 'to'. Either a own list is specified or the node uses built-in lists which are available for several languages.
- Case Converter:
  Converts all terms to either upper- or lower case. I configured it to convert everything to lower case.
- N Chars Filter:
  Removes all terms containing less than a specified amount of characters. All terms consisting of less than three chars were removed.

Another important preprocessing task is stemming. Stemming is used to reduce a word to its root form or word stem, therefore, two words which have the same meaning but a different form can be grouped together. For example the words 'follow' and 'following' have a similar meaning and will be reduced to the same stem by a stemming algorithm. Here, I didn't apply stemming because

this doesn't lead to advantages in the analysis. One reason for this is that the used dictionary was not stemmed neither and stemmed words are confusing in a visualization.

The last step is now to calculate a sentiment measure from the preprocessed words. In order to do this, first all terms which were not tagged, hence probably not containing sentiment information, were removed from the BoW by first converting the tags to Strings (adds an additional tag column containing the tag as String) and then excluding all rows which were not tagged. Then I converted all 'POSITIVE' tags into 1 and all 'NEGATIVE' tags into -1 Integers using a Java Snippet node (allows to execute arbitrary java code). This leads to a scale representing the sentiment with a range of 1 to -1 where a positive value represents a positive sentiment and a negative value a negative sentiment. The last task was to group all terms by their corresponding document and aggregate over the sentiment values by calculating the mean for each document. This was done using a GroupBy node in KNIME. The result is a value for each document indicating its sentiment on the basis of the frequency of 'positive' and 'negative' words contained in the document. If both counts are the same the sentiment measure is zero indicating that the text seems to be neutral. If it contains only positive words its 1 indicating a very positive sentiment and the other way around. Values in between result from other distributions of the sentiment words.

Sentiment Dictionary

On very crucial part of the sentiment analysis is the used lexicon because it decides which words will be interpreted to carry a positive or negative meaning in the examined texts. Often the dictionaries are domain specific because there can't be a universal list for every meaning of every topic. For example if we consider the word 'love' it would generally be associated with a positive sentiment. But if the topic is about tennis sports 'love' means that a player did not score any points yet, hence it wouldn't represent a positive sentiment in this case. There are also some universal lexicons which are not adapted to a special domain. One of these dictionaries is the sentiment lexicon[18] created by Hu and Liu [12] which was crafted over many years and provides a very good starting point for the analysis. This was also the dictionary which I used form my first experiments. It contains about 6800 words which comprise of 2000 positive and 4800 negative words.

There are other dictionaries which cover domains like customer reviews or twitter posts but there is currently no dictionary for the walking/traveling domain. Therefore, I tried to create my own dictionary which should be better suited to the given domain. There are several methods which can be used to create a sentiment directory. One method is Simple WordNet Propagation. It uses WordNet[19] relations in order to generate a dictionary beginning with seed sets of a specific domain. WordNet is a large lexical database in English which

[18] available at `http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html`
[19] `https://wordnet.princeton.edu/`

provides means to link words together based on synonyms, their sense and semantic relations. Starting with the seed sets the algorithm searches for similar words following the WordNet relations. The techniques has several parameters: number of iteration, seed sets and WordNet relations which should be used. In each iteration the algorithm uses the synset (synonym set: refers to words with similar meaning and the list of words created after every iteration of the algorithm) which it created in the previous iteration. This way the seed sets are extended and a wider range of meanings is considered in every iteration of the algorithm. The only thing to do, in order to use this for sentiment lexicon generation, is to use suitable seed sets. Generally, every topic or domain can be used in the seed sets but in this case two seed sets, one with positive and one with negative words, are used. Table 1 shows an example of the algorithm output with two small seed sets. The seed sets should cover the domain so the algorithm can expand them in the correct meaning. Therefore, I manually created two seed sets of words which are associated with a positive and negative sentiment in the walking domain. In order to do this I looked at the two best and worst days (based on my previous sentiment calculations) of his walk an searched for words which could be useful. Additionally, I did some brainstorming and tried to come up with some words on my own. For example terms associated with bad weather or depleted supplies like 'deluge', 'downpour', 'hunger', 'thirst' could certainly indicate negative sentiment. Lastly, Alan has a list[20] of other walkers on his blog. I took a look at them and chose two other blogs (Walk Wales 1027[21] and Seaside Donkey[22]) to search for possible seed words. As both blogs wrote quite a lot I took the texts and extracted all words which are also in the directory of Hu and Liu, resulting in two lists which I manually scanned for suitable words. This was also done in KNIME. Some of the words which I found can be seen in Table 2. Now the algorithm could be run with the created seeds. In order to improve the quality of the results I manually checked the synsets after every iteration and removed words which did not suit the domain. However, one problem is that the WordNet relations sometimes lead to intersections of the synsets, meaning that both lists contain some words which are in both sets. To deal with this problem I removed the overlap from both lists and manually decided for every word if it belongs to one of the sets or if it should be excluded. The final lexicon contains 3300 negative and 2500 positive words. A evaluation of the dictionary follows in the next section.

