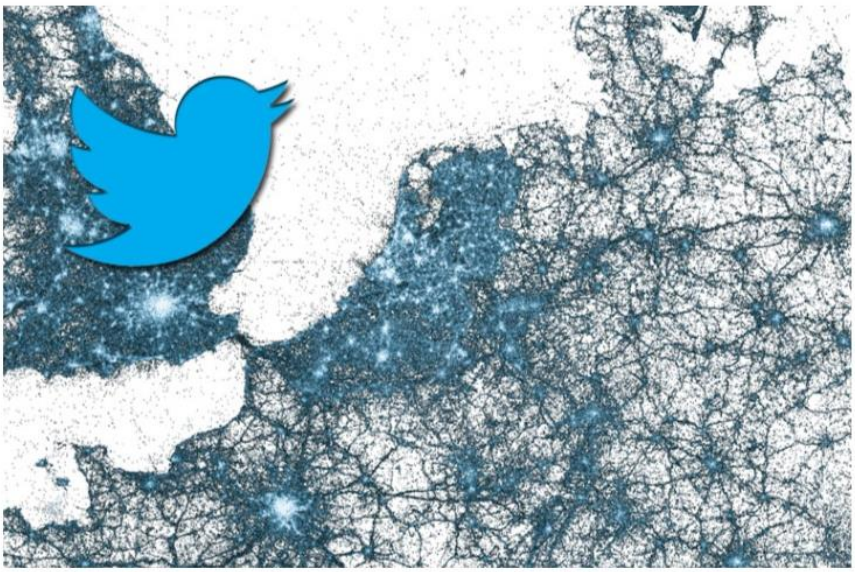
Accessibility of touristic venues in Amsterdam:

A methodology to collect, assess and validate the attractiveness and accessibility of touristic venues from data extracted using Twitter as Urban Sensor: A.M.Sterdam case.

MSc Geo Science | **Geographical Information Management Applications (GIMA)**



Abstract:

Amsterdam attracts every year a growing number of tourists which can generate congestion and overcrowding in the city centre. To tackle this problem, a state-of-art method that enables new ways of collecting large amount of spatio-temporal data to study how people use the urban environment is needed. The research is aimed at develop, implement and validate a method to extract touristic information from Twitter LBSN by using techniques of the Geographic Knowledge Discovery methodology in combination with Python programming and ArcMap platforms. With this method the author assesses the attractiveness of touristic venues generated by spatio-temporal aggregation of tourists. Moreover, he computes the accessibility of touristic venues through the implementation of a gravity model thus finding where and when urban solutions to tackle congestion and overcrowding are needed the most.

Utrecht
ITC-Twente
Wageningen-UR
TUDelft

Student: **Emanuele Panizio**
Supervisor: **Arend Ligtenberg**
Professor: **Arnold Bregt**

Acknowledgements

I am hereby take the chance to gratefully thank the whole of the people who somehow have contributed to the accomplishment of this research work. To start with my supervisor, Mr. Arend Ligtenberg who first suggested me this research topic, and then supported me in the achievement of the objectives set. I gratefully thank the HERE firm, particularly Mr. Nikolai Tihhomirov who supplied me with the log being used in the validation of the final outcome of this work, and Mr. Simon Curtis who was flexible with the internship workload in favour of the accomplishment of this research. In the end, I thank my fiancée and my family helped me morally and financially during the difficulties overcome.

List of abbreviations

AIAMS – Amsterdam Institute for Advance Metropolitan Solutions

AMS – Amsterdam Metropolitan Solutions

AoI – Area of Interest

API – Application Programming Interface

BoW – Bag of Words model

CBS – Central Bureau of Statistics

CSV – Comma Separated Values

DB - Database

FDA – Frequency Distribution Analysis

GKD – Geographic Knowledge Discovery

GPS – Global Positioning System

HTTP – HyperText Transfer Protocol

KDD – Knowledge Discovery in Database

LAE – Landmark Attractiveness Estimation

LAS – Landmark Attractiveness Semantic

LBSN – Location Based Social Network

LDA – Latent Dirichlet Allocation

LSI – Latent Semantic Indexing

MIT – Massachusetts Institute of Technology

NBTC - Netherlands Board of Tourism and Conventions

NLP – Natural Language Processing

NLTK – Natural Language Tool Kit

PDA – Personal Digital Assistant

PoI – Point of Interest

SOF – Spatial Overlapping Frequency

SSE – Sum of Square Errors

SW – Stop-words

TAUS – Twitter as Urban Sensor

TF-IDF – Term Frequency – Inverse Document Frequency

TNO - Netherlands Organisation for Applied Scientific Research

VSM – Vector Space Model

WUR – Wageningen University & Research

Contents

1. Introduction

During the last decade, the parallel development of social networks in human activities and the growing usage of GPS-based portable devices has enabled the study of the human behaviour with regard to a multitude of disciplines. Interestingly, the use of social networks in combination with portable devices such as smartphones, tablets, and PDAs has led to the integration of Geo Location Services within social networks improving location-based capabilities as well as decreasing the costs and time of data collection (Dykes & Mountain, 2003; Sacco et al., 2013 Tasse & Hong, 2014; Demirbas et al., 2010).

The exponential growth of Location Based Social Networks (LBSN), in terms of number of users and user generated data volume, have made LBSN a potential source of information to be used in spatio temporal human behaviour studies (Lee et al. 2013; Ferrari et al. 2010). However, given that the attribute data gathered from those sources is enormous and basically unstructured, if one needs to extract only meaningful information with regard to a specific matter, a dedicated mining approach to filter information out is needed.

1.1. Project Background: the A.M.S. framework

This research is performed within the context of a project named Amsterdam Metropolitan Solution (AMS, 2014). AMS is a design context launched by the City of Amsterdam in 2013 in order to attract international figures to form a world-class institute in the field of applied technology. Amongst the participants of the design contest, a manifold cooperation, which involves academic figures like TU Delft, Wageningen University (WUR) and Massachusetts Institute of Technology (MIT), TNO and societal as well as industry partners has been ranked in the first place in the contest. This cooperation, named *Amsterdam Institute for Advanced Metropolitan Solutions* has the objective to develop and deliver solutions to be able to tackle metropolitan challenges in terms of sustainability and quality of life, including resource and food security, mobility and logistics, water and waste management, and health and wellbeing (AIAMS, 2014). As stated in their report of 2014:

> "*The Amsterdam Institute for Advanced Metropolitan Solutions aims to become a leader in urban innovation, using technology and design to resolve, steer and navigate city flows – e.g. water, energy, waste, food, data and people.*"

Within the AIAMS cooperation, Wageningen University has undertaken a project aimed at identifying and analysing through the use of data extracted from Twitter, the behaviour of tourists in Amsterdam. The testing and piloting location for the AMS project is the city of Amsterdam, in which, the use of Twitter in Amsterdam shows an interesting trend. In the discussion paper of Van de Ven & Neroni (2012) they argue the role of Twitter as a potential data source to support the Central Bureau of Statistics (CBS). According to their work, 11 percent of approximately 300.000 tweets[1] had the word "Amsterdam" embedded in. Moreover, another study published by Helmond on her scientific blog has confirmed the findings of Van de Ven & Neroni (2012) showing the highest Twitter penetration per city in

[1] Tweet is the jargon that describes a 140 character user generated text content of Twitter (post).

Amsterdam and the highest density of tweets per region in North-Holland (See Appendix A and B) (Helmond, 2011).

1.2. Problem Statement

Tourism is a growing industry in Amsterdam. According to the Research and Statistics Office of the City of Amsterdam, in 2013 the Dutch capital received a greater tourist income due to the reopening of museums and various events such as the 400th anniversary since construction began on Amsterdam's world renowned Canal Ring. Furthermore, the Netherlands Board of Tourism and Conventions (NBTC) expects tourism to continue growing in the coming years. Forecasts show that in 2020, with the global economy rising up again, the number of foreign overnighting guests that visit the Netherlands each year is going to increase by 2 percent to approximately 14 million in 2020 with an expected total growth of almost 30 percent compared with 2007 (NTBC, 2013). As most of the touristic attractions are located in the inner part of an urban space, problems of accessibility due to the growing number of newcomers may occur, particularly if these problems concern a compact urban space like Amsterdam.

If measures to control and predict this growth are not established, phenomena of urban congestion and overcrowding, due to the growth in the number of incomers, may arise in space and time, generating problems of accessibility in proximity of the busiest areas. Therefore, it is crucial for the liveability of Amsterdam City to gather information on tourist behaviour such as where the city is most intensively used and perhaps what the main city attractors for this behaviour are. These questions can be addressed through an analysis of the use of the urban space to serve as support to find countermeasures that could tackle the problem.

1.3. The Case Study

Intro

In this work, information mined from Twitter, in terms of geo located point feature data is examined in order to gain a better understanding of aggregation patterns of tourists in proximity of important Point of Interests (PoI) (e.g. museums, monuments, attractions, and so on). Precisely, this work focuses on the analysis of congestion patterns generated by the aggregation of tourists, in space and time, in order to establish whether accessibility issues may arise in proximity of touristic landmarks disseminated in the city centre of Amsterdam. In parallel with the analysis of the aggregation, human behaviour, expressed through the text attribute of Twitter, is also investigated to find a relation between the information included in it and the urban context in which is found.

Why Twitter?

As previously stated, Twitter is the LBSN from which the data is to be collected. The reasons behind the choice of Twitter as the data source for this analysis involve its following promising features:

- Microblogging;
- Vast and costless data collection;

- Privacy issues free;
- Twitter hashtag index;

Twitter is designed as a platform in which users can generate and share contents through *"Microblogging"*. The term indicates the ease of posting short text messages, known as ***posts***, usually enriched with links to web pages, emotional status and/or pictures, in a maximum of 160 characters in each post (20 characters for user name plus 140 for the post) and it serves to communicate the user's experience to others. This aspect is very attractive since it does not requires good writing skills and large content to fill pages like it happens with blogs, therefore users are more encouraged to use it (Demirbas et al. 2010; Bifet, 2013). In addition, Twitter offers the possibility to easily collect vast amount of data without incurring in privacy issues (Muntean et al., 2012). In fact, everyone with a minimum of programming skills can use the HTTP based open source Twitter's Streaming API to share posts of public domain with third party applications.

Furthermore, another reason of the use of Twitter in this case study is its hashtag indexing. The hashtag indexing system of Twitter is deployed in this work to support the extraction of touristic features. Actually, the hashtags system has been launched since 2007 as a method for users to group together tweets on the same topic (Chang, 2010). Thus, tweets linked to a specific argument can be found more easily by searching for hashtag rather than particular keywords in text. Indeed, the hashtag method has enabled the classification of most used hashtags in trending topics. Therefore, it represents a powerful index method that can be combined with topic detection and mining algorithm such as the Latent Dirichlet Allocation (LDA) for instance, to extract hidden patterns in text (Muntean et al., 2012; Blei et al., 2003).

Research components

In this case study, I make use of three components included into Twitter data in order to accomplish the research objectives:

- geo location
- post
- time

The geo location included in Twitter contents is analysed by means of spatial analysis models which serve in the identification and extraction of aggregation patterns that are scattered in urban context. With aggregation patterns, the author intends the number of Twitter occurrences located in proximity of a touristic PoI. This attribute represents the level of attraction that a touristic PoI is able to generate and it is directly proportional to the number of tweets. Hence, the higher the number of tweets in proximity of a PoI is, the higher its value of ***attractiveness***. The attractiveness is the outcome of the method used to study the geo location component and it is used in the assessment of the accessibility: the final output of this study.

Next, the post component serves to assess the existence of a linkage between the urban location and Twitter data. A dedicated methodology is created in order to extract

meaningful[2] information, regarding the relation between the occurrences of tweets at a specific location, the time they are posted, and their meaning (i.e. the topic expressed by the post). In this regard, raw text data need to be pre-processed, removing features that are not needed, and thus, formatting the text in a more conventional (natural) structure. This is done through the combination of Natural Language Processing (NLP) techniques including machine learning tools which are freely available over the web. The NLP approach implements text processing tools to convert unstructured text attribute of tweets into structured data, by removing bad characters which can corrupt the outcome of the Information Retrieval[3] (IR) process (Wikipedia[b], 2015). Whereas machine learning techniques from the NLP approach support the classification of text with regard to what is meaningful information to this particular analysis and what is not (e.g. which tweet is tourism related and which is not?) as well as to discover the *'latent topics'* behind the typical unstructured nature of posts (e.g. what does the user mean by writing that post?). The processes of information mining using text classification techniques and topic modelling tools is described in the following sections.

Finally, the time is the last attribute to be used. The scope of time is to divide the collected dataset of tweets into time periods. Thereafter, for each time period, the measure of the accessibility of touristic venues (or PoI) and the discussed topics are compared to assess the degree of change at different time of the day and of the week. A method to decompose the time is established and it is described in the following sections, together with those established for the other components (See Paragraph 5.3, Section Time).

2. Objectives

2.1. Research objectives

This research seeks to develop, apply and validate a methodology that support the collection of Location Based Social Network contents, like those included in Twitter, in order to analyse the aggregation of tourists in urban context. The study of the aggregation serve to evaluate its effect in urban space and time with regards to the accessibility of touristic venues. Therefore, the purpose of this method is to calculate, and display graphically on map, where and when large number of tourists gather in service areas of touristic venues in which the measure of the accessibility is poor. The outcome of this applied method can be used by local decision makers to support possible urban optimisation measures that may be necessary if large groups of people gather in poorly accessible areas. In this context, LBSN serve to source this analysis with spatio-temporal data used to identify where the aggregation of social networking activities is high and in which poor accessibility could lead to issues of congestion and overcrowding. In the end, with the help of this method, urban planners may be able to grasp how tourists use the urban space at glance and thus drive the attention to where and when important congestion patterns are most likely to occur also in the future. To do so, records, in the form of geo located point features enriched with time and text attributes (i.e. the user text message embedded in the tweet), are extracted from a dense online community,

[2] With the term "meaningful" this research includes touristic information in the form of Twitter posts enriched by geo locations which are extracted from the raw data collected through Twitter.
[3] Also referred as the features extraction. See Paragraph 4.2.5.

namely Twitter and the results further validate with the use of transit information (See Paragraph 5.6.4).

In conclusion, the framework being created attempts to identify the reasons (e.g. Why tourists post at a particular location at a specific time?) and the extent of aggregation patterns (e.g. How attractive is the landmark?) in proximity of touristic landmarks[4] in Amsterdam centre with regards to space and time components. The aggregation of tourists is calculated in terms of number of occurrences (tweets) located in the service area of a touristic venue (calculated through a space partitioning approach) and it is defined in this work as the ***Landmark Attractiveness Estimation*** (LAE). Whereas, the accessibility of touristic venues is obtained by the implementation of a ***gravity model*** which considers the ***attractiveness*** of touristic venues (LAE) and the ***network impedance***. In this case study, the impedance is represented by the walking distance (origin-destination) from a transit stop (origin) in proximity of one or many touristic landmarks (destinations) in which the attractiveness is high. Moreover, an attempt to study the reason(s) that generate high values of attractiveness for tourists is set in order to evaluate if a linkage between the post semantic and the spatial location in urban context exists. This method is defined in this work as the ***Landmark Attractiveness Semantic*** (LAS) and it helps understanding the behaviour of tourists which generates attractiveness. In conclusion, the aim of this research can be summarized in three sub-objectives:

1. **Develop, demonstrate and test a method and application to use social network data to identify and assess meaningful urban aggregation patterns of tourists visiting Amsterdam by means of the Landmark Attractiveness Estimation (LAE)** (number of tweets within the service area of a specific landmark);

2. **Determine the Landmark Attractiveness Semantic (LAS) through a dedicated method which combines supervised and unsupervised machine learning techniques applied to the text message of tweets** (most popular topics discussed within the service area of a landmark)**;**

3. **Implement a method to assess the accessibility of places in space and time in relation to phenomena of congestion and/or overcrowding occurring within the service area of PoI disseminated in the urban space.**

2.2. Research questions

The research objectives can be achieved by answering the following questions and it may be possible to understand how the research steps lead us towards the result:

1. How can unstructured geo-tagged tweets be mined from the Twitter API and translated into meaningful information, to this study, in order to identify tourism aggregations in the City of Amsterdam and thereby assessing the degree of accessibility of touristic venues?

[4] The words venue, Point of Interest (PoI) and landmark are used interchangeably to express an area or object linked to an activity. In this work only typical “touristic” venues are considered.

2. How it would be possible to estimate the attractiveness of spatio-temporal touristic aggregation as well as their semantics (e.g. Why tourists gather in that specific area at that particular time?) of the behaviour of tourists in Amsterdam?
3. What is the impact of the temporal component in the formation of aggregation patterns, hence in the measure of the accessibility of touristic venues, and what approach can be used to evaluate this impact (e.g. time of the day, day of the week to display spatio-temporal hotspots)?
4. What validation approach can be used in order to support the measurement of the accessibility, and verify whether the tourism aggregation is linked to transit usage trends (e.g. is the attractiveness of touristic venue(s) somehow connected to the volume of users at transit stops within a threshold distance from touristic venues?)

2.3. Research outcome expectations

The product of this study is expected to be in the form of a map, generated through the combination of the density of Twitter contents occurring within the service areas of touristic venues in Amsterdam (i.e. the attractiveness of places) and the network distance that separate the landmarks from closest transit stops (i.e. the impedance or the cost of reaching a venue). The purpose of this approach is to identify areas denoting, at the same time, high frequencies of tourism related activities (from Twitter data) happening in poorly accessible urban areas. This information could support successive actions of improvement that the Municipality of Amsterdam could undertake in order to limit and/or tackle issues of urban overcrowding and congestion, which are generated by the growing tourism industry.

Following, the reasons behind the choices undertaken in this work are detailed by evaluating the outcomes of related works on similar topics. Next, a theoretical framework is included to support the understanding of readers that may not be familiar with the jargon and the tools being used in this work. Successively, in Methodology it is argued which choices are made and the process undertaken to support them. Finally, the paper ends with the elucidation of the findings enriched with a Discussion chapter in which conclusions and recommendations for future works are reported by the author.

3. Related works

3.1. Background

This chapter gives background over the outcome of the approaches used in similar topics with regard to problems encountered and solutions applied. The research on the extraction and analysis of user generated data from LBSN for various purposes is extensive and many methods have been found to produce promising results in similar fields such as identification and classification of land use in urban areas, traffic and incident management, political election trends detection and assessment, and so on. Overall, all of the works whose target is to understand the human behaviours in LBSN, apply the geographic location in combination with the text component (i.e. the user text message known as *post*) to be able to study the linkage between the ***user location*** and the ***text semantic*** in space and time,

particularly with regard to the use of the urban space and reasons of using it. The text appears to be a major problem when semantic studies are performed because of its user generative nature. In the context of LBSN, users express feelings - and information in the case of Twitter – in a subjective way (grammatically and sentimentally speaking), producing extremely heterogeneous contents. Excessive content diversity is a shortcoming for studies which involve the understanding of the variables (e.g. events, sentiments, curiosity) that leads to (spatial) aggregation of people because of the difficulty in identifying recurring patterns in text.

A major limitation that seems not being undertaken in the current research is the detection of *spam*[5] in LBSN. In the case of Twitter for instance, many advertising and trend analytic services take place in its public APIs. Spam are characterized by exact same location of occurrences, similar text structure, similarity of information reported and in some cases systematic interval of time between previous and next occurrence (e.g. typical of automatized machine processing such as meteorological stations that output weather condition data on equal time intervals).

In summary, current developments in the research over LBSN data and human behaviour studies are analysed. The findings, in terms of issues and solutions, divide the literature review into three categories, namely Aggregation, Machine Learning, and Accessibility. Each category forms a whole of identified problems and/or innovative solutions that is combined together with the other categories in order to establish a methodology framework.

3.2. Aggregation studies

In a broad meaning, the term aggregation is linked to a human behaviour that gather together a whole of people for some unidentified reasons in the same urban space at the same time. To stay in the touristic context of this study, the aggregation is the tourism behaviour of gathering in proximity of a touristic venue because of the attractiveness generated by that venue. The aggregation has been study in a number researches and the outcomes are here examined with the purpose of introducing important components of this work such as the space partitioning approach to generate the service areas of venues for the computation of attractiveness and the concept of *self-representation* from which to identify the reason(s) that draw people - and more into the specific tourists – to aggregate in space and time. Moreover, bottlenecks of research are argued in order to think of a solution to tackle them.

For instance, in the work of Joseph et al. (2012) they have classified the aggregation of people in accordance to the "latent topic" hidden in the text messages collected from Foursquare by using a topic modelling approach. The findings depict a correlation between aggregation patterns and text semantic, producing an interesting classification of the venues. On a larger granularity, Lee at al. (2013) proposed a crowd-based urban characterization of major cities in Japan by comparing the aggregation patterns retrieved by Twitter with the typology of urban space they were found in. In addition, the work of Cranshaw et al. (2012)

[5] Wikipedia [f], (2015): unsolicited or undesirable electronic messages especially advertising, as well as sending messages repeatedly on the same site.

have revealed a correlation between the user location and the proximity of POI generating "neighbourhoods" over the city in terms of typology of nearby activities. Muntean et al. (2012) and Ferrari et al. (2010) have classified the aggregation via the K-means clustering algorithm. This is an approach that group near occurrences together with regards to the proximity of neighbourhood point data within the clusters itself.

A major aspect of the aggregation techniques, particularly in relation to the space utilization analysis, is described in Lee at al. (2013) and Meyer (2010). They argued that in order to study the aggregation patterns for a target region, this must be subdivided into "Regions of Interests". They described three kind of space partitioning:

- Grid space-partitioning;
- Administrative areas, and
- Clustering-based partitioning.

The outcomes are very diverse and the choice of one partition method over another is dependent by the type of analysis being performed. However, the last method seems to reflect a rather natural space partition as for the case of the Voronoi diagram (e.g. using the K-means cluster centre as the centre of the Thiessen polygons).

Other problems of aggregation analysis have been described by Tasse & Hong, (2014) and Joseph et al. (2012). The former argued that the amount of tweets incorporating the geo location is roughly 10 percent the total amount of tweets relative to a certain area, meaning that the majority of users is still not aware of it or perhaps there is concern about privacy. The latter, instead, argue that the behaviour of users towards posting their location in a particular place is related to the concept of ***self-representation***. The concept is simple, yet fundamental to this approach as it impacts where and how a user share his/her location to the public (See Paragraph 5.1, Assumption 1). A user might avoid posting in places that are commonly seen as 'bad' by the public because of the type of self-representation that others would derive from. For instance, a tourist visiting Amsterdam might not want to share location information when in a fast food or in the Red Light District because of the image that he/she would show to the world.

These bottlenecks can be a limitation to the research on this and related topics and it has to be taken into account. Moreover, also the used K-means algorithm presents some limitations such as the subjectivity in the choice of the number of clusters (K). The K is defined by the user through the evaluation of the outcomes retrieved by means of several attempts on sample data. This generates an analysis strictly related to the variables involved (e.g. location, nature of the topic undertaken, time), hence it is not generally applicable to other contexts (e.g. cities or topics), unless major adjustments are undertaken.

3.3. Machine leaning techniques

Machine learning is the discipline that uses a learning algorithm and example inputs to build a model that make predictions or decisions, converting raw data into information (Wikipedia [g], 2015). In this approach, different machine learning approaches are combined in order to transform raw data collected from Twitter into information, with the purpose of identifying similar patterns in hundreds of thousands of Twitter posts. Moreover, machine

learning is sometimes also combined to the powerful Twitter indexing approach: the *hashtag (#)*.

The researches of Chang (2010), Muntean et al. (2012), Azariah & Australia (2012), and Efron (2010) provide interesting results in the use of the hashtag method in order to find specific contents in Twitter. In particular, with the use of statistical methods Efron (2010) concluded that the incorporation of the hashtag in tweets has led to a better query process. In the work of Muntean et al. (2012), firstly, they extracted top hashtags and texts from tweets into three databases using the LDA algorithm (See Paragraph 4.2.7), thereafter hashtags and texts were transformed in vector features using a Vectors Space Model (VSM) for the successive clustering step. Successively, they implemented clustering algorithms to the databases obtaining positive results on hashtags semantic association. Particularly, they have ranked hashtags in clusters with regard to the hashtag frequency.

Bottlenecks are found in the literature analysis when the machine learning techniques such as text classification and topic modelling are examined. This is indeed considered an emerging field when the text, being under study, is a user generated content like the posts of Twitter. Yet, this approach is very dependent from many factors such as vocabulary diversity per user, spelling mistakes, special characters (e.g. smileys, #, @, & and so on) that is found in the text messages of tweets. This kind of noise in the data influences the outcome of the data mining approach, thus it needs to be removed from text. To tackle those issues, a number of works (Bontcheva et al., 2014; Kaufmann et al., 2010; Muntean et al., 2012; Ferrari et al., 2011) have analysed the validity of the Natural Language Processing (NLP) and Latent Dirichlet Allocation (LDA) topic modelling approaches in combination with the K-means clustering technique in order to normalize and classify the text and its semantics in relation to the spatio temporal location of occurrences.

To end, it is crucial to report a problem that the author could not find in any of the literatures retrieved upon this matter: the translation of LBSN posts. This is a major bottleneck due to the fact that most tasks of the NLP methods work only one language per time. Hence, in the case of touristic information, in which a wide number of languages is found, this should be tackled in order to improve the amount and quality of the collected data.

These approaches could support the spatial identification, classification and assessment of urban aggregation patterns through the use of spatial location and text attributes of tweets. Urban areas in which many LBSN activities take place overtime give an indication of places where factors of overcrowding represent an issue that could influence the accessibility of venues. Therefore, works on accessibility matters are described next.

3.4. Accessibility of places in space and time

As Arafat (2012) reports in his research, the assessment of accessibility varies in existing studies. Linear or network distance, travel time and proximity of attraction points within certain distances are approaches utilised to calculate the accessibility of places. For instance, he argues that the accessibility is expressed as the potential of a particular place to interact with the surroundings, differentiating accessibility and mobility, by linking the former to a destination, and the latter to a mobility network. In each and every case, to be able to measure the accessibility of places, a population count is needed (Arafat, 2012; Handy, S.

2004). The accessibility matrix provided below reports merits and limitations of different approaches to calculate accessibility and it supports the choice of which model to deploy on the basis of pros and cons that each approach requires (Table 1).

	Distance	Network Distance	Opportunity	Gravity General	Gravity Hansen
Merits	–Easy to calculate. –Spatial surface can be generated directly by raster analysis. –Good access estimation for highly connected locations.	–Precise measurement of proximity distance. –Easy to estimate on zonal level.	–Used by many articles as accessibility measurement –Easy to estimate.	–Easy to estimate on zonal level. –Value the attraction and distance in the model. –Considered more precise estimation of accessibility.	–Easy to apply on zonal level –Applied on travel time from travel survey or forecasts.
Limitations	–Poor estimator for poorly connected places and neighborhoods with higher block sizes. –Estimate only distance and does not include attraction in the estimation.	–Complex and time consuming to generate accessibility surfaces by this method. –Estimate only distance and does not include attraction in the estimation.	–The equation will estimate that large attractions are more accessible even though they might be far away.	–Applied on zonal level and complicated to be applied on parcel level.	–Applied on zonal level if the travel time for each zonal pairs is estimated. –Complicated to apply on parcel level.

Table 1: Accessibility matrix - Merits and limitations for different accessibility estimation methods. Source: (Arafat, 2012)

Also in Sink (2010) and Shen (1998) a general connotation and a more precise definition of accessibility are given, respectively. Specifically, in the book of Sink, the accessibility is derived by contextualisation of the **Gravity model** of Newton in an urban space. Sink (2010) describes the basic gravity model as origin- and destination-specific models. Through those models it is possible to predict flows from an origin ***i*** to many destinations ***j***, and vice versa. Handy (1992) argues that the spatial structure of a city and how people move within it are linked by the concept of accessibility. As she states, through accessibility one can measure the ease with which particular activities can be reached as well as the magnitude of these activities. According to Handy (1992), two factors are used to assess accessibility:

- The attractiveness of destination (I_i) and
- The cost of reaching it (d_{ij}).

In this study, the gravity model which takes into account the network of interactions between places (origin to destination on network) is used to evaluate the accessibility of a POI in relation to its attractiveness[6] (I_i) and the distance[7] (d_{ij}) from the PoI to a transit node in

[6] The extension of LBSN activities within the service area of a Point of Interest.
[7] The distance from a transit node to Points of Interest in proximity.

its proximity. Indeed, the action of converging into a place is consequent to the action of reaching that place. This is addressed as the *cost* of reaching a particular location, which represents the distance between departure and arrival of a trip and it is used in the measure of accessibility of places. The general gravity model is preferred to the other methods depicted in Table 1 because of its simplicity of application on zonal level (i.e. service areas with fuzzy boundaries) as expressed in both merits and limitations.

Moreover, the accessibility is evaluated with regard to the temporal component by subdividing the collected tweets in two sets: a set made up by week days (i.e. from Monday till Friday) and the second set which includes weekend days (Friday to Sunday). Both sets are successively subdivided in four time spams for each time of the day (Morning 6-12, Afternoon 12-18, Evening 18-24, and Night 24-6).

In summary, the Voronoi diagram for space partitioning is preferred to the others because of the more natural and accurate subdivision of space which it generates in which the landmarks will serve as the seeds to generate the polygons (i.e. the service areas). Furthermore, a more accurate approach for the identification of the number of topics (K) will be investigated. To end, an ad hoc methodology for the normalization (including translation) of text messages of Twitter will be created in order to extract "knowledge" from the database.

4. Theoretical Framework

The purpose of this Chapter is to explain the background theory as well as existing processes that will support the methodology adopted in this work. The topics discussed are mainly three as they represent the most important techniques adopted:

- **Space and time partitioning approach:** the techniques used to decompose the space and time in a number of service areas for the former and time periods for the latter;
- **Geographic Knowledge Discovery**: a **Natural Language Processing method -** the theory as well as the processes behind the extraction of information from textual data of LBSN such as Twitter;
- **Accessibility of touristic venues in space and time: the Gravity model -** the attractiveness and cost distance variables that leads to the evaluation of the degree of Accessibility of (touristic) venues in urban context.

At the end of the Chapter, applications and tool used to support the method are briefly described and links to documentations will be made.

4.1. Space and time partitioning approaches

Space partitioning

In this work, the attractiveness of places is determined by the number of Twitter occurrences within a specific service area. To be able to define the service area of a certain landmark and thereafter calculate the density in its proximity, the approach that appears to produce satisfactory result is the Voronoi diagram approach (Lee at al., 2013). Actually as discussed in Paragraph 3.2, the Voronoi diagram partitioning generates a more natural

division of space in comparison with the other two approaches described. Moreover, due to the rather natural division, the areas being created can be compared if a ratio such as the density is introduced. The Voronoi diagram, also referred to as Thiessen polygons, is a space partitioning method that divides a plane into polygons from a given set of finite points (seeds). The partition is created in a way that each point within a specific polygon is closer to its seed (i.e. the point from by the polygon was generated) than to any other seed (Figure 1). Therefore, this partition reflects a rather natural subdivision of space in which all tweets that are close to a landmark (i.e. the seed) are considered as tourists visiting that landmark.

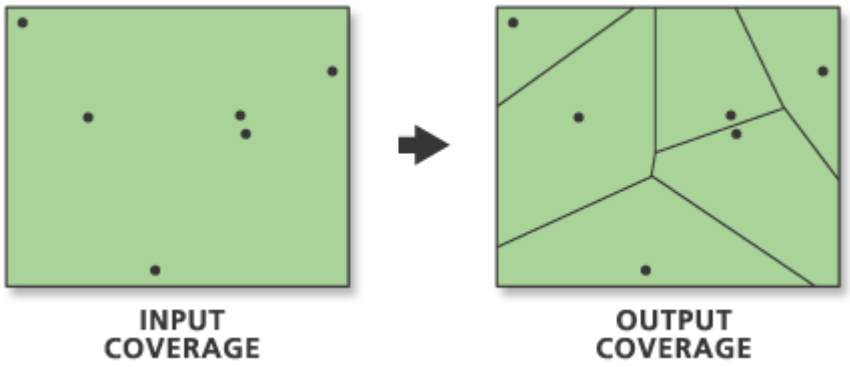


Figure 1: ArcGIS Resource centre - Thiessen polygon tool

Time decomposition

The division of the dataset of Twitter into separate intervals of time is the method that enables the analysis of the time component and it serves to study its influence over the distribution of Twitter content in urban space. The division of time is a task that consider the opening time of the activity or activities that are of interest to a research topic. For instance, in this research the range of activities considered are those related to tourism such as museums, monuments, guided tours, public attractions, shopping and leisure activities. To be able to divide the time accordingly, the opening and closing hours of those activities are obtained by a list of web portals related to tourism in Amsterdam. Among the names, the Iamsterdam official page[8] is a source of information for tourists and not that helps to find a wide range of info, from opening times to events and tips. Moreover, other sources such as thingstodointhenetherlands[9] from which a detailed list of opening times for many different activities was found. The division of time adopted in this work is also compared to those found in Joseph et al., 2012; Dykes et al., 2003) which return in both cases promising outcomes. In fact, they divided the day into four different periods of time: Morning, Afternoon, Evening, and Night, and they studied the effect of the time over the formation and extension of aggregations. With this method, they were also able to visualize how the land use in urban environment changed and where the aggregations were denser overtime.

4.2. Geographic Knowledge Discovery: a Natural Language Processing method

Background

[8] Iamsterdam official web page: http://www.iamsterdam.com/en/
[9] Thingstodointhenetherlads webpage: http://www.thingstodointhenetherlands.com/

The amount of data produced daily is vast and '*ad hoc*' innovative approaches are needed to be able to translated data into information (Figure 2). Among the sources of data, the intertwining between the Web 2.0 and smartphones penetration have led to the transition of users from information consumers to information producers of enormous amount of data.

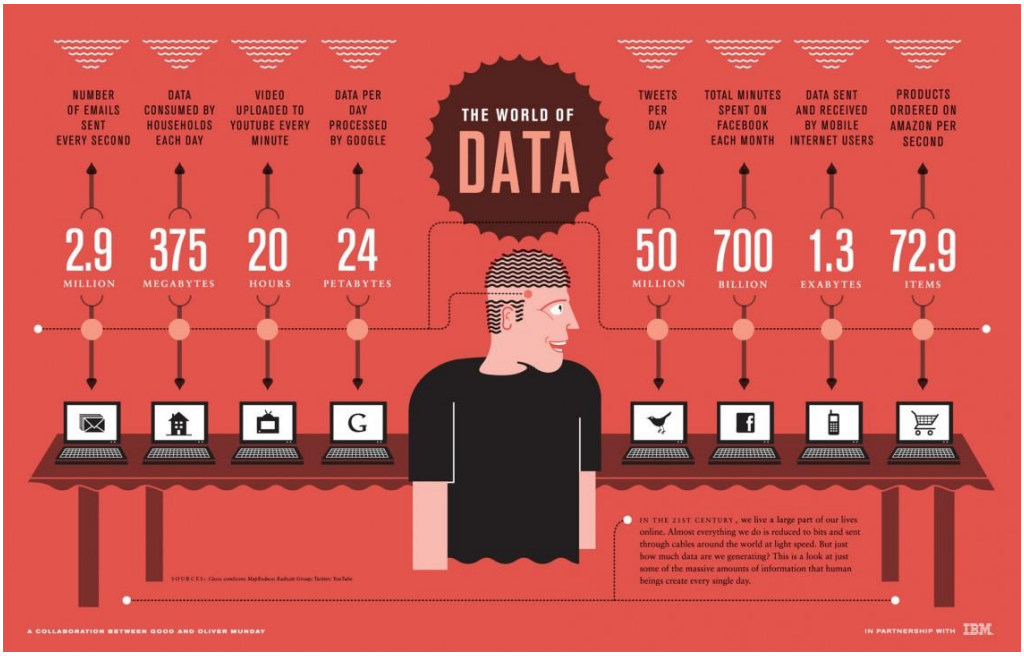


Figure 2: The data penetration over the web. Source: royalonlinemedia.wordpress.com (accessed on 29/01/2015)

Big Data is described by Villars et al. (2011) as a set of technologies and architectures to extract large volumes of heterogeneous data through powerful capturing, discovering, and/or analysing tools. Therefore, a framework of innovative methods and techniques need to be established to extract geo information and knowledge from vast LBSN databases. In this context, the combination of spatial and temporal components with the emerging Knowledge Discovery in Databases (KDD) methods have create a new methodology called Geographic Knowledge Discovery (GKD) (Mennis & Guo, 2009; Laube & Purves, 2006). KDD is a set of methods used to extract high-level information from raw data within large datasets. However, for geographic data, which has unique properties, special KDD and data mining approaches are needed (Miller & Han, 2009).

Geographic Knowledge Discovery

This section describes the current developments in the field of Knowledge Discovery in Databases (KDD) with regard to the tools for text collection from vast archive of data. When the interested data can be also geo located in space, then KDD methodology takes the name of Geographic Knowledge Discovery (GKD). The GKD workflow is similar to the KDD in which data is first gathered and normalized to enhance the accuracy of successive information extraction in relation to the nature of the analysis and for which data is needed. The dedicated GKD methodology that is adopted in this work is the result of a number of techniques, GKD related, combined together in order to extract the information needed out of the vast number of contents produced every day by LBSNs. The method and the connections between them are depicted in the schema below. Bear in mind that this is the general approach used in

Natural Language Processing (NLP) (Figure 3) in order to extract information from data. Later on, a dedicated approach, namely ***TAUS GKD*** is described and displayed by means of a schema (See Paragraph 5.5, Figure 16)

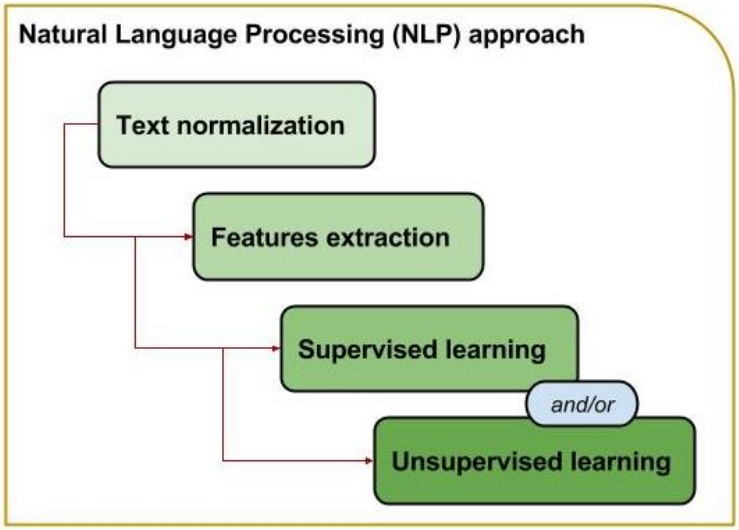


Figure 3: General NLP approach used in Information retrieval (GKD domain). As the shaded colours suggest, the knowledge become more defined as the process advances to the next step.

These tasks above mentioned are strictly linked to the type and accuracy of available data, the pattern (information) to be extracted and the data knowledge *a priori* the extraction phase. In this study, data available from the Twitter Streaming API are processed via a dedicated set of Natural Language Processing (NLP) techniques under the domain of Geographic Knowledge Discovery (GKD) in database. This unique combination of techniques aims to clean the text attribute of tweets, extract important features that help identifying touristic information, and to discover hidden topics from the extracted touristic information. In particular, after reducing posts to a more "natural" language structure, text attributes are deployed into the text classification and topic modelling pipeline. Those approaches help to understand the meaning of the information hidden in posts of Twitter to be able to assess the human behaviour with regards to tourism activities in Amsterdam.

The outcome serves to evaluate the attractiveness of touristic venues in terms of aggregation of tweets in their proximity in order to assess the accessibility of touristic places, thus identifying where and when poor accessibility due to urban congestion may arise. In the next Paragraph, a brief introduction of the main points of data collection from Twitter are described. The description serves to highlight the major issues and limitations in data to better understand the reasons of deployment for the techniques being used.

4.2.1.Twitter and Microblogging

Although the University of Wageningen have already collected since 2013 approximately thirty-four million tweets spatially located in the Netherlands, the author briefly explains the mechanism how the collection of tweets is performed.

Twitter has provided free access to its data through the use of two open-source APIs. It consists of two different parts: the REST API and the Streaming API (Cheng et al., 2011;

Sacco et al., 2013; Mai & Hranac, 2013). The REST API[10] provides programmatic access to read and write Twitter posts. However, there are usage limitations for the REST API[11] in terms of the number of tweets available per user (1500), which make it not suitable to conduct this study (Twitter, 2014). The Streaming API[12] is therefore introduced to enable the unlimited streaming of real-time tweets. Vast amount of data can be then collected by defining an Area of Interest (AoI). In this work, the filter location is set with regard to a bounding box generated by a pair of geographic coordinates enclosing the municipality of Amsterdam, the Schiphol Airport and Bijlmer Arena areas.

A major disadvantage of the Streaming API is its filter mechanism. Although tweets can be filtered by several parameters such as keyword tracking, location tracking, languages tracking and so on, the filter does not allow intersection between parameters. For instance, if one wants to filter tweets by location and keyword tracking, then the filter returns tweets that match the specific location OR the specific keyword, which can be retrieved in tweets at different locations than Amsterdam. For the matter of this research all tweets posted within Amsterdam are queried, many of which are not relevant to the analysis. Thereafter, it is fundamental to extract only the information needed out of the stream, during text normalization and topic modelling. As previously stated in the Introduction Chapter, Twitter is designed as a platform in which users can generate and share contents through "Microblogging". The term indicates the ease of posting small text messages known as tweets, usually enriched with links to web pages, emotional status and/or pictures, in a maximum of 160 characters in each post (20 characters for user name plus 140 for the post) to be able to communicate the user's experience to others. This aspect is very interesting to this research and it is the object of the next analysis step. The embedded text in posts is comparable to a chat style format and its unstructured nature makes the 'hidden topic' (information) extraction very difficult to translate into meaningful text, in most case. Therefore, an approach to 'normalize' the text to a more standardize human-readable grammar structure, hence without slangs, emoticons, spelling mistakes and all other html based special characters. The Natural Processing Language has been created to accomplish this task.

4.2.2. Natural Language Processing

The Natural Language Processing (NLP) is a powerful application that helps to normalize and structure a text document or a set of documents and perform tasks such as text analysis, text classification, and topic modelling (Chowdhury, 2003). The NLP is usually adopted on standard texts such as literatures, papers, books and so on. However, this approach, with some adjustments, has been also successfully implemented on web sources such as email repositories, web pages and social media for tasks like Information Retrieval and Sentiment Analysis. The table below describes the broad workflow adopted in general text mining processing. The procedure can be summarized in three sequential steps as shown in Table 2 below.

[10] REST API: https://dev.twitter.com/rest/public
[11] Connecting and Rate Limiting sections: https://dev.twitter.com/streaming/overview/connecting
[12] Streaming API: https://dev.twitter.com/streaming/overviewsearch

1.Data pre-processing	
Text normalization	**Text analysis**
-Remove punctuation -Remove special characters -Remove web items (if any) -(Translation)	-Words frequency -Hashtags frequency -Language trends -Average n. of words in post
2.Feature extraction	
Text to Features to Vectors	
-Tokenization, Remove stop words, Stemming	
-Feature presence approach	-Feature score approach
3.Machine Learning	
Supervised learning	**Unsupervised learning**
-Define a Training set -Define a classifier -Validation and test sets -Classifier accuracy test	-Vectorization -Dimension reduction -Topics identification -Cluster algorithms

Table 2: Machine learning approach

4.2.3. Data pre-processing

In NLP, data pre-processing is the first step to transform raw text documents into a ***corpus***[13]. Generally speaking, the pre-processing tasks help to improve the performance of the Information Retrieval[14] (IR) process, particularly for noisy data such as those collected from web sources like social media (Muntean et al., 2012). In fact, a major bottleneck of this and related works is that the majority of text messages incorporated into social media posts include a wide range of issues. Special characters (@, emoticons, http-based entities and so on), punctuation, as well as grammar and/or typo mistakes are very common and need to be removed prior to performing any kind of analysis.

The goal of this task is to convert unstructured text from Twitter into structured text data from which one can extract specific words that help to understand what the topic of a text document is about. *Tokenization*, is the process used to break text into its *tokens* (words) in order to perform analysis, such as the distribution of words and hashtags frequency. Normally, in the tokenization process, white spaces, punctuation, and special characters are discarded given that their presence arises redundancy and bias (e.g. in text analysis, '*#travel*' or '*travel… !*' differ from just '*travel*' as they are considered unique entities). However, in this case the deletion of special character does not include the hashtag (#) which is going to enable the next phase of text classification.

[13] The term corpus (plural, "corpora") is used to describe a set of documents or sentences that have been manually labelled with the correct values to serve as 'training' for a model. In the case of Twitter, the corpus is assumed to be the collected tweets, each of which is considered as a small document containing a bunch of terms used to describe a particular topic.

[14] IR is the process of findings specific information from a collection of information resources (Wikipedia[b], 2015)

After the pre-processing task, the tokenized documents are examined in order to identify patterns within the text that can support the text classification, such as the most common recurring words, most popular hashtags and the average number of words in each document.

4.2.4. Text analysis

The method used to identify trends over the data is the Frequency Distribution Analysis (FDA). The FDA serves to identify features (i.e. words and hashtags in Twitter posts) that give an insight of the underlying topic or topics within a group of text documents under analysis. It is an informative approach through which one can investigate the possible presence of latent topics in the text from the frequencies of certain words in it. Moreover, in the case of Twitter data, the powerful hashtag (#) indexing approach can be combined with the FDA in order to produce a first classification method of what could be meaningful to this study and what is not (e.g. by querying hashtags like #travel, #tourism, or #holidays, it is possible to narrow the findings only to those topics mostly related to tourism activities).

In this specific case, the FDA analysis is useful to enables the identification of popular hashtag features in text. Hashtags can be used as the index to group posts by similar topics. Generally speaking, similar topics make use of words whose meaning is connected to the topic they express. Therefore, for the sake of clearness, if tourism related topics are selected via hashtags, words included in those topics can be used as the reference (i.e. the *dictionary* in NLP jargon) to serve as the baseline for the extraction of possible touristic features. This dedicated approach is generally known as the ***features extraction*** process and it is normally represented by a dictionary of unique words which enables text classification and topic modelling techniques.

Interestingly, the frequency of common languages is also calculated in order to evaluate the most popular languages within the text in the case of multi-lingual corpora (this subject is further explained in the Methodology chapter). In fact, a problem behind the language variety is that NLP tools can process only one language at time. Therefore, a translation process has to be set if there is more than one language in the corpus. The feature extraction approach and its modality of deployment are further explained in the following paragraph.

4.2.5. Features extraction

In machine learning, the features extraction is a crucial task from which a number of different techniques can be initialized. Basically, the purpose of the features extraction process is to retrieve specific features of interest (i.e. touristic information in posts) out of a corpus. According to the technique to be performed, textual features are transformed into vectors of identifier (Muntean et al., 2012). To be able to accomplish this transformation, first all unique features are included in a dictionary which is used as a baseline to establish which feature is important within a corpus. Thereafter, the features (i.e. the words) are extracted from the corpus as a whole and weighted according to two main methods, each of which enables a different machine learning technique:

- **Bag of Words model** (BoW), [it enables text classification]

- **Term Frequency – Inverse Document Frequency** (TF-IDF) [it enables topic modelling]

The first approach is known as the *Bag-of-Words* (BoW) model and it is a simple model to classify a text with regard to the occurrence of its words. The outcome of this model, with some manual adjustments[15], is used as the ***features set***[16] to train a classifier[17]. The output of the BoW is in the format of a Python dictionary such as for each word in posts an identifier is assigned as well as a *'1'* to express the presence of that word and *'0'*, otherwise.

The second approach is the *Term Frequency – Inverse Document Frequency*, also known as the 'TF-IDF' in short. The TF-IDF is a numerical statistic that establishes the importance of a word feature in a corpus in terms of a ***score*** assigned to it by the TF-IDF model. This score is linked to the number of times the same word appears in different documents and it takes into consideration that some words are generally more frequent than others (e.g. stop words normally occur more often in a text). Both methods are chosen on the basis of the desired machine learning approach to be performed. Specifically, the former method is often used in text classification, in which the presence (or absence) of a feature is used for feature extraction and *features set* population. In contrast, the outcome of the TF-IDF model, in terms of its scores, is used in topic modelling as the input for the clustering algorithm to group documents by the words with similar scores they contain.

When the extraction of features follows a specific pattern, based on a pre-existent knowledge over the features to be extracted, the approach falls under the *supervised* machine learning domain. The supervised machine learning is basically a classification method that takes a feature set, it assigns labels to each feature according to what is to be extracted and it deploys it as the baseline to train a classifier tool (See Paragraph 4.2.6). The classifier learns how to classify new unlabelled features that are fetched into it after the training task is successfully accomplished. On the other hand, if nothing or relatively little is known about the features being analysed, an *unsupervised* machine learning approach is preferred. The unlabelled data is introduced into a learning algorithm, through which hidden patterns can be identified and visualised. Typically, the unsupervised approach make use of a *cluster algorithm* such as K-Means clustering, which utilizes variables such as distance or similarity to aggregate the unlabelled instances together.

In the next paragraphs, the supervised as well as the unsupervised machine learning approaches are described in terms of their main components, and characteristics of implementation.

[15] Normally the Bag of Words is a representation of a text document of a certain length like a paper, a journal and so on. However, in the case of Twitter posts, the adjustment is done by considering as a document each unique tweet and only few words (max 140 chars) are enough to obtain an outcome, nonetheless.

[16] The features set is a set of words that have been manually labelled on the basis of an existing knowledge over the data under study in order to define a baseline knowledge to train a classifier. Labels represent the Class entity in the Object Oriented jargon and they are manually defined by who performs the classification. Features are the *instances* that belong to a Class in Object-Oriented approach.

