Amsterdam In Motion

Names	Study numbers
Rick Proost	4173619
Daan Schipper	4155270
Ruben Starmans	4141792
Wim Spaargaren	4178068
Thom Hubers	4078543

Introduction

Social data is gathered in large quantities and available on the internet via mediums such as Facebook, Twitter and Instagram. These platforms are widely used by people to share information to the public. Individual tweets, status posts and photos may not be that useful for scientific studies, but when a multitude of these tweets, posts and photos are collected they can be analysed together. With the collected data characteristics can be extracted for different groups of individuals that for example share the same interest, like sport activities. The first challenge here is to collect data and retrieve the latent characteristics that can identify the persons interest in any kind of sport activity. The difficulty is in analysing, categorising and grouping data the right way to be able to extract and come to the right conclusions. Analysing data can give insights on how people are communicating, what they are are interested about, what motivates and drives them and just general information on what they are doing. Since data from the mentioned sources can contain geographic references, characteristics can possibly be identified and defined for certain areas. Another challenge here is spatial analysis and clustering of the collected data.

Objective

The goal of Amsterdam in motion is to identify and characterise neighborhoods in Amsterdam based on sport activities, which requires the previously mentioned data collecting, categorisation, and spatial clustering. The identified neighbourhoods can then be visualised on a map to get an overview of where different sport activities are popular, on what days of the week and on what time of day. This can be useful for event organisers, shopkeepers, or people that want to know where to find companions in there sport activities.

Data Collection

To achieve this objective, data must be collected for the application. This data will be collected from two sources, which are social media platforms.

The first source is Twitter. Twitter is a good choice to collect data from, since tweets contain mostly text about what people are doing. Along with that, a tweet can contain a geolocation of the user, who posted the tweet. Another good reason to use Twitter is that it has a very rich API. To collect relevant data from Twitter, Twitter's streaming API will be used. The streaming API can be given a bounding box, to stream tweets which are send from the area of the bounding box. After this, every tweet will be checked if it has a geolocation and if so, this

tweet will be collected. This is relevant for the application, since it might be possible to derive from the tweet what a user was doing at an exact location in Amsterdam.

The second source is Strava. Strava is a good choice to collect data from, since it is a social media platform for athletes. This platform is used to keep track of sport activities like cycling and running, for this reason, data from Strava contains routes which are represented by geolocations. Just like Twitter, Strava has got a rich API, from which the streaming API will be used. This API can also be given a bounding box, such that data from a specific area can be obtained, which in this case is Amsterdam.

With these two sources data which is relevant for the application, can be collected. Data from strava is relevant, because it contains data about athletes accompanied by their locations. Data from Twitter is relevant, since athletes might use Twitter to tell about their sport activities and if that tweet has a geolocation it can be derived where the athlete was conducting its sport. To derive this information after the collection, data must be analyzed, which will be discussed in the Section Methods.

Methods

This section specifies which methods will be used for the neighbourhood classification. The several steps that have to be taken to extract useful information out of the data are dispayed in a pipeline diagram, after which the steps are elaborated somewhat more.

Overall system architecture

The graph in Figure 1 shows how input data (e.g. tweets) are processed into neighbourhood classifications.

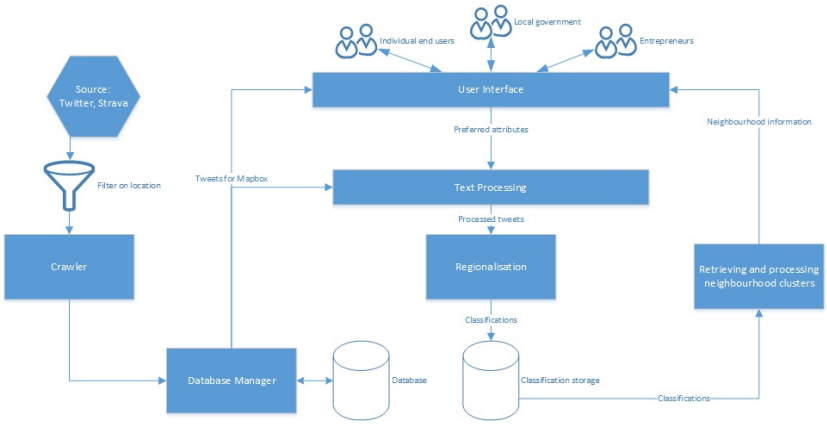

Figure 1: Graphical representation of the proposed system architecture

These tweets can be used for both the processing and identification of neighbourhoods and direct display on the user interface. All relevant (sport-related) messages will be displayed.

The tweets are processed to determine which topics are discussed. Relevant (i.e. matching with a sport related word list) messages are used to classify neighbourhoods. The clusters are also used in the user interface.

Crawler

The crawler uses the Twitter and Strava API. After location based filtering all data is passed to the Database Manager which stores it for later usage.

Database Manager

The storage is provided by a PostgreSQL database.

Text processing

Messages are matched and indexed to a sports-related taxonomy. To indicate whether the message gives information about a sports activity and which activity that could be the BM25 algorhitm or similar is used.

Regionalisation

After the contents have been processed, the matched data is sent to the regionalisation part. This clusters the messages over multple dimensions (spatial, time, sport). The spatial aspect of this is done by the maxp regionalisation algorithm, which is part of the pysal library. Clustering on the other properties is done by a more common clustering algorhitm, of which examples are included in the Python sklearn package.