Study & Evaluation of Results

As mentioned in Section 5.3 the text may not perfectly represent the sentiment. In order to evaluate the results of my sentiment analysis approach I conducted a small study with Alan in which I asked him to give a assessment of some of the days. The study was divided into two parts. In the first part I presented every third day and the task was to recall his sentiment and assess it on a

[20] `http://alanwalks.wales/walkers/`
[21] `http://www.walkwales1027.com/`
[22] `http://seasidedonkey.co.uk/`

Table 1. Table showing the results of the Simple WordNet Propagation algorithm after one iteration for the seed sets shown in the first row. The seeds are examples for possible words which could be used in order to create a sentiment dictionary. The next iteration would use the resulting synsets as new seeds. The *character* after the words specifies the part of speech.

SEEDS	excellent,superb	horrible,awful
SYNSET	brilliant.s,excel.v,excellence.n	atrocious.s,awed.s,awful.s
	excellence.n,excellent.s,good.a	awfulness.n,bad.a,cattiness.n
	luster.n,magnificence.n,nice.a.	reverent.a,terribly.r,unpleasant.a
	superb.s,superior.a	dirty.s,extraordinary.a,frightful.s
		frightfulness.n,grotty.s,hateful.s
		impressive.a,nastiness.n,nasty.a

Table 2. Extract of the seed words used for the Simple WordNet Propagation.

Positive Seeds	Negative Seeds
marvellous,wonderful,glorious	sad,lost,wasted
idyllic,lovely,breathtaking	struggle,dangerous,abyss
optimistic,impressive,amazing	burn,thirst,hunger
comfort,freedom,exciting	tough,stress,trouble
.	.
.	.
.	.

rating scale. The scale was divided into seven options ranging from 'very good' to 'very bad' (with the steps: good, rather good, neutral, rather bad and bad in between). This scale can be translated into my sentiment measure by linearly mapping each option to the range between 1 and -1 (very good $\widehat{=}$ 1, good $\widehat{=}$ 0.67, rather good $\widehat{=}$ 0.33 and so on). The rating scale with this mapping was chosen because it is easier to give a rating with a finite number of options than rating something in real numbers. The second part was about specifying the best and worst days which he remembered. The study was created as an interactive PDF document with fill in forms so it was as easy as possible to undertake the study. I also included clickable links to the blog entries which may help remembering the events. The results of the sentiment analysis were not shown to Alan beforehand so that he was not biased by my findings. Also the days were presented chronologically which may help with the recall of the events as one can remember the sequence of the walk without jumping in time.

After the study was completed the results could be examined. First, the rating scale was mapped to the range of the sentiment measure and written into a CSV file. Alongside the received measure I put the timestamp of the corresponding day so the values can be displayed with the other streams. One problem was that Alan gave more than one assessment for some of the days. Those days typically started very good and ended very bad or the other way around. One more reason may be that some bad events happened but the overall walking was not bad at all. Hence, assigning one single value for the whole day is difficult so he gave two assessments. In these cases I just took the average between both values because it has to be comparable with my results. Also, the characteristics of these days will certainly be contained in the texts as well, thus if half of a day was good and the other half was bad the text should contains positive words in the first half and negative one in the second half resulting in the overall average regarding the calculation of my sentiment measure. For the second part of the study Alan gave two lists of the worst and best days of the walk. I mapped those days to either 1 or -1 and added them to the CSV file. Sometimes he also gave little notes explaining his decisions. Overall Alan rated 38 days which can be compared to the analysis done in KNIME.

First, I compared the study results with the measure derived using the dictionary of Hu and Liu. Both sources were displayed alongside in a bar chart so the values could be easily compared. The calculated values agree with the polarity of the study results except in three cases. Also, the study results are more extreme regarding the amplitude than the calculated values. This is no surprise as it would be very unlikely that a text which was assessed with 'very good $\widehat{=}$ 1' by Alan only contains positive sentiment words. Hence, the values are always lower than the study results. After visually comparing the values I calculated the correlation between them ($r = 0.467$). The correlation is not very high which mostly results from the difference in amplitude but as the polarity of the values is the same in most cases the the calculated values give a rough tendency. Unfortunately, the correlation got worse ($r = 0.347$) using my own

dictionary. Some reasons for the low correlation (results of both dictionaries) could be the following:

- Sentiment dictionary (Hu & Liu) doesn't fit to the domain.
- Text doesn't represent sentiment for a particular day.
- Assessment may be not entirely correct as the study was conducted one and a half year after the walk.
- Simple WordNet Propagation not powerful enough to compete with crafted dictionary of Hu and Liu.