[17] In machine learning a classifier is a tool that group instances in classes following the directions given in the training set.

4.2.6.Supervised machine learning: a text classification method

In supervised machine learning labels[18] and features[19] are manually chosen on the basis of a priori knowledge upon the data. The selection of those attributes is rather subjective and it depends from the information that needs to be extracted (e.g. If I am to extract touristic information, I am going to select posts whose topics regard touristic activities and/or typical venues of tourism). The set of features that is manually labelled is known as the ***features set*** and it is the input of a classifier. The features set is divided into two sub sets: the *training* set and the *test set.* The classifier *'learns'* how to group unlabelled features from a set of manually labelled feature (i.e. the training set) defined in advance. Thereafter, the test set is used to assess the accuracy of the classifier prediction. Therefore, this analysis is rather subjective and its result depends from the questions that need to be answered (e.g. *which posts incorporates touristic features?)*.

The figure below shows the supervised learning workflow (Figure 4). Once classes are defined and features are extracted into the Bag of Words model, the procedure continues with the creation of a dedicated machine learning algorithm, (i.e. the training set) and the choice of a classifier.

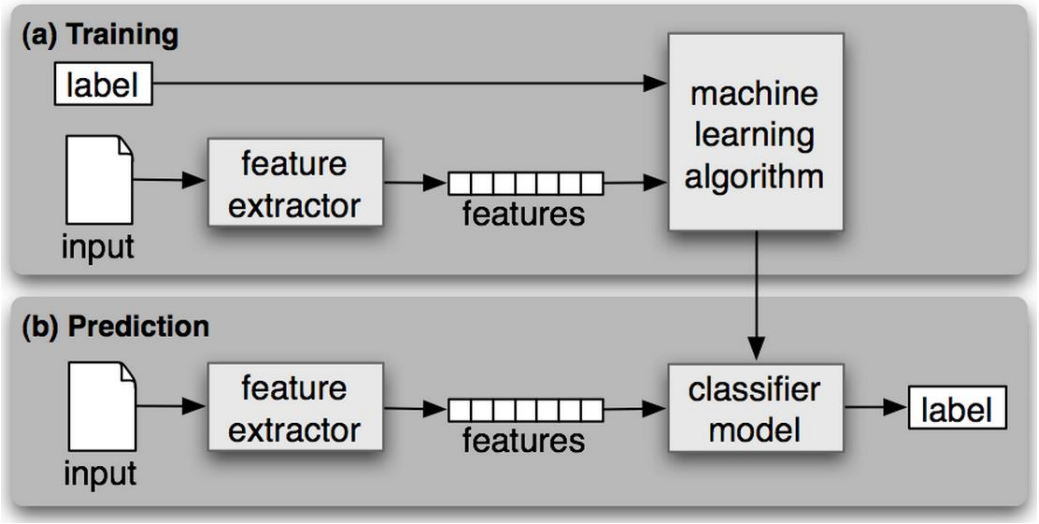


Figure 4: The supervised machine learning workflow. Source: NLTK Documentation @ http://www.nltk.org/book/ch06.html (accessed on 02/02/2015)

A classifier is a tool that returns a classification of text from a manually generated training set. The training set is represented by the selected features (i.e. the BoW model), extracted in the previous phase, which have been manually labelled with the chosen classes. The classifier learns what feature belongs to which established class and it classifies new, non-labelled features accordingly. A validation test is also created in order to check the model performance and its degree of classification accuracy. In text classification, two methods are widely used in literature:

[18] Labels represent the Class entity in the Object Oriented jargon that differentiate the features according to way those are manually assigned to user defined features within a features set.
[19] Features are the *instances* that belong to a Class in Object-Oriented approach.

- **Binary** / it labels extracted features as positive or negative, and
- **Multiclass** / assign one or *n* labels to each feature;

The choice of the classification method strictly depends by the outcome that has to be achieved. If the outcome is represented by mainly two classes of features, one in which there are tourism related features, and another class in which there are non-tourism related features (all the rest), then the binary classification is the best approach. On the other hand, in the multi class approach the number and nature of topics in the dataset is known a priori. Therefore only those features that belong to one or more classes are extracted. A drawback of those approaches is that all the features which do not belong to any of the chosen classes are treated as unknown and therefore they are disregarded from the analysis.

Next to the choice of the classification method, and the creation of the training set through the features extracted from the Bag-of-Words model, the classifier needs to be selected. There exist many classifiers, each of which performs differently in accordance to the type of analysis undertaken. Bayes Classifiers, Support Vector Machine, Artificial Neural Networks, and Decision Tree classifiers are the most commonly used method of text classification. In Pak & Paroubek (2010), the Naïve Bayes Classifier (NBC) is preferred over the other approaches due to its simplicity of implementation and the fact that it performs well when combined to a binary classification (e.g. Sentiment analysis with two classes: positive and negative). After the classifier is trained and its accuracy verified, the test data (i.e. the unlabelled data that need to be classified) is fetched into the "trained" NBC which classifies the unlabelled text on the basis of what it has learnt from the training test. Hence, the better the training data the better the accuracy of the classification is.

4.2.7. Unsupervised machine learning: a topic modelling approach

In contrast to supervised machine learning, the unsupervised machine learning is used when the knowledge of the data prior to the analysis is not extensive or if the number and/or the nature of hidden topics is unknown. Nevertheless, the data can still return a class subdivision by means of the "latent" topic information incorporated inside each document. To be able to find the latent topic in a corpus, an unsupervised machine learning algorithm is necessary. As Blei et al. (2003) argue, Latent Dirichlet Allocation (LDA) is a generative probabilistic model for collections of discrete data such as text corpora. Within the corpora, documents contain random latent topics, each of which is characterized by a distribution over words. At each run the model assign terms to a selected number of topics in different way, yet maintaining similarities in the meaning.

The strength of this approach is the ability to aggregate similar objects together. LDA algorithm returns a distribution of how different objects regard latent topics, as well as a distribution of how different latent topics constitute entities that we can observe. Its original application is topic modelling, where 'hidden themes' are topics, 'objects' are words, and 'observed entities' are documents in a corpus. LDA algorithm is not applied to solely articles or topics, but if the concept of article is extended also to web sites, reviews, tweets and social media in general, LDA can serve to detect important aspects also over a smaller portions of text.

Prior to implement the LDA algorithm, each documents is converted into vectors by means of scores assigned to its words. This approach falls under the domain of Vector Space

Model (VSM) and it can be implemented by means of different, yet similar techniques. Those weights represent the number of times a specific word appears in different documents (i.e. the TF-IDF method) and they are deployed as the parameters of the LDA algorithm. Figure 5 shows the graphical model representation of LDA.

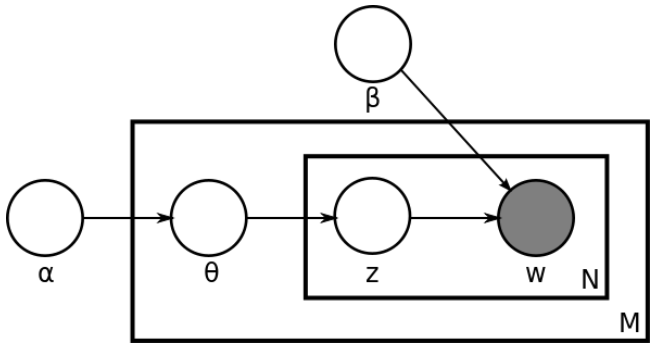


Figure 5: Nodes are random variables, shaded nodes are observation, connectors are the conditional dependencies, and the plates (squares) are replicated components. Source: Blei et al. (2003)

Where:

- Boxes represent plate notations, a method of representing variables that repeat in a graphical model;
- M is the number of documents;
- Plate M is the recurring choice of documents in a corpus;
- N is the number of words in a document;
- Plate N is the recurring choice of topics and words within a document;
- α is the *dirichlet* parameter prior on the pre-document-topic distributions;
- β is the *dirichlet* parameter prior on the pre-topic-word distributions;
- θ_i is the topic distribution of *i;*
- Z_{ij} is the topic for the j^{th} word in document *i;*
- W_{ij} is the specific word or term (the only observed output of the algorithm).

In LDA, a document can be described by the scores of its words and how those scores relate to other latent topics in the corpus. Whereas, each latent topic can be described by the words that are mostly linked with it. For example, a corpus related to tourism activities might include words like 'Van Gogh' or 'portrait' which are linked to a topic *n*, whereas words like 'boat-cruise' or 'tickets' associated with topic *m*. LDA would calculate the degree of association of the two topics with the document according to the weights of the words (e.g. Topic *n* could be about 'museum' while Topic *m* might regard 'trips').

If one considers the association of topics in a corpus, it is likely that a set of documents with similar topics can be aggregated together. The purpose of this approach is to demonstrate whether documents denoting similarities share also similar spatial locations. This is validated and visualized via a clustering algorithm, namely K-Means. The K-Means algorithm is a powerful clustering tool which finds a partition of spaces such that the Sum of

Squared Errors[20] (SSE) between a point and its centroid (mean value) is minimized (Jain, A., 2010).

Next, the method used to evaluate the accessibility of venues in space and time is supported by the implementation of previous approaches in which the *gravity model* and time decomposition have returned positive outcomes. These are described in the following Paragraph.

4.3. Accessibility of touristic venues in space and time: the Gravity model

The measure of the degree of accessibility of touristic landmarks in space and time is the end product of this study and it is performed through the use of a dedicated gravity model on the basis of the works of Sink (2010) and Handy (1992). The gravity model takes into account the network of interactions between places (origin to destination on network) and it is used to evaluate the accessibility of a PoI in relation to its ***attractiveness***[21] (I_i) and the ***distance***[22] (d_{ij}) between a transit stop in proximity of touristic landmarks. The distance represents the *cost* of reaching a particular location and it is used to explain how difficult is to reach that location. Hence, higher distances between transit stops and PoI return areas with poorer accessibility as the distance increases Handy (1992).

In addition to that, the accessibility is calculated considering the temporal aspect of aggregation which is enabled by including the time attribute for each attribute data. This means that a temporal framework has to be established in order to analyse the degree of accessibility in space as well as in time. On the basis of the approaches followed by Joseph et al. (2010), Noulas et al. (2011), Ferrari et al. (2011) and Lee at al. (2013) the author proposes a temporal framework dividing the time span in day of the week (week days and weekend days) and times of the day (mornings, afternoons, evenings and nights). The combination of the dedicated gravity model and the temporal division is thoroughly explained in Methodology (See Paragraphs 5.3)

To end, the methodology and tools utilized to accomplish the analysis are mentioned in the last two paragraphs. These are represented by mainly three types: a programming language application (Python), used to accomplish machine learning tasks and time decomposition, a GIS platform (ArcMap) through which collected and processed data are graphically displayed on map and a database service to create shapefiles from point data to be used by the GIS platform.

4.4. Python and API modules

Python programming language is a powerful tool to seam functionalities of a multitude of applications together on a unique environment. Python enables the user to upload specific modules that work as library of tools which can be accessed from the Python console and utilised to perform different tasks. For instance, to obtain the functionality of the

[20] Errors are also referred as distance given the ability of K-Means to display the errors in a 2D Cartesian space
[21] The extension of LBSN activities within the service area of a Point of Interest.
[22] The distance from a transit node to Points of Interest in proximity.

Twitter Streaming API[23] through which one can access and gather data from, the Tweepy[24] python module has been implemented. The module provides many of the functionalities offered by the Twitter API (e.g. access, stream, filtering and so on). API stands for Application Programming Interface and it can be thought of as a library that includes specifications for routines, data structures, object classes, and variables for a particular web page or web application (Wikipedia[a], 2015).

Among the modules that have been used in support of Python programming language, the most important are

- **Natural Language Tool Kit[25] (NLTK) combined with TextBlob[26]**: libraries that process textual data for common Natural Language Processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and feature extraction.
- **Gensim[27]**: free Python library designed to automatically extract semantic topics from documents through a (machine) fast and user-friendly approach. Among its features, Bag-of-Word, and LDA models are part of this framework.
- **Scikit-learn[28]**: simple and efficient machine learning tools for data mining and data analysis. Amongst its many features, classification and clustering techniques are of most interest to this study and they are implemented in the process of learning from tweets.
- **NumPy[29], Pandas[30] and Matplotlib[31]**: well knowns Python modules used for calculation, database management, and visualization tasks.

4.5. GIS platform

To be able to calculate the accessibility of touristic venues, the author uses a GIS platform for processing and visualization of geographic data: ArcMap, the 10.1 version. The tools which are deployed in order to support the analysis are described as follow:

- **Spatial Overlapping Frequency model**: an ArcGIS implementation to detect the occurrence of spatial point data at same unique locations overtime.
- **Voronoi diagram**[32]: used for space decomposition (also referred to as Thiessen polygons),
- **Spatial Join**[33]: count tweets in polygon to compute the extent of aggregation,
- **Euclidean Distance tool**[34]: deployed to assess the cost distance (d_{ij}) from transit stops to close landmarks,

[23] Streaming API: https://dev.twitter.com/streaming/overviewsearch
[24] Tweepy python: http://www.tweepy.org/
[25] Natural Language Tool Kit - NLTK: http://www.nltk.org/
[26] TextBlob: https://textblob.readthedocs.org/en/dev/index.html
[27] Gensim: http://radimrehurek.com/gensim/index.html, https://pypi.python.org/pypi/gensim
[28] Scikit-learn: http://scikit-learn.org/stable/
[29] Python NumPy library: http://www.numpy.org/
[30] Pandas Data Analysis library: http://pandas.pydata.org/
[31] Python Matplotlib library: http://matplotlib.org/
[32] ESRI Support: http://support.esri.com/en/knowledgebase/GISDictionary/term/Voronoi%20diagram
[33] ESRI Support: http://resources.arcgis.com/en/help/main/10.1/index.html#//00080000000q000000
[34] ESRI Support: http://resources.arcgis.com/en/help/main/10.1/index.html#//009z0000001p000000

- **ArcMap Online base map**[35]: the background map of the study.

The shapefiles used in ArcMap are provided by the database service, namely PostgreSQL. The database aids the process by transforming the geo location coordinates in the datasets collected via Twitter into a geometry field that can be read by ArcMap.

In the next Chapter, the description of the methods as well as the tools and applications utilized to achieve the objectives of this work is illustrated. The Chapter begins with an introduction in which the steps undertaken are thoroughly explained in order to enhance the understanding of the reader upon this new field of Research.

5. Methodology

The methodology to assess the accessibility of touristic venues with regards to the *attractiveness* they generate and the *cost distance* to reach them is described in this Chapter. In addition to that, the *topics* discussed via Twitter are analysed to identify about what users "talk" when posting at a particular touristic venue and if the discussed is somehow related to the time and urban context in which it is found. The schema below show the sequence of stages in which this methodology is subdivided:

1) Data collection (from Twitter API),
2) Data processing (space and time partitioning);
3) Data analysis (SOF+FDA)
4) Tourism information extraction & topic semantic: TAUS GKD method (LAS)
5) Accessibility of touristic venues (LAE+NI)

First, I collect data from Twitter using its API and I divide the space and time component accordingly. Then I analyse the collected data following a two-fold approach using the geo location (SOF) and post (FDA) attributes so as to clean the dataset from spam. Thereafter, I make use of a NLP dedicated method in order to extract only *touristic information* from the wide range of collected (cleaned) data. With touristic information I can assess the attractiveness and the semantic of touristic venues by means of the Landmark Attractiveness Estimation (LAE) and Semantic (LAS), respectively. Thereafter, I calculate the cost distance to reach them from transit stops within a 500 metre buffer. Once, attractiveness $\boldsymbol{I_i}$ and distance $\boldsymbol{d_{ij}}$ are knowns, I evaluate the accessibility as well as the topic semantic of touristic venues with regards to time periods defined in the first stage above. Most important steps are validated through the use of internal (K-Means) and external (HERE data) methods as shown in Figure 6.

[35] ESRI Products: http://www.esri.com/software/arcgis/arcgisonline/maps/maps-and-map-layers

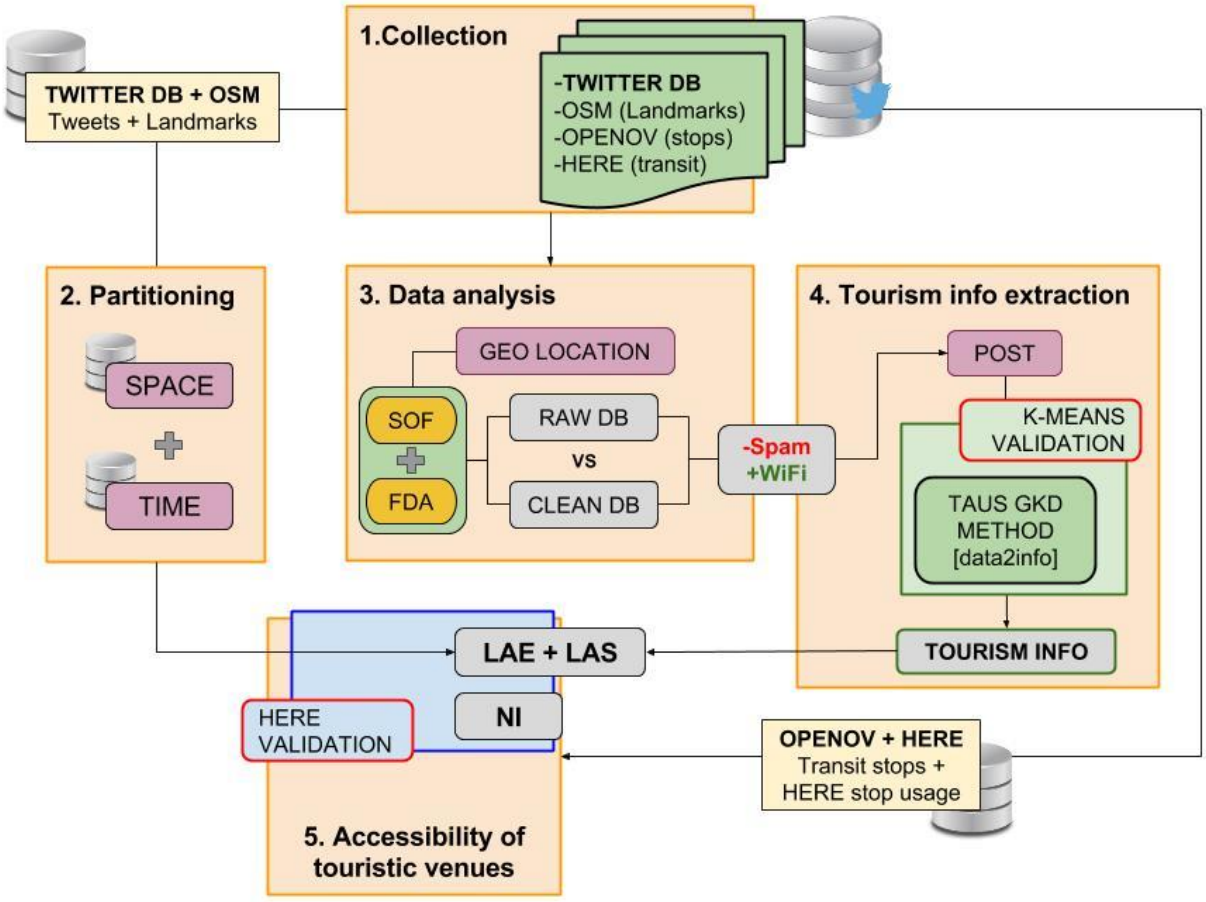


Figure 6: The general methodology schema created to achieve the research objectives. In light yellow, the datasets used at each stage are shown. Steps and their sequence is given by the enumerated orange boxes. Dark yellow represents self-created tools. The results of each step is depicted in grey. Attribute data are shown in purple. Two validations approaches are performed on the most important stages (red boxes).

After the end of the data collection process, the method being built for this research is a three-fold approach in which the *geo location (3)*, the *post (4)* and the *time (2)* attributes represents the main components under analysis. The geo location serves to analyse the trends of tourism aggregation, while the text supports the analysis of the topic(s) being discussed. For each attribute, a set of methods is established in order to achieve the objectives of this work. In particular, during the data analysis I am going to create a specific methodology, namely TAUS GKD solely to extract touristic information and topic semantic.

Bear in mind that the space and time components are processed from external data: the space partition is based on the list of touristic PoI extracted from OpenStreetMap Metro Extract API, whereas the time is decomposed in consideration of approaches used in related works and information upon opening time collected from external websites. The urban space as well as the time period of collection are subdivided to enable the computation of the attractiveness of landmark with the former and the analyse the results at different periods of time through the latter (See Chapters 3 and 4, Paragraphs 3.4 and 4.1). To end, due to the complexity of the TAUS GKD method, I make use of a specification schema to support the understanding of the reader during the explanation of that task (Paragraph 5.5, Figure 16).

The reasons behind the choice of the approaches used in this work to achieve those objectives are supported by means of a number of research assumptions included in the following Paragraph. The assumptions explain major points of the methodology such as those related to the reason of posting for tourists that support the extraction of touristic information, the nature of services from which information are posted, and the willingness of walking to touristic venues which helps in the assessment of the accessibility of touristic venues.

5.1. Methodology assumptions

Assumption 1 - Twitter posts are strictly related to the concept of self-representation desire (Joseph et al., 2010; Lee at al., 2013). Users post one or more activities that identify themselves as being of a certain type. Therefore, for the sake of clarification, a user who likes cultural trips is more incline to share a post which may regard that particular experience (e.g. locating themselves in a museum or taking a picture next to a monument) [linked to Assumption 2].

Assumption 2 – Some textual features, such as the word "*amsterdam*"[36] are assumed to appear in a high number of touristic posts given that the tourist is willing to share the nature of their travel experience. This assumption is confirmed by Van de Ven & Neroni (2012) and it is also related to the Sentiment Analysis study. In fact, in Thelwall et al. (2011) and Bollen et al. (2009) tweet occurrence is driven by popular facts and activities, hence this concept can be extended to those facts and activities related to tourism which are popular worldwide (e.g. *who has gone to Rome without picturing themselves next to the Coliseum?*) [Linked to Assumption 1].

Assumption 3 – Tweets frequently occurring at same spatial location and presenting similarities in the structure of their posts are classified as "*spam"*. Generally, these contents do not relate to human activities (e.g. Twitter analytics such as hashtags trends and ads, or national services such as meteorological stations or emergency services) and they generate bias in the computation of attractiveness (i.e. number of tweets over a specific area), therefore they need to be removed from the dataset. However, not all locations reporting high frequencies are classified as spams and must be removed (See Assumption 4).

Assumption 4 – As Assumption 3 goes, spams denote high frequency of occurrence at unique locations. However, not necessarily many posts occurring at unique locations identify a spam. Generally, touristic venues such as restaurants, museums, attractions and so on provide their customers with free Wi-Fi service. When tourists use these facilities posting contents on Twitter through the Wi-Fi, the location of the internet provider linked to that Wi-Fi is sent to Twitter. Therefore, when collecting posts in Amsterdam, Spatial Overlapping Frequencies (SOF, see Paragraph 5.4.1) in terms of spam (i.e. machine generated contents at unique locations) and free Wi-Fi services (user generated contents at unique locations, see Paragraph 5.4.2) are included in the data. While spams have to be removed, Wi-Fi services

[36] The text is entirely transformed in lowercase in order to reduce word redundancy and improve computational time

are considered a valuable source of information to this work and they are deployed in the calculation of accessibility of places.

Assumption 5 – In the extraction process of touristic features, Twitter hashtags commonly associated with tourism are manually selected in order to extract features (words) from corpus, which are used to train a classifier for entity recognition tasks (See Paragraph 5.5.1.2, Feature extraction). The assumption is that hashtags linked to tourism are likely to include specific words that can teach the classifier to identify touristic activities, thus posts related to those activities.

Assumption 6 - The walking distance threshold away from the most popular landmarks obtained at the end of the LAE approach is calculated for 500 meters, by considering a 5 to 10 minutes range that tourists are willing to walk at an average "***touristic pace***" of 1 m/s[37]. With the "touristic" pace, I intend the walking speed of a person that is involved in touristic actions such as visiting or taking pictures, which is slower than the pace of a person that does not care of visiting the surroundings. I therefore assume that: the longer the distance to be walked, the lower the willingness of tourists to walk hence the poorer the accessibility of a venue.

Assumption 7 – The HERE routing engine log obtained by the HERE maps firm provides information regarding the locations of the departure/arrival stops being used by transit users. Most transited stop locations give an insight of the nodes where the risk of congestion is higher. The purpose of this information is to support and validate the result of the Landmark Attractiveness Estimation, the main variable deployed in the computation of the accessibility of touristic venues. If the network stops in proximity to most attractive landmarks, obtained during the LAE, have had indeed a large number of requests, hence tourists have used them to probably reach one or many touristic landmarks in the surroundings. The validation is therefore true in that case. Bear in mind that the HERE log reports only "requested" trips by the users, but it does not confirm whether the users have physically travelled the requested trip. Hence, for the validity of this information, I assume that if a user requested a transit information it is likely that the trip has been used by the user, eventually.

After the methodology as well as the assumptions to support the choices of the method are in place, I can start with the description of the first step in the approach: the data collection through the Twitter Streaming API.

5.2. Data collection (1)

Data collected from the Twitter Streaming API is a continuous flow of information, therefore a collection scope needs to be set. In this study, the author has collected a total number of approximately 250 thousands tweets collected during the month of December 2014. As explained before, the data have been filtered through the use of the Tweepy location filter, based on the Twitter Place API[38], which allows the collection of all tweets occurring within a pre-defined bounding box. Unfortunately, the location filter of the Twitter API does

[37] From Wikipedia: In the absence of significant external factors, humans tend to walk at about 1.4 m/s (5.0 km/h).
[38] Twitter Place API: https://dev.twitter.com/overview/api/places (see under the bounding box field)

not allow concatenation with the other tracking filter feature (i.e. a keyword tracking filter that retrieve only tweets whose post contains one or many keywords defined in advance). In addition, the location filter reports some outliers (outside the bounding box) showing tweets falling in the sea or in near Countries such England, Belgium, and Germany. These issues are also reported in the recommendation section for future research improvements.

Twitter Streaming API

The first stage of the GDK workflow in this work is performed by means of Python, Twitter Application Manager[39] as well as its Streaming API and the Tweepy module. First step consist in creating a Twitter application to obtain the access credentials necessary for the Twitter Streaming API to gather tweets (Figure 7). Once the application is created and verified through a personal Twitter account, the two access keys and the two access Tokens are incorporated in the Tweepy Python access procedure

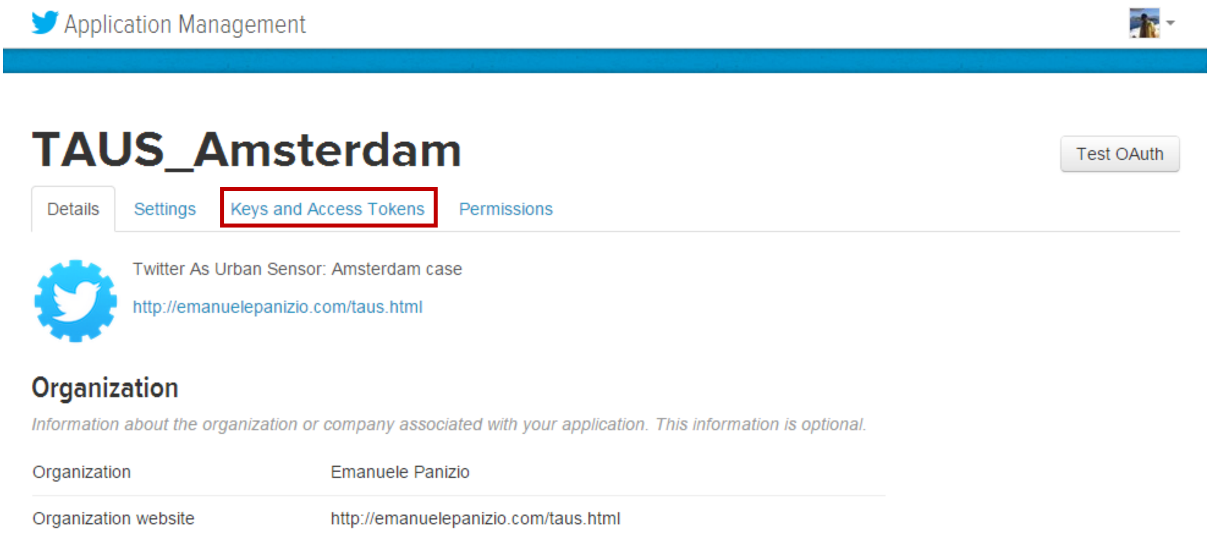


Figure 7: Twitter Application Management_ Twitter as Urban Sensor (TAUS) – the webpage is currently under construction

Consequently, the Keys and Access Tokens[40] (found in the relative tab), are incorporate in the Python script (Figure 8).

```
#access keys
CONSUMER_KEY = '[illegible]'
CONSUMER_SECRET = '[illegible]'
OAUTH_TOKEN = '[illegible]'
OAUTH_TOKEN_SECRET = '[illegible]'

#access procedure
auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(OAUTH_TOKEN, OAUTH_TOKEN_SECRET)
```

Figure 8: Python/Tweepy Access recipe (credential are strikethrough due to privacy matters)

[39] Twitter Application Manager: https://apps.twitter.com

[40] The credential of the TAUS application are not displayed due to privacy reasons. However, it is possible to obtain them for free by creating a own application here: https://apps.twitter.com

Next, the stream of tweets is enabled with the following Python-Tweepy recipe which filters only those tweets that incorporate the geo-location attribute. In addition to that, for all geo located features, coordinates, time of occurrence of the event as well as the text message are also retrieved (Figure 9). The stream is saved in a Comma Separated Value (CSV) file in order to make data export within applications simpler.

```
#stream tweets
class CustomStreamListener(tweepy.StreamListener):
    def on_status(self, status, x=None):
        try:
            #if tweets have coordinates then gather
            if status.geo:
                #needed tweets parameters
                id = str(status.id)
                #screen_name = status.user.screen_name
                xY = str(status.coordinates).split("u'coordinates': [")[1].split(']}')[0]
                x = float(xY.split(',')[0])
                y = float(xY.split(', ')[1])
                timestamp = datetime.datetime.utcnow().strftime('%Y-%m-%dT%H:%M:%S')
                date = arrow.get(timestamp)
                d = date.replace(hours=1)
                ts = d.format('ddd DD-MM-YY HH:mm')
                txt = status.text.encode('ascii', 'ignore')
                txt_noNewline = txt.replace('\n', '').replace('\r', '').replace('\t', '')
                post = txt_noNewline.strip().lower()
                #append to file
                with open('tweets_stream.csv', 'ab') as fp:
                    #tweet format
                    tweet = (id, x, y, ts, post)
                    fp = csv.writer(fp, dialect='excel', delimiter=',', quotechar='"', lineterminator='\n')
                    fp.writerow(tweet)

            else:
                print 'Missing geo location'

        except Exception, e:
            print >> sys.stderr, 'Encountered Exception in ON STATUS:', e
            return True

    def on_error(self, status_code):
        print >> sys.stderr, 'Encountered error with status code in CLASS LISTENER:', status_code
        # Don't kill the stream
        return True
```

Figure 9: Python-Tweepy recipe enabling the streaming

The collected tweets hold the following mutual data model therefore exporting the final result to a database service such as the open source PostgreSQL is seamless (Table 3).

Lon (X)	Lat (Y)	Timestamp	Post
4.763161	52.482938	Fri 28-11-14 15:13	I am at Dam square #Amsterdam http://t.co/96aroempwp

Table 3: Data model of collected tweets (data have been modified to not incur into privacy issues)

This data model is chosen because of the type of data being collected. Each data attribute is needed to perform the research objectives of this work. For instance, the Post attribute is used to assess the behaviour of tourists visiting Amsterdam (See Paragraphs 5.5), whereas the Timestamp attribute is collected to evaluate the effect of time over accessibility trends dividing the dataset in established time periods (See Paragraph 5.3).

Exporting to PostgreSQL database

The information enclosed in tweets is extensive and object oriented as it comes in JSON (JavaScript Object Notation) format. This format is very handy to extract those objects

(information) of interest for this research. Given that tweets are saved in Comma Separated Value (CSV) format and consequently inserted in a spatial database, a mutual data model has to be established. Through the use of Python programming language objects in raw data are extracted and converted in a readable format for PostgreSQL database. **Longitude** and **latitude** (X and Y point geometry), **timestamp** and **post** are the objects considered during the study.

Successively, the "*tweetstream*" table is created in the TAUS spatial database and the CSV file inserted through the "COPY FROM" command. Moreover, to be able to project the geographic coordinates of spatial point entities onto the right location on earth, the geometry column built via X and Y coordinates and the geographical reference code are provided (i.e. the GCS_Amersfoort WKID: 4289 Authority: EPSG) that visualise the point cloud at the right geographic location on Earth (red box Figure 10). Finally, the spatial index is created using the geometry column, in order to improve the outcome of query operations that require a spatial index.

```
--cancel table if it exists
DROP TABLE tweets_stream;

--create a table layout
CREATE TABLE tweets_stream (
        id              int,
        xcoord          real,
        ycoord          real,
        dt              varchar(30),
        pt              varchar(180)
);

--import the csv file containing the table
COPY tweets_stream FROM '..GIMA\Proj\Twitter\tweets_raw.csv' DELIMITERS ',' CSV HEADER;

--change tha table by inserting the primary key and point geometry
ALTER TABLE tweets_stream ADD COLUMN gid serial PRIMARY KEY;
ALTER TABLE tweets_stream ADD COLUMN geom geometry(POINT,4289);

--update the point column
UPDATE tweets_stream SET geom = ST_SetSRID(ST_MakePoint(xcoord,ycoord),4289);

--create the spatial index
CREATE INDEX idx_classified_tweet_geom ON tweets_stream USING GIST(geom);
```

Figure 10: PostgreSQL, table creation, population and geometry insertion

Once the table is created and populated with tweets, the TAUS DB can be exported to a shapefile format to open it in the GIS environment. The process of exporting the dataset is shown in Figure 11 below. The dataset of tweets alone is not enough to accomplish the set of objectives, therefore a number of external map layers such as the background map, the points of interest layer of Amsterdam, and the Amsterdam public transport network (e.g. transit stops and types) are needed.

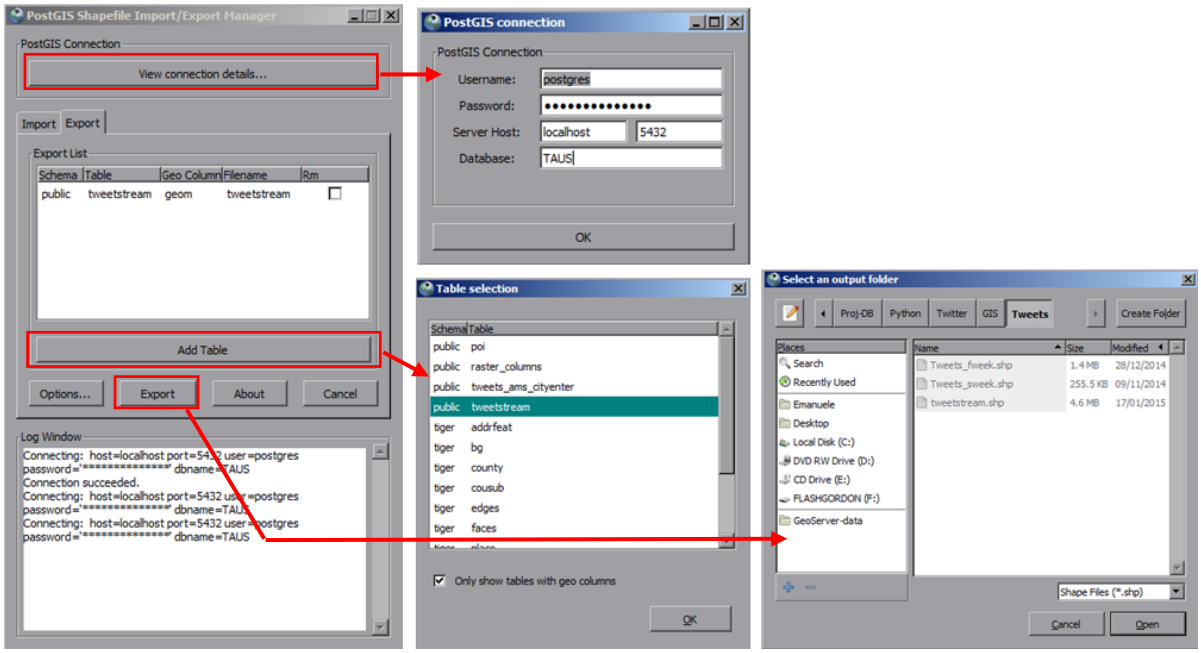


Figure 11: Shapefile exporting procedure (PostgreSQL)

The ArcMap Online service enables to display the background map to easily locate where tweets occur in the city. The layer with the list of landmarks in Amsterdam is downloaded from the OpenStreetMap community, precisely the Metro Extracts API[41] feature. The list includes touristic PoI, like attractions, museums, nightclubs, pubs, but also many which are not tourism related like services and facilities for citizens. Therefore the POI layer is pruned in order to remove the non-touristic features. In addition to the POI dataset, also the Amsterdam public transportation network needs to be gathered and it can be found at the website OPENOV[42].

After tweets are collected and stored in a database with a proper data model and the list of landmarks and transit stops are extracted from OSM and OPENOV, respectively, I can initialize the space and time partitioning. The space is divided by means of Voronoi diagram, also known as Thiessen Polygons whereas the time is decomposed using the timestamp attribute of tweets. The procedures of both partitions are described in the paragraph below.

5.3. Space and time partitioning approach (2)

Prior to the elucidation of the data analysis as well as the introduction of the method I created in the next Paragraphs, I need to clarify how space and time are decomposed to enable the computation of the LAE, LAS and, ultimately, the accessibility of touristic venues in space and time. The space and the time components are crucial parts of the methodology because they impact on the way human activities distributes over an urban area, therefore both attributes need to be carefully partitioned.

Space partitioning

[41] OpenStreetMap – Metro Extracts: https://mapzen.com/metro-extracts/
[42] OPENOV: http://www.openov.nl/

The space partitioning is enabled via Voronoi diagram (or also known as Thiessen polygon space partitioning) which is introduced in Paragraph 4.1. The division is performed onto the touristic venues extracted by OSM in order to generate services areas (the resulting polygons of the partition). The service areas created serve as the "container" to estimate the attractiveness (i.e. number of tweets) as well as the post semantic (i.e. understand the reason(s) why tourists are attracted by specific locations by counting the frequency of semantic labels assigned to each tweet). The list of landmarks obtained from OSM is rather heterogeneous and it includes many types of landmarks that are not of interest for this research like toilets, fountains, benches and so on that need to be discarded. Hence, I retain only touristic landmarks in terms of heritage, leisure and nightlife activities by using facts and statistics retrieved from the official *Iamsterdam* website[43] as supportive reference. In the end, I obtain slightly more than thousands touristic attractions mainly located to the city centre of Amsterdam, upon which the measure of attractiveness is calculated. Hence, popular attractions such as museums, markets, red light district, coffeeshops, pubs and historical buildings are considered. The list of landmarks obtained is the input of the Thiessen polygon tool (also referred to as Voronoi diagram) in ArcMap which divide the space in a number of service areas, one for each of the landmarks considered. The Thiessen polygon tool is implemented by means of Model Builder tool. Tweets (divided into time periods) are added to each service area and counted using the Spatial Join tool as shown in Figure 12:

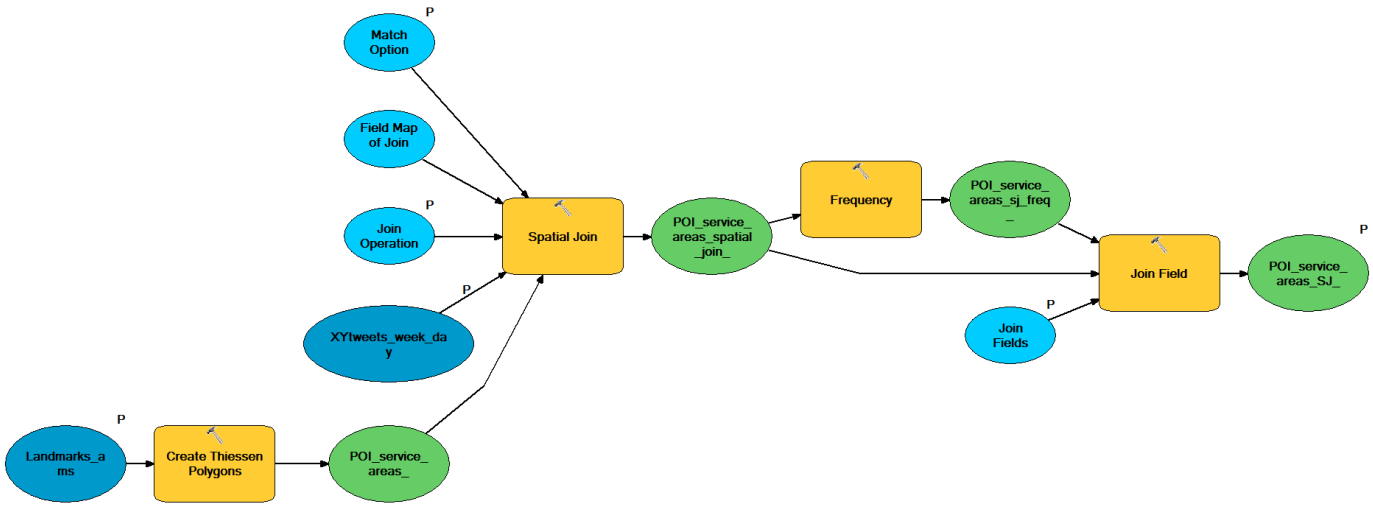


Figure 12: PoI space partitioning. A Thiessen polygons partitioning + Spatial Join approaches

Inside each service area there are all spatial locations closer to its defining PoI than to any other PoI in the surroundings. This particular division makes sure that tweets posted within a service areas are more likely to belong to the landmark in that service area than to any other landmarks around. Moreover, this approach works better in denser urban textures where the distance between PoI is short and more evenly distributed.

Next, I argue the time decomposition approach in which I split the output dataset obtained after the TAUS GKD method is completed. A number of datasets as many time periods are generated. By doing so, I am able to assess the effect of time with regards to the

[43] I amsterdam – What to do in Amsterdam http://www.iamsterdam.com/en/visiting/what-to-do

measures of attractiveness and semantic for each time period and the results compared on map.

Time

The time component is enabled by the timestamp attribute included in tweets. The dataset of tweets is divided into different time periods and in each period there are all tweets which are posted during that specific period of time. The timespan of data collection is of one month, the whole of December 2014. This period includes many common holidays worldwide (Table 4) and therefore the likelihood of tourists visiting Amsterdam may be high. The time component is an important part of the methodology because it impacts on the way human activities distributes over an urban area, therefore it needs to be carefully partitioned as well as the space. For instance, let us consider a group of touristic activities such as culture, shopping and entertainment over a morning and an evening period. During morning, mostly cultural places and shops are open, so the distribution of tourists may be divided on certain extents among these activities. When the evening period is considered, it is more likely that cultural and shopping activities are close, therefore the distribution of tourists in urban environment changes and denser aggregations can be spotted where most leisure activities take place.

Hence, the timespan of collected tweets has to be divided into temporal categories according to the opening and closing times of touristic places such as museums, shops, city attractions, and so on. To be able to partition the time, I make use of website information. Actually, in the Netherlands, there are many websites from which a great source of touristic information like opening/closing times, locations and services can be accessed. However, the opening times (Figure 13 and 14) and holidays (Table 4) in Amsterdam, particularly in the considered period, are very diverse and it is rather difficult to make up an ad hoc time subdivision, so a general framework needs to be established.

Opening hours Amsterdam – **Shops and markets**

Despite one or two exceptions below are the openings hours of shops in Amsterdam:

Opening hours Amsterdam regular	Monday from 1PM till 6PM Tuesday through Saturday 09AM till 6PM
Opening hours Amsterdam shopping night	Thursday till 9PM in the city centre
Opening hours Amsterdam Sunday	12AM till 5PM in the city centre
Opening hours **Albert Cuyp Market**	Monday through Saturday 9AM till 5PM
Opening hours **Waterlooplein Market**	Monday through Saturday 9AM till 6PM
Opening hours **Ikea Amsterdam**	Monday through Friday 10AM till 9PM Saturday 9AM - 8PM Sunday 10AM - 6PM
Opening hours **Villa Arena**	Monday 1PM till 5.30PM Tuesday through Saturday 10AM till 5.30PM (Thursday night till 9PM)

Figure 13: List of opening times of popular shopping areas in Amsterdam. Source: http://www.thingstodointhenetherlands.com/opening-hours-amsterdam-shops-and-museums.php (accessed on 07/02/2015)

Opening hours Amsterdam – Museums

Opening hours Anne Frank House	Winterseason, 9AM till 7PM (Saturday till 9PM) Zomerseason, 9AM till 9PM (Saturday till 10PM)
Opening hours Van Gogh Museum	Monday through Sunday 10AM till 6PM, Friday till 10PM
Opening hours Stedelijk Museum	Tuesday and Wednesday: 11AM till 5PM Thursday: 11AM till 10PM Friday through Sunday: 10AM till 6PM
Opening hours Rijksmuseum	Monday through Sunday 9AM till 6PM, closed 1st of January
Opening hours Nemo	Tuesday through Sunday 10AM till 5PM
Opening hours Jewish Historical Museum	Monday through Sunday 10AM till 5PM
Opening hours Amsterdam Museum	Monday through Friday 10AM till 5PM, Saturday and Sunday from 11AM
Opening hours National Maritime Museum	Monday through Sunday 9AM till 5PM
Opening hours Eye Film Museum	Monday through Thursday 10AM till 10PM Friday and Saturday 10AM till 11PM

Figure 14: List of opening times of popular museums in Amsterdam. Source: http://www.thingstodointhenetherlands.com/opening-hours-amsterdam-shops-and-museums.php (accessed on 07/02/2015)

Day	Holiday
5th Dec 2014	Sinterklaas (Dutch holiday)
24th Dec 2014	Christmas eve
25th Dec 2014	Christmas day
26th Dec 2014	Boxing day
31st Dec 2014	New Year's eve

Table 4: Timetable during public Dutch holidays

As previously stated in Paragraph 4.1, the time period is divided into days of the week and time of the day in order to make a uniform subdivision of time. Although on the one hand this approach is not ideal given that many activities might have a dissimilar timetable (e.g. in some case it ranges from one to a couple of hours), on the other hand the difference in opening/closing time does not impact much the subdivision of tweets in pre-defined time slots with regard to the frequency at which consequent tweets occur (i.e. more than one hour for the majority of consecutive occurrences (Noulas et al., 2011). Table 5 shows the division of timespan in equal interval of time periods:

	Morning	Afternoon	Evening	Night
Week days (Mon-Fri)	6 – 12	12 – 18	18 – 24	24 – 6
Weekend days (Fri-Sun)	6 – 12	12 – 18	18 – 24	24 – 6

Table 5: Timespan partition - four time slots and two week slots

Eight different databases are therefore created using ***Pandas*** and ***Matplotlib*** Python libraries (See Paragraph 6.2). However, due to time limitations and impracticability of clearly showing the effect of time in each of the eight different time spans created, I selected two period that contains the majority of tweets and that are very diverse:

- Mornings and afternoons, in week days;
- Evenings and nights, in weekend days.

In summary, space is decomposed in a number of polygons, generated by the Voronoi diagram in ArcGIS, as many as the number of landmarks considered. Each polygon constitutes the service area of each landmark and it is used to compute the attractiveness I_i as well as most popular topic of the landmark. In the next Paragraph, the data collected via Twitter API is analysed using a dedicated approach which focus on the study of the geo location and post attributes of Twitter.

5.4. Data analysis (3)

The geo location of tweets is utilized to locate the occurrence of tweets overtime in urban environment and serve as the input of two techniques:

- Spatial Overlapping Frequency (SOF) + FDA[44] and
- Landmark Attractiveness Estimation (LAE)

The SOF examines the spatial occurrence of tweets in urban context with regards to unique locations. Moreover, the SOF tool is combined to the FDA tool to verify the nature of SOF being extracted. Actually, a great number of public services such as news feeds, Twitter Analytics share contents via LBSN using a Wi-Fi hotspot. In some cases, the Wi-Fi service is open to customers of a restaurant or a museum, for instance, and it enables to sharing Twitter contents. If posts are uploaded in this way, the geo location displayed in the end content (the one it is downloaded via API) is the location of the Wi-Fi. This is an issue that generates noise in the computation of the attractiveness of a landmark (i.e. LAE), due to the fact that the frequency of tweets is used in that computation. Great sources of non-touristic data such as the SOFs allowed in the computation leads to biased results. Hence the SOF model is deployed to geo locate the occurrence of SOFs while the FDA supports the assessment of their nature. If the SOF is considered a ***spam***[45] (See Paragraph 3.1), it is disregarded in next stages of the study or retained otherwise. Thereafter, the LAE method is deployed to calculate the density of touristic tweets occurring inside the service areas of landmarks in Amsterdam, generated via the Voronoi diagram.

5.4.1. SOF model

Prior to effectively start with the aggregation analysis as well as its semantic investigation, data observation is carried out on the collected dataset. Observations are carried on the two key attribute fields of this work (i.e. the geo location and the text), and they are divided into two consecutive phases: the first phase investigate the collected dataset as it is collected, whereas the second phase is performed after that the dataset is cleaned (e.g. text noise, errors, corrupt data, spams and so on). By doing so, changes during the visualization of data can be identified and analysed. The geo location attribute is examined in ArcMap environment through the use of a model, namely ***Spatial Overlap Frequency*** (SOF), created with the Model Builder tool of ArcMap. The SOF model (Figure 15) takes the X and Y coordinates of each tweet and it calculates its frequency within a certain timespan. The model outcomes a point dataset with the frequencies occurring at unique locations (See Appendix

[44] Frequency Distribution Analysis (FDA) is a Python function I create to analyse the distribution of words in text.
[45] In this work, everything is not somehow related to touristic information is considered noise and addressed as spam, thus it will be removed from further analyses to follow.