User Interface

The messages and regions are sent to the user interface. The different stakeholders can all retrieve they are interested in from it. Mapbox is used to display all this on a map. After the map of Amsterdam is loaded, the areas are plotted on it, including the dots which represent messages that are part of that cluster. A comprehensive interface provides methods for filtering the areas based on time and sport.

Specifications & Execution Plan

In the previous sections the objective is presented and the way the data is collected and how this is processed so it can be used in the application. In this section the different features of the application will be described, and the different requirements to realise these features.

MoSCoW

The MoSCoW method is a prioritisation technique used to reach a common understanding on the importance of the delivery of each requirement. The requirements are divided amongst four categories with each their one priority, as the name suggests. For this application the following requirements are identified and categorised as seen below.

Must haves

- Data crawler on Twitter
- Data crawler on Strava
- BM25 analytics with known taxonomies
- Visual representation of categorised data on Mapbox

Should haves

- Dynamic selection on day of week to visualise data
- Dynamic selection on time of day to visualise data

- Neighborhood selection to represent several attributes regarding the neighborhood

Could haves

- Real-time input of new taxonomies
- Use of Face++ to retrieve additional attributes to gain more insight into users
- Use of Genderize to retrieve additional attributes to gain more insight into users
- Dynamic selection on additional attributes
- Display multiple maps next to each other for easy comparison of different filters

Won't haves

- Intensity (heat) map

First and foremost the application must be able to identify the characteristics of the data gathered from Twitter and Strava. The data, as explained in the Section Methods, is analysed by the BM25 algorithm to rank the relevance to a query. The main priority is to have finish these requirements first so a working prototype can be delivered.

Next the user experience to our platform will be enhanced by giving the user more control over the presentation of the data. The user will be able to view the data based on time of the day and day of week.

The visualisation can later be extended by more extensive analysis of the gathered data to enrich the user experience. With the use of tools such as Face++ and Genderize more attributes will be added to the application, which the user can use to further narrow down his target audience.

In the Table 1 the timeline of the development of the application is given. First the highest priority requirements will of course be developed, and later on in the project the other requirements will be worked on. Throughout the whole project data will be gathered from the different sources.

Week 4	Week 5	Week 6	Week 7	Week 8	Week 9
Idea document	Visualising data on Mapbox	Connecting data stream to Mapbox	Finalising must-have requirements	Integration more attributes	Finalising application
	BM25 analytics	Taxonomy creation	Categorising data on different attributes	Dynamic selection filters	Presentation

Table 1: The timeline of the development of the application

Expected Outcomes

As mentioned earlier in the section objective, the goal of this application is to identify and characterise neighborhoods in Amsterdam. In figure 2 a mockup of the user interface (UI) is shown, it will become visible how the UI could look.

Figure 2: Mockup of the User Interface

The user is presented with a large map of Amsterdam. On this map blue dots are shown which are tweets or strava activities that are found to be relevant by the algorithm. Relevant posts are posts that fall into a sports activity based on the taxonomy of that sport. Around those posts the neighborhood boxes are drawn in different colors to quickly make clear how the neighborhoods of Amsterdam are divided based on sport activities. Because only relevant tweets are shown this can give an indication to the user how active an area is with regards to sports.

In the setting panel the user can customize what will be shown. This will contain a filter on the type of activity and a filter on when a tweet or activity has taken place. The setting to filter when activities were carried out will give insight in activities that only take place at a certain day of the week or even a certain time of the day. There are also show options which toggle areas or labels.

Evaluation & Outlook

Evaluation

In the Section Specifications & Execution Plan a list of MoSCoW specifications was presented. These specifications are a useful and structured manner to evaluate wether or not the application is working as intended. They also describe the importance of each feature. Each feature can then be checked if they are implemented and how well they are working. The must haves are the most important features and are the bare minimum for the application to present the findings on the neighborhoods. The should haves can be inspected on how well they filter and modify the presented data to the user. How much control do they actually give the user? Lastly the could haves describe the tools to further narrow the target group down and can be tested on how easy they are to use and how useful they prove to be.

Besides functional evaluation of the project by the MoSCoW specifications, the project can also be evaluated by testing how well the application performs in different user scenarios. In Table 2 user scenarios for which Amsterdam In Motion could be used in are described next to the possible relevant stakeholders.

Stakeholder	User Scenario
	The government wants to know in what area of the city there is a need for more streetlights

Local government	for runners. The government wants to know what area would be most suitable and in need of a new sports complex.
Entrepeneurs	A company is interested to know where to build their new football/hockey store.
Individual athletes	An athlete is looking to plan their exercise. The best would be to do that on a time of a day of the week the park is least busy.

Table 2: The potential stakeholders with their user scenarios.

Outlook

Other similar projects exist that also try to divide a city into neighborhoods based on data collected from social media, an example of this is LiveHoods. Their focus on how they achieve this is different than ours. LiveHoods uses check-ins to buildings and points of interest (POI's) from the social platform Foursquare to identify certain areas of a city. Using check-ins to buildings and POI's could also possibly be interesting to further expand Amsterdam In Motion. For example, Foursquare could be used to identify important sports complexes around the city. This could help get a better understanding of why an area is popular for a certain sports activity. In addition to that a mapping of important sports buildings and POI's can be created.