I further investigated some of the days and looked at the texts and the sentiment words used for the calculation. Figure 16 shows comparing Wordclouds for the days I looked at. For each day two images are given which were created using the dictionary of Hu and Bing on the one hand and on the other my own dictionary. The color of the words in the Wordclouds represent positive (green) and negative (red) words tagged by the used dictionary. The saturation of the color measures how often the word was seen in the texts. High saturation indicates high occurrence frequency and low saturation the opposite.

- day 1 Cardiff to Newport

Hu & Bing	WordNetPropagate	StudyResult
0.0625	-0.081	1.0

This was the first day of his walk, hence he was very optimistic and enthusiastic about the walk. Therefore, he gave this day a very good rating despite the fact he lost his Garmin device early on. When we look at the Wordclouds (Figure 16 top row) it can be seen that the dictionary of Hu and Bing stresses on very generic words such as 'well' and 'like' but they don't have to be related to the sentiment. Overall, there are a bit more positive than negative words resulting in the slightly positive sentiment. My own dictionary further focus on more meaningful words in this domain like 'stormy' and 'tired' but all in all the text doesn't contain much about the sentiment, hence the study results could not be reproduced in this case.

- day 21 Porth-y-Waen to Llangollen

Hu & Bing	WordNetPropagate	StudyResult
0.013	0.086	0.0 (1.0 ? -1.0)

This was one of the days were Alan gave two assessments. In this case it is especially difficult because he both gave the best and the worst rating. As mention before the assumption is that this results in the average for the calculations which is roughly reproduced by both dictionaries. Again, the dictionary of Hu and Bing is a bit better but when we look at the Wordclouds the considered words are rather generic. In contrast the generated dictionary focuses more on other words which fit better to the domain.

– day 60 Aberystwyth to Aberaeron

Hu & Bing	WordNetPropagate	StudyResult
-0.397	-0.535	0.33

Here was the biggest disagreement between the study and the calculations. While Alan gave a low positive rating, both dictionaries assess the day as pretty negative. This results from the fact that the text contains a lot of negative words which were not associated with his sentiment. When we look at the Wordclouds we can see that there are way more negative than positive words in the text (Hu and Bing dictionary) and when we even remove some of the less significant words (own dictionary) the measure gets even lower. On this day he did not write much about how he was feeling.

In conclusion, the calculated sentiment measure only moderately reproduces the results of the study. Furthermore, the study may also be biased because it was done a lot of time after the walk. Also, the texts do not contain a lot of sentiment information and sometimes even none. Yet, as the polarity and the general tendency is the same for most cases my calculations can be used to complete the data of the study and be seen as a hint for the real sentiment of each day.

8 Visualization

The goal of the visualization was to to allow easy comparison and browsing of the the data streams. Therefore, two major visualizations were created. One showing the calculated sentiment measure and comparing the different lexicons and one showing the quantitative data sources and making a visual comparison.

8.1 Visualization of Sentiment Data

Tools
The first visualization was created using Java and Latex. In addition I included the TikZ[23] package which allows one to draw arbitrary shapes in Latex. It can be used inline normal Latex code and allows to quickly create simple graphics.

First, I created a little Latex/TikZ Toolbox in Java which allows one to write several Latex and TikZ commands to a file and therefore create a valid Latex document. The commands give the ability to initialize and end a document, to begin and end a table, to write text and hyperlinks, to write a arbitrary String, to begin and end the Tikz environment and to draw rectangles and lines. However, the validity of the documents has to be guaranteed by the user because there is no checking if the commands work together. Also, the resulting file has to be

[23] manual available at: `http://www.texample.net/media/pgf/builds/pgfmanualCVS2012-11-04.pdf`