F) and its purpose is to reveal the spatial location of places in which the occurrence of tweets follow a repetitive pattern (i.e. large amount of tweets occurring systematically at exact same location during a certain period of time[46]). This approach allows the author to investigate and analyse the presence of ***spams*** and/or free ***Wi-Fi hotspots***.

5.4.2.FDA: identifying Spams and free Wi-Fi hotspots

However, the analysis of the geo location, alone, is not sufficient to distinguish between spam and free Wi-Fi services, therefore I deploy the Frequency Distribution Analysis (FDA) (i.e. a Python tool that I have created to compute word frequencies during the analysis of text described in the GDK workflow). In this step, the purpose of FDA is to obtain an overview of the topics contained in posts, in terms of absolute features (word and hashtag) counts, total unique words, and the average number of words per tweets. The FDA mainly support the identification of words whose frequency denotes irregular patterns (e.g. relatively high frequencies of particular words and/or hashtags in text). The result of the SOF model is compared before and after the deletion of spams to detect the effect that they are capable of generating (e.g. if posts of highly frequent unique geo location detected through the SOF match with specific words and/or hashtags related to a Twitter advertisement service[47], the tweet is classified as *'spam'* and it can be removed from the data to be analysed improving accuracy of the text classification process).

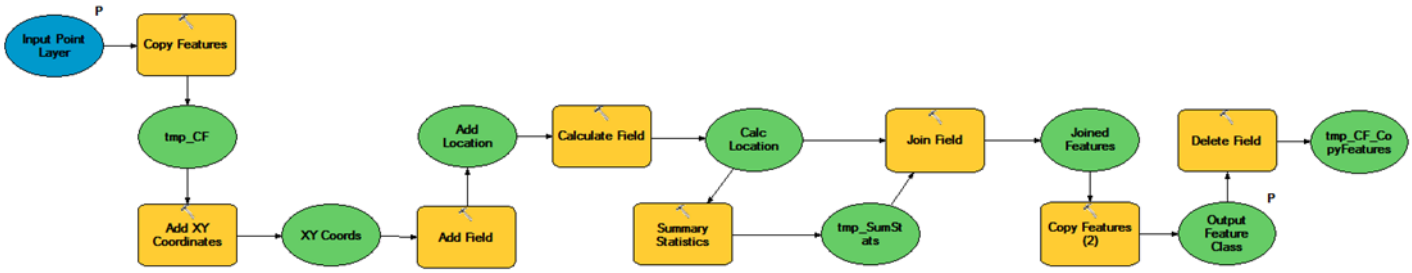


Figure 15: Spatial Overlap Frequency (ArcMap - Model builder)

Spam

As argued in Assumption 3, spam generate noise in the data and they have to be detected and removed. Actually, the computation of the density of tweets used to assess the accessibility of places takes into consideration the number of tweets over a specific service area. Thus, if a spam service is included in the process, the outcome of the density in that area would be untrue because of the frequency peak that a spam normally shows. Hence, they must be removed from the analysis in order to improve computational speed and accuracy of results. Spam can be detected through the SOF model and validated with the FDA methods. In fact, if on the one hand the SOF model geographically locates tweets posted at the unique location over an equal interval of time, on the other hand the FDA exploits the textual features (words and hashtags) of those posts in order to find a connection with the type of SOF and the frequency of particular words and/or hashtags contained in it. In particular, the FDA seeks

[46] The full collection period is of a month and it is selected to show the distribution of SOF on map in December.
[47] A common Twitter advertisement: *"#FeyRij2. #blauwevinkjes3. #Ziggo4. #vote5sos5. #panpsv2014/11/7 10:13 CET #trndnl http://t.co/tOVVBU3TJP"*

to validate that a post is not generated by a human user and the structure of their posts as well as the words they include repeat with a certain frequency. If this is true, the SOF is considered spam and is removed as it can compromise the result of the machine learning process as well as the assessment of density with regard to the spatial aggregation of tweets.

Wi-Fi hotspots

As argued in Assumption 4, besides spams high frequency of tweets could exist also because of free Wi-Fi connection points offered in touristic venues such as museums, attractions, restaurants, bars and so on. In this context, the SOF model is a reliable approach to detect those type of services. In the case for free Wi-Fi services, they actually allow tourists without an internet connection plan, to post their activities on Twitter. Hence, they are useful services that support this work with touristic information. To be able to differentiate spams from Wi-Fi services I need to establish a threshold to classify them accordingly. This is done by using the mean value of calculated SOF as the threshold, defining as spam everything above it and Wi-Fi services, otherwise (See Paragraph 6.3, Section *Raw vs Processes*). Due to the relatively high frequency of spams compared to the frequencies of Wi-Fi services analysed, this represents a reliable approach to discard the first and retain the second.

After that spam are removed from the original dataset, I obtain a *raw* and a *clean* dataset which I can use to graphically compare the effect of spams in map. Actually, by removing noise from data, I can obtain a clearer overview of the information being displayed, and patterns that were hidden due to the influence of spams before, can now be analysed in more detail (See Paragraph 6.3). Following, the output of the data analysis is a dataset absent of spams in which a wide range of information appear.

To end, the wide range of information contained into the obtained dataset needs to be thoroughly cleaned, normalized to English, and pruned to extract mainly[48] touristic information from the database. Hence, a dedicated method is set to extract and *learn* from touristic information.

5.5. Tourism information extraction & topic semantic: TAUS GKD method (4)

The objective of this study is to create a methodology through which to assess the accessibility of touristic venues in the city centre of Amsterdam. Accessibility is influenced by urban congestion which is generated by flow of people gathering – due to the attraction of touristic venues - in specific places at a certain time of the day. In this context, the extent of the attractiveness of venues (i.e. LAE) and their semantic, or LAS, are used to evaluate the attractiveness I_i of landmarks as well as to assess the behaviour of tourists who post their locations within the service area of the landmark they are visiting. In simpler words, I attempt to discover the number of visitors as well as the reason of visiting. For example, in the case of a museum, I assume that the attraction is calculated in terms of number of tweets posted by the customers of the museum, while I assume that the reason of visiting would be mostly "cultural" in a general view.

[48] The Research regarding the extraction of information from LBSN is still at an early stage and results are not always very accurate.

The analysis of LBSN textual attribute falls under the Natural Language Processing approach which is part of the GKD domain. The GKD is a methodology aimed at discovering information from vast datasets in a more efficient and accurate way. To do so, there are a bunch of techniques, tools and models that can be used to perform certain tasks. However, due to the specificity of the information to be extracted in this work, a dedicated GKD methodology is set with regards to the extraction of touristic information from a large number of tweets. The process is as it is described in Paragraphs 4.2.2 to 4.2.7, yet some specific adjustments need to be done to make it fit to the objective of this study: the understanding of the topics discussed by tourists in Amsterdam. The techniques implemented are as follows:

- Natural Language Processing,
- Supervised learning, and
- Unsupervised learning.

The Natural Language Processing (NLP) is a set of tools and methods used to normalize unstructured text data such as those extracted from Twitter into structured data that can be processed through the machine learning abovementioned. In this study, a combination of supervised as well as unsupervised machine learning is needed. Actually, while the supervised machine learning is deployed to extract touristic information from the corpus, the unsupervised machine learning, specifically the LDA algorithm is set to identify the type of touristic information in terms of most popular topics. An introductory explanation of both method is argued as follows:

In supervised learning, as the name may suggest, I already know what is that I need to extract: touristic information. Therefore, I make use of a Naïve Bayes binary classifier to group anything touristic into a class of *'hit'*, whereas the rest is grouped in a class *'miss'*. Thank to this classification, the landmark attractiveness can be computed through the LAE approach and I can proceed with the unsupervised learning technique upon only touristic information (hit). The second approach is made by two different techniques: the Latent Dirichlet Allocation (LDA) and the K-Means clustering algorithms. Both LDA and the K-means are unsupervised learning techniques as they learn from data and not from the knowledge that the author has upon it. However, the number of topics in LDA is arbitrarily decided through several attempts in order to find the best fitting number. Thus, to let the data choose this number, the K-Means is deployed. Actually, the LDA returns the most probable terms linked to the topics found by means of scores that are assigned to each term. By transforming these scores into vectors the K-Means calculated the average Sum of Squared Errors (SSE) and it is able to calculate which number of K-topics best fit the average. Therefore, the K-Means algorithm is used to assess the number of topics for LDA technique, and it shows whether relations, in terms of similarities, exist among the topics retrieved. Topics being extracted via LDA serves as the input of the LAS to be able to assess what is the topic, or percentage of topics, being discussed inside a certain service area.

In summary, a dedicated GKD methodology is therefore needed in order to normalize and extract tweets whose text is linked to touristic activities, and analyse them with a combination of techniques borrowed from the KKD approach. In this work, the TAUS GDK methodology is deployed for Information Retrieval (IR) from large spatial databases and it is the result of a number of different techniques each of which is applied for different purposes.

The sequence and modality of deployment of the TAUS GKD in this case study is shown in the schema depicted in Figure 16, and it is argued in the following Paragraphs.

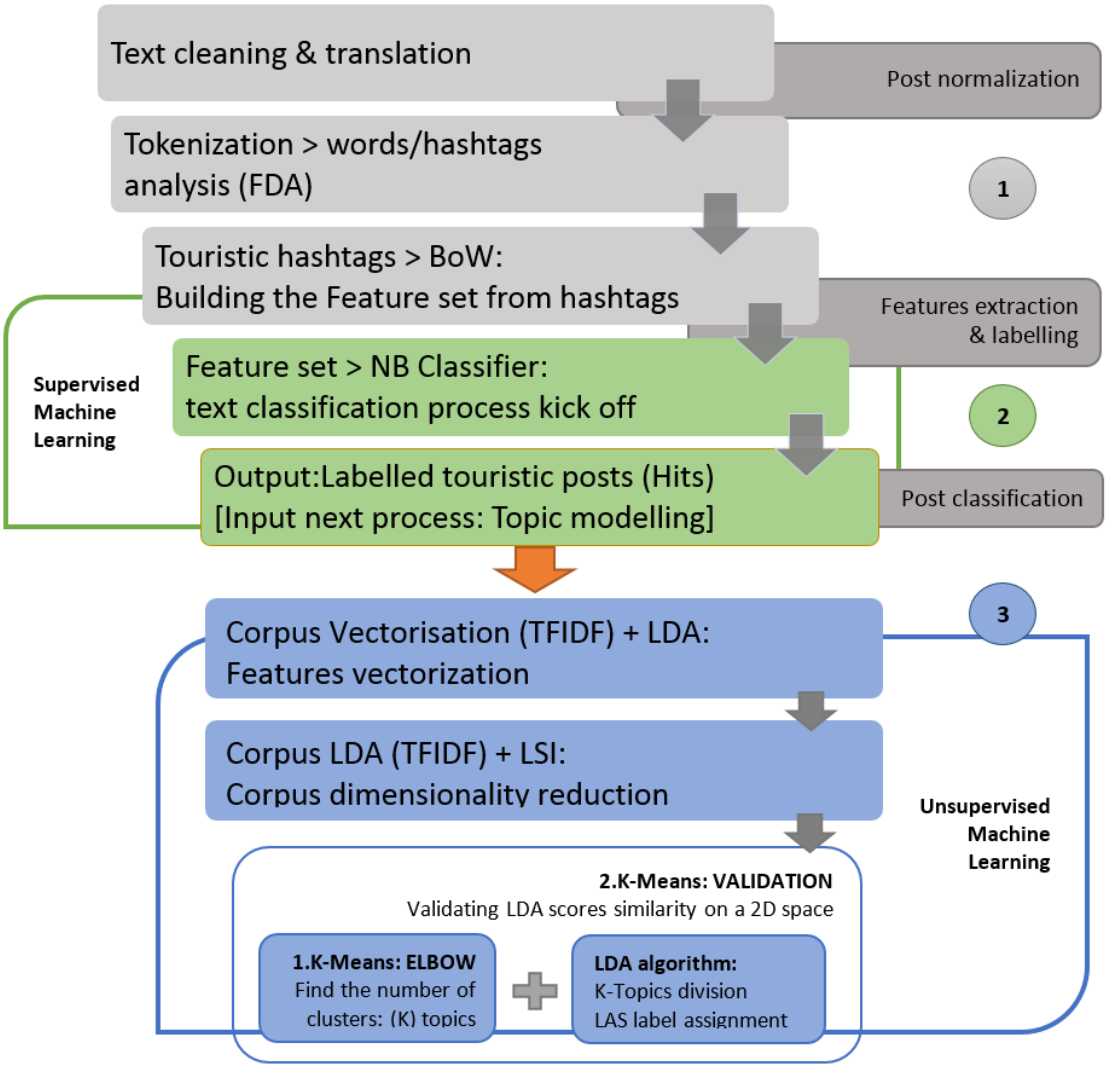


Figure 16: TAUS GKD methodology - Extract touristic information from the database of tweets.

The method is based on Assumptions 1 and 2 set at the beginning of the Methodology Chapter. Both assumptions support the identification of tourism behaviours by offering an indication of the reasons that push tourists to visit specific touristic venues (i.e. popularity of a venue) as for sharing their locations in the very surroundings (i.e. self-representation concept). Therefore, I need to implement a technique to extract information upon most visited touristic places as well as most discussed topics by considering some common attributes of tweets. To do so, Twitter has implemented since 2007 the hashtag (*#*[49]) indexing system to classify its posts in a more efficient way (Chang, 2010). The hashtag is utilized in the TAUS GKD method to query Twitter posts related to tourism activities and the following paragraphs explain how.

[49] Wikipedia[h]: The penetration of hashtag in social media http://en.wikipedia.org/wiki/Hashtag

5.5.1. TAUS GKD method: towards LAS

The following paragraphs guides through the TAUS GKD machine learning process (i.e. also known as Natural Language Processing in a more general view) to be able to extract the topics hidden behind the post attributes of Twitter. As described in the schema above, the process of learning from text is divided into three steps:

1. Post[50] normalization;
2. Feature extraction;
3. Machine learning (Supervised & Unsupervised).

Generally as argued in 4.2.2, NLP is applied on text corpora such as books, articles, papers and so on, commonly grouped as text documents. Each document is normally between one and many pages long and it is often includes few topics. In the case of Twitter, I need to set some adjustments prior to apply the process. Actually in normal NLP, a ***corpus*** is the set of documents that need to be analysed in order to discover hidden information. Documents are represented by a set of sentences which in turn are made of words. In the case of Twitter, I shall consider the corpus as the set of tweets, each of which is considered as a unique document made by words. With this schema in mind I can start describing the various stage of the TAUS GKD approach.

5.5.1.1. Post normalization

Post attributes of Twitter are characterized by extreme noise and bad characters that need to be removed (Table 7). Table 6 shows the overview of the normalization method being described. The process begins with the normalization of text through which Twitter "raw" messages are converted in natural text by eliminating all characters that are not text related (e.g. internet fonts, punctuation, numbers).

Twitter post normalization	
Post cleaning	**Post analysis**
-Remove punctuation	-Words frequency (FDA)
-Remove special characters	-Hashtag* frequency (FDA)
-Remove web fonts	-Language trends*
-Translation**	-Average words/post ratio

Table 6: Tweets pre-processing phase (the hashtag frequency and language trends are performed before the normalization phase given that the hashtag and language diversity must be present in the post in order to be extracted). **Translation is not a usual task in pre-processing and it is used here to normalize the variety of languages in the corpus to English.*

Twitter – A sample of tweets in original format
B1 13407 Amsterdam Valkenisseweg 137 http://t.co/p1Py7imoAf #p2000
@AWMonitor Graag gedaan. PS: Ik ga morgen alvast even goed de Volkskrant lezen. @volkskrant
Bier transport nieuwe stijl. Op naar t #Kluphuis met @duitsenlauret @ Bier&cO http://t.co/GGtyUhJ9XK

[50] To improve the understanding of the method, the Twitter text is referred as ***post***.

amsterdam #leidseplein #iamsterdam #souklan #doaysev #yeilikoru @ Leidseplein http://t.co/FSjVlN1ZEd
Yh man yh man... Dam man ... It's been real tuh bumba
@Fallacy_J AAAIIIT HOSSELAAR @iprolars zwager
Con las loquillas @ amsterdam http://t.co/68j3itEl4j

Table 7: A sample of the average format of Twitter post.

Besides the variety of special characters, translation is also a crucial issue when normalizing the tweets because of the nature of this analysis. In fact, tourists come to Amsterdam from different countries as well as different cities within the Netherlands, and their posts are mostly expressed in the user's native language, particularly if the tweet is related to a specific topic upon which the user express his or her personal experience. Therefore, a method to translate each post from the original language to English needs to be implemented. The NLTK used to offer an API service which used Babylon translator in order to translate text elements. However, for non-identified reasons, this service is no longer available, therefore, another dedicated method has to be found. In this case, the author deployed the Google Translate API offered by another NLP python tool, namely TextBlob (See footnote n. 26). The approach scans the text, evaluate the most probable language in which the tweet is written and it returns the English translation if it detects a different language than English (See Paragraph 6.4.1, Figure 37).

Post translation is also required for the correct deletion of Stop Words (SW), a task incorporated into the next phase: the feature extraction. SW are high frequency text items and they must be removed from text because they do not help explaining the latent topic of a text and their high frequency decreases the accuracy of word frequency analysis as well as increase the computational time (Campbell et al., 2014; Muntean et al., 2012; Bird et al., 2009). Since NLP tasks can be performed on one language at time, the translation uniforms all SW of different languages into English to be able to enable full recognition and deletion through NLP. The language trends are displayed in Paragraph 6.4.1, figure 37.

After the translation is completed and posts are normalized and uniformly expressed in English, the successive post analysis phase can begin. This phase helps to identify patterns that are hidden in the corpora such as the absolute frequency of words and hashtags which helped recognise and verify the presence of spam services, at previous stages (See Paragraph 5.4). FDA of languages provides a classification of the most common languages found in the corpora and in some cases it gives an insight on the Country of origin of users (e.g. that is the case of Dutch, German, Italian, Polish, or Russian as these languages are above all spoken within the Country of origin). In the next Paragraph, the process of extracting features from text is argued. This phase serves to extrapolate a particular set of words that are needed to identify a topic. In this case, the features are words related to touristic activities which in turn return touristic topics.

5.5.1.2. Features extraction

Feature extraction (Table 8) is the process of mining solely those features of interest (i.e. touristic information in posts) out of a corpus. Prior to start with the process, posts are split into words, by means of *tokenization* in order to perform *Stemming* as well as remove SW. While tokenization reduces a post into a bunch of its words, stemming is the action of converting a word to its root (e.g. the stem word of *fishing* and *fisher* is *fish*) improving the

accuracy of word frequency by considering a unique word for all similar words (e.g. fishing and fisher are considered as separate entities even though the relate to a similar meaning).

Feature extraction	
Text to Features to Vectors	
-Tokenization, Remove stop words, Stemming	
-Selecting tourism-related word features (user defined)[51]	
Text classification: text2dictionary	**Topic modelling: text2vectors**
-Bag-of-words *{[Feats], label}*	-TF-IDF *{Words/Tweets Matrix}*

Table 8: Feature extraction phase. Bear in mind that green colour refers to supervised learning, whereas the blue refers to unsupervised learning as depicted in Paragraph 5.5, Figure 16

The ***Bag of Words*** (BoW) model is the input of both supervised and unsupervised learning. However, its format changes with regards to the type of learning considered.

Supervised Learning: Text Classification

In text classification, the extraction of features is carried out through the use of a dictionary, which is created in Python language. The features (i.e. the words included in posts) are selected from posts provided that they also exist in the dictionary, and they are labelled following a user-defined schema (See Paragraph 5.5.1.3). The set of labelled features is then passed into the BoW model. The model considers posts as a group of words disregarding their order but taking into consideration only the word and its presence in the corpus.

In this approach, touristic features are extracted from the dataset through the use of the hashtag indexing system of Twitter. Each hashtag is linked to a certain number of posts expressing one or more topics, for example hashtags such as "#holidays", "#visit", and "#trip" relate to posts associated to tourism activities of different type, and their words can be used as the variables to be extracted (i.e. the features). This method is supported by assumptions 1, 2 and 5 established in 5.1 regarding the attitude of users towards posting on Twitter. Top hashtags are examined via FDA to serve as the "extractor" of touristic as well as non-touristic features to build the ***feature set*** (See Paragraph 6.4.1, Figure 41). Prior to the extraction of hashtags, for each of them, a sample of posts is collected and analysed so as to study the variety of topics expressed. Next, two lists of hashtags, one for each binary class (hit/miss), are created. Hashtags, whose posts depict touristic information, are appended into the ***touristic hashtags list***, whereas hashtags expressing topic of different nature than tourism are appended into the ***non-touristic hashtags list*** (See Paragraph 6.4.2, Figure 42). This method help the analysis to narrow down the number of topics in the dataset, by selecting only a portion of them through hashtag indexing.

Unsupervised Learning: Topic Modelling

[51] Tourism-related features are selected through observatory analysis of the relations between the features and the context within which they are found.

In contrast to the procedure used to extract features in text classification, the features extraction in topic modelling is executed on touristic posts, exclusively. Indeed, the goal of this approach is to find the hidden topic behind tourism related posts, so the text classification process has to end in order for the topic modelling to start. The process of extraction is divided into four sequential steps: First the number of topics in the corpus needs to be found using the elbow method (1). Then, the corpus made by touristic posts is ***"vectorised"*** into the TF-IDF corpus by means of the TF-IDF model (2). Thereafter, the corpus dimensionality is reduced via LSI model transforming the features into an (X, Y) array of vectors (3) (i.e. the scores assigned to each word feature in the corpus). To end, these vectors are displayed into a 2D Cartesian space via K-Means algorithm (4). The steps abovementioned are argued in detail in Paragraph 5.5.1.5, under the *K-Means clustering model* section.

5.5.1.3. Features labelling

The *features set* is a document including features extracted by post attributes connected to touristic and non-touristic hashtag lists mentioned above. After this set is labelled in accordance to the intended classification (i.e. label *hit* for touristic posts, *miss* otherwise), the labelled feature set is created and this includes:

- Posts, the extracted features associated with tourism;
- Labels, manually assigned classification tags that say to what group a post belongs.

The type of label is connected to what kind of text classification is needed. In this approach, a ***binary classification*** (Hit/Miss) is set up. Actually, the binary classification has been successfully used to rapidly detect spams in email accounts, by using a bunch of well-known spams email as the dictionary of features to "train" a text classifier[52]. The procedure followed in this work starts with the manual assignment of the correspondent label to posts in the feature set for which the topic is well known a priori. Thus, for touristic posts a *hit* label is chosen and a *miss* label, otherwise. Next, the classified features set is divided into two sets, namely the *training set* and the *test set*. Those are used to train the Naïve Bayes Classifier (NBC, see Paragraph 5.5.1.4) on what features are of interest and what features are not (See Paragraph 4.2.5). To be able to obtain high accuracy in the classification, the length of both sets is chosen according to a fixed ratio for which the training set is 75 percent of the length of the features set, while the test set, which is normally used to check the accuracy, is 25 percent as the whole of the features set.

In the following paragraphs, the machine learning approach is argued. The machine learning is composed of two separate tasks: the text classification and the topic modelling. As explained in the Theoretical framework paragraphs 4.2.5, 4.2.6, 4.2.7, text classification is a supervised machine learning to classify text in pre-defined classes (labels) with regard to a priori knowledge over features (i.e. training set of known features manually labelled with desired classes). In contrast, in topic modelling, the knowledge over the features is not sufficient to classify the features into labels or the number of labels in the features dataset is

[52] An example of "ham/spam" binary classification approach performed onto a dataset of emails can be found at: http://nbviewer.ipython.org/github/carljv/Will_it_Python/blob/master/MLFH/CH3/ch3_nltk.ipynb.

not known a priori. Hence, a clustering technique is adopted so that the features are grouped according to some similarities among them.

5.5.1.4. Text classification

Once the training set is created and its features (i.e. posts) are labelled accordingly (*tourism=HIT* and *non-tourism=MISS*), this is fetched into a classifier which learns this structure. After the classifier has learnt the classification schema manually established, the accuracy of the classification is assesses over the test set. If the result of accuracy is satisfactory (normally a good accuracy score is above 80-85 percent), then a new set of unlabelled posts, namely *unlabelled set,* is fetched into the classifier to be classified. There exist many classifier and the choice of which classifier to use is linked to the type of analysis to be performed. In this study, Naïve Bayes Classifier (NBC) is chosen because of its ease of use and fast to train with just a single scan. Moreover, it is not sensitive to irrelevant features such as all those tweets in the test set that do not refer to tourism (Bishop, 2006). The chosen classifier uses *Bayes Theorem* to predict the likelihood that a post belongs to a certain label. The Bayes Theorem is expressed as:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

Equation 1: Bayes Theorem. Source: Wikipedia[c], 2015 @ http://en.wikipedia.org/wiki/Bayes%27_theorem (Accessed on 08/02/2015)

Where:

- A is the label and B are the tweets,
- P(A) and P(B) are the probabilities of A and B, and
- P (A|B) is the probability of A (labels) if B (tweets) is True (1).

Once, the NBC outputs the classified dataset in which touristic information are separate from non-touristic information I can proceed to the next stage of unsupervised learning: Topic modelling.

5.5.1.5. Topic modelling

The final step of the machine learning approach is the topic modelling. The goal of this approach is to detect the latent topics behind each touristic related tweets in the corpus and perform a group analysis of these topics to be able to detect patterns in the corpus of posts. This approach made use of the Gensim library of Python (See Paragraph 4.4) which is designed to automatically extract topic semantic from documents through a relatively fast and user-friendly algorithm implementation. Gensim implements many learning algorithms such as dictionary of documents (i.e. list of unique words in posts), Term Frequency – Inverse Document Frequency (TF-IDF), and Latent Dirichlet Algorithm (LDA) algorithm approaches via Python seamlessly and it is used in this work to model the statistical distribution of topics. In topic modelling, the dictionary of unique words is converted into a vector space by using the function ***doc2bow***. The function counts the number of occurrences of each distinct word, converts the word to its integer word identified through an ID and returns the result as a

sparse vector. The TF-IDF model of Gensim is then used to represent each term in the vector space with regard to how important is a word included in a post within the Twitter corpus.

In the TAUS GKD methodology, the topic modelling approach is made up by two distinctive unsupervised learning techniques. The first approach transforms textual information into vectors and it clusters them according to vector similarities. The second approach assigns scores to word features in text and it groups data into topics by its most weighted word features. The techniques are respectively:

- K-Means clustering model (Vector modelling), and
- Latent Dirichlet Allocation (LDA) model (Topic/terms modelling).

K-Means clustering model: identifying the number of clusters (Elbow method)

To enable the K-Means clustering algorithm, scores are assigned to words in the corpus with regard to assumption 4 and 5. Higher weights can in fact be assigned to words whose occurrence in tweets is higher, hence very relevant. The Frequency-Inverse Document Frequency (TF-IDF) is an approach to assign scores to tweets. It represents tweets as vector of identifiers such as relevancy in the case of the words in tweets. Each vector corresponds to a term in a tweet and if a term is found, a non-zero weight is assigned to that vector.

To establish the importance of a term in a post, a weight is assigned to it by means of the following formula implemented through the combination of Gensim and Scikit-learn Python libraries:

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \times \mathrm{idf}(t, D)$$

Equation 2: Term Frequency - Inverse Document Frequency formula. Source: Wikipedia 2015 @ http://en.wikipedia.org/wiki/Tf%E2%80%93idf (accessed on 08/02/2015)

As stated in the explanation of the formula in Wikipedia[e] (2015):

> *"A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights*[53] *hence tend to filter out common terms. Since the ratio inside the idf's log function is always greater than or equal to 1, the value of idf (and tf-idf) is greater than or equal to 0. As a term appears in more documents, the ratio inside the logarithm approaches 1, bringing the idf and tf-idf closer to 0."*

Scores, in terms of vectors, can be clustered together through a clustering technique which groups them according to similarities that each score (word feature) share with others in proximity. The algorithm to be used is the K-Means clustering algorithm provided by the Scikit-learn Python module, and the similarity that scores share among each other is called "*Inertia*" of the cluster, which is provided by the K-Means algorithm itself, and it represents the the Sum of Squared Errors[54] (SSE) between each score and the centre of the cluster that

[53] In this work, I refer to weights as the ***scores*** that the TF-IDF and LDA models returns over word features.
[54] This work refers to the errors of the Sum of Squared Errors (SSE) as the ***distance*** between the score and the centre of its cluster.

the score belongs to. The SSE is calculated over a defined range of K-values (e.g. 1 to 10) using the following formula:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

Equation 3: Sum of the squared distance (SSE) between each member of the cluster and its centroid. Source: Polytechnic of Milan @ http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html (accessed on 19/04/2015)

Where $\left\| x_i^{(j)} - c_j \right\|^2$ is the chosen distance between the score *x* and the centre of the cluster C_i.

However, prior to execute the K-Means clustering algorithm over the scores of word features, two separate tasks need to be done:

- Find the most probable number of hidden topics in the corpus,
- Reduce the vector space to two a 2-D space to be able to plot the vectors.

The first is the number represents the threshold number of K-topics in corpus that includes the majority of posts within it. A way to find the "best" value of K, hence overcoming the issues described in the related works Chapter, is through the ***Elbow method***[55] (Tibshirani, 2001). By plotting the value of K against the Sum of Squared Distance, the error decreases as K gets larger. In fact if K increases, the size of clusters decreases, as for distortion (i.e. the error). Therefore the ***Elbow method*** returns the most probable K value as the SSE decreases abruptly producing an "elbow effect" that can be seen when results are plotted in a graph (See Paragraph 6.4.4, Figure 47).

The second task makes use of the Latent Semantic Indexing (LSI) model of Gensim a decomposition tool to reduce high dimensional corpora space to a 2-D (Cartesian) space in which the x and y coordinates (i.e. the scores of each word feature in posts and its label, respectively) can be plotted (Wikipedia[d], 2015). The corpus dimension is represented by the number of unique terms, found by the LDA algorithm, within the corpus itself. The decomposition is necessary if one wants to make use of a clustering technique over large corpora. By using Python language and Matplotlib Python module (See Paragraph 4.4) for graphical representation, the scores reduced to a 2D plane are plotted by means of a scatterplot and the clustering result is shown in Figure 54, Paragraph 6.4.4.

In summary, the Elbow of the K-Means is a crucial input when the number of topics (i.e. an important parameter of the LDA model) has to be defined to be able to implement the Landmark Aggregation Semantic (LAS)[56] (i.e. what tourists express through posting over

[55] A good explanation of the Elbow method is found in a web tutorial for txt mining and it is cited as follow: """when we increase the value of K, the value of "within-cluster-sum-of-squares" will drop as we have more clusters hence smaller distances to centroids. But each successive increase in K will not give the same drop. At some point the improvement will start to level off. We call that value of K the elbow and use that as the "good" value of K."""" (GitHub, 2015)

[56] LAS method: a combination of supervised and unsupervised machine learning for behaviour discovery.

Twitter). Furthermore, in this context the K-Means algorithm is introduced as a validation tool to confirm whether the vector scores that the LDA algorithm assigns to the terms of the topics obtained (i.e. the LAS result) are similar, hence clustered in space.

The LDA model: identifying the topics for Landmark Attractiveness Semantic (LAS)

In this section, the Latent Dirichlet Allocation (LDA) model implementation of Gensim is applied over the corpus of tweets in order to find the latent meaning behind posts, hence discover what are the topics discussed by tourists visiting Amsterdam. The inputs of the LDA model are mainly three: a Dictionary of unique words, a serialized corpus containing the word identifiers followed by the value which depicts its presence in the dictionary (*'1.0'* which is very similar to the Bag-of-Words representation) and the number of topics to be found.

The dictionary is the baseline *features set* used by the LDA model to establish what the words of interest are so to retrieve touristic information from a larger number of words included in the corpus. Therefore, for the sake of accuracy, words such as stop words and extremes words need to be removed. With the term extreme words, the author refers to those words whose frequency in the corpus is either extremely high (e.g. Amsterdam, Holland and Netherlands) or low (e.g. typo mistakes, wrong translation, misspelled words and so on). It is important to find the right ratio that filters out the noise[57] from the dictionary so as to obtain higher accuracy of results. The LDA model takes a serialized[58] corpus which is represented by two files: one describing the documents, and another describing the mapping between words and their ids (Gensim, 2015). The corpus consists of all features that are included in both corpus and dictionary, which are transformed, post after post, into the Bow model. The third input is the number of topics to be found. This parameter can be found through the use of the ***Elbow method*** described in the previous paragraph (Tibshirani, 2001).

In short, the LDA model outputs a set of most popular ***terms*** grouped by a pre-defined number of ***topics***. Terms are chosen according to scores assigned by the LDA algorithm. The approach established in this work assigns a label to each post according to which topic the post is in, thus clustering them together if posts have similar scores (See Paragraph 6.4.4, Figure 54). When the corpus is reduced to a 2D Cartesian plane and the most probable number of clusters is obtained, the methods continues by assigning each posts in corpus, whose score falls above the mean value of all scores, to the rightful cluster with regard to their scores. By doing so, the end product of this method is the Twitter dataset in which two significant labels are introduced:

1. A ***classification label***, showing what is touristic information and what is not (Text Classification);
2. A ***cluster label***, identifying the nature of the cluster(s) hence the semantic of the post linked to that cluster.

The final output includes all those tweets classified via the NBC as being tourism related which enables the implementation of the Landmark Attractiveness Estimation of

[57] Noise regards those words whose frequency is extremely high or low such as stop words and typo mistakes, respectively.

[58] The serialized corpus is created in Gensim through the Blei.serialize module and it has the following data model: {ID: id, Value: value}. Blei.serialize @ https://radimrehurek.com/gensim/corpora/bleicorpus.html

touristic PoI. In addition to that, touristic posts are enriched with the LDA ***cluster label*** which is the essential attribute for the assessment of the LAS. The cluster label enables to calculate, in percentage, the proportions at which each label appears within the service areas of each landmarks, hence compute the landmark semantic. For instance, when a service area of landmarks is selected, the count of each unique labelled tweets that is contained in it returns the distribution of cluster labels. The most popular label returns the most popular topic being discussed at that landmark.

Therefore, within the service area of each landmark a number from 1 to *n* clusters is obtained according to a certain proportion which is related to the number of occurrence for each unique label inside that service area. Bear in mind that the LDA algorithm is a *generative approach* that introduces randomness in the computation. Therefore, despite a certain degree of similarity in every run, a post can be assigned to one or more clusters differently at each model run. Actually, the purpose of the LAS is to offer a general indication of the most prominent topic for each landmark and not the exact topic discussed. If higher accuracy of results needs to be achieved, multiple runs should be performed in order to get an average of the results which would be closer to reality. However, this is out of the scope of this research, but it is included in the Discussion Chapter.

Finally, after that the label(s) are assigned to the clusters, they are included in the output of the TAUS GKD methodology. By doing so, I enabled the variables for the next phase, allowing the investigation of aggregation patterns with regards to the time, the extension (LAE) and the semantic assessment (LAS) of touristic landmarks in Amsterdam City. In addition to that, the service areas in which the measure of the attractiveness I_i is above the average value are selected to be the input of the LAS. Those landmarks denoting attractiveness above the mean value are implemented in the next step for the identification of issues in the measure of accessibility of touristic landmarks.

5.5.1.6. Landmark Attractiveness Semantic

The Landmark Attractiveness Semantic (LAS) follows the completion of the LDA and space partitioning processes and it is calculated over the densest service areas obtained in the end of the LAE analysis. In those areas, the topic labels obtained are measured in relation to the percentage of occurrence of unique topic labels obtained in the end of the TAUS GKD method (See Paragraph 5.5.1.5). The resulting LAS is then compared to the location of the PoI considered to check if relations between the context and the LAS occur. The output is a pie chart for each of the service areas of study. The result of this stage is a label or a set of labels, manually assigned to each topic of the LDA, through which it is possible to generally identify a popular topic discussed inside the service area of a landmark. This may help in the identification of the tourism behaviour and I could reveal interesting patterns such as a particular event or fact that could have appended in the time period considered. Bear in mind that the output of the LAS and the LDA are both in terms of a text label. The difference is that the LDA outputs top terms related to a topic, whereas the LAS method considers those top terms in order to manually assess the term semantic for each of the topics. Therefore I produce a single general label that identifies the nature of each topic. This label is assigned to one or many touristic posts by comparing their LDA scores with the threshold LDA score: the average score value (i.e. sum of scores by number of scores).

Next, with touristic information as well as space and time partitioning in place, I can begin with the evaluation of the accessibility of touristic venues of Amsterdam City in space and time.

5.6. Accessibility of touristic venues in space and time (5)

The measure of the accessibility is calculated through the use of the gravity model discussed in Paragraph 3.4. This approach takes into consideration the attractiveness of places, represented by the number of tweets within a certain service area, and the network impedance expressed as the cost of reaching a PoI from a network destination (i.e. closest transit stops in proximity of the PoI), which in this case is represented by the distance that a tourist walks from the transit stop to the PoI. Space and time divisions are explained in more detail below. I consider the distance from the transit stop to the landmark given that the location of landmarks changes because of the factors involved such as time of the day, day of the week. Whereas, stop locations are rather stable and they represents a reference from which to travel.

5.6.1. Landmarks Attractiveness Estimation (LAE)

Once time periods are established and service areas created, the computation of the PoI attractiveness can begin. To enable the calculation of the density, I count the number of tweets in each polygon by means of the Spatial Join tool, selecting tweets by the location they are posted (i.e. within the service area of each landmark). The output of the join is a new feature set including a count field that serves to assess the PoI attractiveness as the ratio of tweets count (nominator) by the extension of the service area (denominator) in which they are located. The analysis of the LAE is subdivided over the two temporal periods defined in the time decomposition and they are overlaid one to another to evaluated spatio temporal changes in the LAE.

The LAE method is implemented in ArcMap environment over the datasets which are obtained by the space and time decompositions as argued in Paragraph 5.3. The different time periods serve to analysis the changes in accessibility of touristic landmarks in the next phase. The approach argued in this paragraph is implemented in ArcMap by means of a model constructed in model builder environment (Figures 17 and 18).

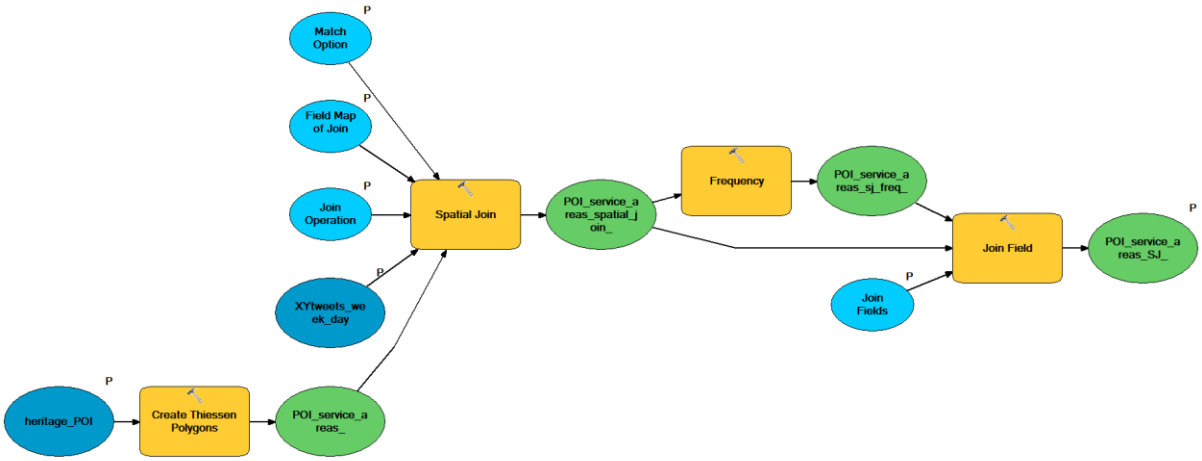


Figure 17: ArcGIS model to implement the space partitioning approach previously discussed. The model also performs a Spatial Join in order to count the frequency of tweets in each of the service areas created for each touristic landmark being considered

Thus, to be able to assess the extent of the LAE, the service area of each touristic landmark is calculated by means of Thiessen polygon (Voronoi diagram) implemented in ArcMap. Through the Thiessen polygon space partition the urban space is distributed among the landmarks. Therefore, high densities of tweets which occur mainly in the city centre are overlaid onto Thiessen polygons which in contrast are smaller in the city centre and scattered as the distance from the centre increases. This leads to a true representation of the reality, in which high density patterns occur in denser urban textures.

To end, a classification of congestion spots is produced and the densest areas are spatially and temporally located. The outcome of this process is a density map in which two datasets at different time periods illustrates where and when the measure of the LAE is higher, hence where and at which period of time the concentration of tourists posting Twitter contents is higher. This returns an indication of the most popular areas in which congestion and overcrowding should be carefully analysed.

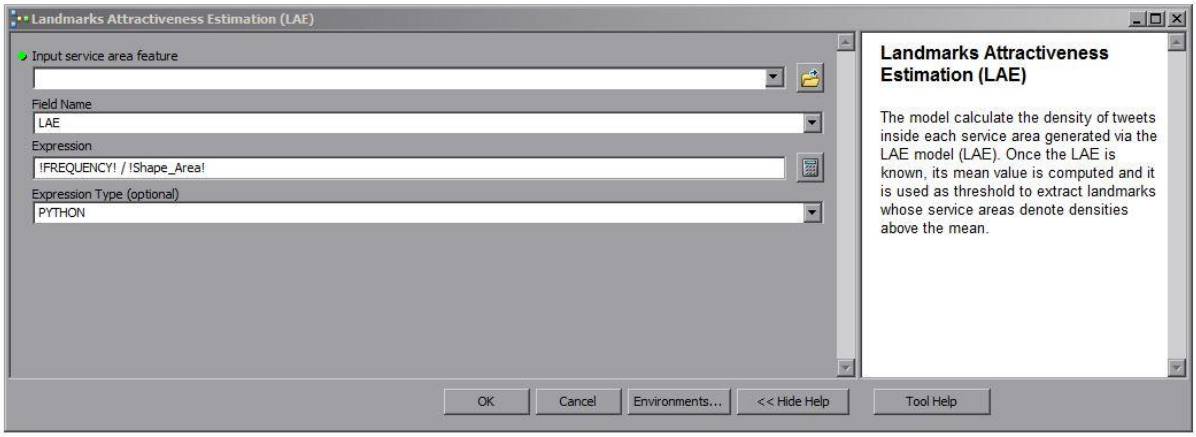


Figure 18: The ArcGIS model I create in order to implement the Landmark Attractiveness Estimation. The frequency of Twitter contents are divided by the measure of the service area obtained in the space partitioning approach.

In the next phase the outcome of the LAE is combined with the network impedance. The impedance is represented by the distance from closest transit stops in proximity of

landmarks and it is displayed by a multi-buffer raster layer computed by means of Euclidean Distance tool in ArcMap. The tool shows different walkable (Euclidean) distances over the urban texture of Amsterdam city as argued in assumption 6.

5.6.2. Network Impedance (NI)

The computation of the network distance takes into consideration two inputs: the landmarks whose LAE is above the average and the Amsterdam transit network obtained from the OPENOV online repository. Specifically, I use the transit stops from which to compute the ***distance d_{ij},*** or the *network impedance*. Then I compute the travel cost surface in raster format by means of the ***Euclidean Distance*** tool found in the ArcMap Spatial Analyst toolbox (Figure 19). The tool create a raster surface whose cells hold the value of the distance from the cell to the nearest PoI. I then use the Reclassify tool in relation to assumption 6 to classify the raster values so that I have a scale from *"easy to travel"* (short distance: class 1) to *"difficult to travel"* (long distance: class 5).

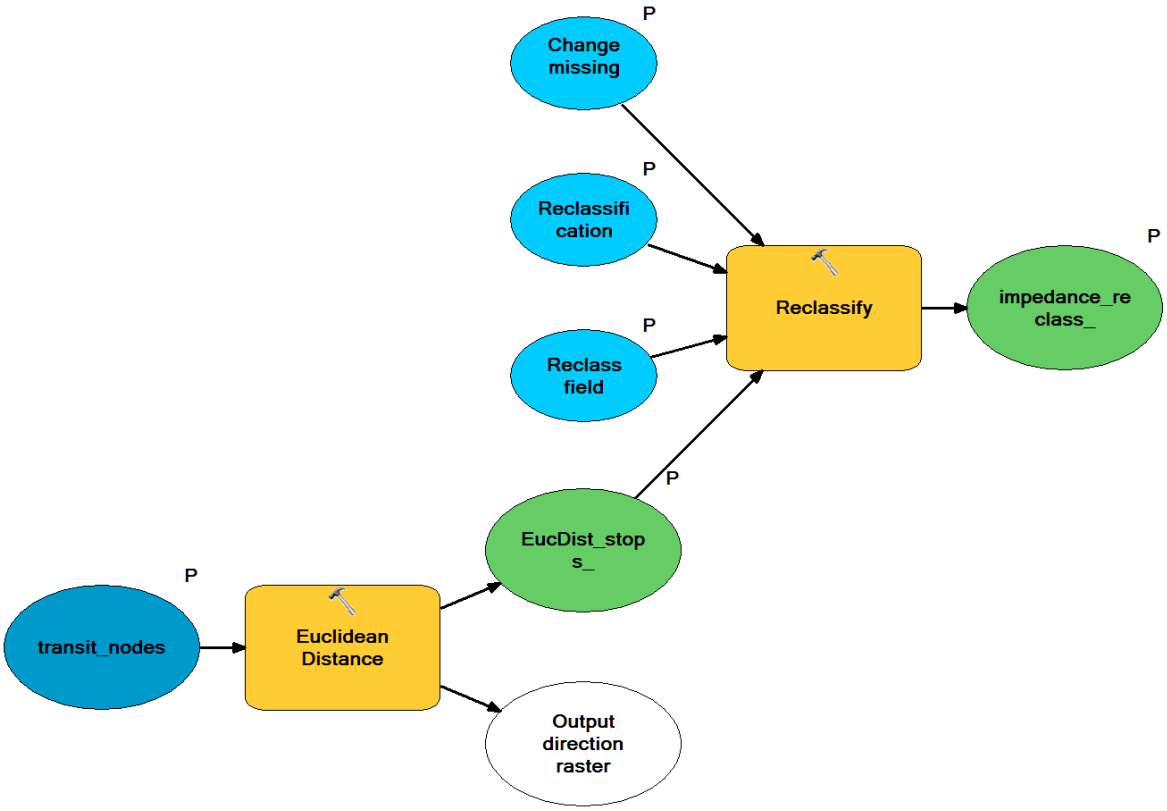


Figure 19: Network impedance model of ArcGIS. The model outputs a raster layer showing a "reclassified" multi-buffer feature set showing a 1 to 5 class range. 1 is assigned to a highly accessible locations (i.e. within 100 metres from a transit stop), 5 is assigned to the poorest (i.e. 500+ metres away).

Next, the landmarks that depict densities above the average are extracted and overlaid onto the classified buffer zones created by the Euclidean Distance and Reclassify tools from which I can obtain the value of the raster (from 1 to 5) and assign it to the landmark in accordance to its location. For example, if a landmark is in a zone 4 or 5 that could indicate an area with poor accessibility. Furthermore, given that I selected only those landmarks with densities above the average, the large number of tourists could generated phenomena of overcrowding and/or congestion if the landmark is close to poorly accessible areas.

For the sake of clearness, the transit stops I have implemented in the computation of the network distance refer only to “aboveground” transit systems such as busses and trams. The reason behind this choice is explained by the structure of the Amsterdam metro system. Actually, besides metro stop in Central Station (which is already indirectly considered as tram stops are located next to the metro entrance) there is only one metro stop which is located in Nieuwmarkt area. However, its proximity to Central Station and the costly fare discourage tourists to use the metro for such a short trip.

5.6.3. Accessibility of touristic venues

The final outcome of this work is the evaluation of the accessibility of places in urban space and time. The process described above is repeated for both datasets at different time period: one considering the week days during day time (6h-18h), and the other is related to weekend days during night time (18h-6). This subdivision allows the evaluation of the effect of the temporal component over the occurrence of tweets. The accessibility is analysed over an accessibility index obtained via the Reclassify tool of ArcMap. The process starts with the creation of distance bands from transit stops to touristic landmarks and it is obtained through the Euclidean distance tool in ArcMap. Then, the output is reclassified via the Reclassify tool in ArcMap which create an accessibility index from 1 to 5. 1 is used for highly accessible touristic venues which are within 100 metres to a transit stop and 5 is assigned to all areas farther away (500 metres as it is established in assumption 6).

The map being created displays the urban areas in Amsterdam where the measure of the accessibility is compared with the locations of most attractive touristic landmarks by means of spatial overlay. Actually those are the landmarks that show high frequency of Twitter activities coming from tourists, hence issues of overcrowding and congestion are likely to occur there due to the large amount of people. This approach is performed on both time period being considered so as to enhance the understanding of the use of the urban context as well as the degree of accessibility of touristic venues at different times.

In this context, to support as well as validate the choice of the transit stops, a log containing coordinates of stop locations is obtained from the routing application of the company HERE maps. This is a list of requests sent by customers of the HERE Transit routing engine application that have used the routing app to retrieve transit indication prior to a trip, or displacement in Amsterdam City at the same period of Twitter contents extraction.

5.6.4. HERE Transit: a validation approach

The HERE routing engine log provides information regarding the locations of the departure/arrival stops being used by transit users. Hence, most transited stop locations give an insight of the nodes where the risk of congestion is higher. The purpose of this information is to support and validate the result of the Landmark Attractiveness Estimation, the main variable deployed in the computation of the accessibility of touristic venues.

I decided to consider the use of external data from HERE firm because of the degree of internationality shown by users of the HERE transit routing engine. In fact, by using HERE transit information I am able to reach a very diverse customers from all over the world. This

could not be the case if I am to use data from the Dutch transit service 9292.ov[59]. In fact, the 9292 transit app is mostly known and used by locals, therefore despite the larger amount of data, I could not verify whether the information was either touristic or not. In contrast, this is the case for the data obtained by HERE given that the application is used worldwide because of its extensive coverage which spreads over countries such as North and South America, UK, Italy, France, Spain and so on.

The dataset received by the company is a CSV format log and it comes with a number of attributes that are not of interest to this research and therefore are removed. Specifically, I retain solely the geo location attribute of the stops of each trip, which are requested during the month of December and the timestamp, from which to extract the two selected time periods used in the computation of the accessibility (day time on week days and night time of weekends). The stops extracted from the HERE log are added to the ArcGIS application and they are overlaid onto the network stop coordinates obtained from OPENOV. In particular, the stop information of HERE do not refer to a particular stop location but it is rather a user requested information performed in the Transit application of HERE, therefore they do no coincide with the actual location of network stops. To overcome this issue, I create buffers around the network stops of OPENOV and I count all stops falling within 150 metres ray buffer. The count of destination stops obtained represents the frequency at which users of the transit app have requested that particular destination in the month of December. The values obtained following the approach described above gives an insight of the most transited stops during the month of December. In the end, this information serves to validate whether the network stops in proximity to most attractive landmarks, obtained during the LAE, have had indeed a large number of requests, hence tourists have used them to probably reach one or many touristic landmarks in the surroundings.

6. Results

In this chapter findings are shown, described and compared with the list of research assumptions argued in Chapter 5. Results are shown following the methodology established in Chapter 5 Figure 6. Python Matplotlib and Pandas modules support the visualization of the outcomes of the LAS approach. Whereas, ArcMap platform supports the computation and visualization of Spatial Overlapping Features (SOF, Paragraph 5.4.1), as well as the LAE which returns insights over accessibility issues in space and time. Figures, tables and graphs are reported as the process is explained so to keep track of the workflow. The results are discussed as follows:

1. Data collection,
2. Space and time partitioning,
3. Data analysis,
4. TAUS GKD: LAS identification and assessment, and
5. Accessibility of touristic venues.

Following, a discussion over the conclusions as well as the limitations, with regards to application used and methodology adopted, are argued in the last Chapter of this paper.

[59] 9292.ov official page: http://9292.nl/over-9292

6.1. Data Collection (1)

The data collection process uses the Streaming API of Twitter in order to collect tweets according to the specified location set in the API filter (Figure 20). Tweets are streamed into a Python array, which is a 4-dimentional list including:

- USER_ID (Long integer),
- X,Y coordinates (decimal degrees),
- TIMESTAMP (YYYY-MM-DD T HH:MM:SS format), and
- POST (text attribute)

```
# Create an instance of the MyStreamer class
stream = MyStreamer(APP_KEY,APP_SECRET,OAUTH_TOKEN,OAUTH_TOKEN_SECRET)

# Tell the instance of the MyStreamer class what you want to track
# The location in terms of bounding box extremes is defined
stream.statuses.filter(locations=[4.730935,52.295111,5.014335,52.428306])
```

Figure 20: The bounding box extremes location filter. (Python recipe)

The 4-D array is appended to a CSV file by means of Python and the end result is displayed as follow in Figure 21:

```
542079164086751232,4.929627,52.364663,2014-12-08T22:11:57.309000,You should come to amsterdam
```

Figure 21: An extracted tweet from the CSV file. Note that attributes are separate by comma.

After the CSV file is transformed into a shapefile via PostgreSQL, the transformed data is visualized into ArcMap onto a baseline map layer included in the ArcMap Online service. The point data is exported into a geodatabase and geographically located on earth by means of the WGS 1984 - Web Mercator Auxiliary Sphere projection (WKID: 3857 Authority: EPSG) that is the baseline map projection. In addition to the point data cloud obtained, the layer is enriched with external layers such as the Amsterdam transportation network from OPENOV website and touristic venues layer obtained from the OpenStreetMap (OSM) Metro Extracts API. The outputs showing the obtained datasets (tweets, transit, and venues) are shown in Appendices C, D and E.

The collected datasets are first subdivided time and space wise in the next Paragraph to enable LAS analysis and the effect of the influence of time. Thereafter, the Twitter dataset is analysed in the following methodology step under the two major aspects: the geo location (SOF) and the post attributes of Twitter (FDA).

6.2. Space and Time partitioning (2)

To be able to understand how landmark attractiveness and semantic implementation works, I argue the results obtained in the decomposition of the urban space and the time period of collection. First the space is subdivided into service areas, one for each landmark. Then, the time periods are established considering the time subdivision plan created in Methodology.

Space partitioning

The space is divided using the Thiessen polygons approach. As the distance between landmarks becomes smaller and smaller going inside the inner centre of Amsterdam, areas become also smaller. In contrast, the more the inner centre is considered, the higher Twitter activities are. Therefore, using this method I obtain a natural partition of the space and a more accurate representation of the landmark attractiveness. The method which uses the Thiessen polygon tool to divide the urban space is shown in Figures 22 and 23 below. The method takes the list of touristic venues as the input seeds from which to generate the service areas of Thiessen polygon. Then, the second input is the Twitter dataset, which has already been divided according to the time decomposition defined in the following section of this Paragraph.

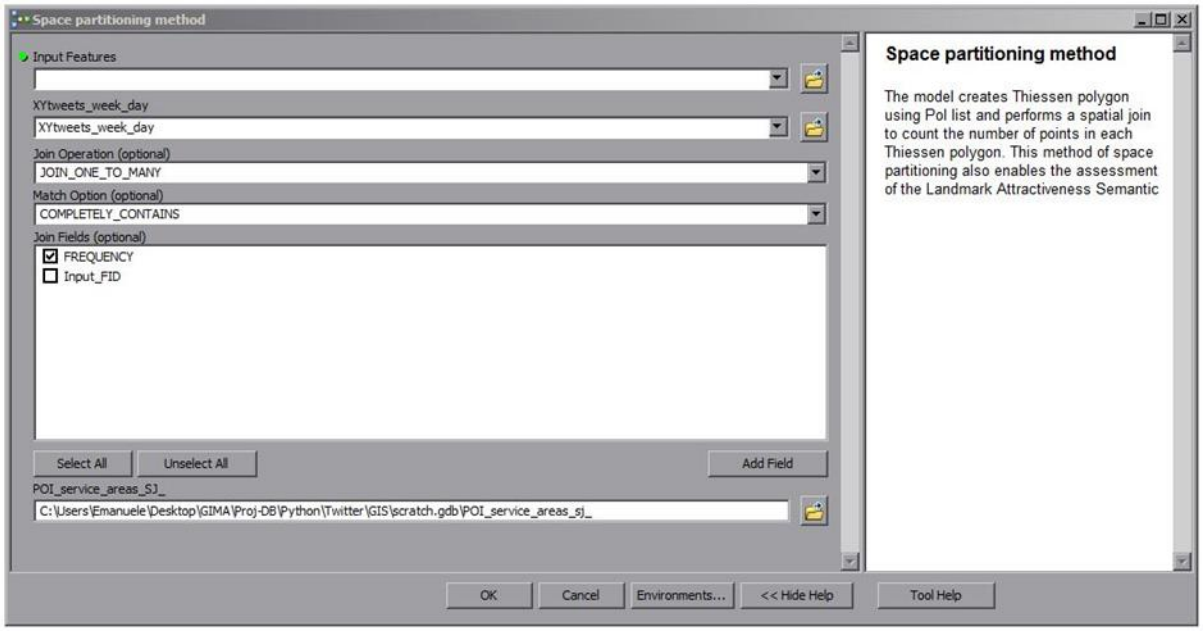


Figure 22: The ArcGIS tool interface I created to construct the Thiessen division. Note that the Spatial Join tool is included in the computation to merge tweets in each service area created and count the frequencies.

The Spatial Join is embedded into the application I created in order to compute the frequency of tweets within the service areas for each of the input landmarks. In fact, the polygons being created are not useful without the frequency of tweets occurring in it. The Spatial Join merges the tweets "*completely contained*" in each service area to enable the computation of that information. Thus, by using the frequency of tweets and the size of the service area in which they are included I can assess the Landmark Attractiveness Estimation (LAE), which is explained in the following section. In addition to that, once service areas are obtained, the assessment of the landmark semantic (LAS) can be performed for each service area. Next, I discuss the time decomposition.

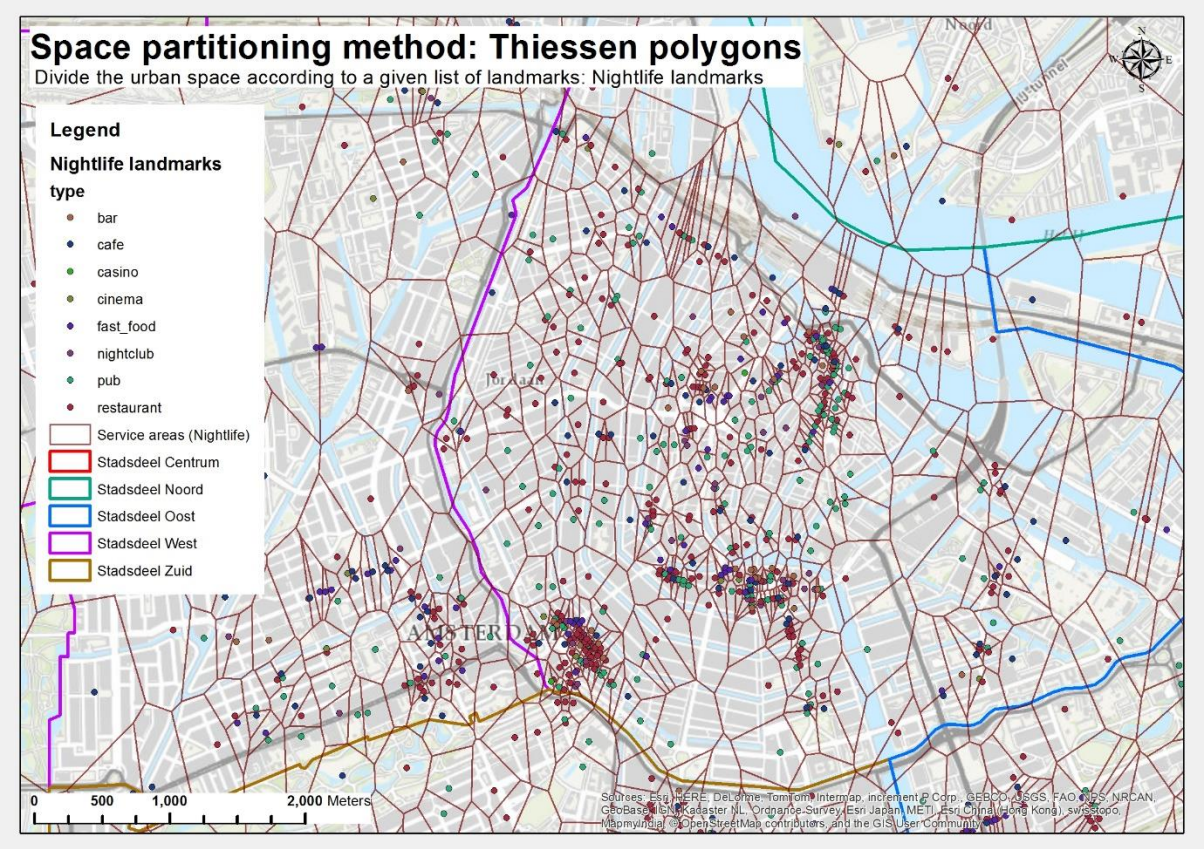


Figure 23: The division of urban space in Amsterdam using the nightlife landmarks as sample. Same process and similar results are obtained using landmarks related to Amsterdam heritage.

Time decomposition

The decomposition of the time component into the considered temporal periods is enabled in Python environment through an application created with the Pandas library. The division is performed for each time slot as mentioned in Table 5, Paragraph 5.3 (Time). Hence eight time slots (Morning Afternoon Evening Night, MAEN) and two temporal periods (week days: Monday-Friday; and weekend: Friday-Sunday) are considered. The application computes the number of tweets in each of the temporal periods and it returns the percentage of that number compared to the total number of tweets. Moreover, the average number of tweets per day and the number of tweets for each day of the week is also computed.

Not surprisingly, the majority of tweets is found during the day especially during afternoon and evening times. Actually, those are the periods during which the majority of urban activities are open. The distribution of tweets throughout the month shows interesting patterns on the 5th of December, which is a holiday celebrated exclusively in the Netherlands: *Sinterklaas*. On that day, the number of tweets obtained is lower than any other day. This means that many public activities, probably with available Wi-Fi connection, could have stay closed. The new year's evening is the highest peak observed and this could be explained by the celebration of the end of the year during which many visitors from inside and outside the Netherlands come to Amsterdam to spend there the last day(s) of the year (Figure 24).

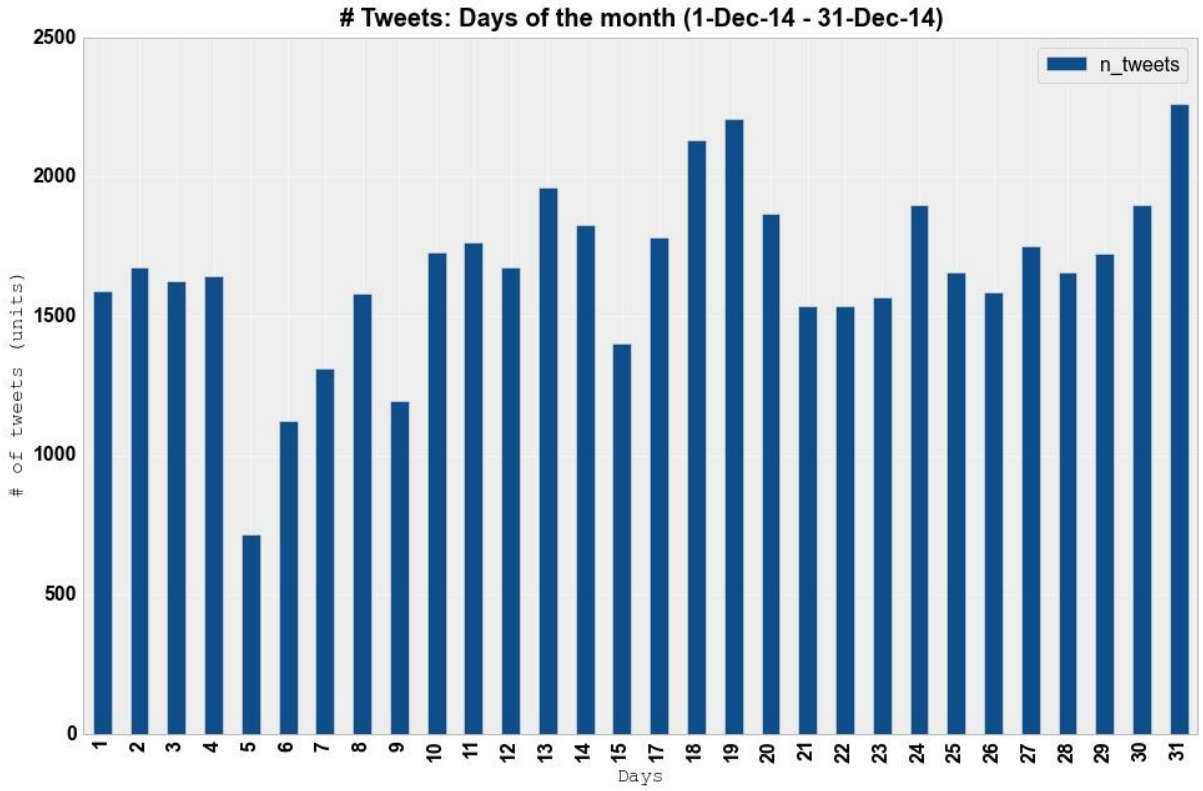


Figure 24: Distribution of Twitter activities in Amsterdam during the month of December.

The division of the month in its week days is used to support the time decomposition, and the days of the week in which the distribution of Twitter contents is high. In the figure below, Wednesdays show the highest percentage of posts. This is the result of some main holidays like New Year's and Christmas evenings both occurring on Wednesdays. Interesting drops are spotted on Tuesdays due to a problem during data collection which stopped the stream of tweets during Tuesday, the 16th. Moreover, also on Fridays the distribution of tweets is relatively low as there are only three Fridays in the month of December and one of those is the 5th: Sinterklaas.

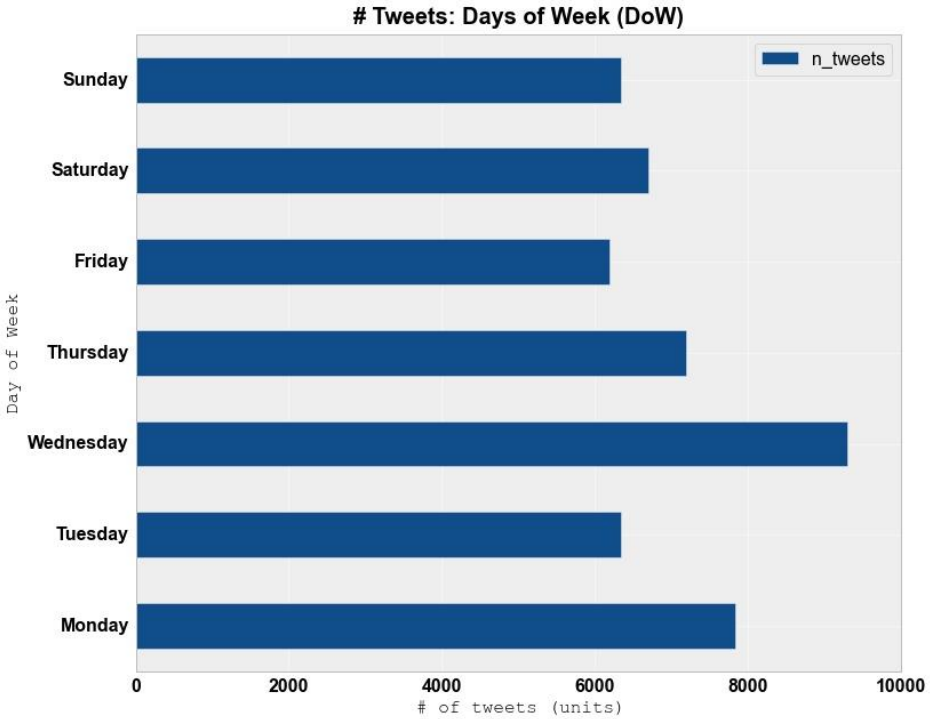


Figure 25: Distribution of Twitter activities during week days (Mondays - Sundays)

```
Create time series
# of posts during week:     73.0% (36847)
# of posts during weekend:  38.0% (19223)



Assessing the number of posts during week days at each time serie (MAEN):
# of posts during week (M):  21.0% (7775)
# of posts during week (A):  33.0% (12172)
# of posts during week (E):  38.0% (14096)
# of posts during week (N):  7.0% (2908)

# of posts during week (M+A, Mon-Fri):   54.0%  (19911)
# of posts during week (E+N, Mon-Fri):   46.0%  (16974)



Assessing the number of posts during weekend days at each time serie (MAEN):
# of posts during weekend (M):  17.0% (3303)
# of posts during weekend (A):  36.0% (7009)
# of posts during weekend (E):  37.0% (7143)
# of posts during weekend (N):  9.0% (1828)

# of posts during weekend (M+A, Fri-Sun):   53.0% (10289)
# of posts during weekend (E+N, Fri-Sun):   46.0% (8955)
```

Figure 26: Time decomposition results (% - absolute numbers)

Due to the limited time available and the difficulty of showing the influence of time for each of the time division made, I separate the time component into two different periods: the first period consider the week days from Monday to Friday during day time in which the majority of touristic activities are available. The second period includes the weekends (Friday to Sunday) during night time in which many touristic activities, but those related to nightlife, are closed. Those are two distinct time periods in which diverse ranges of activities are considered.

Once space and time are divided accordingly. I can proceed with the computation of the density with regards to the Landmark Attractiveness Estimation (LAE) as well as the Landmark Attractiveness Semantic (LAS). The former approach is enabled by the creation of service areas upon with I can estimate the Attractiveness I_i, whereas the latter is enabled by the space partitioning and the outcomes of the TAUS GKD method I have created, which are argued in the next Paragraph

6.3. Data analysis (3)

As first in the methodology process, following the data extraction process and the space and time partitioning, the analysis of collected data is discussed. The investigation process takes into consideration two main data attributes, namely the geo location and the post. The former is investigated via the SOF model, the latter through the FDA tool. Bear in mind that the FDA tool is also implemented during the TAUS GKD method, after the post attribute is cleaned.

Spatial Overlapping Frequency (SOF)

As explained in 5.4.1, the SOF model is deployed to show the distribution of spatial overlapping tweets on map occurring at different spatial locations. This computation serves to identify spams and Wi-Fi services to be able to remove the first and retain the second. The map below illustrates the distribution of SOF in Amsterdam related to the data collected during the month of December (Figure 27).

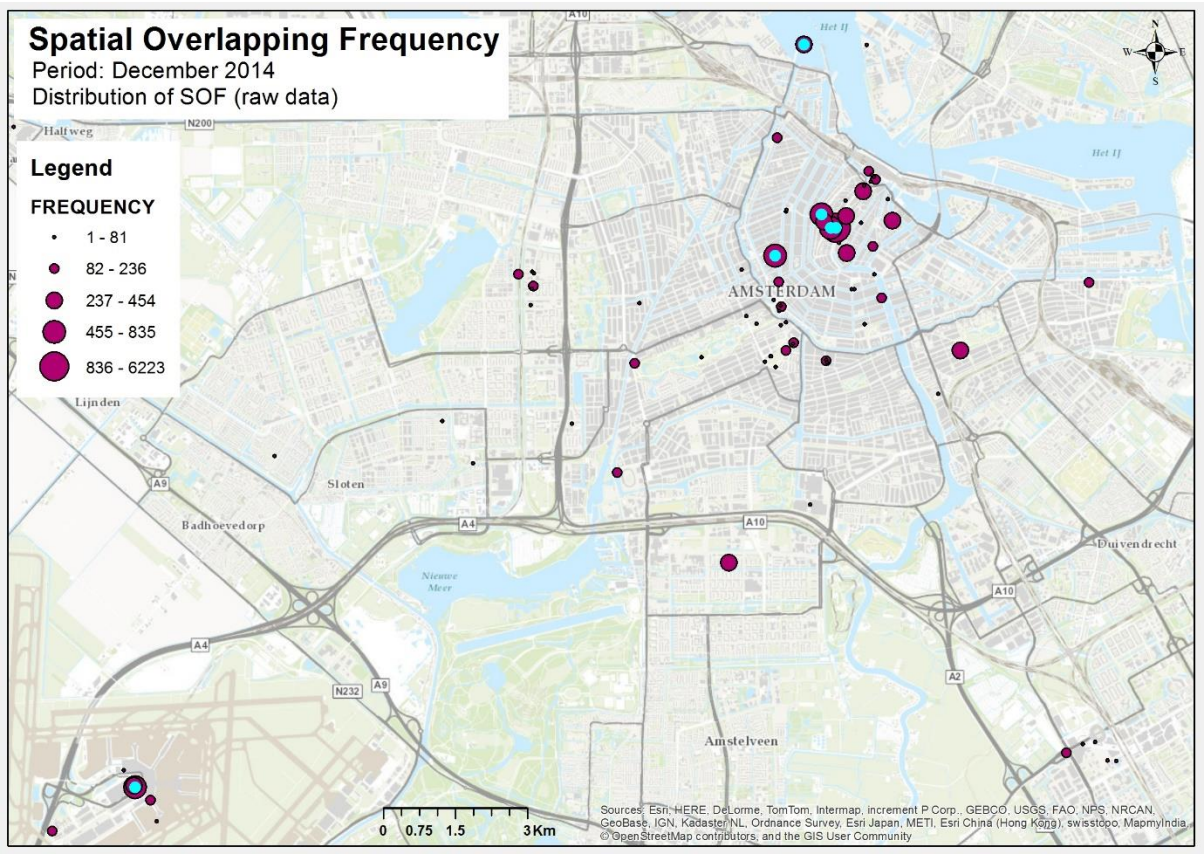


Figure 27: Spatial Overlapping Frequency of tweets in Amsterdam (Dec 2014)

The selection illustrates where the frequency is higher than the average value. Not surprisingly, peaks are located at Schiphol Airport, which is the starting and ending point of the majority of tourists visiting Amsterdam, and in the city centre where a bunch of locations are identified. However, only through the analysis of the findings of the SOF model it is not clear whether these high concentrations of tweets shown above are because the popularity of activities in proximity, which attracts many tourists over there, or if it is the result of noise generated by automatized processes: ***spam*****,** such as Twitter analytics, news or advertisements services. Thus, if it is indeed the case of spams, more interesting patterns could remain hidden to the analysis due to the noise they introduce in data. To be able to identify what generates those high frequencies, the post attribute of SOFs is analysed. Usually, automatized contents denote homogeneous text format or specific words and/or hashtags and this can be identified through the FDA tool.

Next, I investigate the post attributes at those locations highlighted in Figure 27, particularly I am interested to analyse the city centre given that the airport is solely a transit node for tourists who come to visit Amsterdam. However, an investigation of the airport area is also carried out for the sake of completeness.

Frequency Distribution Analysis - FDA

Following, the process continues with the observation of the text attributes of selected data (Table 9, Figures 28 and 29) in order to detect possible word/hashtag trends in the text attribute which can be compared to the findings of SOF. I am interested to find a link between locations with high density of tweets extracted via SOF and specific structures of text, such as high frequency of particular words or hashtags. Word/hashtag frequencies are shown in Figures 28 and 29. Both table, and graphs show interesting patterns:

Word features analysis on raw text of Twitter posts		
Operation type	Before processing	After processing
Number of collected tweets	248.823	74.983
Tweets with hashtag	66.773	##
Words count	2.105.155	398.727
Unique words	181.899	44.851
Hashtags count	103.925	23.143
Unique Hashtags	24.841	##

Table 9: Word feature analysis on raw data

First of all, there is a great deal of noise in the data and this is explained by the difference between the word count and the unique word count, which means a large number of duplicate words. This is confirmed by the graph in Figure 28. Words such as '*in*', '*de*', and '*the*' are stop words, which are features that appear in the majority of posts and need to be

removed given that their frequency can hide more important patterns. In addition, Figure 28 confirm assumption 2 showing “Amsterdam” as the most frequent word in the dataset.

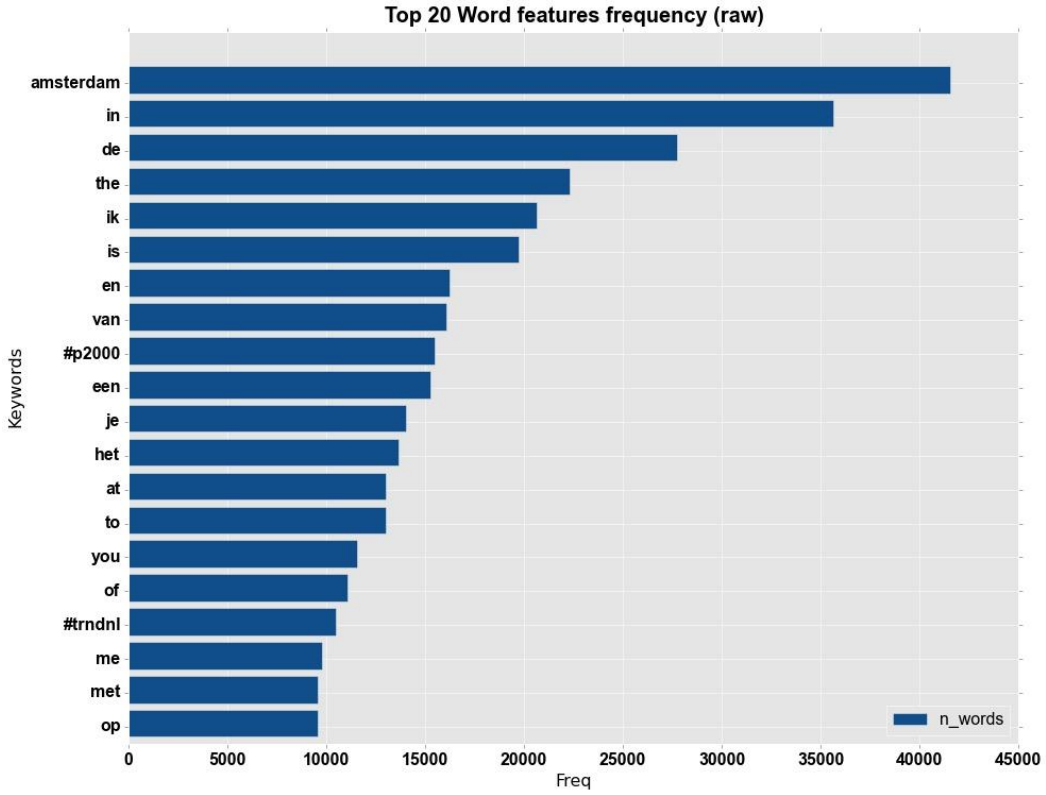


Figure 28: Word frequency of raw Twitter data

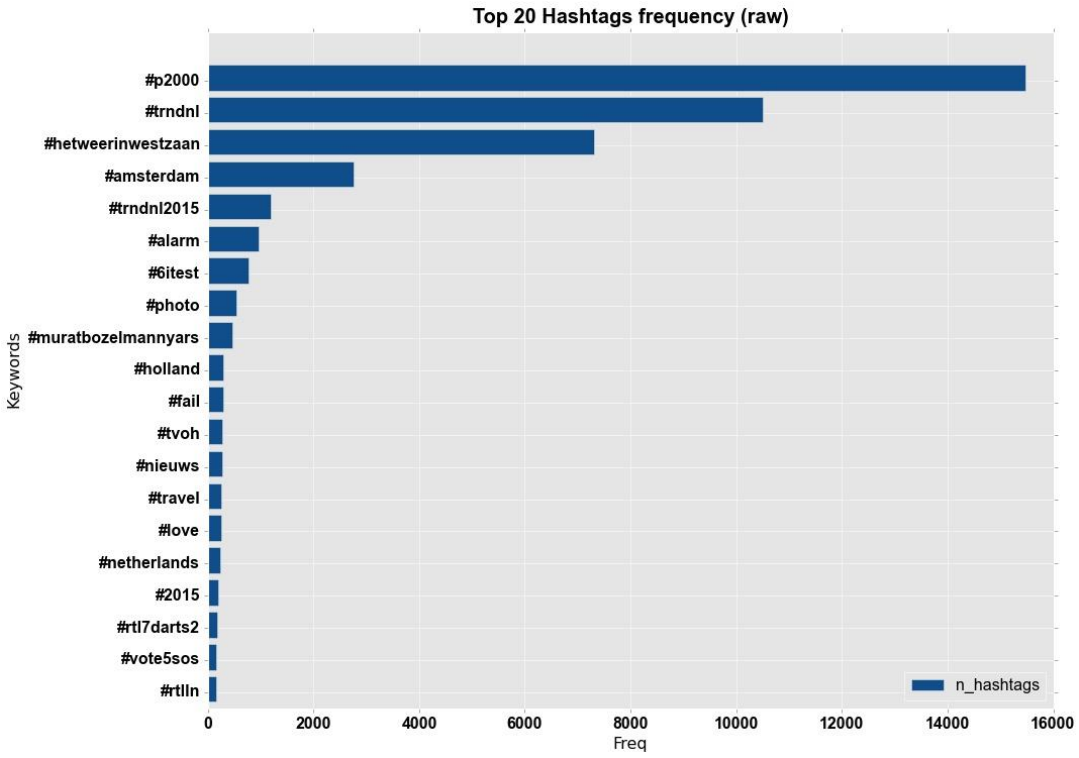


Figure 29: Hashtag frequency of raw Twitter data

The frequency of some hashtags in Figure 29 is as high as the frequency of stop words in Figure 28, thus most of them could be machine generated and could be classified as spams. Let us consider the first three items as sample data, namely *#p2000*, *#trndnl*, and *#hetweerinwestzaan*.

The hashtag ***#p2000*** appears in the majority of posts and it is machine generated data (Figure 30). The figure shows the distribution of this information over the city and the frequency coincides with one of those above the average, which are marked in red. Thereafter, I investigate the type of information posted as shown in Table 10.

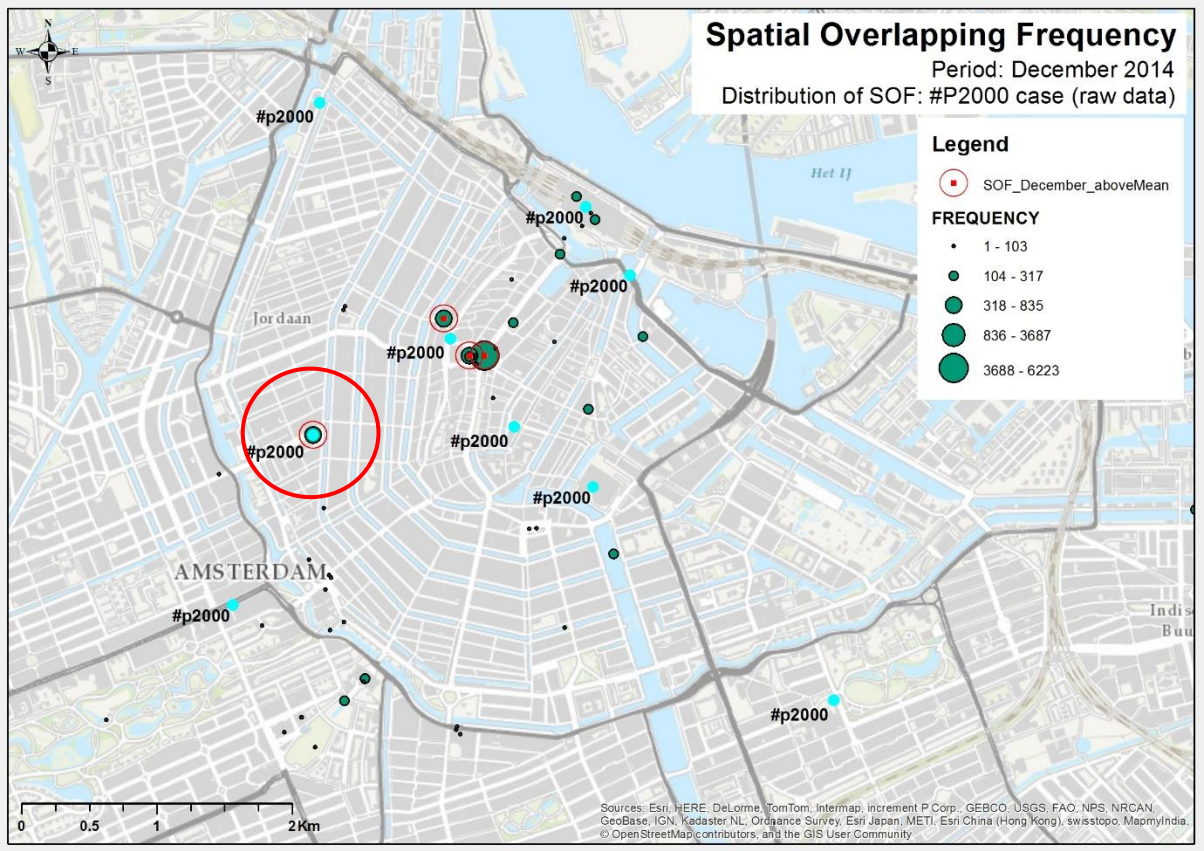


Figure 30: SOF locations of #p2000 hashtag. The red circle highlights the SOF location above the mean frequency value which match with the #p2000 spam.

POST	FREQUENCY
B2 Amsterdam Elandsgracht 117 Meldkamer Ambulancedienst (Achterwacht REGIO: PUR Contact MKA geen spoed) http://t.co/4TAk2ZZak5 #p2000	597
B2 Amsterdam Elandsgracht 117 Meldkamer Ambulancedienst (Contact MKA:) http://t.co/Dh1y5W5k08 #p2000	597
B2 Amsterdam Elandsgracht 117 Meldkamer Ambulancedienst (Achterwacht STAD: OOS Contact MKA geen spoed) http://t.co/mYhhXxffgg #p2000	597
B2 Amsterdam Elandsgracht 117 Meldkamer Ambulancedienst (Contact MKA:) http://t.co/Z8cmFOdyAW #p2000	597
B2 Amsterdam Elandsgracht 117 Meldkamer Ambulancedienst (Contact MKA:) http://t.co/Pwk2RPTzHT #p2000	597
B2 Amsterdam Elandsgracht 117 Meldkamer Ambulancedienst (Contact MKA:) http://t.co/5ZRGighx43 #p2000	597
B2 Amsterdam Elandsgracht 117 Meldkamer Ambulancedienst (Contact MKA:) http://t.co/y4E4WY5Dfc #p2000	597
B2 Amsterdam Elandsgracht 117 Meldkamer Ambulancedienst (Contact MKA:) http://t.co/OqWuOpbBne #p2000	597
A1 13103 Amsterdam Elandsgracht 117 Celle complex Hoofdbureau http://t.co/XwjH7FhnzP #p2000	597
B2 Amsterdam Elandsgracht 117 Meldkamer Ambulancedienst (Contact MKA:) http://t.co/VxmfeiJcMa #p2000	597
B2 Amsterdam Elandsgracht 117 Meldkamer Ambulancedienst (Contact MKA:) http://t.co/y3KxlacRlr #p2000	597
B2 Amsterdam Elandsgracht 117 Meldkamer Ambulancedienst (Contact MKA:) http://t.co/fvVnURepXL #p2000	597
B2 Amsterdam Elandsgracht 117 Meldkamer Ambulancedienst (Achterwacht REGIO: ZAA Contact MKA geen spoed) http://t.co/o4eiAQLZm3 #p2000	597

Table 10: A sample of posts at the location within the red square. A stands for Ambulance whereas B stands for Brandweer (i.e. Firefighter in Dutch)

According to the information displayed in Table 10, #p2000[60] seems to be a Dutch monitoring system for emergency services which generates tweets anytime an emergency call to ambulance, police, or firefighters and so on is dialled. As a confirmation, the locations found with the SOF model match with the location of emergency calls within Amsterdam which are found at the website http://watiserloos.in/. This service is therefore classified as spam and is removed prior to the text analysis given that its topic does not regard touristic information.

Next in line, the ***#trndnl*** hashtag is under the microscope and it is displayed into the map through the same procedure used in the previous spam identification task (Figure 31). In this case, there is only one location in which the service appears and it is in the heart of Amsterdam city: Dam Square (i.e. the cyan selection). In Table 11, the typology of posts generated by this service are displayed.

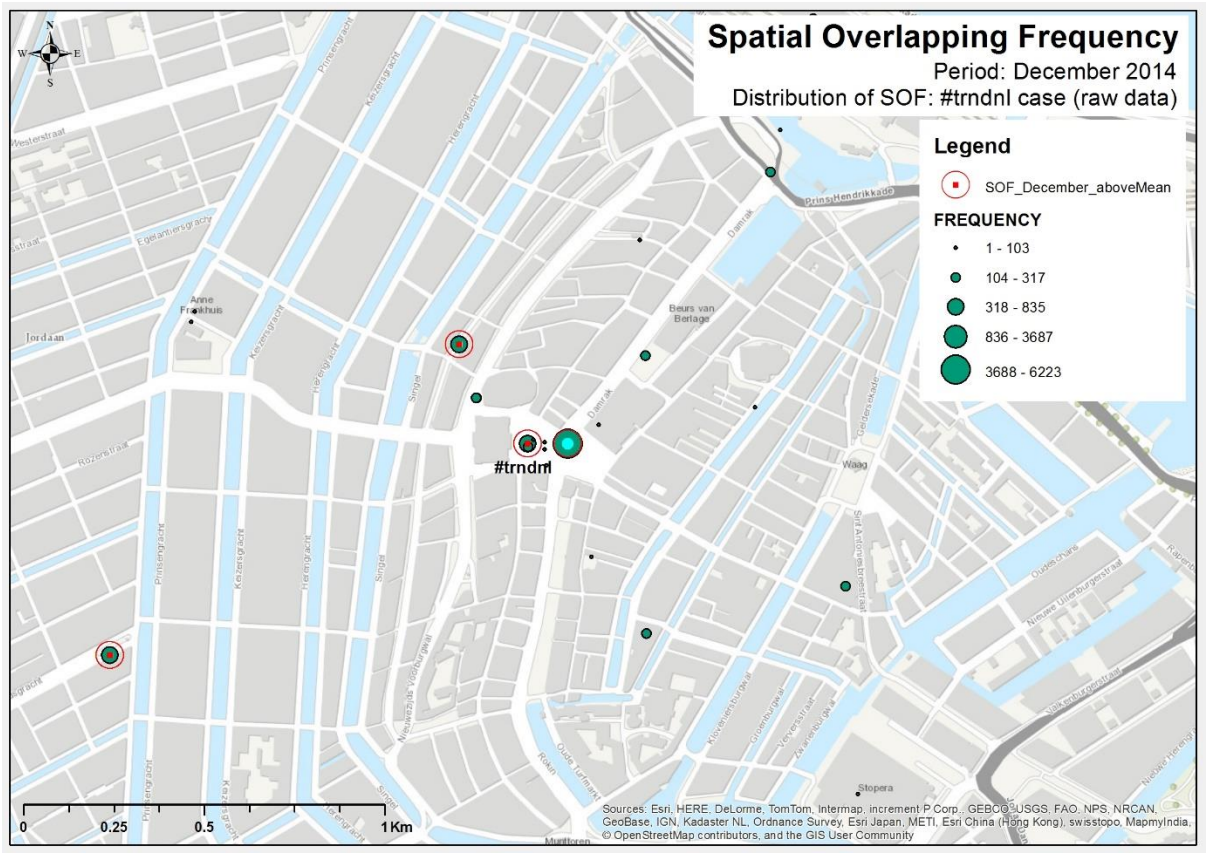


Figure 31: SOF locations of #trndnl hashtag

The information displayed in the Table 11 below clearly identifies the presence of a Twitter analytic service which generate posts over topics, word and hashtag trends posted by users within the city. The high frequency being detected reports approximately 200 posts each day that is roughly 8 posts every hours. If this information is considered in the calculation of the density of tweets, it could compromises the accuracy and reliability of the result giving a wrong indication in that area. Hence, also this spam is removed from the dataset.

[60] #p2000 service in Amsterdam - (Elandsgracht): http://watiserloos.in/melding/10068085/b2-amsterdam-elandsgracht-117-meldkamer-ambulancedienst-(contact-mka-).html

POST	FREQUENCY
6. Sint7. Nederland8. Goedemorgen9. Amsterdam10. Zwarte Piet2014/12/1 00:49 CET #trndnl http://t.co/tOVVBUluRT	6223
1. #MTVStars2. #adoaja3. #tegenlicht4. #ikvertrek5. #twegro2014/12/1 00:49 CET #trndnl http://t.co/tOVVBUluRT	6223
6. Sint7. Nederland8. Amsterdam9. Ajax10. Waarom2014/12/1 01:15 CET #trndnl http://t.co/tOVVBUluRT	6223
1. #MTVStars2. #adoaja3. #tegenlicht4. #ikvertrek5. #Jinek2014/12/1 01:15 CET #trndnl http://t.co/tOVVBUluRT	6223
Trend Alert: #Jinek. More trends at http://t.co/tOVVBUluRT #trndnl http://t.co/d2jratE7Ww	6223
The longest Trends for Sunday 30 in Netherlands was 16 characters: http://t.co/TcHzcqPhGk #trndnl	6223
6. Sint7. Nederland8. Amsterdam9. Ajax10. Waarom2014/12/1 01:28 CET #trndnl http://t.co/tOVVBUluRT	6223
1. #MTVStars2. #adoaja3. #tegenlicht4. #jinek5. #ikvertrek2014/12/1 01:28 CET #trndnl http://t.co/tOVVBUluRT	6223
On Sunday 30 'Zoeterwoude' was Trending Topic in Netherlands for 12 hours: http://t.co/TcHzcqPhGk #trndnl	6223
6. Sint7. Nederland8. Amsterdam9. Ajax10. Waarom2014/12/1 01:53 CET #trndnl http://t.co/tOVVBUluRT	6223
1. #MTVStars2. #adoaja3. #tegenlicht4. #jinek5. #ikvertrek2014/12/1 01:53 CET #trndnl http://t.co/tOVVBUluRT	6223
6. Sint7. Nederland8. Amsterdam9. Ajax10. Waarom2014/12/1 02:06 CET #trndnl http://t.co/tOVVBUluRT	6223
1. #MTVStars2. #adoaja3. #tegenlicht4. #jinek5. #ikvertrek2014/12/1 02:06 CET #trndnl http://t.co/tOVVBUluRT	6223
39% of the Netherlands's Trends for Sunday 30 were hashtags: http://t.co/TcHzcqPhGk #trndnl	6223

Table 11: A sample of posts at the location in the centre highlighted in cyan. Repetitive patterns as well as words appear in text.

In the last part of the SOF investigation, I consider the hashtag service ***#hetweerinwestzaan***, which already from the name could be thought of as a possible meteorological station (i.e. the weather in Westzaan would be the English translation from Dutch). In this case, the location of this service is far away the city centre of Amsterdam as shown in Figure 32. Interestingly, this information clearly display the limitation of the filter offered by the Tweepy API as, despite the fact that the area falls outside the chosen bounding box, tweets are still collected.

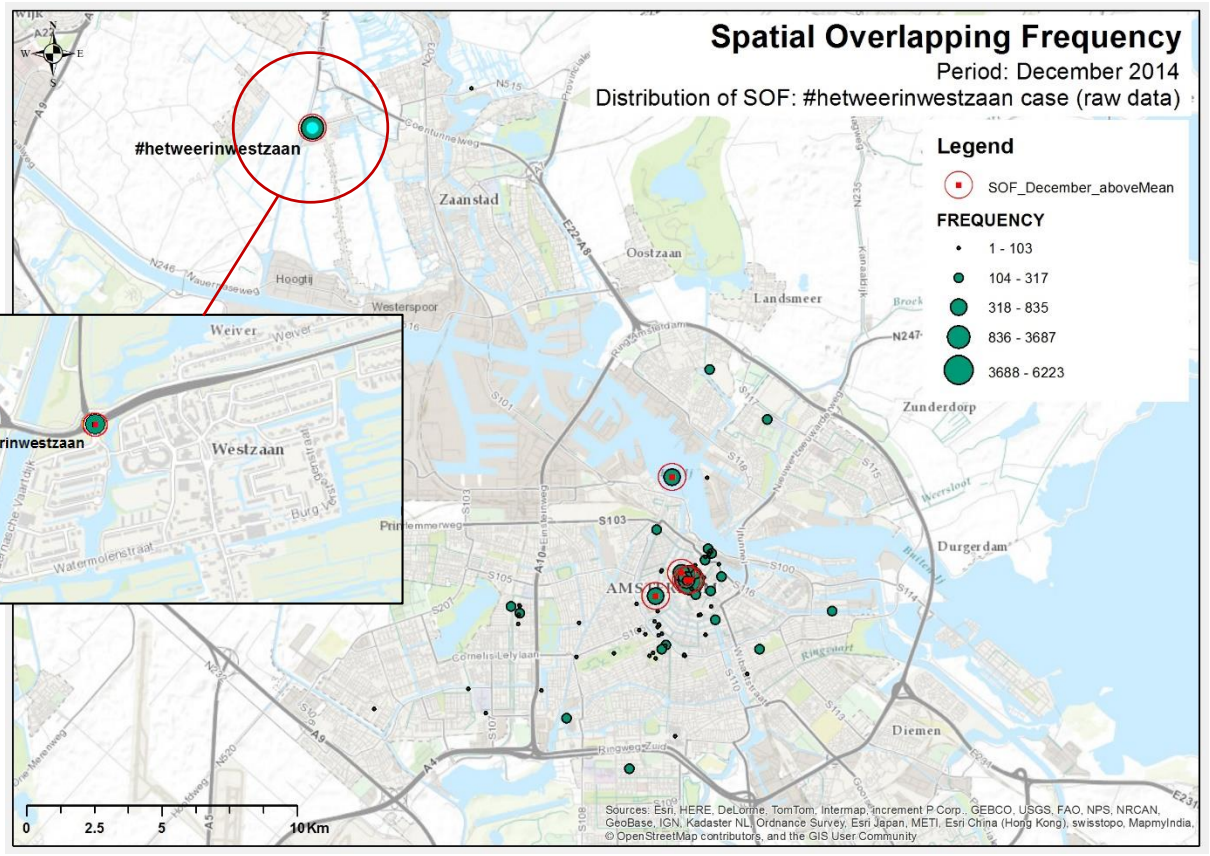


Figure 32: SOF locations of #hetweerinwestzaan hashtag

Furthermore, the post investigation confirms the finding as information over weather conditions are reported following a structured schema which is illustrated in Table 12.