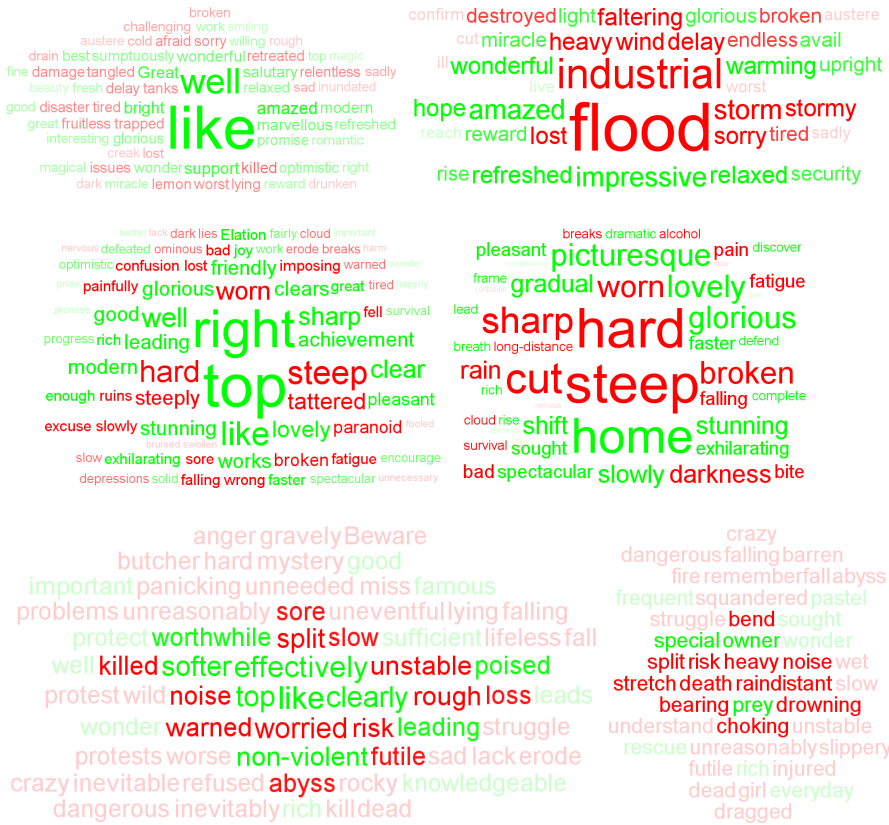

Fig. 16. Wordclouds derived from the blog texts using different sentiment dictionaries. The *red* words correspond to negative and the *green* words to positive terms found in the texts. The saturation of the color indicates the occurrence frequency were high saturation means high frequency and low saturation means low frequency. The *left column* shows Wordclouds using the dictionary of Hu and Bing and the *right column* the same using the created dictionary. The rows show the days in the following order: *first row*: day 1 Cardiff to Newport, *second row*: day 21 Porth-y-Waen to Llangollen, *third row*: day 60 Aberystwyth to Aberaeron. Comparing both Wordclouds for each day shows the difference in words the dictionaries are using.

compiled by hand by the user to create a PDF file. In the next step I created a table containing the relevant information for the visualization which is the name of each day, a link to the corresponding blog page and the sentiment measure. This table was then used together with the Toolbox to create the visualization. An example of the resulting visualization can be seen in Figure 17. For the general structure I used a simple table where every row shows information about each day. There are five columns: the date, the name of the blog entry, a clickable link to the blog and two bars showing the sentiment measure for two different dictionaries. The bars are horizontal and the range of the bars (indicated with the not filled area) represent the range of the sentiment measure. The left border was mapped to 1, the right to -1 and in the middle is a line indicating zero, so if the bar is left of the zero the sentiment is positive and if it is right the sentiment is negative. The color of the bars also show the polarity of the sentiment (orange $\widehat{=}$ positive, blue $\widehat{=}$ negative) and the length the value of the measure. The bars for both dictionaries are placed adjacent to each other so both values can be easily compared. There is no labeled axis because it is not that important to know the exact value but to see the tendency and to compare the values.

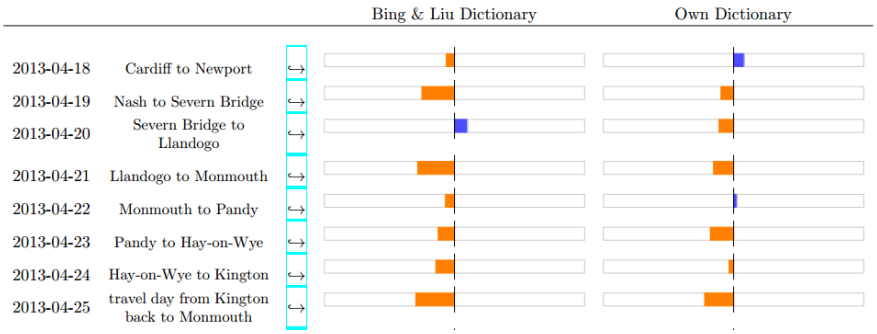

Fig. 17. Extract of the visualization of the sentiment analysis result. In this example the results from the Dictionary of Bing & Liu are compared with my own dictionary. Column description from left to right: date of the day, start and destination, link to blog, bars for sentiment measure.