POST	FREQUENCY
Baro 1012 0mb-Stabiel. Temp 2 4c (-0 3). Hum 78%. Rain last 24h 0 0mm. Wind 7 8kph ENE / gust 14 0kph. UV 0. #hetweerinwestzaan	3687
Baro 1012 0mb-Stabiel. Temp 2 3c (-0 3). Hum 77%. Rain last 24h 0 0mm. Wind 10 0kph ENE / gust 16 6kph. UV 0. #hetweerinwestzaan	3687
Baro 1011 0mb-Daalt langzaam. Temp 2 3c (-0 3). Hum 77%. Rain last 24h 0 0mm. Wind 9 9kph ENE / gust 16 2kph. UV 0. #hetweerinwestzaan	3687
Baro 1011 0mb-Daalt langzaam. Temp 2 3c (-0 3). Hum 77%. Rain last 24h 0 0mm. Wind 8 2kph ENE / gust 16 2kph. UV 0. #hetweerinwestzaan	3687
Baro 1012 0mb-Stabiel. Temp 2 3c (-0 3). Hum 77%. Rain last 24h 0 0mm. Wind 8 9kph ENE / gust 14 0kph. UV 0. #hetweerinwestzaan	3687
Baro 1012 0mb-Stabiel. Temp 2 3c (-0 2). Hum 77%. Rain last 24h 0 0mm. Wind 9 5kph ENE / gust 16 9kph. UV 0. #hetweerinwestzaan	3687
Baro 1012 0mb-Stabiel. Temp 2 2c (-0 3). Hum 77%. Rain last 24h 0 0mm. Wind 9 9kph ENE / gust 16 9kph. UV 0. #hetweerinwestzaan	3687
Baro 1011 0mb-Daalt langzaam. Temp 2 3c (-0 2). Hum 78%. Rain last 24h 0 0mm. Wind 9 0kph ENE / gust 20 5kph. UV 0. #hetweerinwestzaan	3687
Baro 1011 0mb-Daalt langzaam. Temp 2 2c (-0 2). Hum 78%. Rain last 24h 0 0mm. Wind 11 2kph ENE / gust 19 8kph. UV 0. #hetweerinwestzaan	3687
Baro 1011 0mb-Daalt langzaam. Temp 2 2c (-0 2). Hum 77%. Rain last 24h 0 0mm. Wind 9 3kph ENE / gust 16 6kph. UV 0. #hetweerinwestzaan	3687
Baro 1011 0mb-Stabiel. Temp 1 9c (-0 1). Hum 76%. Rain last 24h 0 0mm. Wind 8 6kph E / gust 16 9kph. UV 0. #hetweerinwestzaan	3687
Baro 1011 0mb-Stabiel. Temp 1 9c (-0 1). Hum 76%. Rain last 24h 0 0mm. Wind 8 9kph ENE / gust 16 6kph. UV 0. #hetweerinwestzaan	3687
Baro 1011 0mb-Stabiel. Temp 1 9c (-0 1). Hum 76%. Rain last 24h 0 0mm. Wind 10 6kph E / gust 15 8kph. UV 0. #hetweerinwestzaan	3687
Baro 1011 0mb-Stabiel. Temp 1 9c (-0 1). Hum 76%. Rain last 24h 0 0mm. Wind 7 8kph E / gust 12 6kph. UV 0. #hetweerinwestzaan	3687
Baro 1011 0mb-Stabiel. Temp 1 9c (0). Hum 76%. Rain last 24h 0 0mm. Wind 8 5kph E / gust 16 2kph. UV 0. #hetweerinwestzaan	3687

Table 12: A sample of posts at the location highlighted in cyan. Repetitive patterns as well as words appear in text.

The combination of SOF and FDA approaches seems to return promising results in the identification of spams in the dataset. Other spams such as ***#trndnl2015***, ***#alarm***, ***#6iTest*** and so on could also be identified by going over the list of the top hashtags retrieved with the FDA techniques. Although many of them are reported in the chart in Figure 29 above, that represents only an extract which shows the top 20 hashtags, thus it does not include all those being removed. The full list of spam service is removed during the text classification process.

Raw vs Processes – Comparing datasets prior and after spam deletion

To end with this part of the analysis, I argue the results once spams are removed from the dataset and I compare them with the original (raw) input in order to display the effect of the noise generated by spams. Besides, the FDA tool is performed also during the analysis of the processed dataset in order to investigate new possible patterns in posts, such as free Wi-Fi services exploration. The map below shows the comparison between the SOF of collected (raw) dataset and the SOF of the cleaned dataset (i.e. without spams). Specifically, I compare the spatial distribution of SOF whose frequency is above the relative mean value (i.e. the mean value is calculated separately for both original and cleaned datasets) and I display them on map as follow (See also Appendix F):

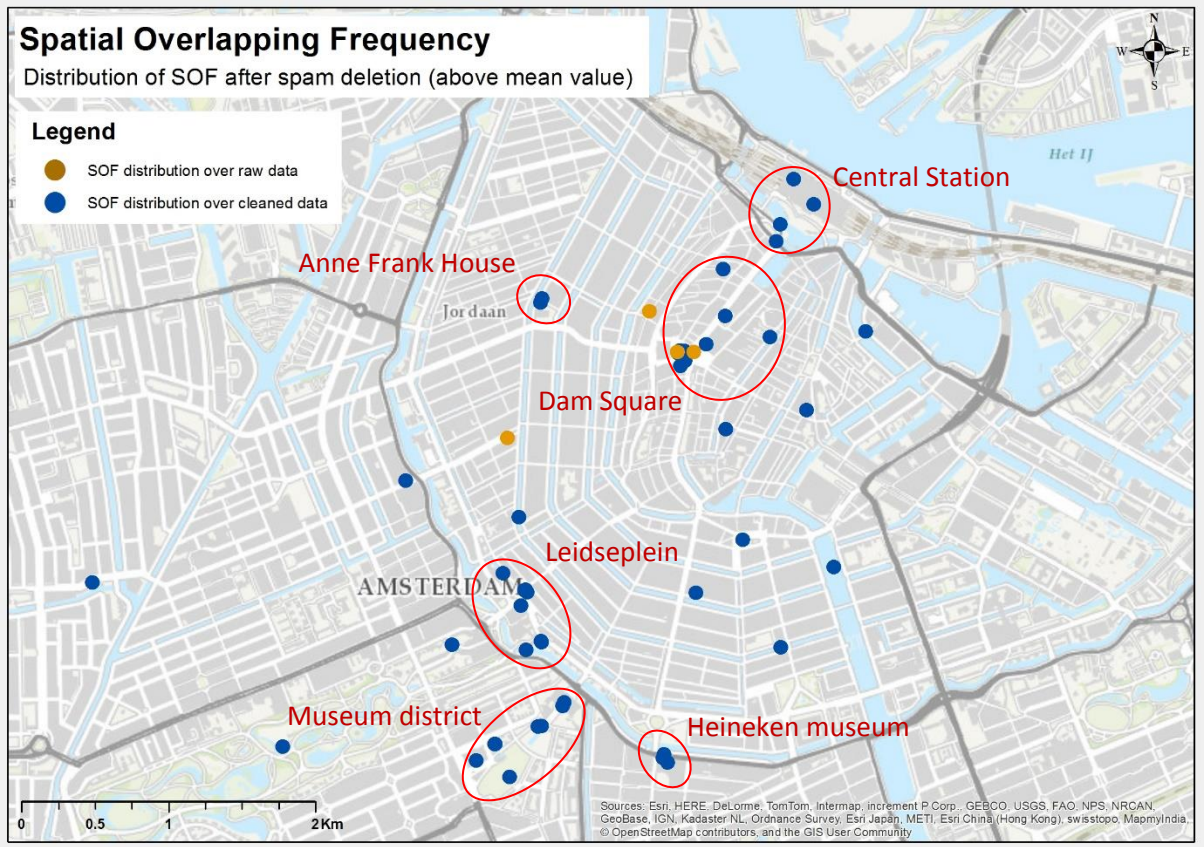


Figure 33: SOF raw vs clean. The cyan markers display the locations where SOF are still high. However those are user generated contents posted via Wi-Fi open services.

The distribution of SOF, absent of the influence of spams, shows a more uniform pattern with clusters forming in popular touristic areas of the city. The findings reveal the footprint that spams produced in terms of noise during visualization. In fact, patterns that were hidden before, such as aggregation of SOF highlighted in red, are now shown above all in proximity of touristic venues in Figure 33.

Tweets at those highlighted locations have been investigated and interesting outcomes were found. The majority of tweets at locations being analysed express touristic information. However, it is interesting to illustrate the source of these SOF so I selected a sample area to carry out a more detailed investigation. The chosen area is the museum district and it is displayed in Figure 34. I have assigned a random post as the graphical label of each SOF and I can demonstrate that the information included in posts is somehow linked to the location from which that Twitter content is posted.

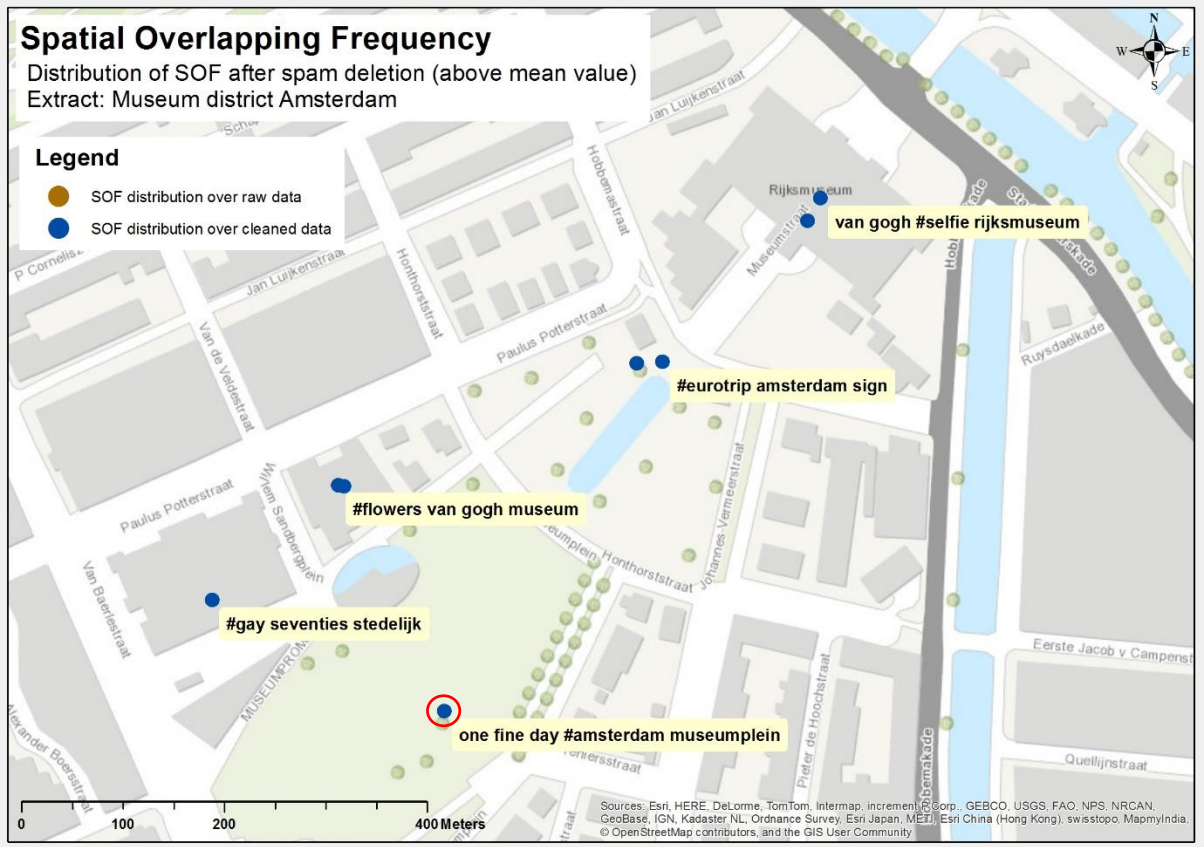


Figure 34: Extract from the distribution of SOF after spam deletion (Figure 25). Museum district area

Interestingly, the unique location displayed (rather than many unique locations within the same building) suggests the presence of an open Wi-Fi service that tourists can access for free as for the Rijksmuseum[61], for instance. This is true for a building, but what if the location of SOF is in an open space (red circle) away from any Wi-Fi coverage? The question is answered by the temporal component. Actually, during the month of December that green area (i.e. Museumplein) is equipped with an ice skating[62] complex, including some facilities like cafes and a restaurant. Probably, the SOF location coincides with that of the restaurant offering a free Wi-Fi access point for its customers. A sample of tweets posted from that location are displayed in Table 13.

[61] The presence of a free Wi-Fi service is confirmed by the FAQ on the official website of the Rijksmuseum: https://www.rijksmuseum.nl/en/organisation/frequently-asked-questions/visiting.
[62] See the official webpage of the event @: http://iceamsterdam.nl/en.

iceskating

TS	POST	FREQUENCY
Tue 09-12-14 14:56	festive feeling museumplein	44
Tue 09-12-14 18:10	getting pushed around like old mantoddler nmsterdam museumplein	44
Wed 10-12-14 14:21	freezing say photo took classical museumplein	44
Thu 11-12-14 19:54	#sunnyday #amsterdamtrip #eurotrip #us #luv museumplein	44
Fri 12-12-14 13:32	back amsterdam good weather still beauty city de volta amsterdam com uma	44
Fri 12-12-14 18:35	beeing cool #orcold #visionsoul museumplein	44
Sun 14-12-14 17:45	#iceskating #amsterdam #iamsterdam #museumplein museumplein	44
Wed 17-12-14 09:57	absolutely impossible monopolize letters given point time museumplein	44
Wed 17-12-14 10:03	absolutely impossible monopolize letters given time haha museumplein	44

Table 13: A sample of posts enriched with time and frequencies to validate the type of information of the SOF.

Last to be investigate but not least, is the SOF above the mean value located at the Schiphol Airport (See Figure 27 above). During the exploration, the highest frequency of user generated tweets, posted by people arriving, departing and/or transiting the hub, is noticed (Table 14). In addition, the text structure of those posts has a mixed character which helps to distinguish an open Wi-Fi connection service. The extension of the signal allows customers to stay connected from the aprons until just outside the terminals. In Figure 35 and Table 14, this is clearly illustrated through the findings of the SOF model at Amsterdam Schiphol Airport.

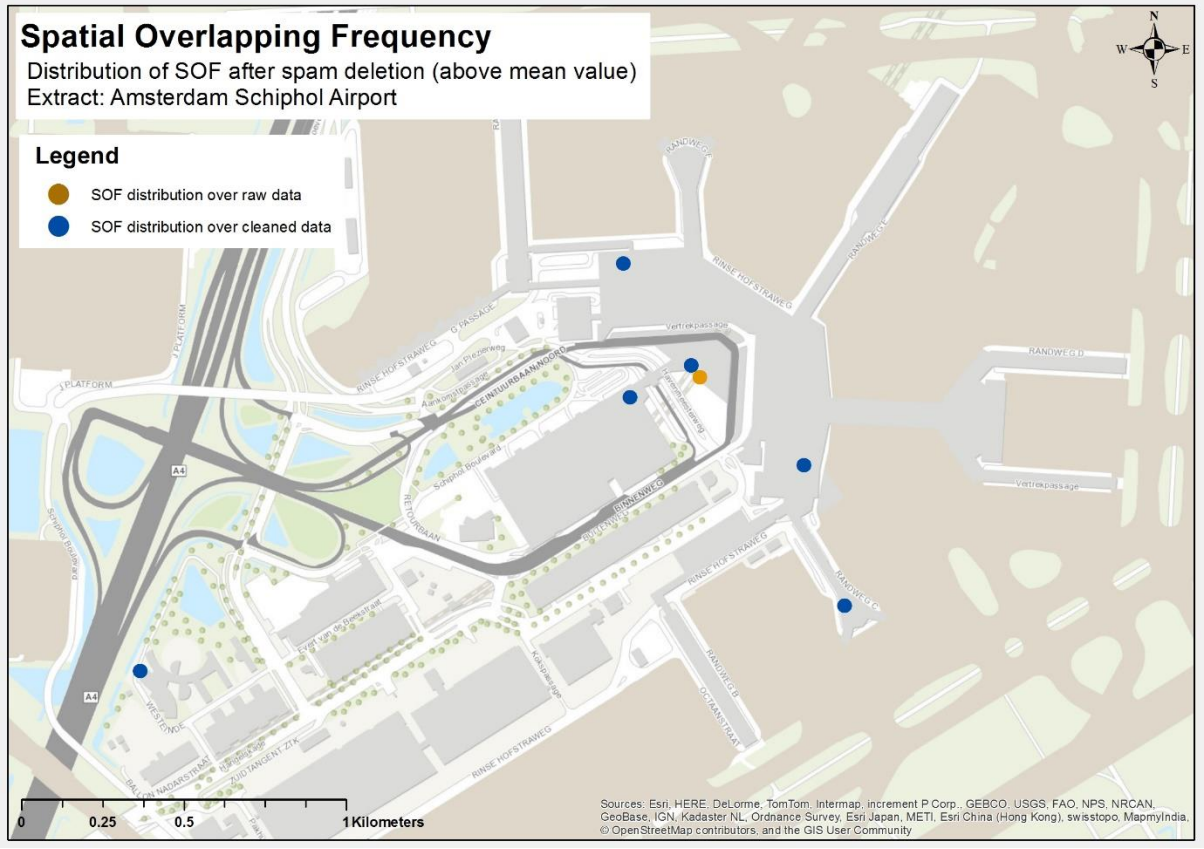


Figure 35: Spatial Overlapping Frequency in Amsterdam Schiphol Airport. Five more SOF are shown after the spam deletion.

POST	FREQUENCY
I'm at Amsterdam Airport @Schiphol (AMS) in Schiphol Noord-Holland https://t.co/nknB0Q05fs	835
Ready to boarding soon @ Amsterdam Airport @Schiphol (AMS) in Schiphol Noord-Holland) https://t.co/HUjqLstmSU	835
I'm at Amsterdam Airport @Schiphol (AMS) in Schiphol Noord-Holland https://t.co/ZYgGBcVZan	835
Amsterdam mailov! (@ Amsterdam Airport @Schiphol (AMS) in Schiphol Noord-Holland) https://t.co/CXO3zXmtmi	835
I'm at Amsterdam Airport @Schiphol (AMS) in Schiphol Noord-Holland https://t.co/MZfhCsSEhZ	835
I'm at Amsterdam Airport @Schiphol (AMS) in Schiphol Noord-Holland https://t.co/JGP1IlX0RC	835
Let's recruit IT professionals in Bucharest for #ING (@ Amsterdam Airport @Schiphol (AMS)) https://t.co/jS9CArxcj5	835
On our way to Berlin. (@ Amsterdam Airport @Schiphol (AMS) in Schiphol Noord-Holland) https://t.co/tssRLv4lew	835
Off to Barcelona for HP Discover (@ Amsterdam Airport @Schiphol (AMS) in Schiphol Noord-Holland) https://t.co/chNSpQ7c7J	835

Table 14: A sample of tweets posted from the terminals of Schiphol Airport. The user generated contents are highlighted in blue, while the output of the automatized service is highlighted in red.

In conclusion, the findings of the data analysis depict a strong relation between the locations of tweets and the topic they express. Particularly, the frequency as well as the text structure attributes support the identification of spams such as #p2000, #trndnl and many other services that are not linked to touristic activities. The results confirmed the negative effect of spams over the distribution of the frequencies, at specific locations, detected via SOF model. Moreover, assumption 4 is also confirmed by the results shown in Figure 35 above. Each and every museum in the considered area offers free Wi-Fi connection for its customers thus it is explained why SOF are located within each building. In that context, the SOF model also represents a powerful approach to distinguish high frequency generated by spams from those which are generated by Wi-Fi services. This is done by using the mean value of calculated SOF as a threshold, defining spam everything above it and Wi-Fi services, otherwise.

A major drawback in this approach is found in the use of the FDA to retrieve the distribution of top hashtags in text. By doing so, a number of news services, (i.e. such as radio stations and touristic information services using Twitter to post news) showing high values of SOF but posting content of various nature, are not identified due to the large lexical diversity of words contained in it. Those services have to be manually removed (Figure 36).

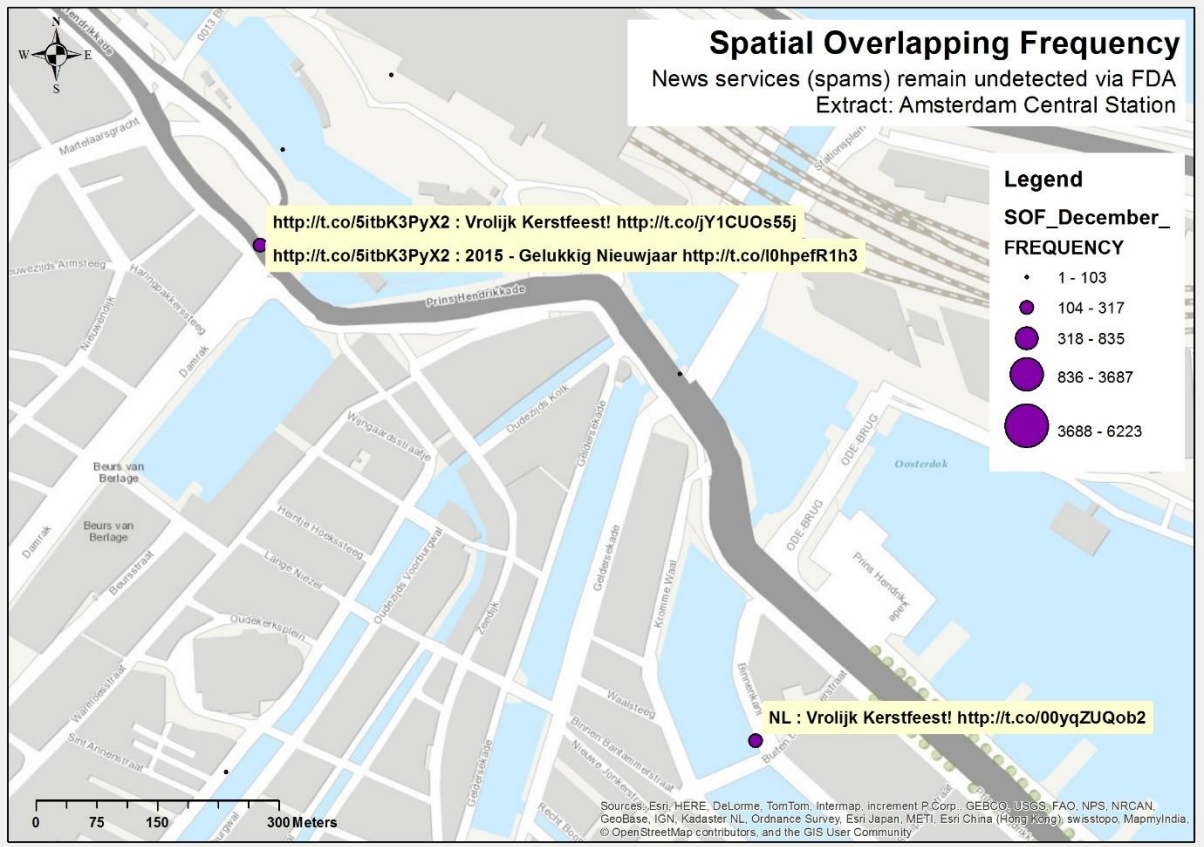


Figure 36: Extract of the SOF in Amsterdam city centre. FDA does not identify word frequency for news services

In the end, the output of this phase is a dataset in which spams are disregarder meanwhile contents uploaded via Wi-Fi are retained. The data is deployed in the next step of this methodology in which a dedicated combination of text processing, features extraction, text classification and topic modelling tasks is created. Through the method, namely TAUS GKD I classify the text attribute with regards to what is touristic information and what is not using the NBC tool, and I assess the behaviour of tourists, who visit Amsterdam during the month of December, by means of the LDA algorithm.

6.4. TAUS GKD method: LAS identification and assessment (4)

In this part of the work, I study the text attribute of tweets (i.e. the post), with the purpose of discovering meaningful patterns in terms of tourism behaviour in urban environment. By meaningful pattern, I refer to tourism aggregation trends occurring in different areas of Amsterdam at different time of the day and the week. This returns a visual indication of the densest locations where crowds of people gather together due to the attractiveness I_i of touristic venues in the near proximity. The objective of this stage is to assign a label or labels to each aggregation in order to identify the most probable reason that leads to such a converging behaviour. The TAUS GKD methodology is as depicted in Figure 16, Paragraph 5.5.

6.4.1.Text normalization

In the first step, I use text processing techniques of the Natural Language Processing (NLP) approach to clean up the text from mistakes, punctuations, web characters and so on in order to obtain a more "natural", homogeneous text. Moreover, due to the fact that a multitude of languages is found (Figure 37), I also need to normalize the text to a unique language by means of a Python function supported by the TextBlob library (Figure 38). Actually, some tasks of the NLP process do only work on a language per time, such as stop word deletion for instance. In addition to that, the same topic could be expressed in different languages thus creating redundancy of information. Redundancy of words influences the accuracy of the Naïve Bayes Classifier (NBC) hence has to be removed.

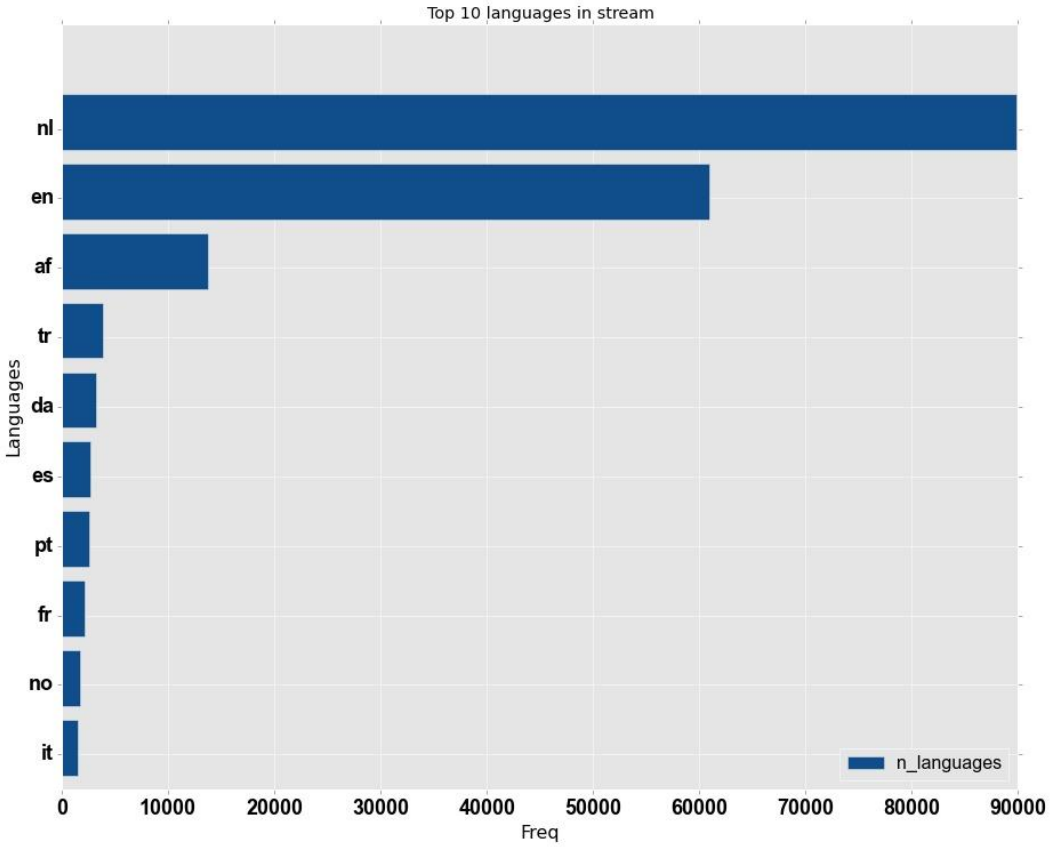


Figure 37: Different languages identified in posts. Use the link below for referencing the language code to the language name: https://cloud.google.com/translate/v2/using_rest#language-params.

```
# Languages_list
langs = []
# Function to detect non-English post, and translate them to English
def translator(text):
    try:
        tokens = TextBlob(text)
        lang = tokens.detect_language()
        langs.append(lang)
        if lang != 'en':
            return tokens.translate(from_lang=lang, to="en")
        else:
            return tokens
    except Exception, (e):
        print 'Error in Translator:', str(e)
```

Figure 38: Python recipe - Translation and language identification function

After the text is translated, I create a Python function with the support of the NLP tools to clean the text (Figure 39). The NLP is enabled in Python though the use of a dedicated library, called NLTK. The function takes raw text as its input and it removes new line characters ('\n') often present in web fonts, non-alphanumeric characters, words shorter than three letters, and stop words. Thereafter, it reduces each word to its root through the deployment of the WordNet dictionary, internal to the NLTK library. This approach considerably enhances the accuracy of classification by reducing words with similar meaning to just one. For example, the verb to go can be found in text in several forms such as gerund, past and participle past tenses, and so on. During classification process, all those forms are considered as different words even though they relate to the same verb. The reduce words to their roots the *Word Lemmatize* NLTK object is enabled. In addition, the Treebank object supports the identification of the English parts of speech (e.g. nouns, adverbs, verbs) helping the Word Lemmatize object to improve the reliability of lemmatization.

```
# NLP utilities
lmtzr = WordNetLemmatizer()
stopwords = stopwords.words('english')
stopwords.extend(unicode(['also', 'us', 'this', 'not']))
nonan = re.compile(r'[^a-zA-Z ]')
shortword = re.compile(r'\W*\b\w{1,2}\b')


# Treebank: PARTS OF SPEECH IDENTIFICATION
tag_to_type = {'J': wordnet.ADJ, 'V': wordnet.VERB, 'R': wordnet.ADV}
def wordnet_pos(treebank_tag):
    return tag_to_type.get(treebank_tag[:1], wordnet.NOUN)

# Clean tweet from characters not used in the text classification
def text_cleaner(txt):
    """ Clean text from english stopwords and lemmatize (word_conj to root) the output test """
    try:
        new_line_rem = re.sub('<(.|\n)*?>', '', txt)
        clean_text = nltk.word_tokenize(shortword.sub('', nonan.sub('', new_line_rem)))
        filtered_w = [w for w in clean_text if not w in stopwords]
        tags = nltk.pos_tag(filtered_w)
        return ' '.join(lmtzr.lemmatize(word, wordnet_pos(tag[1])) for word, tag in zip(filtered_w, tags))
    except Exception, e:
         print 'Error in Txt cleaner:', str(e)
```

Figure 39: Python recipe - Text cleaning function

After the post pre-processing task is complete, I assess the FDA over the cleaned text and I obtain the following words and hashtags frequencies (Figures 40 and 41, respectively). While the words frequency offers an overview upon the effectiveness of the cleaning function

of Python, the hashtags frequency supports the identification of the variables (i.e. hashtags) that are deployed in the next phase of features extraction and labelling.

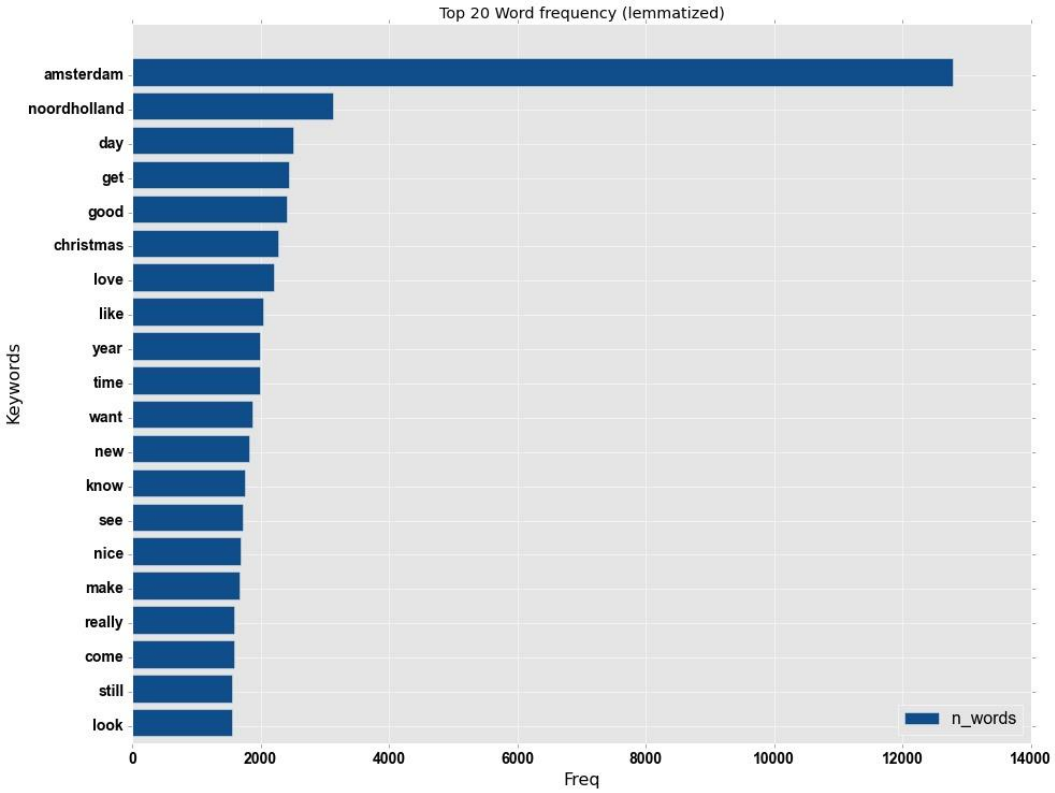


Figure 40: FDA of words in corpus.

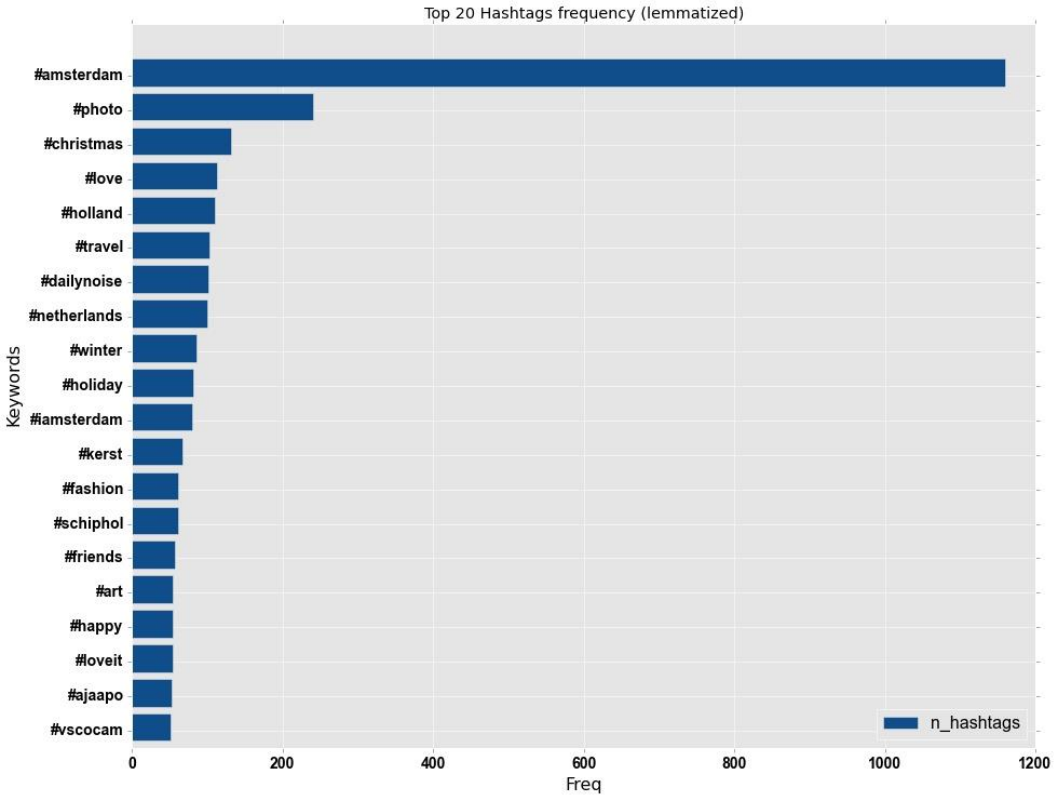


Figure 41: FDA of hashtags in corpus.

In the next step the method adopted to carry on the features extraction and features labelling is argued. The accuracy of extraction is directly proportionated to the accuracy of the classification. To improve the understanding of the jargon of next phase, I address the text attribute of tweets as the ***post***, posts are also referred as the ***documents***, and a collection of document is the ***corpus***, whose words represent the corpus ***features***.

6.4.2. Features extraction and labelling

This part describes the procedure adopted in order to extract those features that serve as baseline to teach the classifier. The Twitter hashtag indexing system is considered during the extraction of features. The hashtag itself is already a classification of tweets into different topics, and I show an example to be able to prove it. By querying the hashtag '*#holiday*' found in the corpus: a sample of the obtained results is displayed in Table 15:

ID	Posts linked to "#holiday":
1	welcome cheers #serefe #happyhours #holiday #beers #fun #joy #love #weekendescape
2	amsterdam gives good face pretty jadore #beautiful #weekendbreak #love #like #holiday
3	#hermitage #museum #amsterdam #iamsterdam #holiday #happy #winter #love #art #morning
4	#christmas #tree #little #magazines #glamour #love #magazine #holiday #fun #amsterdam
5	wine dinner thing edible hotel bookinamsterdamn #holidaytravelproblems

Table 15: Example of posts obtained by querying #holiday hashtag.

A bunch of topics, all related to holiday activities are identified in the selection. For instance, posts 1 and 2 seem to express a weekend trip to Amsterdam to celebrate a particular event, perhaps with some drinks (hinted by #beers in post 1), post 3 identifies a typical touristic activity related to a cultural topic (hinted by #hermitage[63] and #museum hashtags), and so on. This approach validates assumptions 1 and 2 argued in Paragraph 5.1, reporting the way people choose to self-represent themselves as a cultural type (post 3) or fun-loving type (post 1, 4). In this context, also assumption 5 is validated as shown with the hashtag-post relationship displayed in Table 15. I identify possible candidate hashtags from which to extract touristic features through FDA (Figure 41 above). For each hashtag[64], I manually select those linked to touristic topics following the same method described above and displayed in Table 15 (Figure 42).

[63] Hermitage museum: Amstel 51, 1018 EJ Amsterdam, Netherlands, https://www.hermitage.nl/en/.
[64] For illustrative reasons I only display the top 20 hashtags frequencies. However, the selection of touristic and non-touristic hashtags is performed on a bigger sample (top 50)

```
# Hashtags related to touristic activities
#from which to extract the features (HIT)
touristic_hashtags = ['#holiday',
                      '#eurotrip',
                      '#trip',
                      '#rijksmuseum',
                      '#instatravel',
                      '#vacation',
                      '#iamsterdam',
                      '#canal',
                      '#city',
                      '#fun'
                      ]
```

```
# Hashtags related to non-touristic activities
#from which to extract the features (MISS)
non_touristic_hashtags = ['#kerst',
                          '#dailynoise',
                          '#ajaapo',
                          '#ajavit',
                          '#kerstcrisis',
                          '#endorphins',
                          '#ajautr',
                          '#art',
                          '#fashion',
                          '#loveit'
                          ]
```

Figure 42: List of manually extracted touristic (top) and non-touristic (bottom) hashtags from which features are extracted.

Top hashtags such as *#amsterdam*, *#photo*, *#holland* and so on, are not included in the analysis due to the wide diversity of topics as well as features that they include. The reason behind this choice is that the classification of posts is based on the features included in a dictionary which serves as baseline for successive features extraction. Therefore, the more specific the features in the dictionary are, the better the outcome of the classification is. For the same reasons, I first remove all words that appear only once as well as stop words[65], as displayed in the "bag_of_words" Python function.

Once features are extracted from the selection of posts, which is done via hashtag FDA, a dictionary of features (i.e. variable "word_features" in Figure 43 below), is used as the reference to create two sets: the training and test sets in the format of the Bag of Words (BoW) model (Figure 43).

```
def bag_of_words(training_data):
    """ Classify the feature set in "hits" and "misses" for tourism and non tourism information, respectively """
    try:
        randoc = [(w, k) for w, k in random.sample(training_data, len(training_data))]
        word_features = nltk.FreqDist(chain(*[i[0] for i in randoc]))
        word_features = word_features.keys()[:]

        numtrain = int(len(randoc) * 75 / 100)
        train = [({i:(i in tokens) for i in word_features}, tag) for tokens,tag in randoc[:numtrain]]
        test = [({i:(i in tokens) for i in word_features}, tag) for tokens,tag  in randoc[numtrain:]]

        print '\n'
        print 'Features dictionary: %d tokens, sample %s' % (int(len(word_features)), word_features[:8])
        print 'Feature set:   %s posts' % str(len(training_data))
        print 'Training set:  %s posts' % str(int(len(train)))
        print 'Test set:      %s posts' % str(int(len(test))), '\n'
        print 'Training data sample: \n', training_data[1],'\n', training_data[2],'\n'
        print "Hits (touristic topics):", len([n for n in training_data if n[1] == 'hit'])
        print "Misses (other topics):", len(training_data) - len([n for n in training_data if n[1] == 'miss']), 2*'\n'
        return train, test, word_features
    except Exception, e:
        print 'Error in Bag_of_words:', str(e)
```

Figure 43: Python recipe - Building the Bag of Words model divided into train and test sets. Format: {[features list], label}

The function assigns to all features extracted from the *touristic hashtags list*, the label ***"hit"***, whereas ***"miss"*** is assigned to those features extracted from the *non-touristic hashtags list*. The BoW model is a part of the classification process and it represents the input of the NBC in the next step: the text classification. As explained in Paragraph 5.5.1.2 (Table 8), the features extraction process is a dedicated approach, and it is applied in both text classification

[65] Despite the deletion of stop words in previous text cleaning step (Figure 39), I make sure that those are removed also at this stage to make sure the features set is as accurate as possible.

and topic modelling techniques by means of different *features vectorization techniques*. Although the input of both techniques is similar, it returns different results, therefore the outcome of the features extraction for topic modelling is argued in the relative section.

6.4.3. Text classification

The objective of the text classification is to distinguish posts linked to tourism activities from those that are not, classifying them as ***hit*** or ***miss***, respectively. This is done by means of the Naïve Bayes Classifier (NBC). The input of the NBC is the output of the BoW model Python function, enriched with the ***"hit"*** and ***"miss"*** labels which I call the ***features set***. Prior to its classification, I proportionally divided the features set into two subsets: the ***training set*** and the ***test set*** (75 and 25 percent, respectively). As the names of the sets suggest, the training set is the actual set from which the NBC learns how to classify new unlabelled posts, while the test set is deployed to assess the accuracy of the classifier. Moreover, to avoid the deleterious effect of words with count zero, which could appear in some of the features, the ***Laplace smoothing estimator*** is set when the classifier is defined enhancing the accuracy of the NBC. This is a customary approach used to obtain probability values rather than maximum likelihood (Lucena et al., 2013). The result of the classification are depicted in Figure 45 and 46. Background information are displayed in Figure 44 with regards to the size and a set of samples of the dictionary of features extracted from touristic and non-touristic posts. Proportional amount of features are considered: 75 percent of the features is assigned to the training set, the remaining features are assigned to the test set for the computation of the NBC accuracy. A sample of the features set format is also displayed (i.e. 'post', 'tag') together with the amount of touristic posts and not, that are found in the features set (Figure 44).

```
Vocabulary:    1732 tokens
Vocabulary sample: [u'gatekeeper', 'ciao', 'palladiumamsterdam', 'four', 'swag', u'paris', u'buyuk', u'bike']

Feature set:   655 posts

Training set:  491 posts
Test set:      164 posts

Training data sample:
('zaandam holiday holland bestoftheday instalike zaandam railway station', 'hit')
(u'christmas come love christmas holiday december', 'hit')

Hits (touristic topics): 372
Misses (other topics): 283
```

Figure 44: The parameter "vocabulary" represents the dictionary of features used to build the BoW.

As shown, the NBC returns a number of information that can be investigated to assess the classification performances. The most important are shown in Figure 45: the accuracy of the classifier and the most informative features considered by the NBC during the classification process. In the NLTK documentation, the accuracy of a classifier is satisfying if it is in a range between 85 and 100 percent.

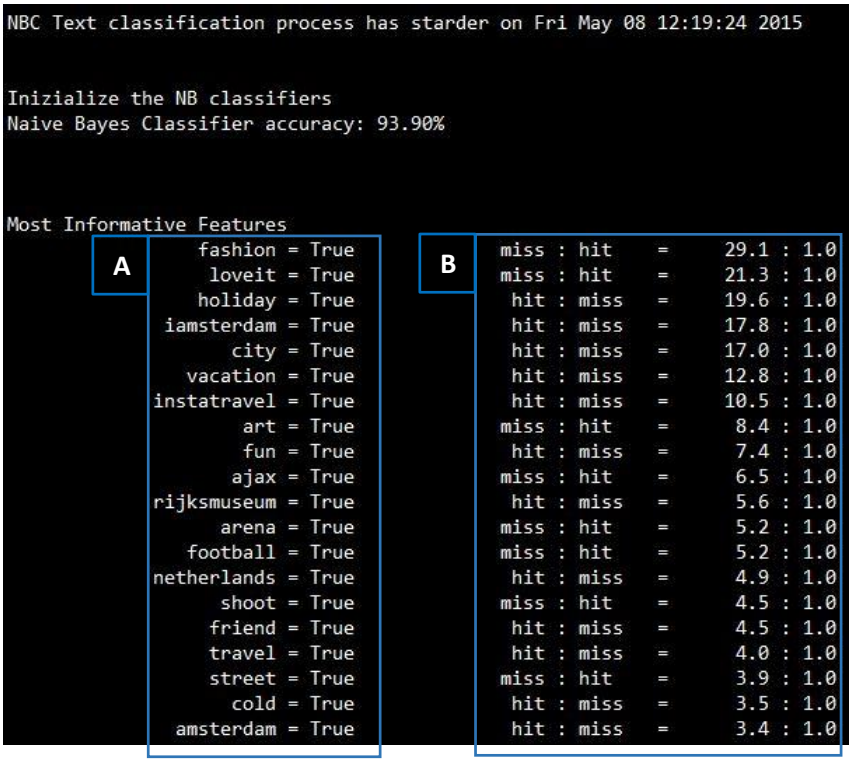


Figure 45: The accuracy of the NBC calculated by comparing the classification results of the train set over the test set. Most informative feature shows the words that the classifier has deployed in the classification task.

The frequency of features is used to assess which of them is useful to the NBC in the classification process and it is depicted in column B in Figure 45 ("*hit: miss*"). So for instance, if the features "holiday", "IAmsterdam", "city", and "vacation", which are hits as shown in column B, are contained in an unlabelled post, then the probability that the post is considered a touristic information is higher.

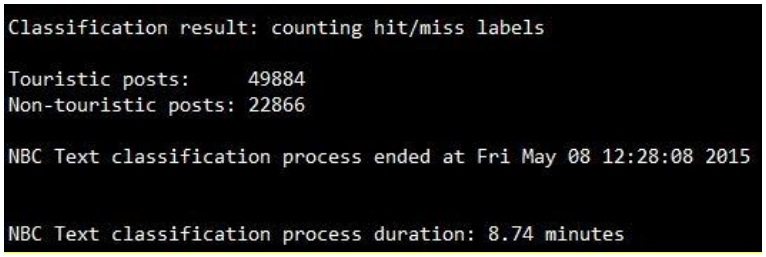


Figure 46: Text classification result – Touristic information represent the majority in the corpus dataset (approx. twice as the size of other information)

The end result of the classification process returns a classified dataset in which there is a number of approximately 50 thousands tweets labelled as touristic information and slightly above 20 thousands non-touristic features. It is interesting to this and following research to consider the low number of useful information that can be extracted from Twitter an important limitation. Actually, after removing noise, spams and useless data attributes, in this work only a quarter of the data originally collected could be used in the analyses. Therefore new ways of collecting, cleaning and/or extracting features should be found.

Next, with the classified corpus, I am finally able to select all touristic features that are going to be deployed in the final part of the TAUS GKD methodology: the semantic analysis or

also referred to as **LAS**. Through this dedicated method, I am able to identify hidden topics in the corpus via LDA algorithm implementation. In particular, I use the K-Means clustering algorithm to implement the ***Elbow method*** and to validate the findings of the ***LDA algorithm*** implementation. Hence, via the K-Means and Elbow, I assess the number of K-topics ***K***, so as to tackle the limitation which it was previously found in related works. As previously stated also in the Methodology Chapter, the K-Means algorithm is introduced as a validation tool to confirm whether the vector scores that the LDA algorithm assigns to the terms of the ***K-number*** of topics obtained are similar, hence clustered in space. In fact, once the results of the LDA are returned in terms of vector similarity, I can show these vectors assigned to each ***K*** topic by the LDA, onto a Cartesian 2-D space and evaluate their (spatial) similarities.

Thereafter, to enable the computation of the LAS, I can classify each post in the corpus by using the average value of the scores as the threshold to establish into which cluster a post falls. Bear in mind that the output of the LAS and the LDA are both in terms of a text labels. The difference is that the LDA outputs top terms related to a topic, whereas the LAS method considers those top terms in order to manually assess the term semantic for each of the topics. Therefore I produce a single general label that identifies the nature of each topic. This label is assigned to one or many touristic posts by comparing their LDA scores with the threshold LDA score: the average score value (i.e. sum of scores by number of scores).

6.4.4. Topic modelling

The techniques used in the topic modelling process are mainly four and are deployed as follows:

1. Elbow – Find the most probable number of topics,
2. TF-IDF + LDA – Find the semantic of topics (topic modelling of text data),
3. (TF-IDF + LDA) + LSI – Vectorization and corpus space decomposition (to 2D),
4. K-Means clustering of vectors (LDA method validation)

The outcomes of this step are represented by steps 2 and 4, specifically the LDA algorithm (2) decomposes the BoW into a number of topics which is defined by the author through a technique called Elbow method (1). The K-Means clustering algorithm aggregate vectors, which are computed through the LDA algorithm and reduced to two dimensions by the LSI model, so as to be plotted into a 2D Cartesian space (See Figure 16, Paragraph 5.5 for more details over this method).

Elbow (1)

The Elbow method uses the squares of the distance of each point in a cluster from the centroid of that cluster (i.e. also referred to as the "within-cluster-sum-of-squares" value) to determine how dense the cluster is. This process is repeated over a K-range from 1 to 10 to be able to find the right number of topics: K. Actually, as K increases, the number of clusters raises and point-centroid distances are smaller, hence the "within-cluster-sum-of-squares" value drops. However, its drop is not always constant and it levels off at a certain K-range giving the value of K equal to 5 topics (Figure 47).

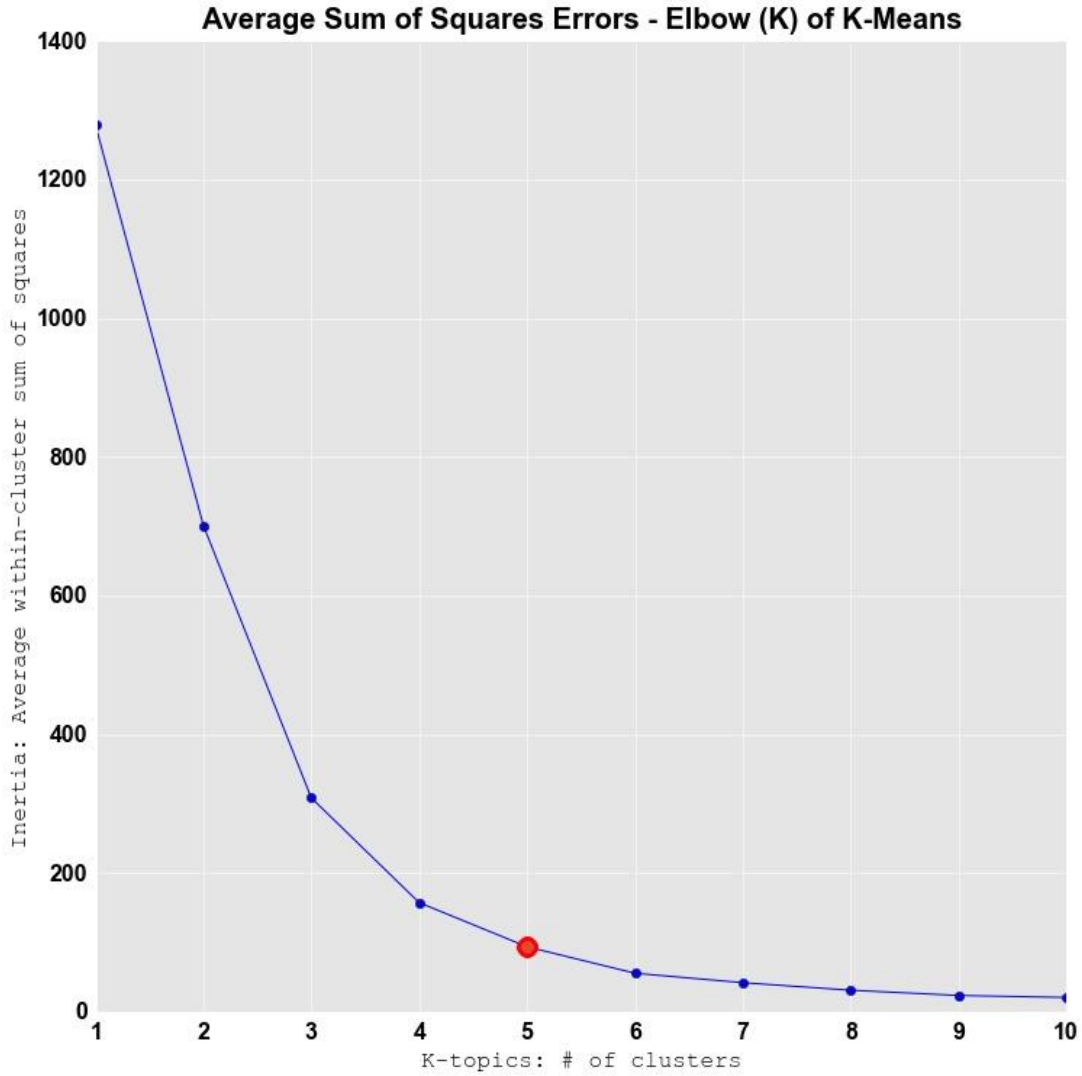


Figure 47: Elbow method - Finding the most suitable value of K-topics in the corpus through the analysis of the SSE.

LDA (2)

Once the value of K is known, the LDA computes a user defined number of most popular words for each of the five topics (i.e. top 10 terms) from which I can logically assess and assign the topic semantic (i.e. the LAS). Figure 48 shows the settings of the LDA algorithm performed on the corpus of tweets. The corpus needs to be vectorised via the TF-IDF model (hollow blue box in Figure 48) in order to enhance the accuracy of the LDA corpus transformation (hollow yellow box). In addition to that, also the LDA model of Gensim give the possibility to adjust the parameters of the model to balance accuracy over processing speed. Among the parameters available in the LDA model, the "*alpha*" is the most important as it influences the number of topics that are associated to each term. For instance, a large value of alpha leads to many topics being associated to each term, whereas smaller values of alpha return fewer topics. In this particular approach, I let the LDA model chooses the best value of alpha so that the most suitable number of topics for each term is chosen accordingly.

```
# Call the Blei corpus
corpus_blei = corpora.BleiCorpus(os.path.join(MODELS_DIR, "corpus.lda-c"),
                                 os.path.join(MODELS_DIR, "corpus.lda-c.vocab"))

# Vectorise the corpus
tfidf = models.TfidfModel(corpus_blei) #, normalize=True
corpus_tfidf = tfidf[corpus_blei]

# Initialize the LDA model (ALPHA=1.0 / len(corpus))
lda = models.LdaModel(corpus_tfidf, num_topics=5, id2word=dictionary, iterations=100, alpha='auto', passes=10)

lda_corpus = lda[corpus_tfidf]

# Results overview
rand_num = random.choice(range(1,len(input_dict)+1))

print 'Which LDA(Blei) topic maximally describes a document?\n'
print 'Document format:           ' + str(input_dict[rand_num])
print 'Corpus format:             ' + str(corpus_blei[rand_num])
print 'Topic probability mixture: ' + str(lda[corpus_blei[rand_num]])
print 'Maximally probable topic: topic #' + str(max(lda[corpus_blei[rand_num]],key=itemgetter(1))[0])
print 2*'\n'


print 'Which LDA(TF-IDF) topic maximally describes a document?\n'
print 'Document format:           ' + str(input_dict[rand_num])
print 'Corpus format:             ' + str(tfidf[corpus_blei[rand_num]])
print 'Topic probability mixture: ' + str(lda[tfidf[corpus_blei[rand_num]]])
print 'Maximally probable topic: topic #' + str(max(lda[tfidf[corpus_blei[rand_num]]],key=itemgetter(1))[0])
print 2*'\n'
```

Step 2

Figure 48: LDA model parameters setting (red box). The first parameter to be defined is the corpus. Thereafter, there is the number of topics in which I want the LDA model to divide the corpus. The dictionary serves to keep a matching structure within words and LDA scores. Iterations and passes are the parameters which define how accurate the LDA results will be. Obviously, higher values return slower responses. In the blue box, the vectorization of the corpus is enabled via the TF-IDF model of Gensim.

With the most probable number of topics in text, I can assess the semantic for the four topics in the corpus. The assessment of the semantic is a subjective task due to the fact that it has to be extracted from the most popular terms present in each topic. It is interesting to report the presence of some extreme words such as “amsterdam”, “netherlands” and “noordholland” which are generated by Wi-Fi connections like in the Schiphol case investigated in Paragraph 6.3 (Figure 34, Table 14). Those words are removed from the semantic analysis to improve the understanding of each topic (Figure 49). The results of the manual topic semantic assignment is then displayed in Table 16 below.

```
# Print topics and the highest probability words
extremes = ['amsterdam','netherland','noordholland']
print
print "Print topics and the highest probability of words in topics", '\n'
for ti in xrange(lda_blei.num_topics):
    words = lda_blei.show_topic(ti, 10)
    tf = sum(f for f, w in words)
    print "Topic #%d" % int(ti), '\n'
    print('\n'.join('  {:>}: {}'.format(int(100. * f / tf), w) for f, w in words if w not in extremes))
    print
    print
```

Figure 49: Python recipe to extract top terms of topics. Highlighted in red is the part of the script that excludes extremes from the list of terms.

Topic ID[66]	Top 10 terms in topics (LDA)	LAS labels
Topic #0	Thank, day, great, beauty, go, one, best, time, nice , holiday	***Good times***
Topic #1	Pic, museum, redlightdistrict, canal, hard, old, eat, omg, na, eve	***Sightseers***
Topic #2	Sleep, please, bulldog, many, stop, bed, ask, win, coffeeshop, sit	***Smocking time***
Topic #3	Love, happy, year, new, good, people, like, nice, still, fun	***New Year visit***
Topic #4	Schiphol, airport, holland, christmas, station, central, home	***Schiphol traffic***

Table 16: Topic semantic assessment - Terms which are assigned to topics support the identification of the latent meaning behind the topic. In this case the latent meaning is identified by the LAS label. Hence, the label Sightseers refer to words such as pic (pictures, photos) museum, red light district, canal, hard (for hard rock café) which are all words attributable to a sightseers (or tourists) visiting the city. Bear in mind that the label is manually chosen hence the outcome is rather subjective.

Table 16 shows the approach I used to determine LAS labels from the top LDA terms. Because of the generative approach used by the LDA algorithm, the way how terms and their order are linked to topics is slightly different at each run, however, the meaning remain similar as described in Paragraphs 4.2.7 and 5.5.1.5. Therefore, to argue the assignment of labels I make use of example. For instance, I assign the **"*Good times*"** label to Topic #0 given that the majority of terms refer to nouns such as "day", "time" and adjectives like *"beauty"*, *"great"*, *"nice"*. In contrast, I link Topic #1 to the label ***"Sightseers"*** because of terms such as *"pic", "museum", "canal", "omg"* which suggest actions, places and expressions that tourists could do while visiting the heritage of a city. This approach is used to assign a label to each of the topics. Bear in mind that in LDA, as previously suggested in Paragraph 5.5.1.6, the label is manually chosen by the author on the basis of the top terms, hence the outcome is rather subjective. The label is in important variable in the identification of the LAS in the next Paragraphs.

As for validating the results of the LDA algorithm, I also assess the most discussed topic in the corpus (Figures 50 and Table 17) and this should mention words such as Schiphol and/or airport due to the high frequency of SOF, and tweets, detected at Schiphol Airport (See Paragraph 6.3, Figure 34 and Table 14).

[66] The LDA algorithm indices begin from 0, therefore for 4 topics the range would be 0 to 3.

```
# MOST DISCUSSED TOPIC IN CORPUS
# We first identify the most discussed topic, i.e., the one with the highest total weight
# First, we need to sum up the weights across all the documents
weight = np.zeros(lda_model.num_topics)
for doc in mm:
 for col, val in lda_model[doc]:
      weight[col] += val
max_topic = weight.argmax()
words = lda_model.show_topic(max_topic, 10)  # Get the top 25 words for this topic
print("Top 10 features in Most Discussed Topic (MDT)"), '\n'
print('\n'.join('  {:>}: {}'.format(int(100. * f / tf), w) for f, w in words if w not in extremes))
print
print
```

Figure 50: The most discussed topic in corpus – a Python recipe to compute the sum of the highest scores assigned by LDA to terms (.argmax( )).

Top terms in Most Discussed Topic	
Terms	**Scores (%)**
Schiphol	9%
Airport	9%
Holland	6%
Christmas	5%
Station	4%
Central	4%
Home	4%
Topic name: ***Schiphol traffic***	

Table 17: Top terms in the MDT in corpus. As indicated also via SOF analysis the airport is where the highest values of frequency of tweets are located.

The findings obtained via the Python recipe in figure 50 are computed using the scores assigned to each term by the LDA algorithm. The scores are normalized using the percentage of the sum of the scores (Table 17). Indeed, the topic with higher scores refer to Schiphol and the pattern of words being identified is very similar to those observed in posts collected at Schiphol airport (Paragraph 6.3, Table 14). Interesting to notice is the word Christmas which is recurring in many tweets together with *"happy"*, *"new"* and *"year"* terms like it occurs in Topic #3: ***"New year visit"***. This is of course explained because of the time of data collection during December.

Next for validation matters, I visualize the similarity of scores using a 2D Cartesian plane to see how the similarity between scores look in space. This is done by using the vector scores assigned by the LDA model to text data in combination with the LSI model to reduce the corpus dimensionality to two dimensions (x, y), which can be clustered via K-Means according to the SSE (i.e. the average sum of squares errors).

(LDA + TF-IDF) + LSI (3)

At this stage, the vectorised TF-IDF corpus is processed by using the LDA (3) model, implemented through Gensim library in Python, which transforms Twitter posts into vectors by considering the similarity of frequencies of all words in each post. The LDA returns a multi-dimensional array of scores (i.e. a *n*-dimensions for how many unique *n*-words the post contains) assigned to each post, one is the real score and the other is the estimated score, both assigned to each and every word of the same post (Figure 51). In this way, a corpus with

high-dimensionality is created and it not possible to display it on a 2D plane unless a dimensionality reduction model is implemented.

```
# project to 2 dimensions for visualization
lsi = models.LsiModel(corpus_tfidf, id2word=dictionary, num_topics=2)

# write out coordinates to file
fcoords = open("vct/topic_coords.csv", 'wb')
for vector in lsi[corpus_tfidf]:
    if len(vector) != 2:
        continue
    fcoords.write("%6.12f\t%6.12f\n" % (vector[0][1], vector[1][1]))
fcoords.close()
print(2*'\n')
```

```
In [5]: 0.01733,0.03324
   ...: 0.90416,0.34471
   ...: 0.02865,0.08696
   ...: 0.00112,0.00539
   ...: 1.15291,-0.02292
   ...: 0.00622,0.01775
   ...: 0.06136,0.28590
   ...: 0.72640,0.21758
   ...: 0.73485,0.21888
   ...: 0.09401,0.10074
   ...: 0.00272,0.00108
   ...: 1.86434,0.19694
   ...: 0.02082,0.06872
   ...: 0.03306,0.04309
```

Figure 51: LDA (TF-IFD) to LSI corpus transformations - The result is written into a csv file (extract on the right) to be used for clustering purpose in the next step.

The Latent Semantic Indexing (LSI) (3) is then implemented in order to decompose the LDA TF-IDF vectorised corpus into a 2D array which can be fitted afterwards into a K-Means algorithm (4) for clustering (Figure 51). The K-Means groups similar scores together on a Cartesian space and it returns by default, cluster labels and cluster centroids.

K-Means (4)

The K-Means algorithm is implemented in Python through the Scikit-learn library as described in Paragraph 4.4, to accomplish two purposes: the first is to assess the number of clusters that best fits the vectors of the LDA corpus being passed with regards to the SSE (See Paragraph 5.5.1.5). The second is to validate graphically the outcome of the LDA scores distribution. The implementation is rather easy as shown in Figure 52, and the model returns mainly three results: labels, centroids, and of course, division of clusters. Labels and centroids are deployed in legend to improve the readability of the resulting scatter plot. The distribution of topics are plotted in the scatter plot and vector similarities can be examined (Figure 54).

```
NUM_TOPICS = 4

X = np.loadtxt("vct/topic_coords.csv", delimiter="\t")
kmeans = KMeans(NUM_TOPICS).fit(X)
y = kmeans.labels_
centroids = kmeans.cluster_centers_

print "KMeans labels: ", list(set(y))
print "KMeans centroid: x{'\n'}  y{'\n'}".format(centroids[:,0], centroids[:,1])
print
```

Figure 52: Python recipe - Clustering the 2D vectors on a Cartesian space via K-Means algorithm

```
Topic clustering: LDA-LSI-KMeans pipeline (started at Thu May 21 21:59:31 2015)
Clustering the 2D space with KMeans algorithm: Inertia of the cluster

Notes:
The inertia of the cluster is defined as the sum of squared differences of each point to its cluster centroid
(This value is provided directly from the Scikit-Learn KMeans algorithm).

The elbow of the inertias is:  5

Clustering data with MiniBatchKMeans(batch_size=500, compute_labels=True, init='k-means++',
        init_size=100, max_iter=100, max_no_improvement=None, n_clusters=5,
        n_init=3, random_state=None, reassignment_ratio=0.01, tol=0.0,
        verbose=0)
```

Figure 53: Result of Python run showing the assessment of the Elbow and the parameters used in the K-Means algorithm.

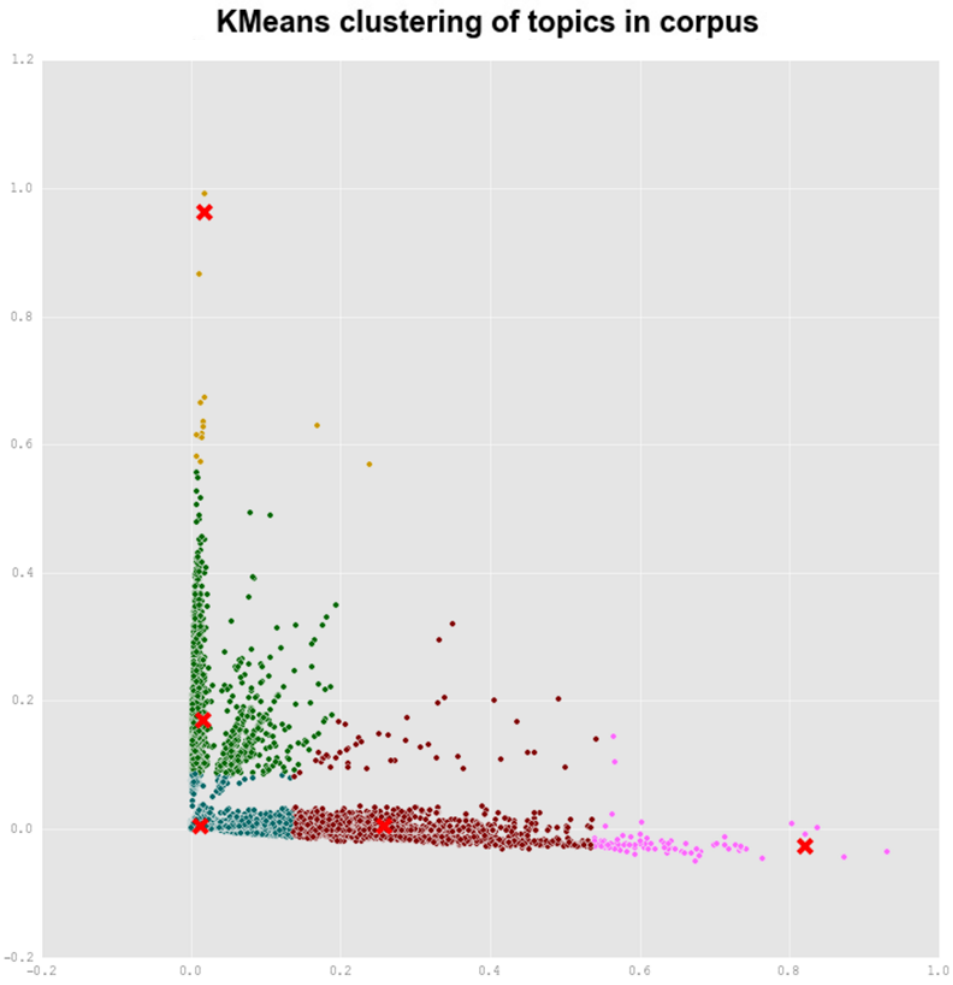


Figure 54: Scatter plot showing the spatial similarity of posts in terms of their computed vectors. The axes represent the scores of the LDA algorithm.

The scatter plot illustrates the spatial aggregation of similar scores into the 5 clusters, which best fits their spatial distribution as returned by the Elbow. In addition, the plot is interpreted by means of the distance among clusters: clusters spatially close denote similarity of topics, whereas cluster spatial distant denote dissimilarity of topics. The distribution of scores on the axes shows a dense character in mainly four topics, whereas the yellow topics shows a sparse character which makes it an outlier topic. The finding of the scatter plot verify the evidence of strong similarity in four out of five topics retrieved with the combination of K-Means and LDA methods. The outlier topic probably occurs due to the text noise that is still

present in the corpus such as the presence of words for which I did not find the way to remove completely.

In the end, the findings of the four methods are combined and fetched into a CSV dataset. This is later transformed into a point feature class via PostgreSQL to be processed in ArcMap. This is part of the last step of the LAS approach in which LAS labels which are assigned to tweets, are spatially located in the urban context of Amsterdam and their occurrence is used to assess the LAS in the service areas of landmarks.

6.4.5. LAS results

In order to divide the LDA corpus in the number of clusters found, I use the average of LDA scores as a reference to compare each score attribute with the mean score threshold value, grouping posts by score similarities. In this way, I obtain a partition of the corpus into a number of different clusters each of which contains similar posts labelled accordingly to the threshold (Figure 55). Note that since I iterate over the list of posts as many times as the cluster number (5), a post can be assigned to one or many (max 5) clusters, resulting in a number of multi-labelled post attributes (Table 18).

```
# Find the threshold, let's set the threshold to be "sum_scores/lenght_scores"
# To prove that the threshold is sane, I average the sum of all probabilities:
scores = list(chain(*[[score for topic_id,score in topic] for topic in [doc for doc in lda_corpus]]))
threshold = sum(scores) / len(scores)
print len(scores)
print "Setting the threshold as clustering baseline through mean probabilities of the topics"
print "Threshold score (mean): %.6f" % float(threshold), '\n'


tagged_docs = defaultdict(list)
print("Document cluster: assign labels to the clustered posts")
print

for scores, post in zip(lda_corpus, corpus_txt):
    for cl, score in scores:
        if score > threshold:
            tagged_docs[post].append(str(cl))
```

Figure 55: Python recipe - Classify posts in corpus by similarity of scores compared to the threshold (mean score)

To make use of the label assigned to each cluster, I need to assign it also to each post so that I can analyse it later during the LAS step. The number of posts included in each LAS label found are shown in the Table 18 below. In the Table the combination of topics is depicted and it is clearly shown the vast presence of the *"Sightseers"* label which was manually assigned to the topic #1 as reported in Table 16 above.

LAS labels	Multi_label	N. of posts
Sightseers	***1***	***17599***
Sightseers - Good times	***1-0***	***6290***
Sightseers - Good times - Smocking time	***1-0-2***	***1204***
Sightseers - Good times - New Year visit	***1-0-3***	***16***
Sightseers - Good times - Schiphol traffic	***1-0-4***	***164***
Sightseers - Smocking time	***1-2***	***15022***
Sightseers - Smocking time - Schiphol traffic	***1-2-4***	***1114***
Sightseers - New Year visit	***1-3***	***2625***

Sightseers - New Year visit - Smocking time	***1-3-2***	***359***
Sightseers - New Year visit - Schiphol traffic	***1-3-4***	***12***
Sightseers - Schiphol traffic	***1-4***	***5479***

Table 18: LDA topic subdivision. Assessing the size of each topic cluster

The scores divide the corpus of posts into 11 different clusters as shown in Table 18. However, the majority of posts are included in only two cluster, those related to topic 1 (***Sightseers***, Table 16) and those related to topics 1 and 2 (***Sightseers + Smoking time***, Table 16). The multi-label distribution of topics is an interesting outcome due to the rather short nature of posts in Twitter. However, this is probably explained by the presence of hashtags in many posts which in some case are linked to different topics. Hence this leads to a multi-label classification of the posts in the dataset. Through the use of Pandas library in Python I analysed the proportion in which posts in topics are distributed on a bar graph and pie chart displayed in Figure 56 and 57, respectively.

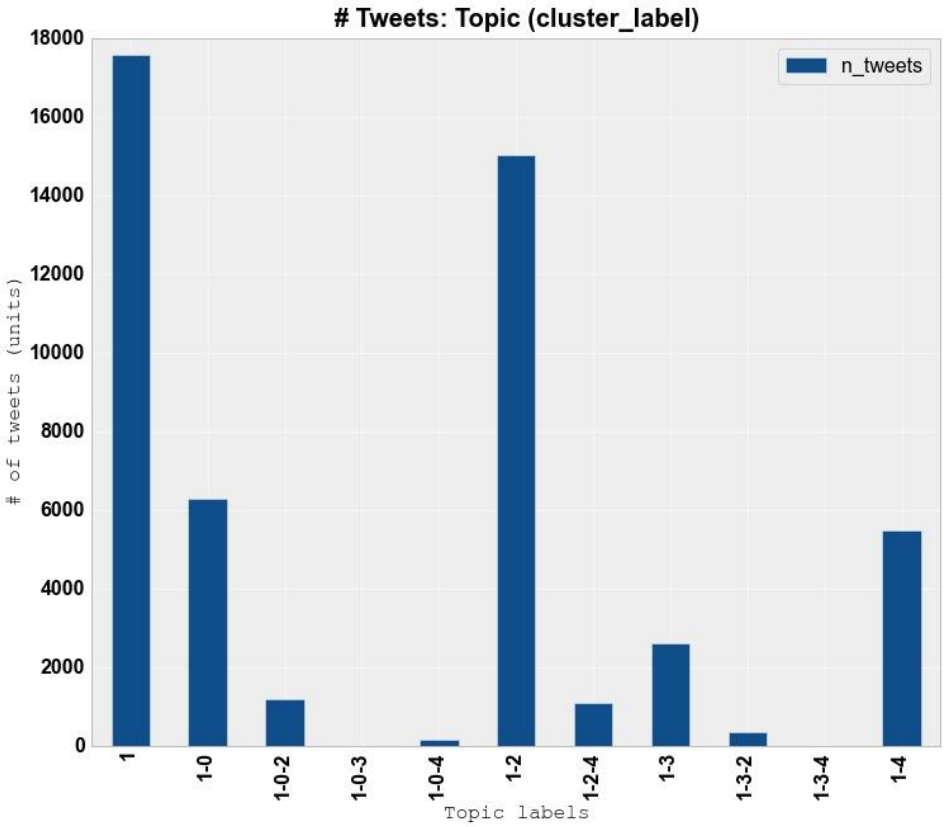


Figure 56: Distribution of topics over the corpus. Mainly, four homogeneously distributed topics are obtained. Multi-label topics contains very few posts.

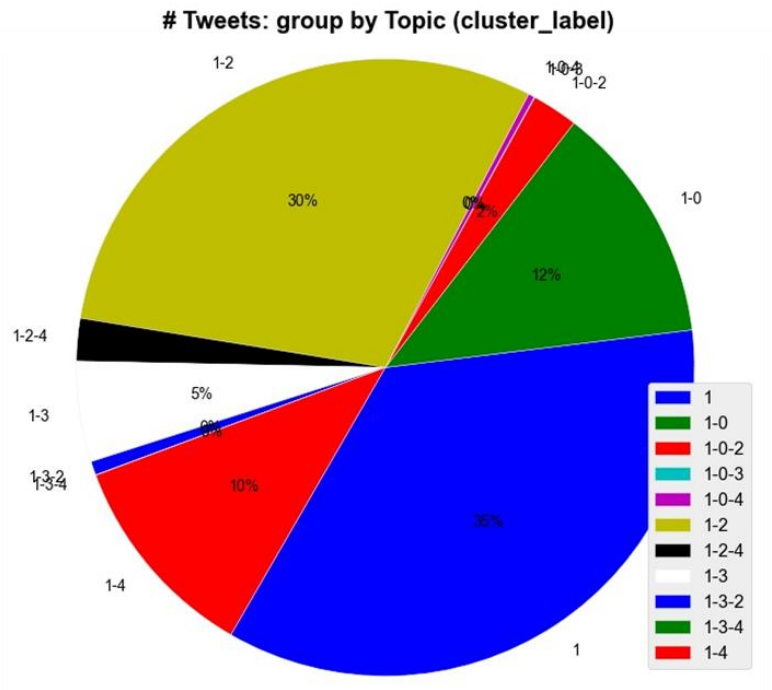


Figure 57: Pie chart showing the distribution of post frequencies for each LAS label, considering the whole of the corpus.

To perform the LAS analysis, service areas of landmarks and attractiveness values are calculated in ArcMap. The former is enabled through Thiessen polygon creation tool via ArcMap, whereas the latter is computed via Spatial Join tool performed on the service area just created. (See Paragraph 6.5, Figure 61). Once, the variables for the computation of the density are available I calculate the densest service areas in terms of the number of tweets within it. For each service area, I compute the frequency of each LAS label so as to assess which is the most popular label or labels discussed within each touristic landmark. The LAS labels are selected and counted in proportion to the total number of occurrences within each service area. By doing so, I obtain a local semantic assessment of the most probable topic found within each service area. The results are as follow:

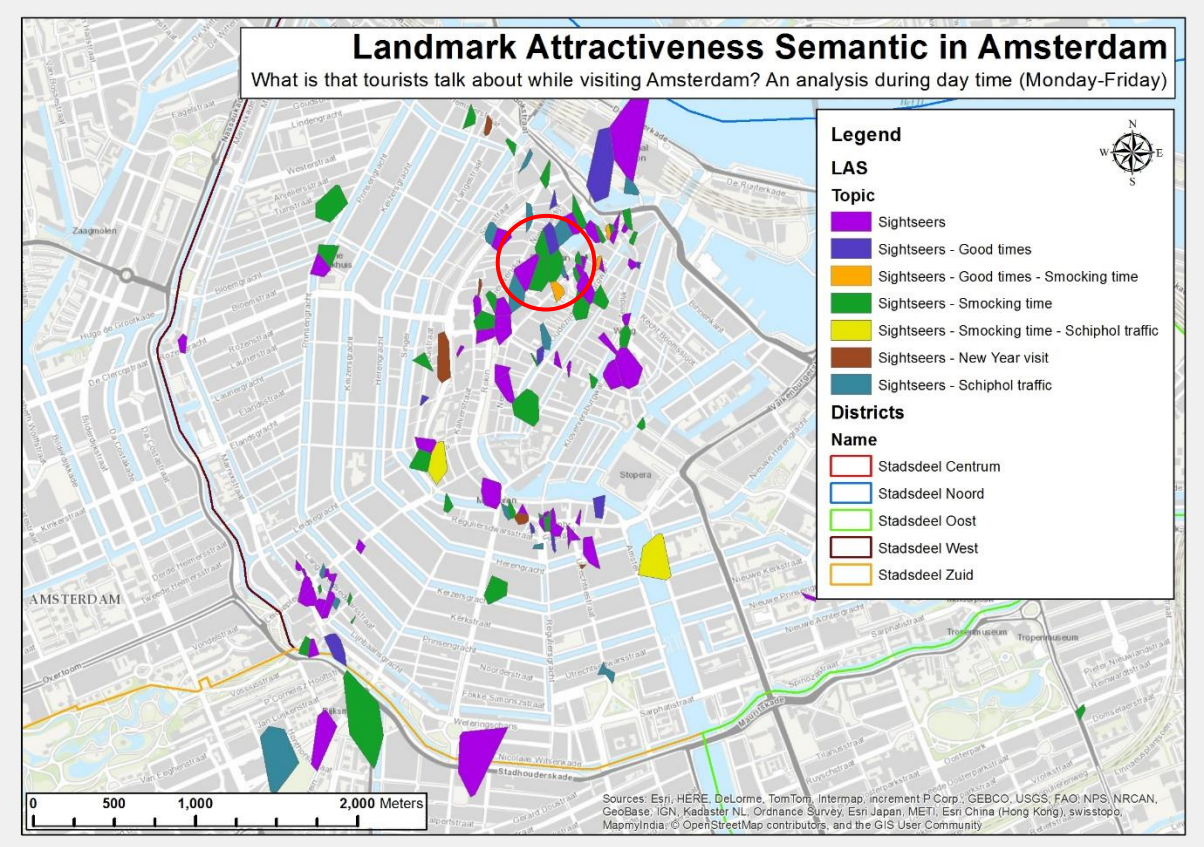


Figure 58: LAS analysis in Amsterdam. The distribution of topics onto the most attractive touristic venues (LAE above the mean) during day time. According to Figures 56 and 57 the most popular topic is Sightseers (purple).

The map shows the service areas of touristic venues above the average values of LAE. The purple polygons are the venues in which "Sightseers" topics are found. Interestingly, those are located in mostly in proximity of museums and important landmarks such as Heineken experience, Anne Frank house, Trippenhuis, Dam square and so on. The distribution seems very heterogeneous with some clusters forming at specific locations. For instance, the cluster highlighted (red circle) shows a combination of *Sightseers* and *Smoking time*, and it is located in proximity of the coffeeshops area. However, these results are strictly dependent by the space subdivision, hence the more accurate the list of PoI is, the better the accuracy of the LAS result. Moreover, due to the small scale of visualization and zonal[67] character of the LAS, it is rather difficult to compare the time at this scale. Therefore, I select a random area to be able to display in more detail the differentiation of LAS results in relation to the time (See appendices I and J).

[67] With "zonal" I mean that the LAS is strictly influenced by the area of study. For instance, in an area in which the majority of services have an entertainment use, in that case the LAS would return topics like "good time".

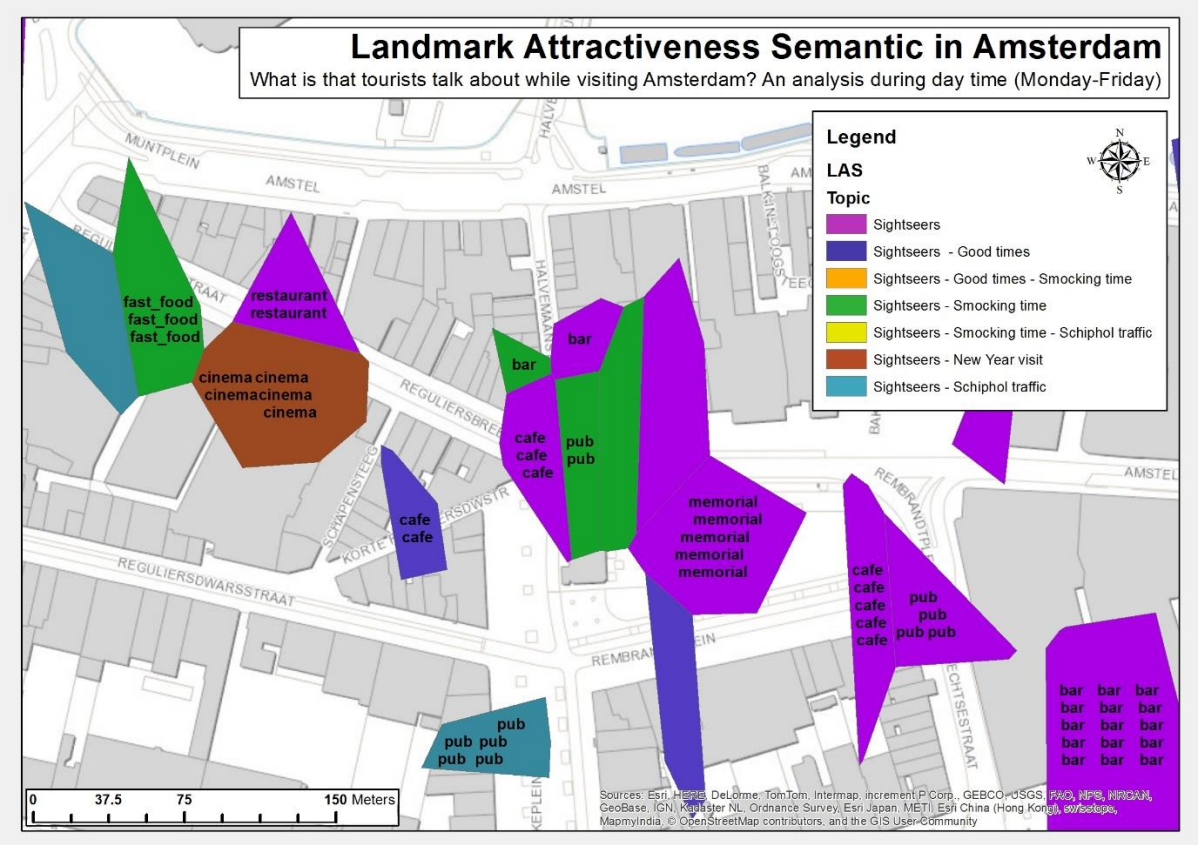


Figure 59: LAS analysis in Amsterdam. The distribution of topics onto the most attractive touristic venues (LAE above the mean). The map shows a detailed representation of Rembrandtplein during day time.

In Figure 59 a detailed extract of the area of Rembrandtplein is extracted from the map above (Figure 58). At this scale, it is also possible to check the use of the service areas above the mean value of LAE in relation to the most popular topics being discussed during the day (in this case). The majority of topics are related to general touristic activities as it is suggested by the *sightseers* LAS label. It is possible that tourists visiting the square, decided to take a break and sit in one of the many café and pubs in the areas.

It is important to say that the area of Rembrandtplein has a mixed character according to the time of the day. In fact, during the day the square is a touristic venue visited due to the popularity and because it is in proximity of many other popular attractions, such as the Flower market and the Munt tower. During the night, the land use of the area changes to rather entertainment given that a large number of leisure activities such as discos, nightclubs, coffeeshops, clubs and so on open to customers. In Figure 60 which depict the analysis of the LAS during night time this can be seen.

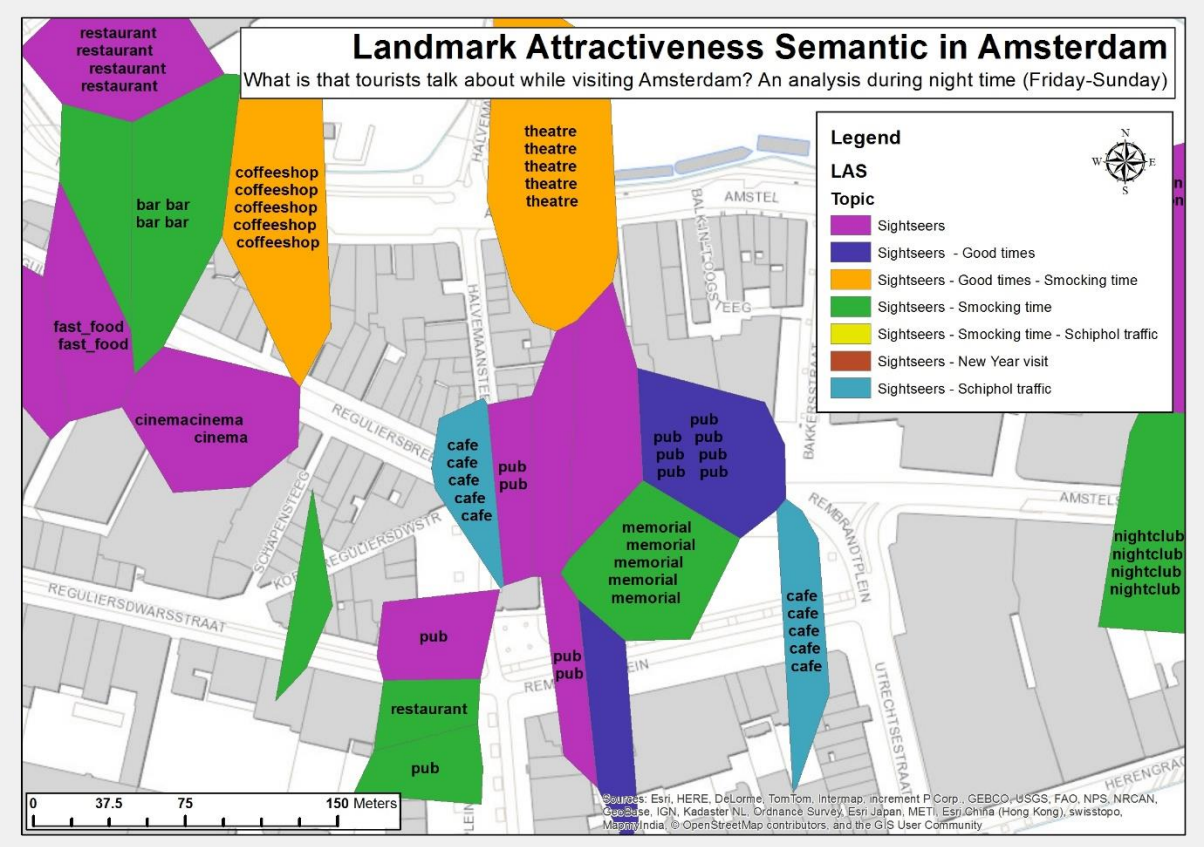


Figure 60: LAS analysis in Amsterdam. The distribution of topics onto the most attractive touristic venues (LAE above the mean). The map shows a detailed representation of Rembrandtplein during night time.

Although the majority of services show a general character as shown by the Sightseers LAS label, some of the areas change their semantic to a more entertainment one. For instance, the Sightseers - Smocking times and Sightseers – Good times - Smocking times labels begin to appear. Interesting, it is the change of LAS from day to night in the centre of the square. If during day time the most discussed topic was Sightseers which mostly denotes the action of visiting a place, during the night the LAS changes to Sightseers - Smocking times given the large number of tourists “having fun” while sitting on the benches of the square.

In conclusion, the LAS is the end product of the TAUS GKD method through which I was able to assign LAS labels to each of the topics obtained via LDA algorithm implementation. A discussion upon the obtained results is reported in Chapter 7: Discussion. Next, the results relative to the measure of accessibility of touristic venues is argued in the Paragraph below.

6.5. Accessibility of touristic venues (4)

This Paragraph shows the results obtained in the Space partition and Time decomposition, Landmark Attractiveness Estimation (LAE) and Network Impedance (NI) processes and it combines them to evaluate the accessibility of touristic venues of Amsterdam in space and time.

Landmark Attractiveness Estimation (LAE)

The frequency of tweets inside each service area being created is given by the implementation of the Spatial Join tool in ArcMap. The tool counts each unique attribute linked to the ID of each Thiessen polygon and, and it returns the count of tweets though which I can assess the density (Expression field) that I compute in a new field called LAE (Figure 61)

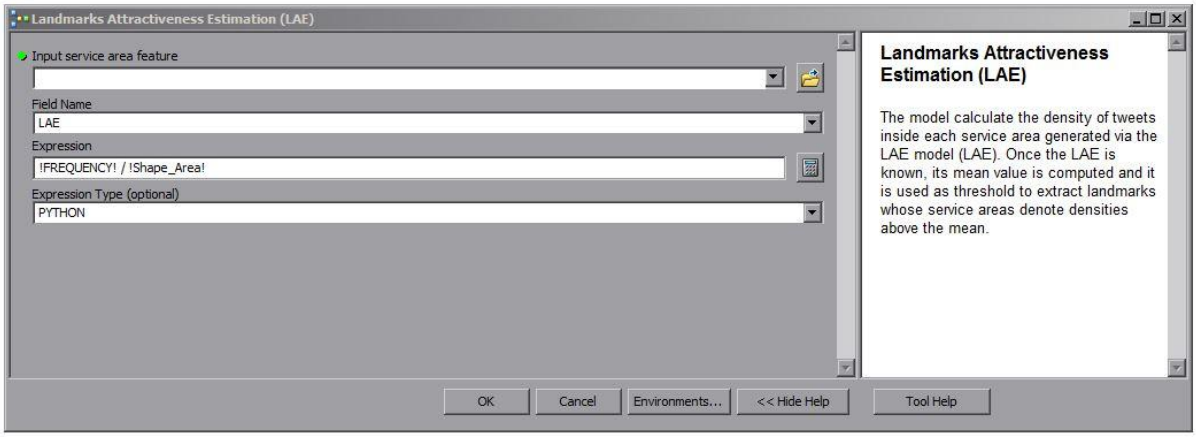


Figure 61: Estimation of the density of LBSN activities for each landmark – LAE.

At this stage, I am interested to display the landmarks in which the concentration of Twitter activities, hence number of touristic posts in the considered periods, is higher than the average. This is a good approach to close the scope of the analysis by focusing on the areas in which the density is considerably high. To do so, I consider all those landmarks whose density is above the mean value of the frequencies obtained as displayed in Figures 62 and 63 below.

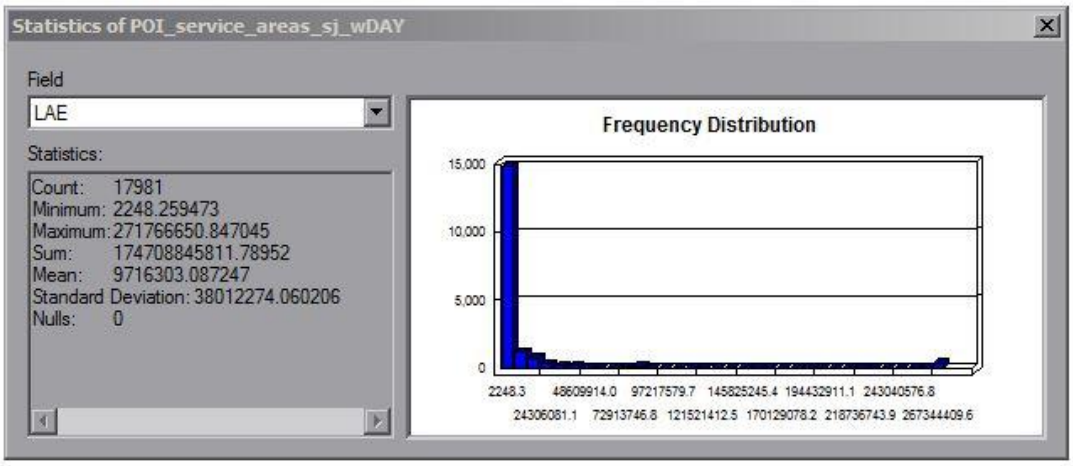


Figure 62: Statistics over the computation of the LAE retrieved during the day time from Monday till Friday.

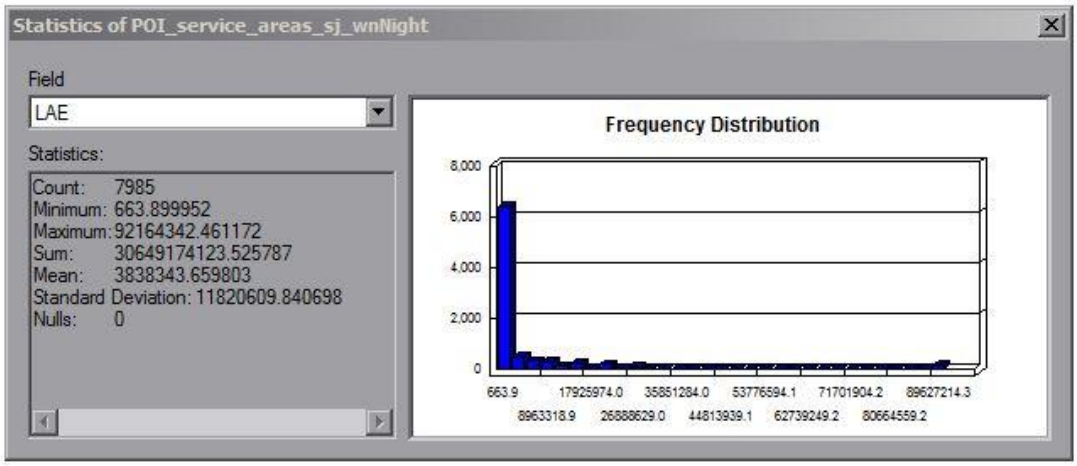


Figure 63: Statistics over the computation of the LAE retrieved during the night time from Friday till Sunday.

This is repeated, with same regards as above, for both datasets created to display the distribution of tweets in each time period. Therefore, I produce two separate maps: one displaying the distribution of tweets during day time (6h-18h) in week days from Monday to Friday, and the other showing the distributions during night time (18h-6h) in weekends (Fridays-Sundays). It is interesting to see the frequency distributions of the statistics generated via ArcGIS (Figures 62 and 63 above). A peak is present in both visual representations and that is related to the open Wi-Fi services located in Dam square. Actually, in that tight amount of space, the percentage of touristic activities, such as the historical buildings, museums, shopping facilities and restaurants is very high. This is confirmed by the consideration of the time component. In fact, all these activities are open to the public in the selected time period, hence tourists crowd the area all day and night returning such distribution (Figures 64 and 65).