8.2 Visualization of Quantitative Data

Tools

For the second visualization I used the Software Tableau Desktop[24]. It is a commercial Software (free for Students) for a wide range of classical visualizations. It can connect to databases or read data from text files and provides a lot of

[24] `http://www.tableau.com/de-de`

different visualizations for a lot of data types. They include all standard visualizations like barcharts, scatterplots, lineplots, heatmaps, maps for location data and many more. After a suitable chart type is found the user can customize the charts and their appearance in many ways and also combine several visualizations for the same data. After the visualization is ready it can be included in a website or distributed as a file which can be viewed using a viewer software from Tableau.

The general workflow of Tableau can be divided into several steps. The first step is to connect the program to your data sources. If these data is stored in databases the software can directly connect to them. In our case everything is stored as simple text files which can directly be loaded into Tableau. However, some adjustments needed to be made so one can easily work with the data. The biggest problem was that every data source was not yet synchronized. One way of doing this would have been to put all sources into one big csv file where each row shows the values for all streams at one specific point in time. As the data sources strongly vary in size, sampling rate and that sometimes the data is equally distributed and sometimes it is not this was not possible. Generally, there are two ways in Tableau to bring different sources together. The first is to do a classical join on the tables, as mentioned before with the one big csv file, and the second is a feature called 'data blending'. This option allows the user to put sources together using the scale of one primary attribute which is the same in the sources. The difference to the joins is that the sources are placed on the same scale but are not directly connected. Yet, to bring them into relation the data is aggregated using a specified interval. Here, I chose the time as shared primary attribute so the data gets placed on a timeline independently. Then I configured the aggregation to calculate the mean and Tableau automatically calculates a value for the specified time granularity independent of the sampling rate or the distribution. Hence, I calculated timestamps for every row of every data source using Java. I used the start date of every file and the sampling rate to determine correct timestamps for every row. Furthermore, I created one file containing all data if there were more than one file containing the measurements. The last preprocessing step was to extract the coordinates and timestamp of the GPS data and also put them into a text file because there is currently no support for GPX files in Tableau.

After the data was connected with the software the visualizations could now be created. Overall, I created three different visualizations which look at different aspects of the data. The first one includes a map showing the location traces and line charts for all of the numeric data. As there is a lot of data I included a time filter which allows the user to choose a specific time period using a slider with two indicators, one for the start date and one for the end date. After the user chooses a time period the views automatically update to show only data from the configured time period. Additionally, after the user chooses the time he can select a section of the map by dragging a rectangle over the location traces. Then the line charts get updated accordingly to only show data which corresponds to

the selected locations. By default the line charts are configured to show the data divided by month, day and hour. Hence, Tableau automatically aggregates the data for every hour in order to get one value. Yet, as this is useful if the user wants to look at a big time period this may not accurate enough to watch the data for one day only. Therefore, I included radio buttons which allow to chose the data granularity. The default option, as mentioned before, divides the time into month, day, hour and the second option additionally displays values for every minute of the hour. This is especially useful if the user wants to look at a small time period. The radio buttons allow the user to easily switch the granularity so they can quickly chose the view which is more appropriate to the selected amount of data. As there are multiple sources for some of the data streams (acceleration and elevation) I included a drop-down menu to chose which of the sources should be displayed in the line charts. The last option allows the user to chose which streams should be displayed in general. There are checkboxes where the user can select which streams should be displayed and which not. Last but not least the calculated sentiment measure was mapped to color of the line on the map. Each part of the route which corresponds to one day was colored according to the value of the sentiment. The most negative value was mapped to red and the most positive one to green. So the color range diverges between this colors and changes in color intensity. The neutral sentiment was mapped to white. This mapping easily distinguishes which part of the route belongs to a different day and to see how the tendency of sentiment was for this particular day.

The second visualization shows an area chart which shows the values of every stream. The areas are displayed on top of each other to create a stacked bar chart. This visualization has the same control option as the first visualization including the date filter, the data granularity radio buttons and the drop-down menu to chose the sources for acceleration and elevation. The goal of this visualization was to spot peaks in data streams where all streams have very high values. These peaks could correspond to special events that happened during the walk. To ensure that all streams have the same influence all values were normalized in the same range.

The last visualization shows a comparison of the different data sources for the acceleration and elevation. There are two line charts which show the values for the elevation from the GPS sensors versus the elevation from OpenStreetMap and the acceleration from the EDA sensor versus the acceleration from the ECG sensor. Again, the filter for the time and the selection of the data granularity was included. Example pictures of all three visualization are shown in Figures 18 - 20.

8.3 Visual Analysis Examples

Elevation Accuracy
As already mentioned in Section 5.1 the GPS sensors manage to correctly measure the elevation most of the time. This can be easily observed by looking at the comparing line chart in the third visualization shown in the upper part of Figure

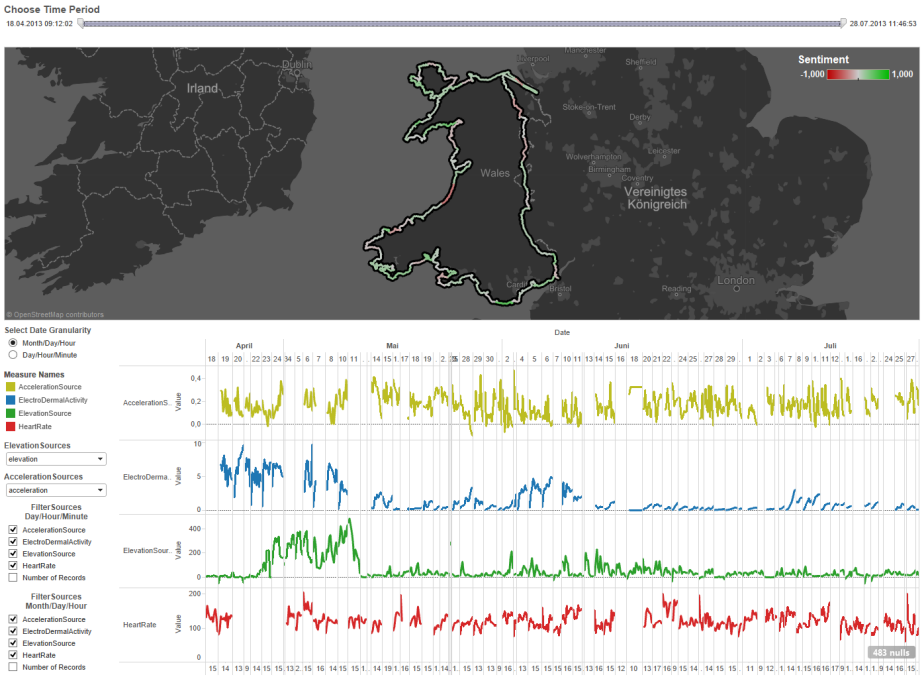

Fig. 18. Image showing the first visualization as explained in Section 8.2.

Fig. 19. Image showing the second visualization as explained in Section 8.2.

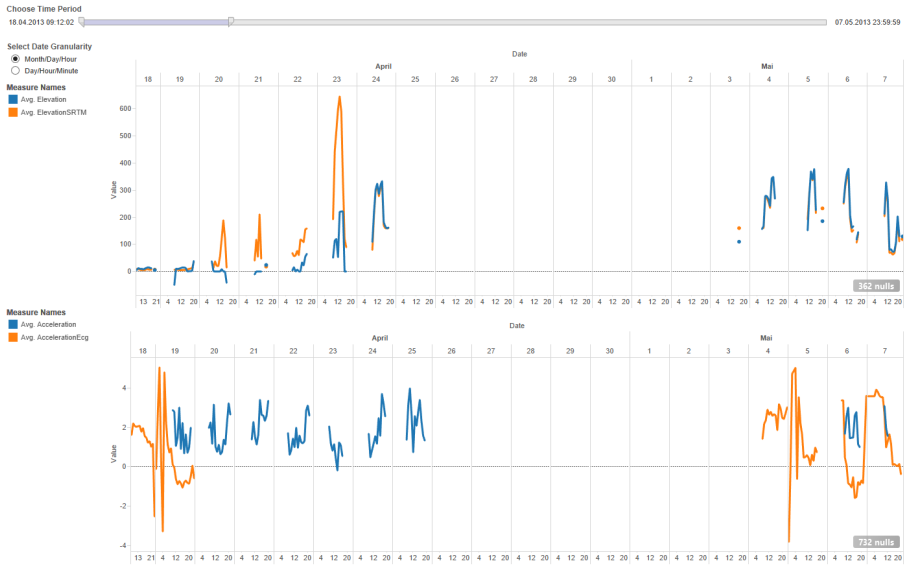

Fig. 20. Image showing the third visualization as explained in Section 8.2.

20 or in Figure 11. It can be seen that both sources concur with each other on most of the days, however, there is a small period of time in the beginning where there are major differences. Disregarding this small period the rest of the data, which is not covered by the line chart, concurs. However, it would be interesting to find the reason for the accuracy loss in that period of time. As the elevation was extracted from the cleaned and merged GPS data the elevation data also contains measurements from both sources. If we look at the date of the wrong measurements we see that this is exactly the time period where Alan lost his Garmin device (see Section 5.1), hence only the smartphone app recorded the location data. Now we can conclude that the accuracy of the smartphone elevation data is very bad compared to the Garmin device as the measure values are way lower (sometimes differences of 400 meters) than they actually are and the Garmin Device seems to be very accurate (supposed OpenStreetMap yields the correct elevation).