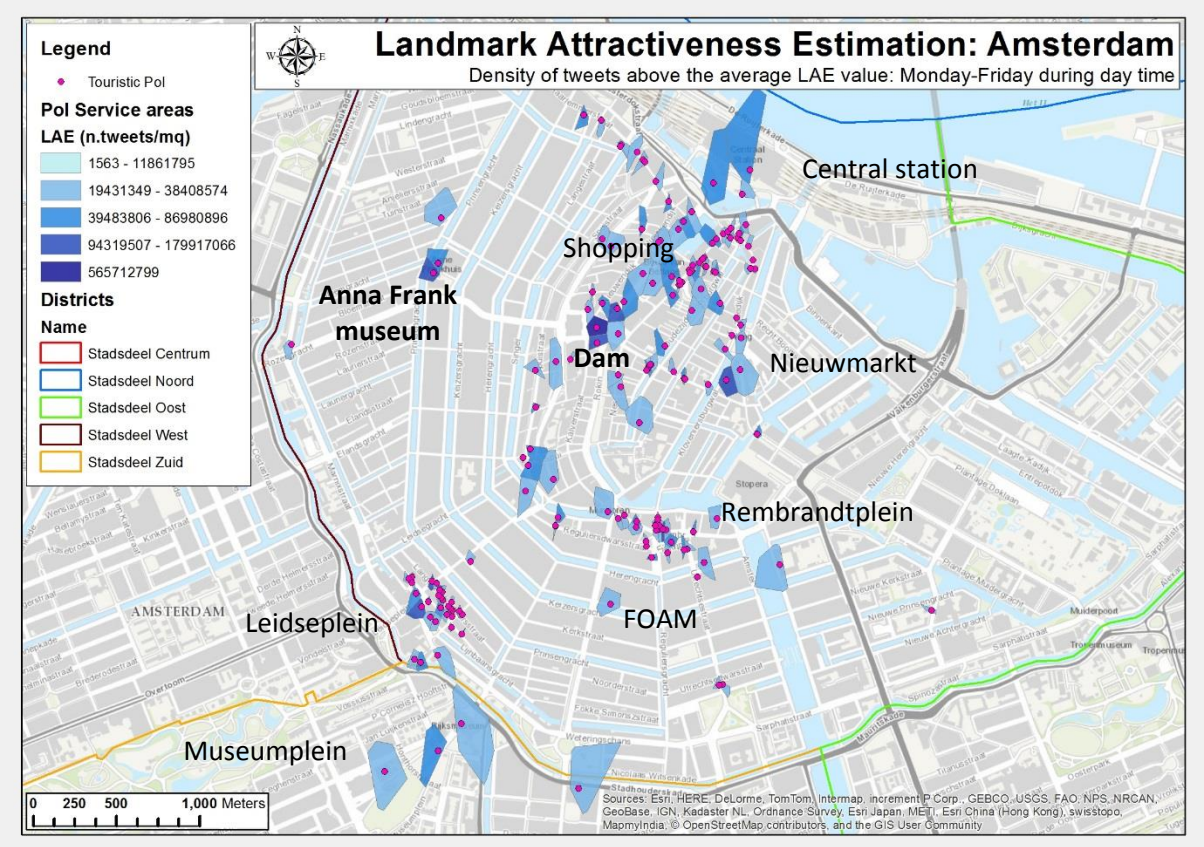


Figure 64: LAE obtained during week days in day time. Densities are distributed mainly in the city centre. Peaks are shown at Dam Square and at the Anne Frank Huis museum. The value of the LAE is a density ratio expresses as number of tweets inside the service area (See appendix G).

The map shows a heterogeneous distribution of LAE results during day time. The highest aggregations are located in the city centre of Amsterdam around the area close to the King's palace and more on the west side, in proximity of the Anne Frank House museum. Those are among the most popular touristic attractions in Amsterdam. High densities are also seen in the shopping street as well as in the proximity of the central station, and in the Nieuwmarkt area. High frequencies of attractiveness are shown across the whole of the city centre due to the fact that all activities are open. Therefore, tourists distribution of tweets are seen almost everywhere. However, worldwide popular attractions such Anne Frank House museum still draw the attention of the majority of tourists in Amsterdam. Also the wax museum, namely Madam Tussauds[68], in Dam square seems to attract many tourists during day time, probably because of its presence in all major cities worldwide.

Having said that, these findings become clearer when I consider the weekend time period spanned during evenings and night hours (Figure 65). By then, the majority of the activities (i.e. all but nightlife businesses such as restaurants, bars, pubs, discos, coffee shops and red light district) are closed. As shown in Figure 65, here the distribution of attractiveness above the mean is higher in the areas where a wide range of touristic venues linked to nightlife activities exists. Leidseplein, Rembrandtplein, Nieuwmarkt, Red Light District are among the most popular entertainment areas. These areas are packed of touristic venues

[68] Madam Tussauds, Amsterdam: https://www.madametussauds.com/amsterdam/en/

running particularly during night time such as discos, pubs, fast food, theatres, coffee shops, sex and so on.

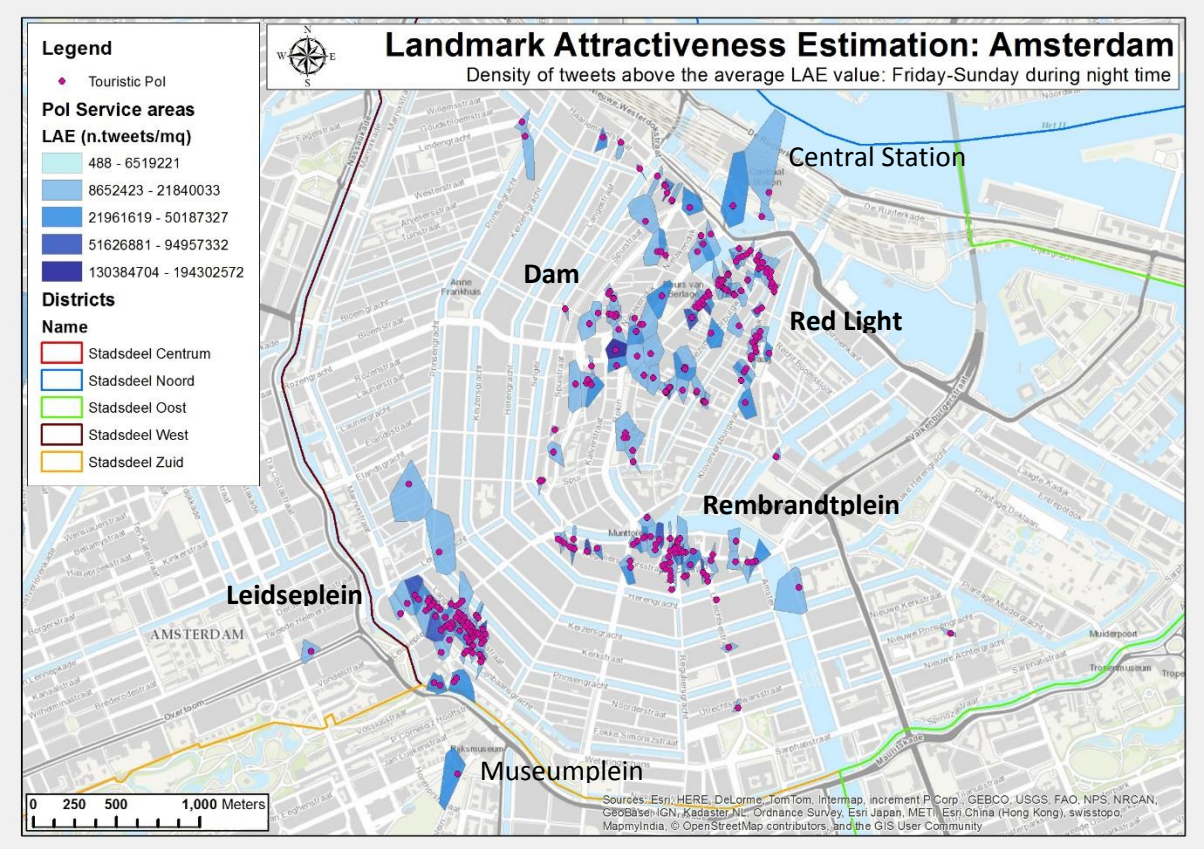


Figure 65: LAE obtained during weekend in night time. Densities are distributed mainly in nightlife areas such as the Red Light District, coffee shops area, Rembrandtplein and Leidseplein. A peak is still shown at Dam Square due to the presence of Wi-Fi services that allow tourists to connect (See appendix H).

The findings show interesting patterns linked to the type of activities present within specific areas. High values of attractiveness I_i are spotted in Dam Square in which particularly during the Christmas period is often busy with events and exhibitions that attract many tourists and not. Several touristic landmarks in areas mainly covered by leisure activities such as the Red Light District, Rembrandtplein and Leidseplein, also show relatively high attractiveness. In proximity of Central Station noticeable values of I_i are depicted. Actually, after 1 am the regular transit system switch to the night service and the majority of busses run from there without transiting in the city centre. Therefore, everyone has to reach the station in order use the night bus service. Last location that needs further description is the service area which appears at the museum district. As previously shown during the data analysis Chapter, in that area in December there was the ice skating facility in place. Thus, also that area is interesting to this research as it clearly shows the presence of a touristic activity that attracts many people, both during day and night time, as shown in Figure 64 and 65. Thus, the time appears to be a very sensitive variable which should be divided and analyzed with caution. Indeed, with a different decomposition of time most of the patters revealed so far could have remained unknown.

Next step towards the completion of the objectives of this work is the assessment of the network impedance in terms of the distances that separate a transit stop to a range of touristic activities in the surrounding. The calculation is argued in the next section.

Network Impedance (NI)

The network impedance, as argued above is the measure of the distance from a transit stop to a variable range of landmarks in proximity. This attribute represents the willingness for tourists to walk in order to reach an activity or a range of activities. I assume that tourists who travel by public transport across Amsterdam are willing to walk an average distance of 500 meters (approx. 5-10 min walking) as argued in assumption 6. The parameters used in the model run are displayed in Figure 66 below.

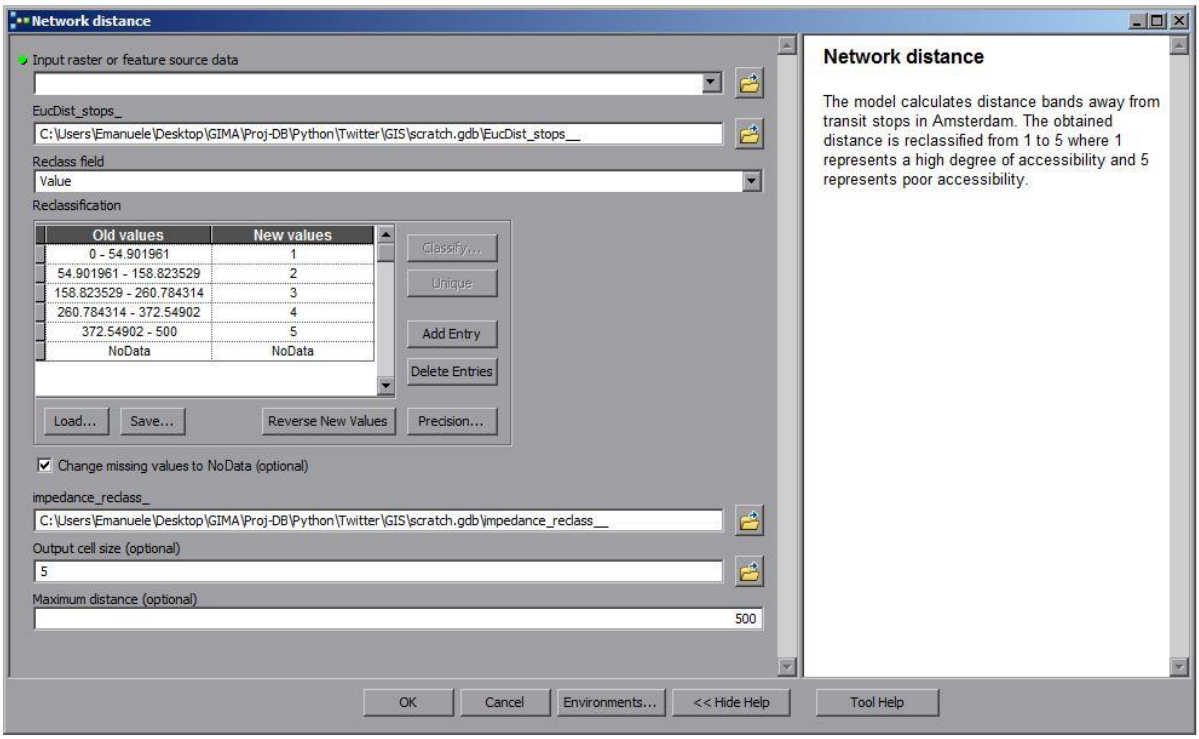


Figure 66: Network impedance model run. The set of parameters needed by the model to perform the computation of the buffer areas. The reclassification tool is also included to transform the values obtained in the raster into 5 classes. Those are used to assess the measure of accessibility of touristic landmarks retrieved above.

The model requires an input feature which is represented by the network of transit stops retrieved from OPENOV, and the threshold distance upon which to calculate the service areas of transit stops. The model returns a raster data layer in which transit stops are the centre of multi-buffer features holding the distance value at each raster cell (Figure 67).

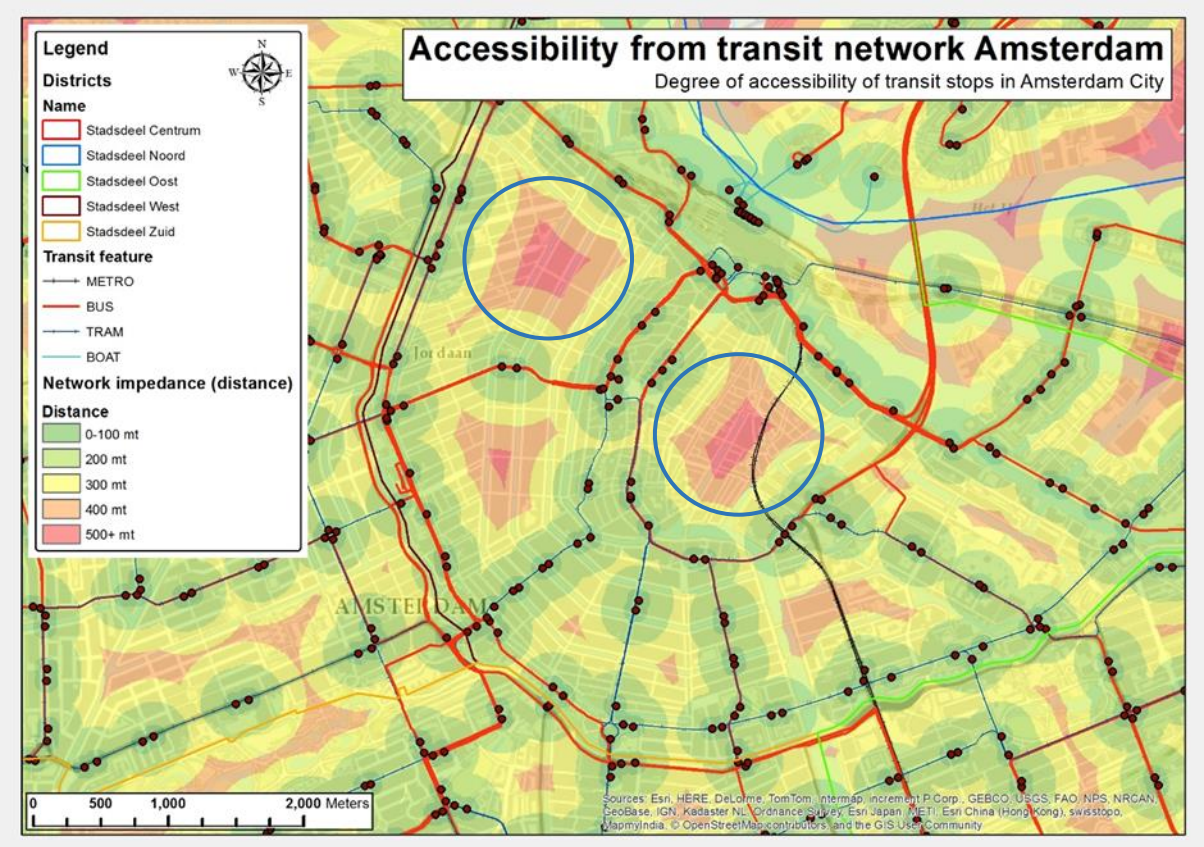


Figure 67: The output result of the Network impedance model in ArcMap. First the Euclidean distance tool computes the distance (500 metres) away from each transit stop in terms of a multi-buffer raster feature. Then the Reclassify tool generate a classification of the multi-buffer in a range from 1 (highly accessible areas) to 5 (poorly accessible areas). See appendix D for a larger image.

Not surprisingly, noticeable poor accessibility measurements are seen in proximity of the old district of Amsterdam whose structure of the urban texture does not allow the location of transit stops anywhere close. Mainly, two area denote poor accessibility: one is detected in the Jordaan district nearby the Anne Frank House museum and the other coincides with the popular touristic area of Nieuwmarkt (blue hollow rings).

To be able to establish a classification range upon which the accessibility of touristic venues can be evaluated, the output raster feature is reclassified using a class range from 1 to 5 which represents the ***Accessibility Index***. The index assigns a value of 1 for highly accessible areas (i.e. within 100 metres) due to their proximity to the transit stop. Whereas an index class 5 is assigned to areas in which the accessibility to touristic venues is rather poor due to the increased distance to be walked (i.e. equal to or greater than 500 metres). In this regard, the touristic landmarks obtained in the previous stage, whose LAE was above the average, are deployed to assess their degree of accessibility. Indeed, the accessibility of touristic venues should be assessed for all those landmarks that are visited by a large number of tourists given that those represent areas most at risk of congestion and overcrowding.

Accessibility of touristic venues

The measure of the accessibility of touristic venues in Amsterdam is the last step of the analysis performed in this work. The landmarks denoting high values of LAE are

considered at this stage because of the high frequency of touristic posts, hence number of tourists that they report. However due to the scope of this work, a more in depth analysis is performed solely on those touristic landmarks with high values of LAE that are located in poorly accessible areas. In Figure 68, the service areas of touristic landmarks denoting LAE above the average (displayed in shades of blue) are overlaid on top of the reclassified raster obtained at the end of the Network Impedance computation. The measure of the LAE refers to the density ratio of the number of tweets inside each service area. The network impedance layer shows the buffers related to the distance to be walked and the Accessibility index, which ranges from 1 (within 100 metres walking distance) to 5 (500+ metres walking distance) displays the degree of accessibility of touristic venues whose LAE is above the average.

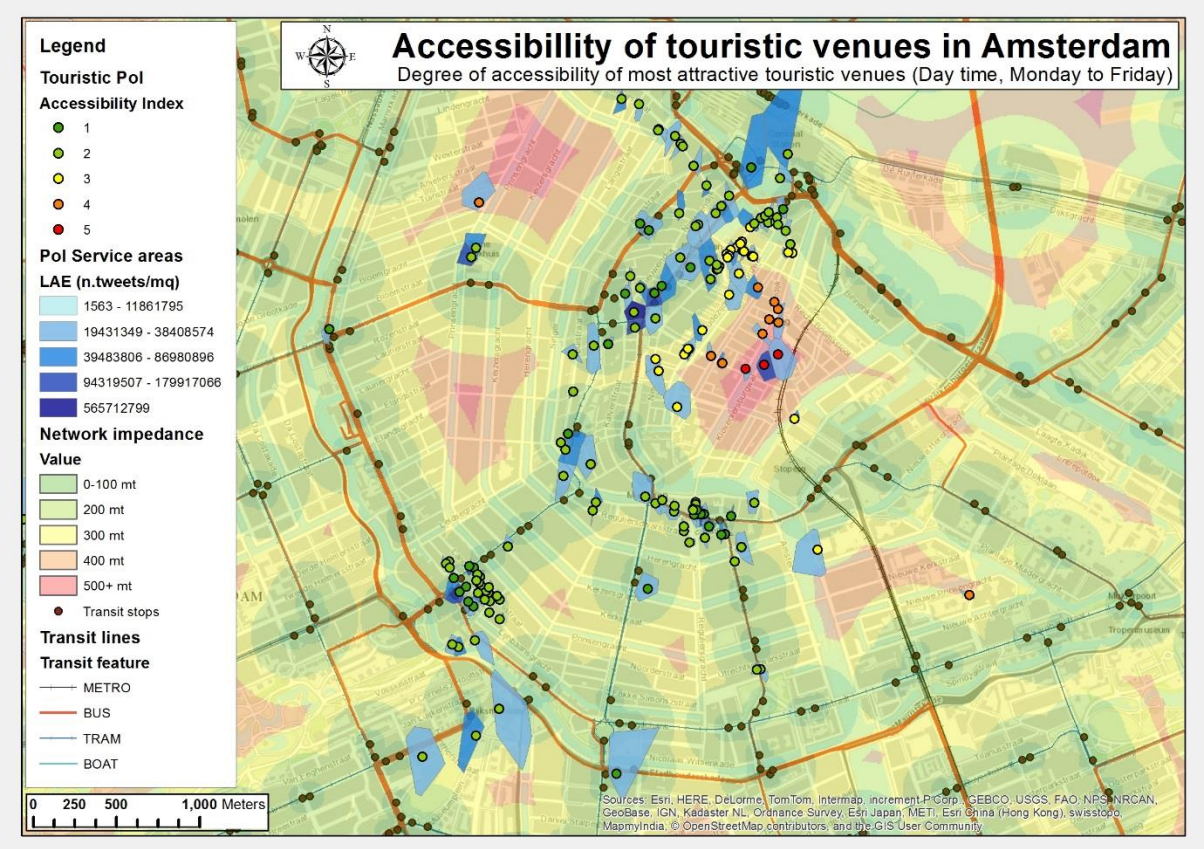


Figure 68: Accessibility of touristic venues in Amsterdam during week days in day time. The accessibility classes are assigned to landmarks that show high values of LAE, hence high frequency of people gathering in proximity.

The map above shows the assessment of accessibility relative to the time period that includes week days (Monday-Friday) during day time hours (6h-18h). As shown in the map, a group of touristic attractions showing high values of LAE are located in a poorly accessible, yet very busy area of Amsterdam: Nieuwmarkt. The range of activities is wide with theatres, museums and monuments as well as café, restaurant and other nightlife attractions. In particular, that area is in proximity of other two popular touristic attractions for which Amsterdam is famous worldwide to be the capital of freedom: the Red Light District (RLD) and the coffeeshops street (Warmoestraat). Those are among the oldest areas in Amsterdam in which the urban texture does not allow public transports in the surroundings. In Figure 69 below, a detailed map of the area is extracted to report the assessment of the accessibility in more detail.

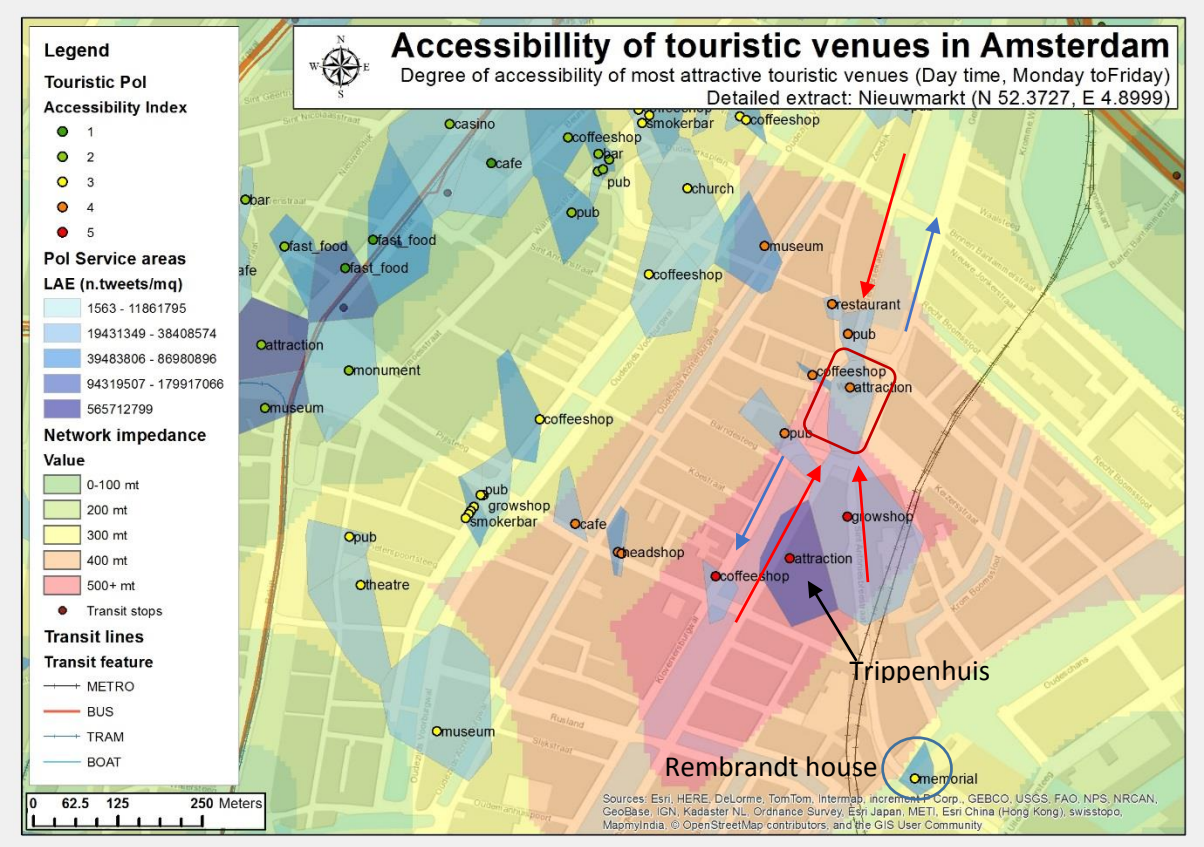


Figure 69: Detail of the accessibility in the Nieuwmarkt area (Week days during the day). The accesses to the area are limited and the structure of the street network in combination to the crowds of tourists visiting the area could cause problems of congestion and overcrowding. Blue arrows are the road exits from the area, while the red ones are the road entrance to the area.

The area it is not served by trams or busses due to the limited amount of space available. The access to that area is enabled by many streets and alleys for pedestrians, but if cars and roads are considered, there exists mainly three ways to access the area (i.e. red arrows) and two ways to exit it (i.e. blue arrows). Either way follows the street around the square (dark red square). This is a clear example of area in which overcrowding and congestion may arise due to the large number of tourists that reach the area to visit the nearby touristic attractions such as the Rembrandt house museum[69] and the Trippenhuis[70], particularly if I consider the local road traffic. In fact, cars could be slowed by crowds of tourists crossing streets, taking pictures and so on, therefore urban planners should carefully evaluate the dynamics happening in this area and provide solutions to relieve the effect of overcrowding and congestion.

[69] Rembrandt House museum: http://www.rembrandthuis.nl/

[70] Trippenhuis: http://www.iamsterdam.com/en/visiting/what-to-do/attractions-and-sights/places-of-interest/trippenhuis-trip-house

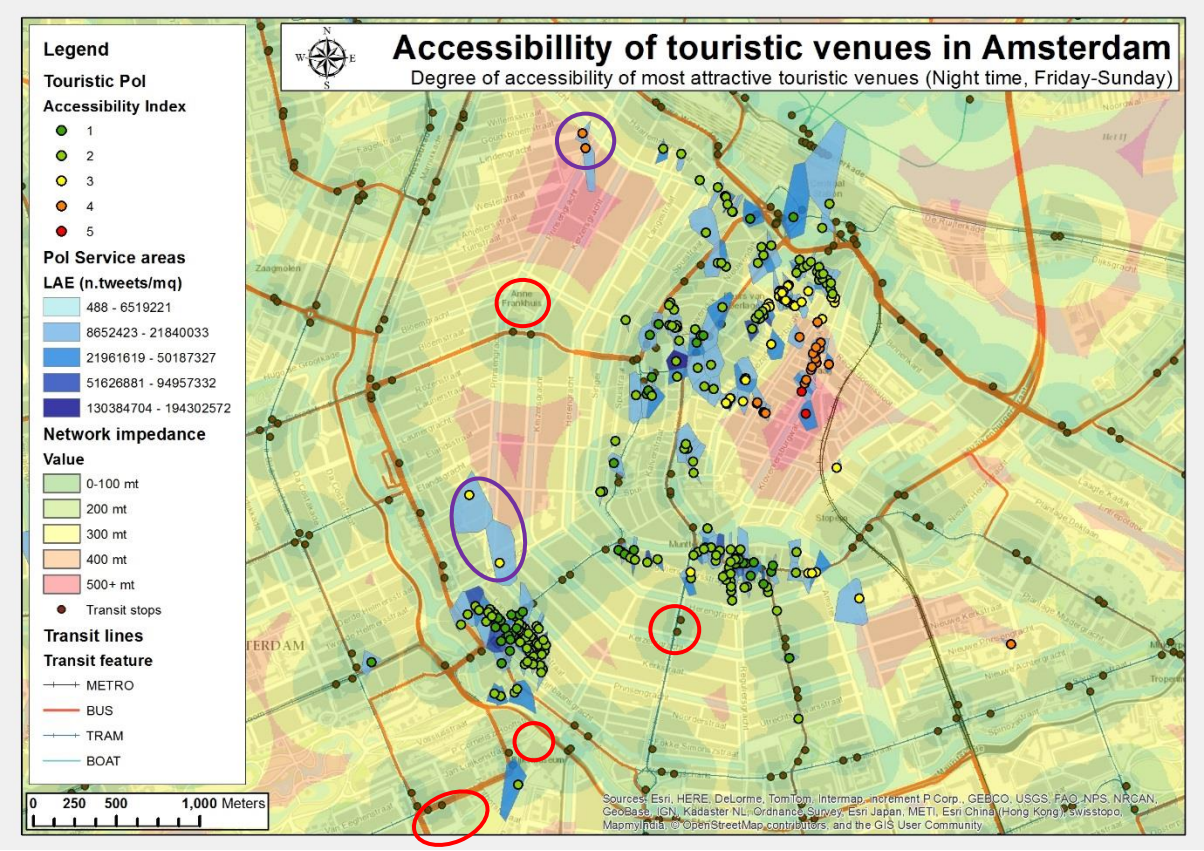


Figure 70: Accessibility of touristic venues in Amsterdam during weekends in night time. The accessibility classes are assigned to landmarks that show high values of LAE, hence high frequency of people gathering in proximity.

In contrast, Figure 70 shows the assessment of accessibility relative to the time period that includes weekend days (Friday-Sunday) during night time hours (18h-6h). The resulting map looks similarity to the previous accessibility map displayed above probably at glance. However, when I look more in detail, there are a number of interesting patterns I noticed that need further argumentation.

First of all, although the value of the LAE is lower than the one obtained during day time, a wider number of touristic activities particularly those related to nightlife appear on map. This can be seen in the areas of Leidseplein and Rembrandtplein, in which the concentration of nightlife activities is higher. Also in Nieuwmarkt a similar pattern, but in minor entity can be seen with a bunch of nightlife attractions appearing as above the average (Figure 71). Second, the effect of time can be clearly seen when I look at the museum district (Van Gogh, Stedelijk, and Rijks museums), the Anne Frank museum and the FOAM museum. During day time, all museum are shown as being above the average LAE mean. However, during night time these activity are closed, hence they are not displayed (Figure 67, red circles). Furthermore, a bunch of pubs and restaurants are revealed where there was nothing during the day as highlighted with purple rings.

To be able to see the effect of the time over the distribution of LAE I consider the area around Nieuwmarkt due to its poor accessibility index. I want to show the differentiation of the type of touristic activities that are obtained during night time in the area (Figure 69). By comparing the two detailed maps in Figures 69 and 71, it is possible to detect the changes in the type of landmarks with regards to the measure of attractiveness (LAE). In fact, Figure 71

shows a wider range of activities, most of which are exclusively open during night such as nightclubs, café and pubs. Therefore, the area is crowded of tourists during the day as well as during the night generating congestion and overcrowding. This information is helpful for urban planners and decision makers that need to assess the use of the city at different time of the day and different days of the week. (See appendices K and L)

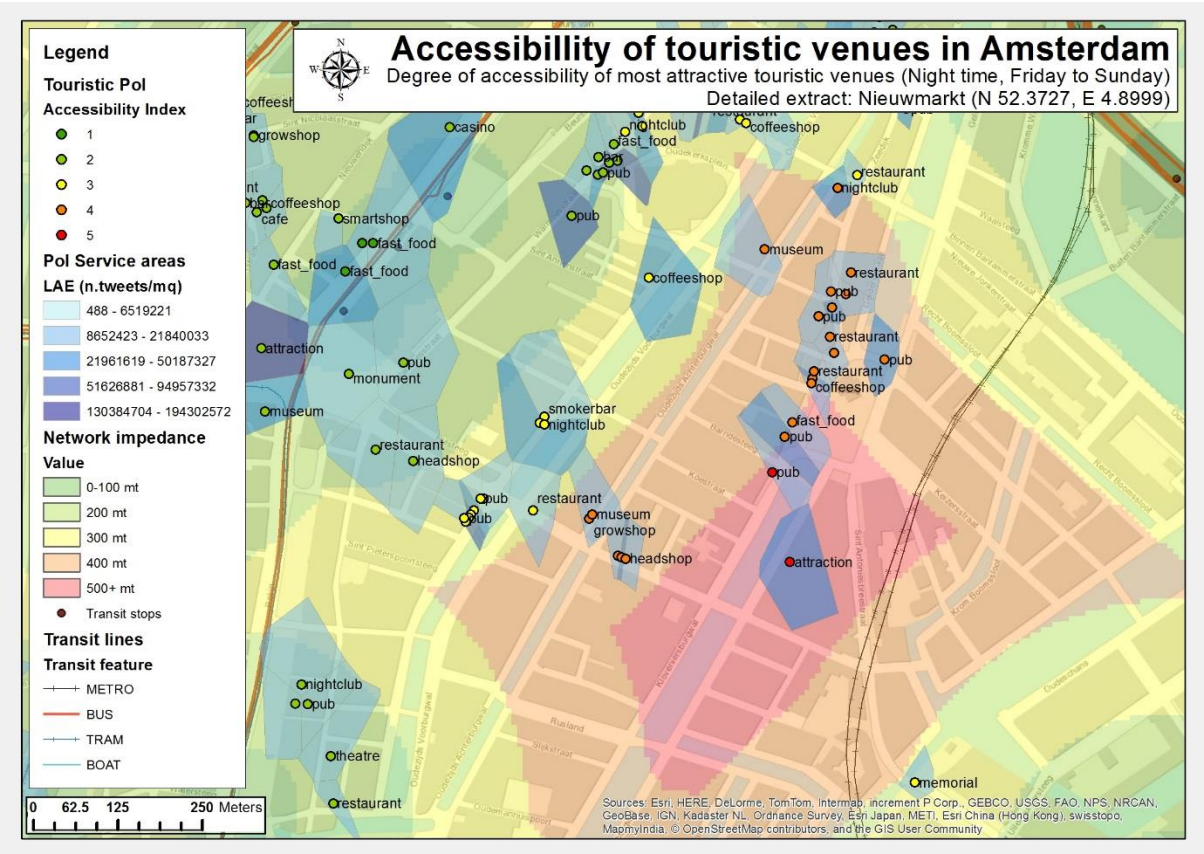


Figure 71: Detail of the accessibility in the Nieuwmarkt area (Week days during the day). The accesses to the area are limited and the structure of the street network in combination to the crowds of tourists visiting the area could cause problems of congestion and overcrowding.

In the last Paragraph, I describe the results of the validation approach that I implemented in order to confirm the findings of the attractiveness, hence accessibility of touristic venues in space and time.

6.6. HERE Transit: a validation approach

In this Paragraph, I performed an analysis of the data obtained by the HERE maps firm related to the usage of transit stops in Amsterdam in the period of December 2014. The dataset is obtained in the form of a CSV log and is processed via Pandas library of Python in order to extract the information needed. Figure 72 shows the cleaned dataset to perform the validation. The attribute data that are needed in this approach is the departure and arrival locations of transit stops for each requested route and the timestamp to select the same periods of time used in the computation of the accessibility. With those attributes data, I am able to compute the frequency of requests from the users of HERE as displayed in the FREQUENCY field (Figure 72).

here_validation

	lijnnr	plaats	haltenaam	Freq	X_DEP	Y_DEP	X_ARR	Y_ARR	LANG	TS	FREQUENCY
	369	Amsterdam	Amsterdam, Seineweg	6	4.81892	52.386538	4.84154	52.386639	nl-NL	2014-12-08T10:51:38	5
	61	Amsterdam	Amsterdam, Seineweg	28	4.81892	52.386538	4.84154	52.386639	nl-NL	2014-12-08T10:51:38	5
	69	Amsterdam	Amsterdam, Seineweg	62	4.81892	52.386538	4.84154	52.386639	nl-NL	2014-12-08T10:51:38	5
	369	Amsterdam	Amsterdam, Seineweg	6	4.81892	52.386538	4.84154	52.386639	nl-NL	2014-12-08T10:51:38	5
	61	Amsterdam	Amsterdam, Seineweg	28	4.81892	52.386538	4.84154	52.386639	nl-NL	2014-12-08T10:51:38	5
	69	Amsterdam	Amsterdam, Seineweg	64	4.81892	52.386538	4.84154	52.386639	nl-NL	2014-12-08T10:51:38	5
	369	Amsterdam	Amsterdam, Naritaweg	6	4.83728	52.38892	4.82091	52.388031	en	2014-12-15T08:34:10	2
	61	Amsterdam	Amsterdam, Naritaweg	28	4.83728	52.38892	4.82091	52.388031	en	2014-12-15T08:34:10	2
	69	Amsterdam	Amsterdam, Naritaweg	64	4.83728	52.38892	4.82091	52.388031	en	2014-12-15T08:34:10	2
	369	Amsterdam	Amsterdam, Naritaweg	6	4.83728	52.38892	4.82091	52.388031	en	2014-12-15T08:34:10	2
	61	Amsterdam	Amsterdam, Naritaweg	28	4.83728	52.38892	4.82091	52.388031	en	2014-12-15T08:34:10	2
	69	Amsterdam	Amsterdam, Naritaweg	62	4.83728	52.38892	4.82091	52.388031	en	2014-12-15T08:34:10	2
	21	Amsterdam	Amsterdam, Haarlemmerweg	85	4.84761	52.38501	4.866473	52.375693	nl-NL	2014-12-31T08:20:24	31
	21	Amsterdam	Amsterdam, Van Hallstraat	87	4.86986	52.386902	4.878083	52.358837	nl-NL	2014-12-18T12:17:40	18

Figure 72: The HERE log obtained after processing the original dataset. The location of departure stops and arrival stops for each requested trip are displayed. Language information are useful to investigate the language trends of the users of the app.

That is computed by first creating a 150 metre buffer feature around each of the transit stops from OPENOV. Then, by using the Spatial Join tool I can count the number of HERE requests occurring within the service area of each transit stop (the 150 metres buffer). With the frequencies in place, I can assess the HERE transit usage during day time (Monday-Friday) and night time (Friday-Sunday) following the same time partition used in the computation of the accessibility. The resulting maps are displayed as follow:

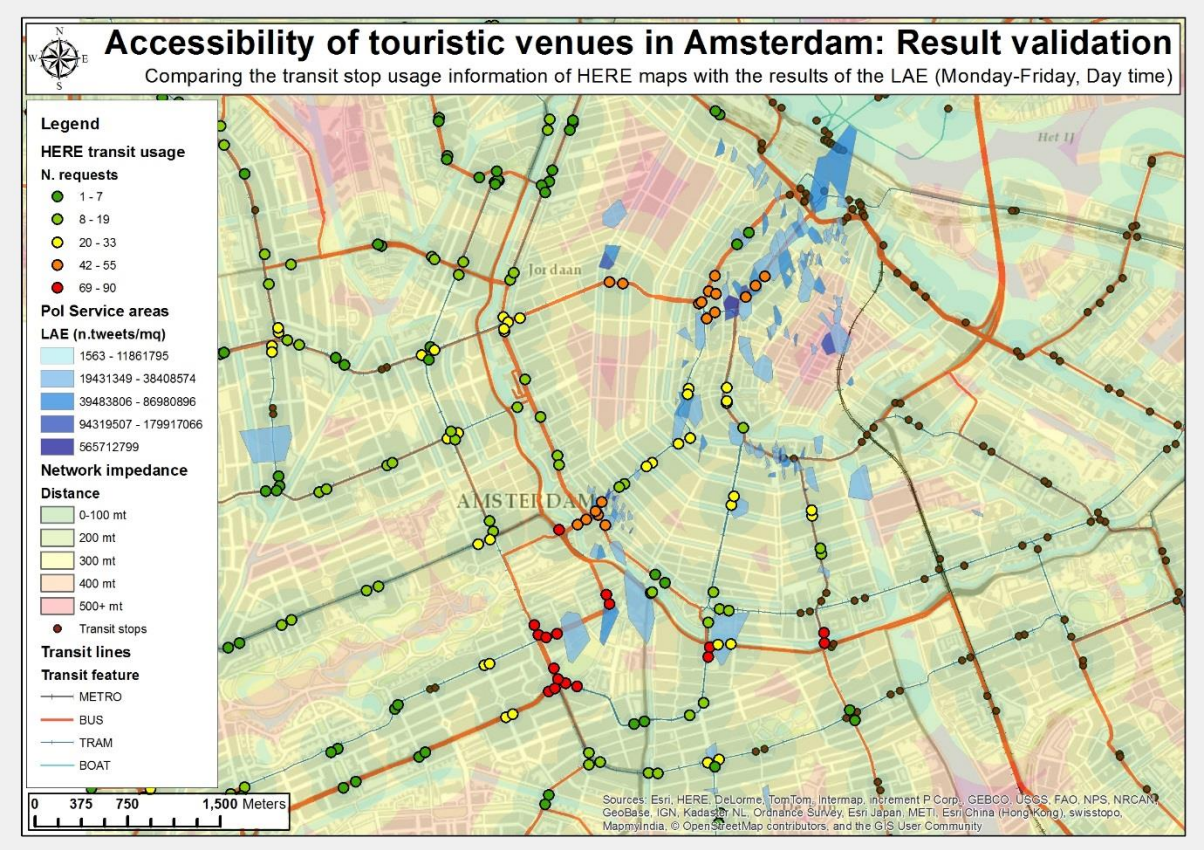


Figure 73: Transit usage trends in Amsterdam during day time on week days. The usage index goes from 1 (low usage) to 5 (high usage). This information support the evidence of congestion trends in the areas in which the measure of the LAE is above the average.

The map returns a visualization of the most congested stops with regards to the information extracted from the dataset of HERE. The usage index assigns an index of 5 to the areas in which the frequency of requested routes from HERE users is high (red) and 1 to the areas denoting low frequency of HERE requests (green). The vast majority of stops with high usage are indeed in proximity of areas in which the value of the LAE is high hence the validation is confirmed. In fact, busy areas such as Museumplein, Dam square, and the Anne Frank museum, attract large number of tourists hence the frequency of requests by users of the HERE transit app also shows high values.

This is entirely confirmed by the visualization of the same information of Figure 73, but during the night of weekend. During night time, the stop network usage moves towards the areas in which nightlife activities take place such as in Leidseplein area. Dam square area maintain the high frequency of requests overtime due to the mixed type of touristic activities that are available both during the day and night. Moreover, it is interesting to notice the decrease of usage in the areas that were rather busy during the day. For instance, three interesting changes are noticed in Figure 74: the "Museumplein" (red circle), "Stadhouderskade" (blue circle) and "Westermarkt" (green circle) transit stops report a considerable changes in the usage frequency in comparison to the results obtained during day time. This is explained by the fact that the land use of those areas is not very mixed, therefore, when the main activity is closed, the attractiveness of those areas drops.

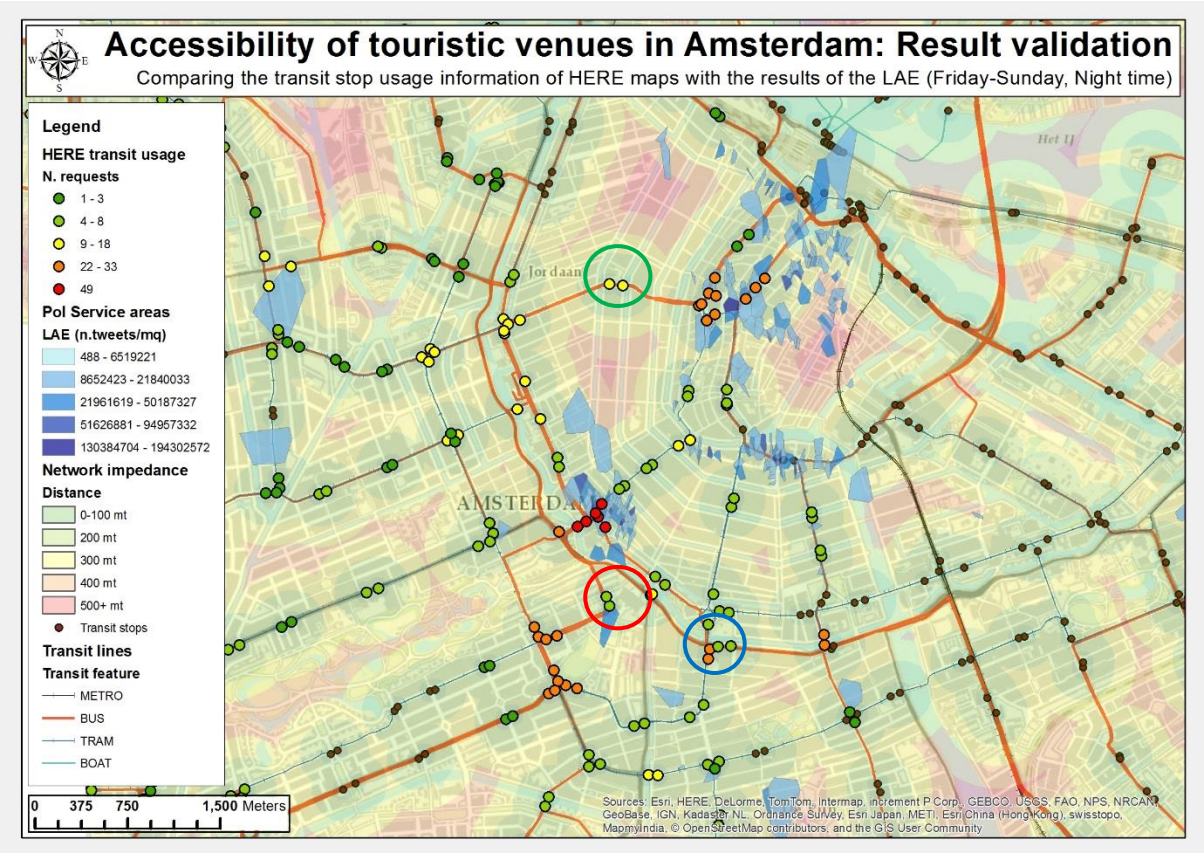


Figure 74: Transit usage trends in Amsterdam during night time of weekends. The usage index goes from 1 (low usage) to 5 (high usage). This information support the evidence of congestion trends in the areas in which the measure of the LAE is above the average.

The last remark is in the absence of usage trend information in the areas east of the city centre. For some reasons that I could not identify, the usage information relative to those areas is missing from the HERE dataset, therefore I could not evaluate the validity of this approach over there.

Following, in the last Chapter of this work, I discuss the conclusions over the application of this methodology. This is done by answering the research questions formulated during the explanation of the research objectives in the Chapter 2.

7. Conclusions and Recommendations

In this Chapter, I describe the conclusions relative to the research objectives and the obtained results. Moreover, a discussion over several disadvantages in the set of methods, techniques and tools being implemented is included in Recommendations. This will support future researches with limitations that could be addressed by means of different approaches. Conclusions are based on the research questions established in Chapter 2 and the results obtained are used to support and validate the provided answers.

7.1. Conclusions

The list of research questions established in Chapter 2 and the relative answers are displayed as follow. The research questions are represented in order to keep track of the elucidation for each of the questions addressed.

Q1. How can unstructured geo-tagged tweets be mined from the Twitter API and translated into meaningful information, to this study, in order to identify tourism aggregations in the City of Amsterdam and thereby assessing the degree of accessibility of touristic venues?

In order to identify tourism aggregation from Twitter data, three main attributes included in tweets are considered: the geo location, the post and the time components. Those are transformed into information by using a dedicated extracting method. This method uses techniques such as the SOF model and the FDA tool to detect and remove noise such as spam in the data. Thereafter, the method considers the post in order to extract touristic information from the data by means of the TAUS GKD method. The TAUS GKD is an ad hoc method which implements tools and techniques from the Natural Language Processing approach such as features extraction, text classification and topic modelling to extract information over tourism activities in Amsterdam. To be able to assess the attractiveness as well as the accessibility of touristic venues I collected the list of touristic venues as well as the list of transit stops in Amsterdam from OSM and OPENOV, respectively.

The information being extracted is used to calculate the aggregation of tourists in proximity of touristic venues in relation to the attractiveness that they generate. The proximity is defined as the service area of a touristic venue and it is calculated by means of space partitioning via Thiessen polygon tool in ArcMap. The attractiveness $\boldsymbol{I_i}$ is computed as the density of Twitter contents inside the service areas of those venues: the Landmark Attractiveness Estimation (LAE). The measure of the accessibility of touristic venues is given by the accessibility index. The index considers a set of distance bands (0-500 metres) from a

transit stop to touristic venues (i.e. network impedance) to evaluate the degree of accessibility of a specific landmark.

As the results in Chapter 6 depicts, the accessibility of touristic venues in Amsterdam is rather good, particularly in the busiest touristic areas of the city such as Dam square, Leidseplein, Rembrandtplein and Museumplein. In those areas, the transit network is present and close (i.e. within max 200 metres) to the major touristic activities. However, there are two major areas, namely Nieuwmarkt and Jordaan in which the transit network is not in the immediate surroundings due to the compactness of the urban context in those areas. Above all in the area of Nieuwmarkt, without a fast way to access (and leave) the site, this can be an issue that may lead to congestion and overcrowding phenomena, particularly if aggregation of tourists are combined with local congestion such as the road traffic. In fact, in that location many touristic venues have been identified as highly attractive, hence large number of tourists have visited the area posting Twitter contents from there.

Q2. How it would be possible to estimate the attractiveness of spatio-temporal touristic aggregation as well as their semantics (e.g. Why tourists gather in that specific area at that particular time?) of the behaviour of tourists in Amsterdam?

As explained by answering Q1 above, the measure of the attractiveness of touristic venues is calculated using the density of tweets through a technique called LAE. In order to enable this technique, I had to divide the urban space and the time following a specific schema. The space is partitioned by the Thiessen polygon implementation whereas the division of time enabled via a process that uses Python programming language, the time component of tweets and touristic information, regarding opening times of activities, collected from external web sites.