Correlation of Heart Rate and Elevation

Another interesting aspect would be to investigate if there is a correlation between the heart frequency and the elevation. As the change in elevation can be considered as steepness of the terrain it would make sense that changes in height (especially positive) increase the heart rate because walking uphill can be very exhausting. A first approach to this question would be to calculate a correlation between the overlap of the streams. Unfortunately, the correlation is very low ($r = 0.119$). However, this is not really surprising as the heart rate also depends on a lot of other factors and not only the elevation and there may be some time

offset in the data. Furthermore, the heart rate may not be as accurate due to the problems described in Section 5.2. Yet, when we look at the line charts of both streams in the first visualization (Figure 18) there are definitely some time periods where a correlation can be identified (see Figure 21).

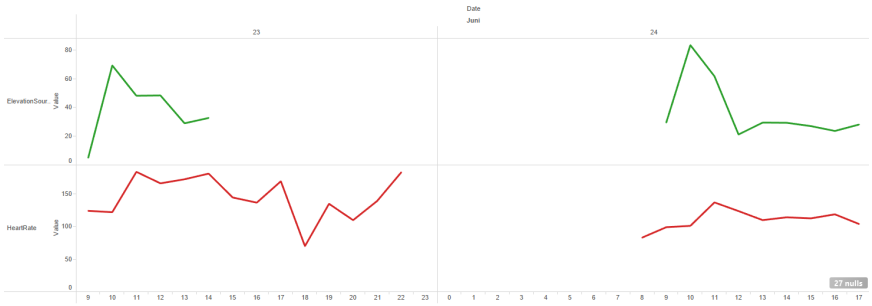

Fig. 21. Line plot showing the heart rate (*red line*) against the elevation (*green line*). It can be seen that after the elevation starts to increase also the hear rate increases.

Incident Detection
Now the task is to find exceptional situation on the walk which affected the data and thereby can be identified. One indication for such a situation could be a high value of all streams at a specific point of time. In order to find a point with this characteristic we can look at the second visualization and search for the highest peak of the bar charts. The upper part of Figure 22 shows the stacked bar chart of the isolated day where the peak occurred. We can see that the peak is around 12 p.m. on the 6^{th} of May. Now we have a look at the first visualization (lower part of Figure 22) and see that there is a peak in EDA at approximately this point of time, which is responsible for the overall peak. Hence, we conclude that this date could be potentially interesting for further investigation. In order to get more information about the events of this day we can read the corresponding blog entry. In his summary he writes about an 'unexpected meeting' which could be the cause of the data characteristics. While reading the text I found out that he saw some wolves around midday and met a man with a large dog, that also turned out to be a tamed wolf. However, Alan didn't realize this at the beginning so it is very likely that this situation could be the cause of the peak in the data.

9 Conclusion

In this project I dealt with the problem of processing and analyzing a raw real world data set which comprises of a multitude of different sources. All of the streams were introduced and all of their special properties were illuminated. It was shown how to deal with the different formats and how to synchronize the

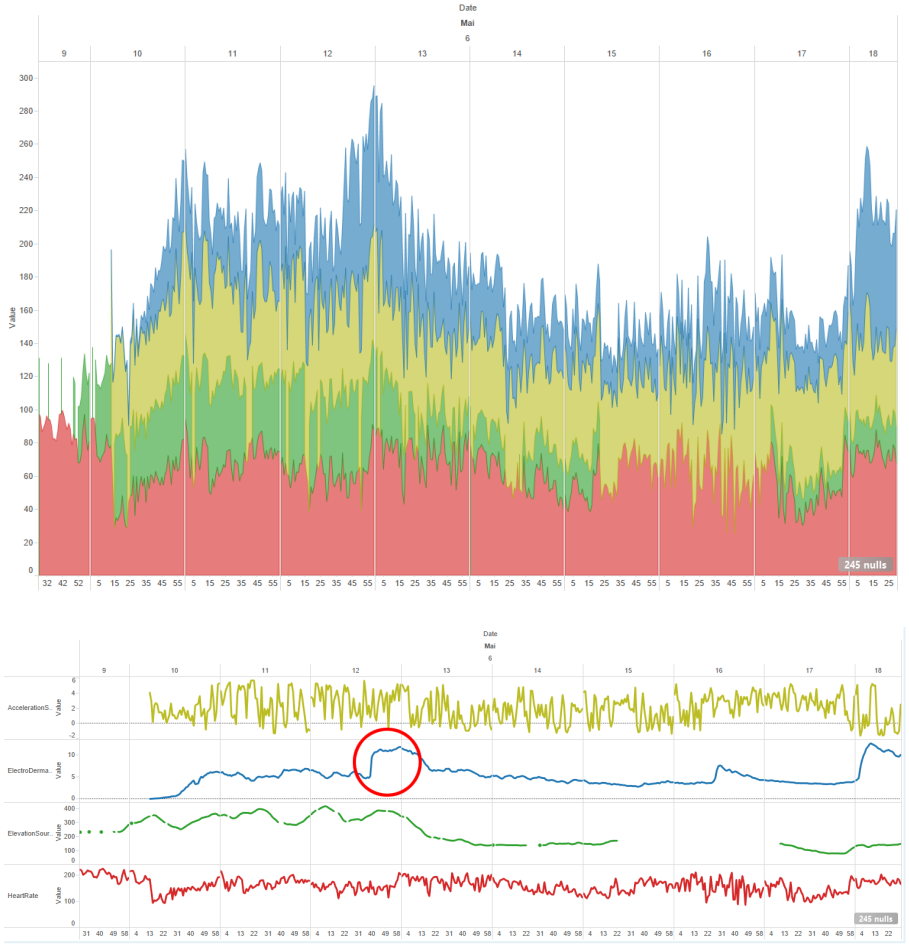

Fig. 22. Two charts showing the data of a special situation of the walk. The upper chart shows the stacked bar chart visualization for the situation described in Section 8.3 Subsection 'Incident Detection'. The bar chart shows a peak around 12 p.m. . The lower image shows a line chart for the same data and the same time period. The *red circle* marks a peak in EDA at the same time as the peak in the upper bar chart occurred.

streams in order to bring them into a relation and to display them in one visualization. Furthermore, the sources which I focused on in my project were assessed in terms of data quality. Regarding the GPS data it was shown how to deal with several problems and I introduced some tools which a capable of cleaning such data. However, there is still room for improvement in cleaning the ECG data. For the analysis I introduced methods and tools to calculate a heart rate signal from a ECG stream and showed an approach to extract a sentiment measure from text data. Additionally, I tried to create a domain specific sentiment dictionary which could be used for sentiment analysis in a long-distance-walking context. Unfortunately, it could be shown that the Simple WordNed Propagation approach is not powerful enough to create a better dictionary than common dictionaries which aren't domain specific. Hence, a more sophisticated technique may be able to outperform this general dictionary. Furthermore, it was shown that the given text data is hard to assess in terms of sentiment as the analysis only produced mediocre results. Lastly, several visualizations were created which allow browsing and a visual correlation of the data. One visualization for showing the results of the sentiment analysis and one for visualizing the quantitative data of the set.

References

1. Dawson, Michael E., Anne M. Schell, and Diane L. Filion. "7 The Electrodermal System." Handbook of psychophysiology 159 (2007).
2. Reisner, Andrew T., Gari D. Clifford, and Roger G. Mark. "The physiological basis of the electrocardiogram." Advanced methods and tools for ECG data analysis (2007): 1-25.
3. Ungless, Gary. "The Actiwave User Guide." (2008) Chapter 6.
4. Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23):e215-e220 [Circulation Electronic Pages; http://circ.ahajournals.org/cgi/content/full/101/23/e215]; (2000).
5. Moody, George B. "WAVE: Waveform analyzer, viewer, and editor [Computer software]." Available from http://www.physionet.org/physiotools/wag/wave-1.htm (2005).
6. Moody, George B. "Convert EDF (European Data Format) file to MIT format header and signal files [Computer software]." Available from http://www.physionet.org/physiotools/wag/edf2mi-1.htm (2008).
7. Moody, George B. "sqrs: Single-channel QRS detector [Computer software]." Available from http://www.physionet.org/physiotools/wag/sqrs-1.htm (2010).
8. Zong, Wei, George B. Moody. "wqrs: Single-lead QRS detector based on length transform [Computer software]." Available from http://www.physionet.org/physiotools/wag/wqrs-1.htm (2010).
9. Moody, George B. "gqrs: A QRS detector [Computer software]." Available from http://www.physionet.org/physiotools/wag/gqrs-1.htm (2013).
10. Moody, George B. "ihr: Generate instantaneous heart rate data from annotation file [Computer software]." Available from http://www.physionet.org/physiotools/wag/ihr-1.htm (2004).

11. Moody, George B. "tach: Generate heart rate vs. time signal with evenly spaced samples [Computer software]." Available from http://www.physionet.org/physiotools/wag/tach-1.htm (2002).
12. Hu, Minqing, and Bing Liu. "Mining and summarizing customer reviews." Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004.