For the computation of the semantic of tourism aggregations, I created an ad hoc methodology, named TAUS GKD, which takes the post attribute as input and it returns a set of labelled topics assigned to each service area: the Landmark Attractiveness Semantic (LAS). The TAUS GKD implements a set of tools and model to normalize the Twitter post to a natural English text structure. Moreover, the method make use of techniques to extract touristic features using the hashtag index of Twitter. The extracted hashtags are used to classify the posts with the Naïve Bayes classifier tool. Thereafter, the method implements a generative model and a clustering techniques, namely Latent Dirichlet Allocation (LDA) and k-Means clustering algorithms, respectively, in order to find the number of K topic in the Twitter dataset (Elbow of K-Means) and generate as many topics via LDA as the value of the K. For each topic found, a textual label is manually assigned on the basis of top terms generated by the LDA for each topic. This approach is supported by the use of Python language and specific libraries, namely Gensim, NLTK and Matplotlib. In the end, the LAS approach returns a distribution of labels that can be used to evaluate which is the topic or concatenation of topics most discussed inside the service area of a touristic venue. The LAS is implemented to study the relation between the attractiveness generated by touristic venue, the urban context in which the attractiveness is found and the time at which it occurs. Therefore, with this method is possible to assess what attracts people to a particular area and if the time influence this reason.

The result is very dependent to the time and space partitioning choices made as well as to the subjectivity of the LAS labels manually established. Therefore, the results of this approach should be considered as an indication of the topic semantic hidden in the dataset of posts and not a precise definition of the topic. Indeed, the development of the research upon this matter is still at an early stage and it misses solid background references in some cases. Moreover, the documentation related to tools and models used to achieve this objective is not always clear and extensive given the early stage of research and the large amount of variables involved in the process. Thus, a range of issues found during the implementation of the TAUS GKD method is included in the following Paragraph to discuss upon possible recommendations.

Q3. What is the impact of the temporal component in the formation of aggregation patterns, hence in the measure of the accessibility of touristic venues, and what approach can be used to evaluate this impact (e.g. time of the day, day of the week to display spatio-temporal hotspots)?

The impact of the temporal component represents a powerful approach to identify changes in the location of tourism aggregations and in the behaviour of tourism. In this approach, the day and night time is evaluated in relation to the week days from Monday to Friday and the weekend days from Friday to Sunday. This type of temporal decomposition is set on the basis of touristic information over opening times available on web sites and it differentiates touristic activities typically open during day time like public attractions, museum and touristic services and those mainly operational during night time such as clubs, pubs, discos and so on. The results displays changes overtime whether a touristic attraction may be open or not as argued in the Museumplein area during the Wi-Fi identification task (the ice skating). In that case, during night time, the attractiveness generated by museums located in that area was absent. However, high frequency of Twitter activities were spotted on the lawn where the ice skating facility which was operational in the month of December.

Hence the result of the time analysis brought to light the existence of a strong relation between the time and the attractiveness of touristic venues. This is clearly depicted by comparing Figure 71 showing the distribution overnight in the weekend and Figure 69 which shows the diurnal distribution of tweets. In the former, despite the lower number of tweets considered in the dataset, a wider range of touristic PoI was detected. In addition to that, this approach also gives the indication of the type of use that an area has overtime showing which range of activities is operational at what time.

Q4. What validation approach can be used in order to support the measurement of the accessibility, and verify whether the tourism aggregation is linked to transit usage trends (e.g. is the attractiveness of touristic venue(s) somehow connected to the volume of users at transit stops within a threshold distance from touristic venues?)

To perform the validation of the method being deployed, I used transit information relative to the usage of the transit stops in Amsterdam which was provided in log format by HERE maps. Here is international leader in the field of maps and navigation services and it has competitors of caliber such as Google, Tom Tom and Apple. The use of HERE transit data was preferred to the use of Dutch routing engines such as 9292ov, due to the degree of internationality that the HERE firm denotes in terms of its customers.

The stops usage is extracted by the log dataset and decomposed into the same intervals of time considered before. The outcome is merged in the attribute table of network stops as frequency value (i.e. number of requests within a 150 metres buffer). The result is used to validate the measure of the LAE (over the same time period) so as to verify whether transit stops in proximity of highly attractive touristic venues showed high frequency of usage. This was confirmed during the comparative analysis of Figures 73 and 74. In either period of time, the measure of the transit usage was directly proportional to the measure of the attractiveness. Therefore, the accessibility of touristic venues is also indirectly validated for this particular approach given that the LAE is the main variable used to select the touristic landmarks upon which the accessibility is calculated.

In light of this research, I consider the usage of Twitter as a potential source of data to support urban studies. By implementing a consistent approach to be able to detect and discard noise, and an accurate selection of the specific features to extract, it is possible to use the social network as urban sensor. A sensor through which it is possible to derive information from the behaviour of its users in space and time. The results obtained proof that a link between the attractiveness generated by a touristic venue and the urban context exists given that topics linked to the type of activities were detected in different areas of Amsterdam (e.g. clusters of “smoking times” topics in proximity of the coffeeshops area).

Having said that, as argued in the answer to Q2 the current research over the extraction of semantic from the post attribute still denotes some limitations in terms of text cleaning and normalization when noise has to be removed. Moreover, the LDA algorithm and its implementation on LBSNs is still at an early stage and improvements are still needed in order to create an accurate fit between the algorithm and the data. Noticeable is the effort offer by the Gensim library that enables the implementation of the algorithm in Python for large corpora without incurring in memory problems due to the large amount of processed information.

In the end, the use of the temporal attribute represents also a major strength due to the fact that it enables the study of the 4th dimension. Through the time, I was able to study the way tourists use the city, and how the attractiveness of touristic venues as well as tourism activities change overtime. In the next Paragraph, I discuss the recommendation that are needed in order to improve the outcome of this and future research. Actually, among the limitations considered some could represent the topic for future studies in this field.

7.2. Recommendations

In this last Paragraph, I describe a number of drawbacks that I have come through along the process of research. The nature of the issues is explained and a brief view of the author is argued so as to raise discussions for future studies on the limitations observed. For clearness, the limitations are divided by the research process they have been found in order to show which of the processes used is the most critical in terms of further developments. Hence four sections similar as they appear in methodology are discussed:

1) Data collection and processing (space and time partitioning);
2) Data analysis (SOF+FDA)
3) Tourism information extraction & topic semantic (LAS)

4) Accessibility of touristic venues (LAE+NI)

Data collection and processing

During the collection of data from external sources such as Twitter contents via the Streaming API and the list of PoI from Metro Extract API of OpenStreetMap, I have come across two main issues. The first is represented by the functionality of the filter offered by the Streaming API of Twitter, the second is the quality and accuracy of the source data being provided by OSM.

Actually, the filter of the Streaming API is enabled by the Tweepy Python library, however, the filter of Tweepy is based on the functionalities of the one of Twitter hence they are equivalent. The limitation noticed is in the concatenation of two types of attributes at the same time. In fact, it is not possible to concatenate, with the use of Boolean operations, more than one filter parameter per time. For instance, it is not possible to intersect words tracking AND location at the same time so as to produce an intersection between specific words occurring at a specific location. Also in documentation[71] of Twitter this issue is mentioned under the *Location* section. This is a major limitation in data retrieval because it enables the useless and harmful collection of noise for which workaround solutions have to be implemented.

The second issue is identified during the collection of PoI from OSM Metro Extract API. Despite the fact that the service allows to extract a wide range of PoI from the core map of OSM, a large number of specific landmarks such as important touristic attractions is missing in the retrieved data and it had to be added manually. For instance, the list of coffeeshops which was not present in the OSM dataset, was introduced by using an external source extracted from the Place API of Google. This is also a major disadvantage given that the list of PoI is an important variable used in the space partitioning and LAE tasks. It is important to state that I also attempted to use the Place API of Google to extract the list of PoI, however the dataset obtained was rather similar to the one collected via OSM hence I assume both APIs use the same source to display the PoI on map. Moreover, the Place API of Google sets an extraction limit that makes the collection rather slow and time consuming.

Due to the little available resources to the author, a third issue occurred during the collection of Twitter data. Actually, during the data collection from the Streaming API, a fairly stable internet connection is needed in order to keep the streaming of tweets flowing into the database. The effect of this shortcoming could be identified with drops in the number of tweets collected per day. Specifically, on Tuesday, the 16th of December, the connection was aborted and all tweets in that day are missing from the dataset.

In conclusion, I recommend an in depth analysis to find a way of concatenating different attributes via the filter of the Streaming API in order to obtain specific and more accurate outcome for each topic and location queried. Moreover, I suggest the use of a different source of data with regards to the collection of PoI. Perhaps, the municipality of the city being object of study could give this kind of support by providing a more complete dataset of landmarks. By doing so a subdivision of the space closer to reality hence a better

[71] Streaming API, Location section: https://dev.twitter.com/streaming/overview/request-parameters#locations

computation of the attractiveness and accessibility of touristic venues could be achieved. To end, the collection of data should be set using a stable connection in order to avoid drops in the flow.

Data analysis

In the analysis process upon the collected data of Twitter, a great number of spam and Wi-Fi services were found. These were detected through a combination of the Spatial Overlapping Frequency model (SOF) and the Frequency Distribution Analysis tool (FDA) I created via Python. The former was applied to the geo location attributes of Twitter, while the latter was applied to post attributes. The SOF model is a powerful tool used to detect noise such spam in the data, however the model alone was not sufficient to determine what the nature of the noise detected was. Hence, the FDA tool was implemented to analyse the occurrence of words in terms of their frequency and it enabled the identification of spam and free Wi-Fi services such as those used by the customers of public services like restaurants, museums, shops and so on. Basically, the FDA exploited the repetitive structure of text in posts and in case of extremely high frequency of certain words the tool identified a spam, a Wi-Fi service otherwise.

The limitation of this approach, particularly for what concerns the FDA tool, is that some kind of noise in data such as hotspots services posting news feeds (being very active on Twitter) could not be identified via FDA due to the heterogeneous text structure they showed. In fact, although the SOF model detected the SOF generated by these services, the FDA was not able to spot them, thus they were manually deleted from the datasets after a long analysis of the SOF model output.

In contrast, Wi-Fi services identified in Amsterdam have provided a reliable and accurate source of touristic information to this research as for the tweets collected from the Wi-Fi of museums, restaurants, pubs and so on. Therefore, I believe that if the Wi-Fi coverage is extended to more areas of Amsterdam, better outcomes for the computation of the landmark attractiveness could be obtained. Moreover, I would also recommend the introduction of temporary “touristic” mobile traffic data packages that can be purchased with a small amount and at the cost of being “tracked”. Thus, tourists could connect to the internet using the internal GPS of the phone enabling the collection of more accurate data relative to the tourism behaviour.

In conclusion, I recommend an adjustment of the SOF + FDA approach with regards to a better way to identify the nature of SOF via a more reliable approach. Furthermore, to improve the possibilities of using Twitter data I also recommend to establish a mobile plan in order to provide tourists with a temporary internet connection, without the need of using Wi-Fi, and so to improve the coverage of Twitter data in the urban environment. Perhaps, a pilot project could be set to evaluate the implementation onto a bigger scale.

Tourism information extraction & topic semantic

During the information extraction and topic semantic analyses, a number of issues and limitations have been identified in terms of text processing, tools, model outcomes and the subjectivity in the evaluation of some results.

In text processing and implemented tools, the major issue and limitation of this and all related works upon this matter is in the heterogeneous nature of the text attribute of Twitter. The post is a short message that the user is free to populate with any sort of data, web links, emoticons, pictures, retweets and so on. In this particular study, also the language diversity represents a problem given that most of the process implemented to normalize the text can be performed onto a language per time. To tackle those issues and limitations, a complex function to clean the data through the NLTK library of Python was introduced, and a several translation approaches implemented. Although the cleaning function used an internal dictionary and entity recognition tools to check the correctness of each word, some typo mistakes and elongated words (e.g. *"hellooooo", "doneeee", "ahahah"*) could not be deleted from the analysis. Moreover, the translation task, also available in the NLTK library was rather difficult to set due to the unavailability of the Babylon translator tool (part of the NLTK library) that was non-operational for some reasons. To overcome the issue, I introduced the TextBlob translator service through which I could translate the posts to English. However, due to the text problems mentioned before, the accuracy of translation was not always reliable and in some cases it could not be retrieved.

Another limitation identified in this approach is in the results of the Latent Dirichlet Allocation (LDA). The LDA is a *generative approach* that introduces randomness in the computation. Therefore, despite a certain degree of similarity in every run, a post can be assigned to one or more clusters differently. Actually, the purpose of the LAS is to offer a general indication of the most prominent topics for each landmark and not the exact topic discussed. If higher accuracy of results needs to be achieved, multiple runs should be performed in order to get an average of the results which would be closer to reality. However, this was out of the scope of this research also because of the short time available. Therefore to be able to obtain a better outcome of the Landmark Attractiveness Semantic, in terms of the labels assigned to the topics obtained via LDA, the results of a number of runs should be evaluated and compared.

In conclusion, a complex method, namely TAUS GKD was set to tackle all these issues described above. The method was also validated using the Sum of Squared Errors (SSE) into the K-Means clustering algorithm so as to verify whether the scores assigned to words by the LDA were indeed similar when those were added onto a 2D plane. However, in the author's view a noticeable drawback of the method is the subjectivity introduced when labels were manually assigned to topics of the LDA according to the terms returned for each topic by the LDA. This could be resolved by the computation of the average labels upon multiple run, nevertheless.

To end, the results of the validation approach (K-Means) are also slightly biased by the noise in the text as typos and elongated words could not be removed from the analysis. Hence also this limitation strengthen the need of solutions to the problem of text processing.

Accessibility of touristic venues

In the last step of this work I assess the accessibility of touristic venues by comparing the attractiveness $\boldsymbol{I_i}$ of touristic landmarks with the cost distance $\boldsymbol{d_{ij}}$ (network impedance) by means of the Gravity model of Handy (1992). The distance, as argued in assumption 6, is set as an arbitrary threshold to create buffer zones around the network of stops. Those zones are

then classified in order to create an index upon which to assess the accessibility of touristic venues.

A limitation of this approach is represented by the arbitrary choice of the distance threshold that needs to be made to be able to implement the approach. If a different distance is considered, then also the computation of the accessibility would change. This limitation could be tackled by implementing an analysis of related works on the computation of the accessibility of venues and use the results obtained in similar compact urban textures, perhaps in other cities of the Netherlands. However, due to time constrains this could not be performed and it is therefore left to future research. Moreover, I also consider the limitation of the validation approach used to confirm the findings of the attractiveness as well as their index of accessibility. Transit usage trends collected from HERE maps, mainly contain information over "requested" routes information and not actual network displacements. In fact, the log provided by HERE solely register the requests that users send to the routing engine when they are to find a particular route to reach a destination. Unfortunately, the log does not confirm if the users have actually used the transit system.

In conclusion, although the approach validated the information over the attractiveness of landmarks in both time periods considered, a more reliable approach, perhaps using GPS tracking data or actual transportation trends, could be implemented for a more accurate validation procedure.

8. Reference

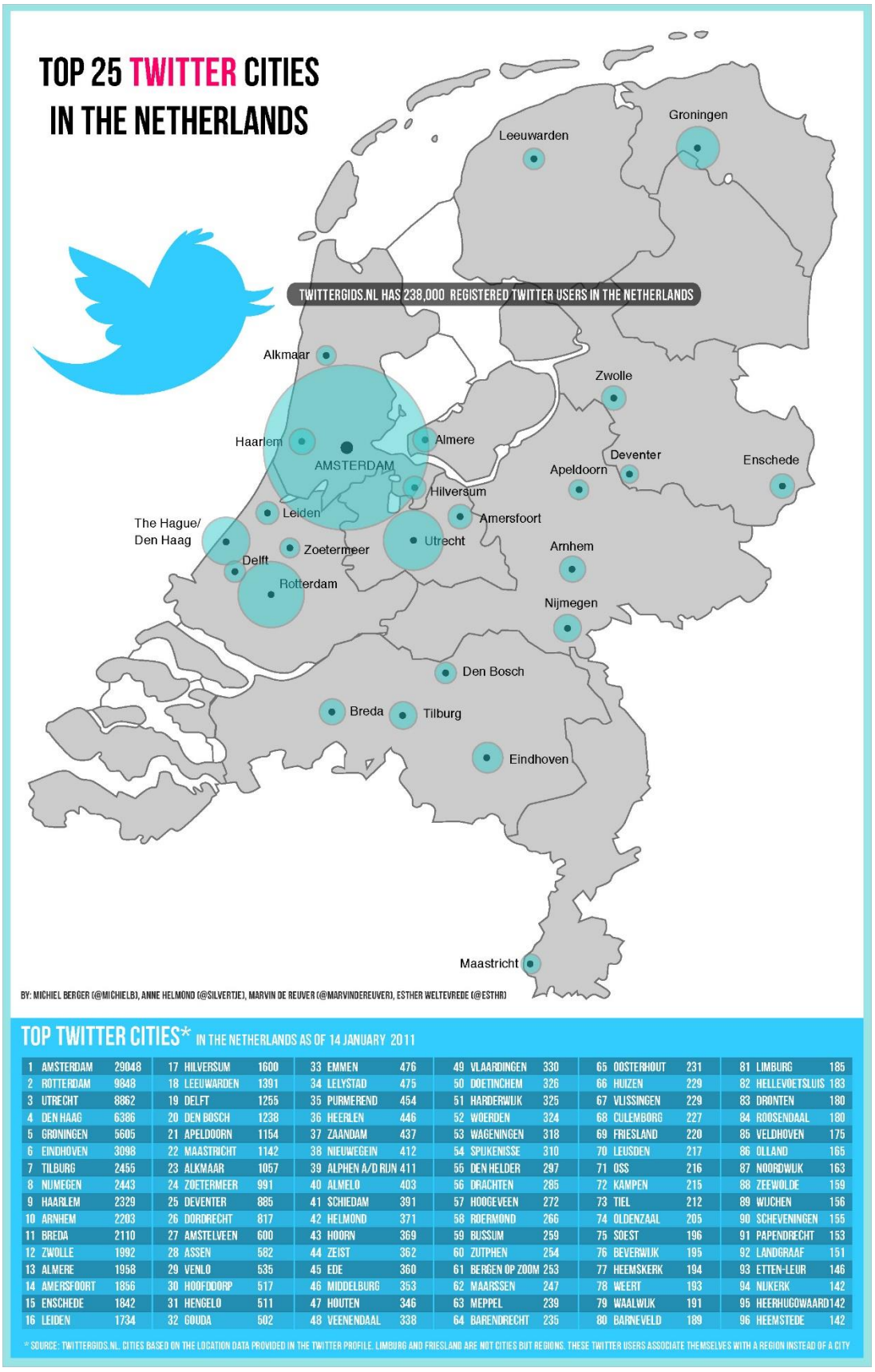
1. Amsterdam Metropolitan Solutions (AMS). (2012). Ministry of Economic Affairs (EZ). Retrieved September 2014 from http://www.amsterdam.nl/publish/pages/521972/update_amsterdam_metropolitan_solutions.pdf.
2. Application Programming Interface. (2015). Wikipedia[a]. Retrieved January 2015 from http://en.wikipedia.org/wiki/Application_programming_interface.
3. Arafat, A. N. (2012). Tools to Create Network-Based Accessibility Grids for Land Use Modelling. ESRI UC (2012) Retrieved January 2015 from http://www.birzeit.edu/node/116326.
4. Azariah, D. R., & Australia, W. (2012). Beyond the blog: The networked self of travel bloggers on Twitter. *PLATFORM: Journal of Media and Communication*, *4*(1), 63-78.
5. Baldoni, R., DAmore, F., Mecella, M., & Ucci, D. (2014, June). A Software Architecture for Progressive Scanning of On-line Communities. In *Distributed Computing Systems Workshops (ICDCSW), 2014 IEEE 34th International Conference on* (pp. 207-212). IEEE.
6. Bayes' theorem. (2015). In Wikipedia[c]. Retrieved February 08, 2015, from http://en.wikipedia.org/wiki/Bayes%27_theorem.
7. Bifet, A. (2013). Mining Big Data in Real Time. *Informatica (Slovenia)*, *37*(1), 15-20.
8. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. "O'Reilly Media, Inc.".
9. Bishop, C. M. (2006). *Pattern recognition and machine learning* (Vol. 4, No. 4, p. 12). New York: springer.
10. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. The Journal of machine learning research, 3, 993-1022.
11. Blei's LDA-C format. (2015). In Radim Řehůřek – Gensim. Retrieved April 19, 2015, from https://radimrehurek.com/gensim/corpora/bleicorpus.html.
12. Bollen, J., Pepe, A., & Mao, H. (2009). Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *arXiv preprint arXiv:0911.1583*.
13. Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M. A., Maynard, D., & Aswani, N. (2013). TwitIE: An Open-Source Information Extraction Pipeline for Microblog Text. In *RANLP* (pp. 83-90).
14. Campbell, J. C., Hindle, A., & Stroulia, E. (2014). Latent Dirichlet Allocation: Extracting Topics from Software Engineering Data.
15. Chang, H. C. (2010). A new perspective on Twitter hashtag use: diffusion of innovation theory. *Proceedings of the American Society for Information Science and Technology*, *47*(1), 1-4.
16. Cheng, Z., Caverlee, J., Lee, K., & Sui, D. Z. (2011). Exploring Millions of Footprints in Location Sharing Services. *ICWSM*, *2011*, 81-88.
17. Chowdhury, G. G. (2003). Natural language processing. *Annual review of information science and technology*, *37*(1), 51-89.
18. Cranshaw, J., Schwartz, R., Hong, J. I., & Sadeh, N. M. (2012, June). The Livehoods Project: Utilizing Social Media to Understand the Dynamics of a City. In *ICWSM*.

19. Demirbas, M., Bayir, M. A., Akcora, C. G., Yilmaz, Y. S., & Ferhatosmanoglu, H. (2010, June). Crowd-sourced sensing and collaboration using twitter. In World of Wireless Mobile and Multimedia Networks
20. Detailed Vision and Roadmap. (2014). Amsterdam Institute for Advanced Metropolitan Solutions (AIAMS). Retrieved September 2014 from http://www.tudelft.nl/fileadmin/Files/tudelft/actueel/Nieuws/Detailed_vision_and_roadmap_AMS.pdf.
21. Dykes, J. A., & Mountain, D. M. (2003). Seeking structure in records of spatio-temporal behaviour: visualization issues, efforts and applications. *Computational Statistics & Data Analysis*, *43*(4), 581-603.
22. Efron, M. (2010, July). Hashtag retrieval in a microblogging environment. In*Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (pp. 787-788). ACM.
23. Elbow for K-Means cluster. (2015). GitHub. Retrieved April 19, 2015, from http://nbviewer.ipython.org/github/nborwankar/opendatasci/blob/master/notebooks/D3.%20K-Means%20Clustering%20Analysis.ipynb.
24. Ferrari, L., Rosi, A., Mamei, M., & Zambonelli, F. (2011, November). Extracting urban patterns from location-based social networks. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks* (pp. 9-16). ACM.
25. Handy, S. (2004). Planning for accessibility: In theory and practice. Proceeding from the Access to Destination Conference. Minneapolis, MN: University of Minnesota.
26. Handy, S. L. (1992). Regional versus local accessibility: neo-traditional development and its implications for non-work travel. *Built Environment (1978)*, 253-267.
27. Hashtag. (2015). In Wikipedia[h]. Retrieved April 11, 2015, from http://en.wikipedia.org/wiki/Hashtag.
28. Helmond, A. (2011). Twitter NL Visualizations. Anne Helmond. New Media Research Blog. Retrieved October 2014 from http://www.annehelmond.nl/2011/01/19/visualization-of-the-top-25-twitter-cities-in-the-netherlands/.
29. Information Retrieval. (2015). In Wikipedia[b]. Retrieved January 30, 2015, from http://en.wikipedia.org/wiki/Information_retrieval.
30. Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern recognition letters*, *31*(8), 651-666.
31. Joseph, K., Tan, C. H., & Carley, K. M. (2012, September). Beyond local, categories and friends: clustering foursquare users with latent topics. In*Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (pp. 919-926). ACM.
32. Kaufmann, M., & Kalita, J. (2010, July). Syntactic normalization of twitter messages. In *International conference on natural language processing, Kharagpur, India*.
33. Laube, P., & Purves, R. S. (2006). An approach to evaluating motion pattern detection techniques in spatio-temporal data. *Computers, environment and urban systems*, *30*(3), 347-374.
34. Lee, R., Wakamiya, S., & Sumiya, K. (2013). Urban area characterization based on crowd behavioral lifelogs over Twitter. *Personal and ubiquitous computing*, *17*(4), 605-620.
35. LSI. (2015). In Wikipedia[d]. Retrieved February 08, 2015, from http://en.wikipedia.org/wiki/Latent_semantic_indexing.

36. Lucena, D., Brito, G., Formiga, A., & Pessoa–PB–Brazil, J. A Probabilistic Programming Approach to Naive Bayes Text Classification.
37. Machine learning. (2015). In Wikipedia[g]. Retrieved February 13, 2015, from http://en.wikipedia.org/wiki/Machine_learning.
38. Mai, E., & Hranac, R. (2013, January). Twitter Interactions as a Data Source for Transportation Incidents. In *Proc. Transportation Research Board 92nd Ann. Meeting* (No. 13-1636).
39. Mennis, J., & Guo, D. (2009). Spatial data mining and geographic knowledge discovery - An introduction. *Computers, Environment and Urban Systems*, *33*(6), 403-408.
40. Meyer, W. D. (2010). *Analyzing Crime on Street Networks: A Comparison of Network and Euclidean Voronoi Methods* (Doctoral dissertation, University of Illinois at Urbana-Champaign).
41. Miller, H. J., & Han, J. (Eds.). (2009). *Geographic data mining and knowledge discovery* (pp. 3–32). London: Taylor and Francis.
42. Muntean, C. I., Morar, G. A., & Moldovan, D. (2012, January). Exploring the Meaning behind Twitter Hashtags through Clustering. In *Business Information Systems Workshops* (pp. 231-242). Springer Berlin Heidelberg.
43. Noulas, A., Scellato, S., Mascolo, C., & Pontil, M. (2011). An Empirical Study of Geographic User Activity Patterns in Foursquare. *ICWSM*, *11*, 70-573.
44. Pak, A., & Paroubek, P. (2010, May). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *LREC*.
45. Preferred walking speed. (2015). In Wikipedia[e]. Retrieved January 28, 2015, from http://en.wikipedia.org/wiki/Preferred_walking_speed.
46. Sacco, D., Motta, G., You, L., Bertolazzo, N., & Chen, C. (2013) Smart Cities, Urban Sensing and Big Data: Mining Geo-location in Social Networks.
47. Sacco, D., Motta, G., You, L., Bertolazzo, N., & Chen, C. (2013). Smart Cities, Urban Sensing and Big Data: Mining Geo-location in Social Networks.
48. Shen, Q. (1998). Spatial technologies, accessibility, and the social construction of urban space. *Computers, environment and urban systems*, *22*(5), 447-464.
49. Sink, T. (2010). Gravity model. In B. Warf (Ed.), *Encyclopaedia of geography.* (p. 1362). Thousand Oaks, SAGE Publications, Inc. doi: http://dx.doi.org/10.4135/9781412939591.n538.
50. Spamming. (2015). In Wikipedia[f]. Retrieved February 13, 2015, from http://en.wikipedia.org/wiki/Spamming.
51. Tasse, D., & Hong, J. I. (2014). *Using Social Media Data to Understand Cities*. Technical report, Carnegie Mellon University.
52. Tf-idf. (2015). In Wikipedia[e]. Retrieved February 08, 2015, from http://en.wikipedia.org/wiki/Tf%E2%80%93idf.
53. The REST API. (2014). Twitter. Developers section (Developers\Documentation\REST APIs). Retrieved October 2014 from https://dev.twitter.com/rest/public.
54. Thelwall, M., Buckley, K., & Paltoglou, G. (2011). Sentiment in Twitter events.*Journal of the American Society for Information Science and Technology*, *62*(2), 406-418.
55. Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(2), 411-423.
56. Van de Ven, M., & Neroni, J. (2012). Twitter as a potential data source for statistics.
57. Villars, R. L., Olofson, C. W., & Eastwood, M. (2011). Big data: What it is and why you should care. White Paper, IDC.

Appendices

Appendix A – Twitter penetration (per city)

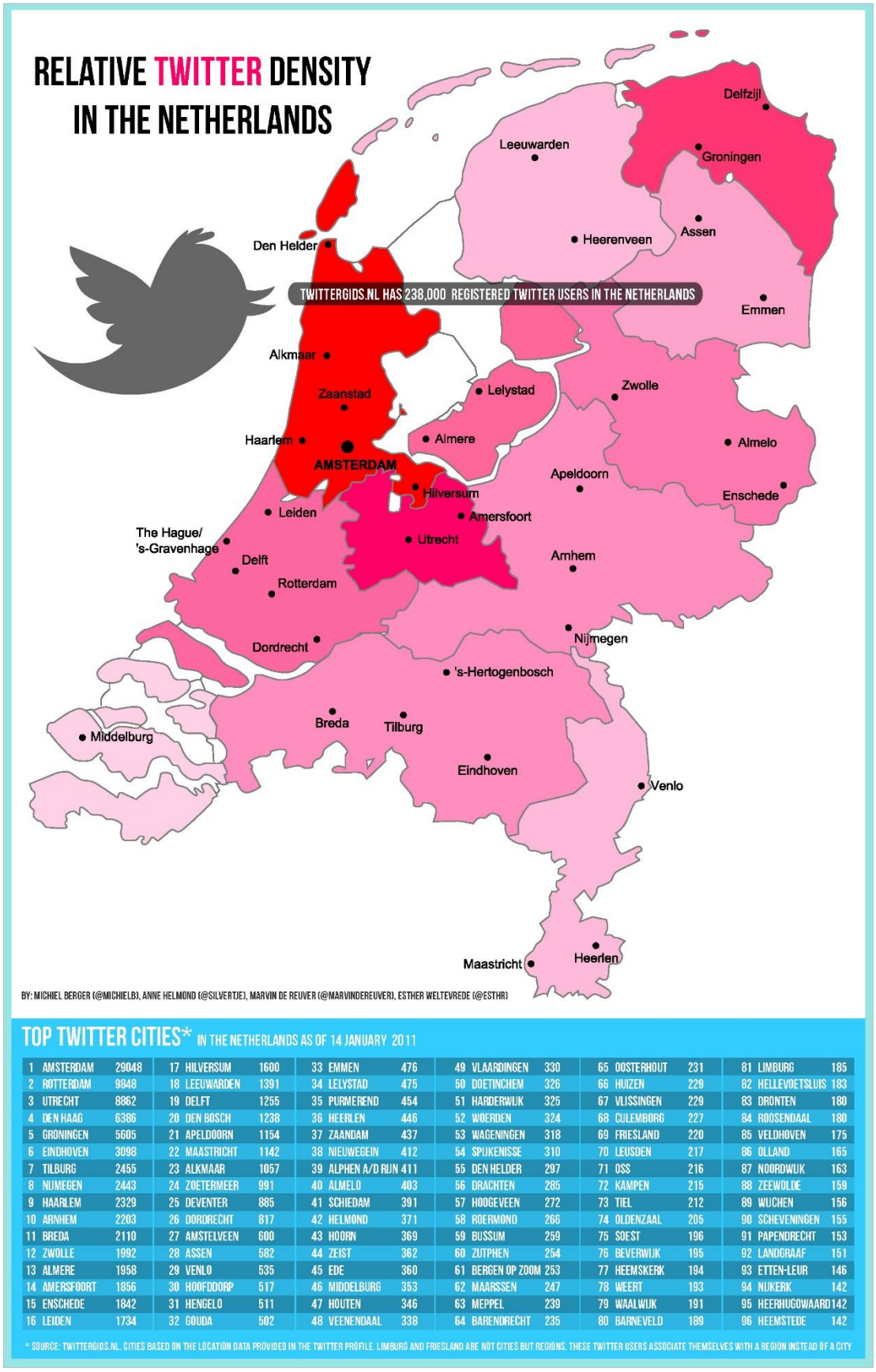


TOP TWITTER CITIES* IN THE NETHERLANDS AS OF 14 JANUARY 2011

1	AMSTERDAM	29048	17	HILVERSUM	1600	33	EMMEN	476	49	VLAARDINGEN	330	65	OOSTERHOUT	231	81	LIMBURG	185
2	ROTTERDAM	9848	18	LEEUWARDEN	1391	34	LELYSTAD	475	50	DOETINCHEM	326	66	HUIZEN	229	82	HELLEVOETSLUIS	183
3	UTRECHT	8862	19	DELFT	1255	35	PURMEREND	454	51	HARDERWIJK	325	67	VLISSINGEN	229	83	DRONTEN	180
4	DEN HAAG	6386	20	DEN BOSCH	1238	36	HEERLEN	446	52	WOERDEN	324	68	CULEMBORG	227	84	ROOSENDAAL	180
5	GRONINGEN	5605	21	APELDOORN	1154	37	ZAANDAM	437	53	WAGENINGEN	318	69	FRIESLAND	220	85	VELDHOVEN	175
6	EINDHOVEN	3098	22	MAASTRICHT	1142	38	NIEUWEGEIN	412	54	SPIJKENISSE	310	70	LEUSDEN	217	86	OLLAND	165
7	TILBURG	2455	23	ALKMAAR	1057	39	ALPHEN A/D RIJN	411	55	DEN HELDER	297	71	OSS	216	87	NOORDWIJK	163
8	NIJMEGEN	2443	24	ZOETERMEER	991	40	ALMELO	403	56	DRACHTEN	285	72	KAMPEN	215	88	ZEEWOLDE	159
9	HAARLEM	2329	25	DEVENTER	885	41	SCHIEDAM	391	57	HOOGEVEEN	272	73	TIEL	212	89	WIJCHEN	156
10	ARNHEM	2203	26	DORDRECHT	817	42	HELMOND	371	58	ROERMOND	266	74	OLDENZAAL	205	90	SCHEVENINGEN	155
11	BREDA	2110	27	AMSTELVEEN	600	43	HOORN	369	59	BUSSUM	259	75	SOEST	196	91	PAPENDRECHT	153
12	ZWOLLE	1992	28	ASSEN	582	44	ZEIST	362	60	ZUTPHEN	254	76	BEVERWIJK	195	92	LANDGRAAF	151
13	ALMERE	1958	29	VENLO	535	45	EDE	360	61	BERGEN OP ZOOM	253	77	HEEMSKERK	194	93	ETTEN-LEUR	146
14	AMERSFOORT	1856	30	HOOFDDORP	517	46	MIDDELBURG	353	62	MAARSSEN	247	78	WEERT	193	94	NIJKERK	142
15	ENSCHEDE	1842	31	HENGELO	511	47	HOUTEN	346	63	MEPPEL	239	79	WAALWIJK	191	95	HEERHUGOWAARD	142
16	LEIDEN	1734	32	GOUDA	502	48	VEENENDAAL	338	64	BARENDRECHT	235	80	BARNEVELD	189	96	HEEMSTEDE	142

* SOURCE: TWITTERGIDS.NL. CITIES BASED ON THE LOCATION DATA PROVIDED IN THE TWITTER PROFILE. LIMBURG AND FRIESLAND ARE NOT CITIES BUT REGIONS. THESE TWITTER USERS ASSOCIATE THEMSELVES WITH A REGION INSTEAD OF A CITY

Appendix B - Twitter penetration (per region)

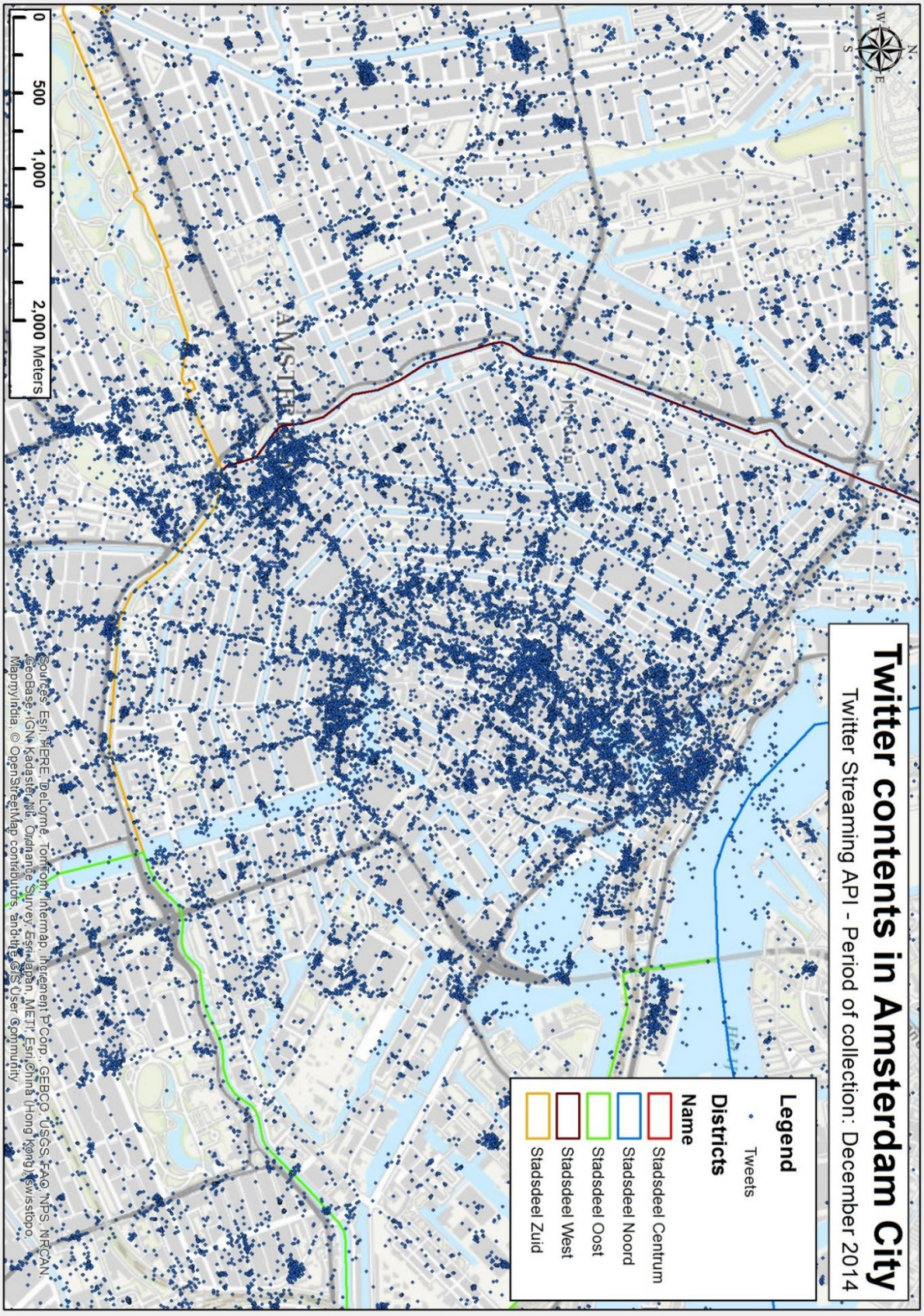


TOP TWITTER CITIES* IN THE NETHERLANDS AS OF 14 JANUARY 2011

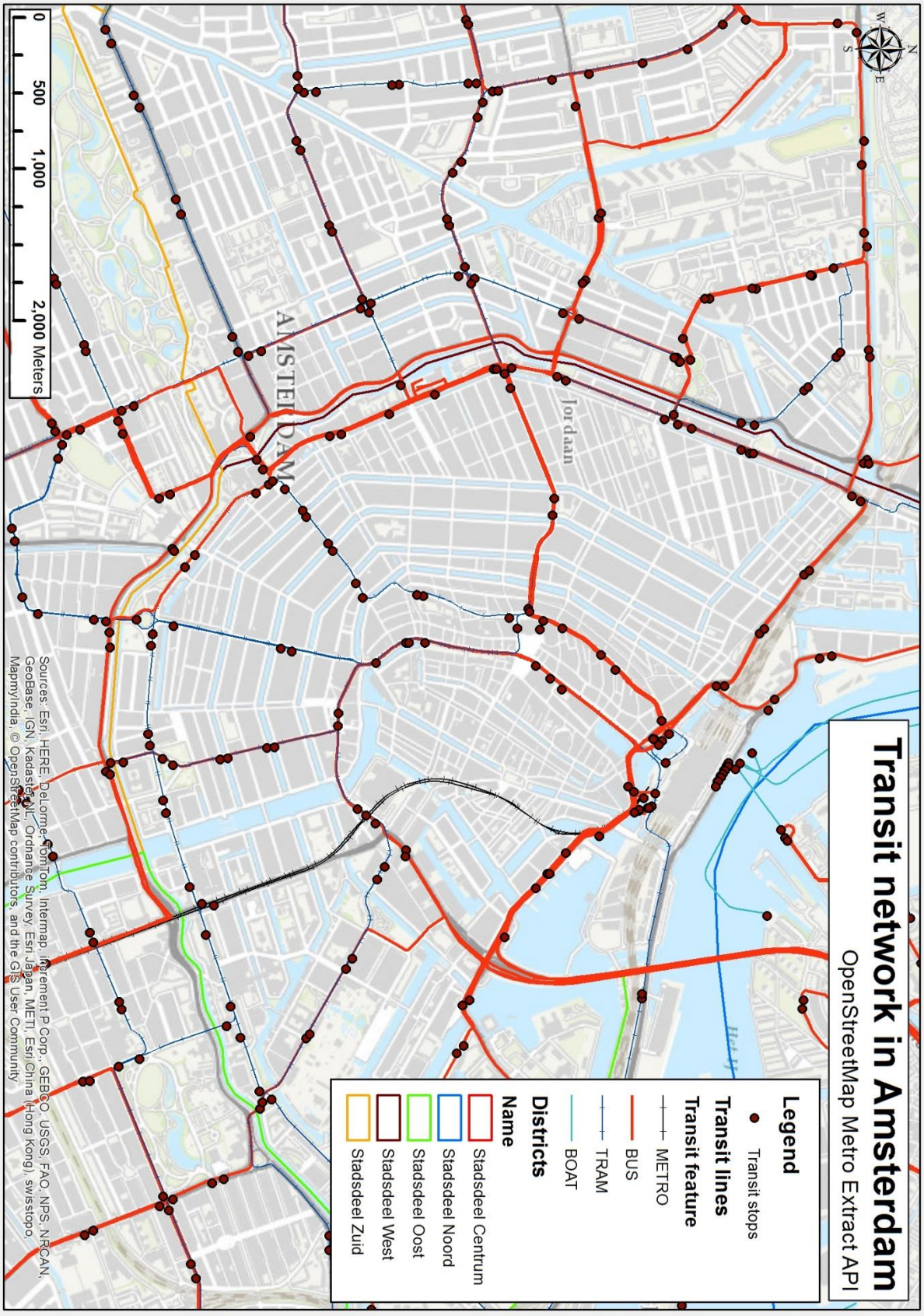
1	AMSTERDAM	29048	17	HILVERSUM	1600	33	EMMEN	476	49	VLAARDINGEN	330	65	OOSTERHOUT	231	81	LIMBURG	185
2	ROTTERDAM	9848	18	LEEUWARDEN	1391	34	LELYSTAD	475	50	DOETINCHEM	326	66	HUIZEN	229	82	HELLEVOETSLUIS	183
3	UTRECHT	8862	19	DELFT	1255	35	PURMEREND	454	51	HARDERWIJK	325	67	VLISSINGEN	229	83	DRONTEN	180
4	DEN HAAG	6386	20	DEN BOSCH	1238	36	HEERLEN	446	52	WOERDEN	324	68	CULEMBORG	227	84	ROOSENDAAL	180
5	GRONINGEN	5605	21	APELDOORN	1154	37	ZAANDAM	437	53	WAGENINGEN	318	69	FRIESLAND	220	85	VELDHOVEN	175
6	EINDHOVEN	3098	22	MAASTRICHT	1142	38	NIEUWEGEIN	412	54	SPIJKENISSE	310	70	LEUSDEN	217	86	OLLAND	165
7	TILBURG	2455	23	ALKMAAR	1057	39	ALPHEN A/D RIJN	411	55	DEN HELDER	297	71	OSS	216	87	NOORDWIJK	163
8	NIJMEGEN	2443	24	ZOETERMEER	991	40	ALMELO	403	56	DRACHTEN	285	72	KAMPEN	215	88	ZEEWOLDE	159
9	HAARLEM	2329	25	DEVENTER	885	41	SCHIEDAM	391	57	HOOGEVEEN	272	73	TIEL	212	89	WIJCHEN	156
10	ARNHEM	2203	26	DORDRECHT	817	42	HELMOND	371	58	ROERMOND	266	74	OLDENZAAL	205	90	SCHEVENINGEN	155
11	BREDA	2110	27	AMSTELVEEN	600	43	HOORN	369	59	BUSSUM	259	75	SOEST	196	91	PAPENDRECHT	153
12	ZWOLLE	1992	28	ASSEN	582	44	ZEIST	362	60	ZUTPHEN	254	76	BEVERWIJK	195	92	LANDGRAAF	151
13	ALMERE	1958	29	VENLO	535	45	EDE	360	61	BERGEN OP ZOOM	253	77	HEEMSKERK	194	93	ETTEN-LEUR	146
14	AMERSFOORT	1856	30	HOOFDDORP	517	46	MIDDELBURG	353	62	MAARSSEN	247	78	WEERT	193	94	NIJKERK	142
15	ENSCHEDE	1842	31	HENGELO	511	47	HOUTEN	346	63	MEPPEL	239	79	WAALWIJK	191	95	HEERHUGOWAARD	142
16	LEIDEN	1734	32	GOUDA	502	48	VEENENDAAL	338	64	BARENDRECHT	235	80	BARNEVELD	189	96	HEEMSTEDE	142

* SOURCE: TWITTERGIDS.NL. CITIES BASED ON THE LOCATION DATA PROVIDED IN THE TWITTER PROFILE. LIMBURG AND FRIESLAND ARE NOT CITIES BUT REGIONS. THESE TWITTER USERS ASSOCIATE THEMSELVES WITH A REGION INSTEAD OF A CITY

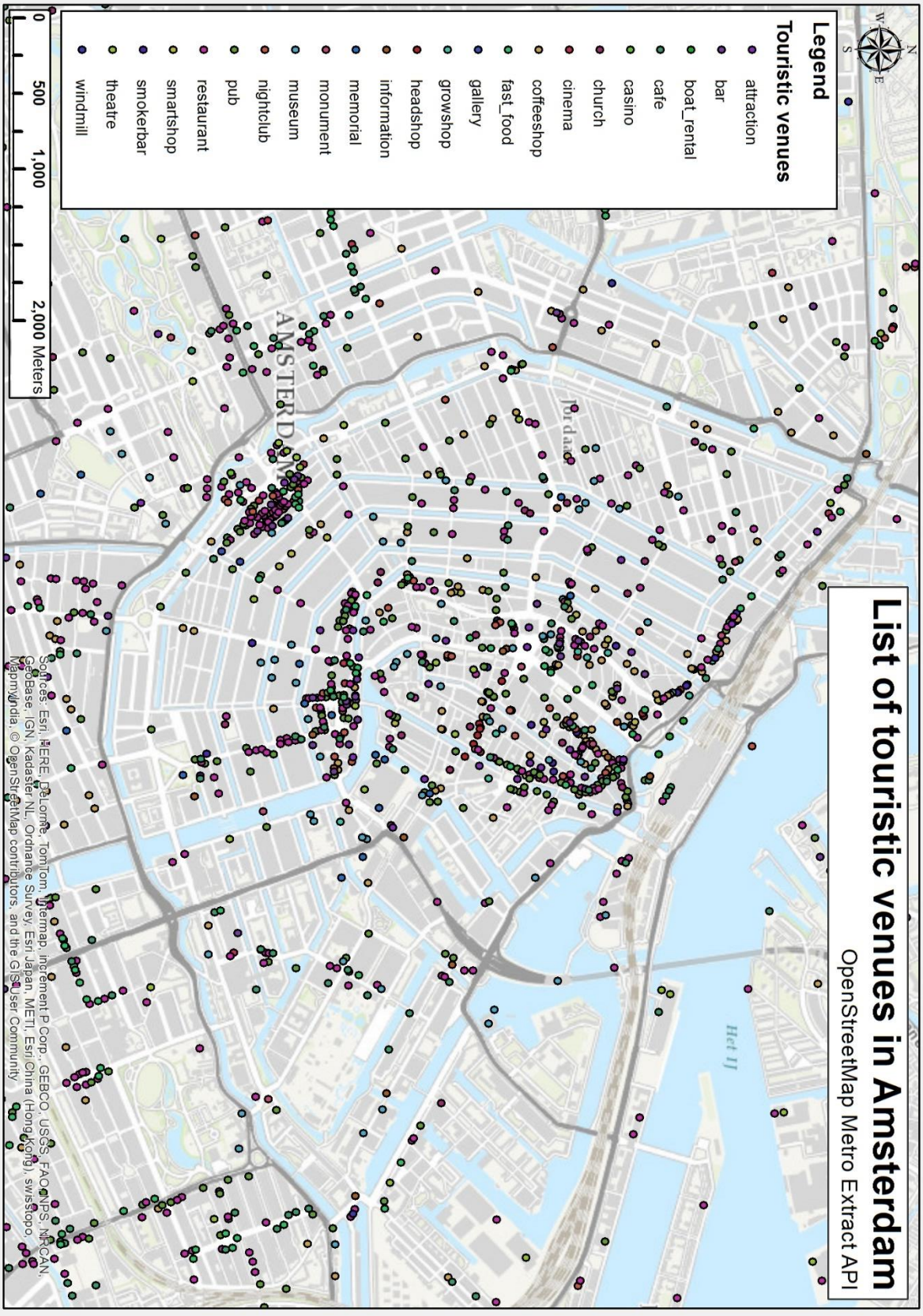
Appendix C – Tweets locations in space



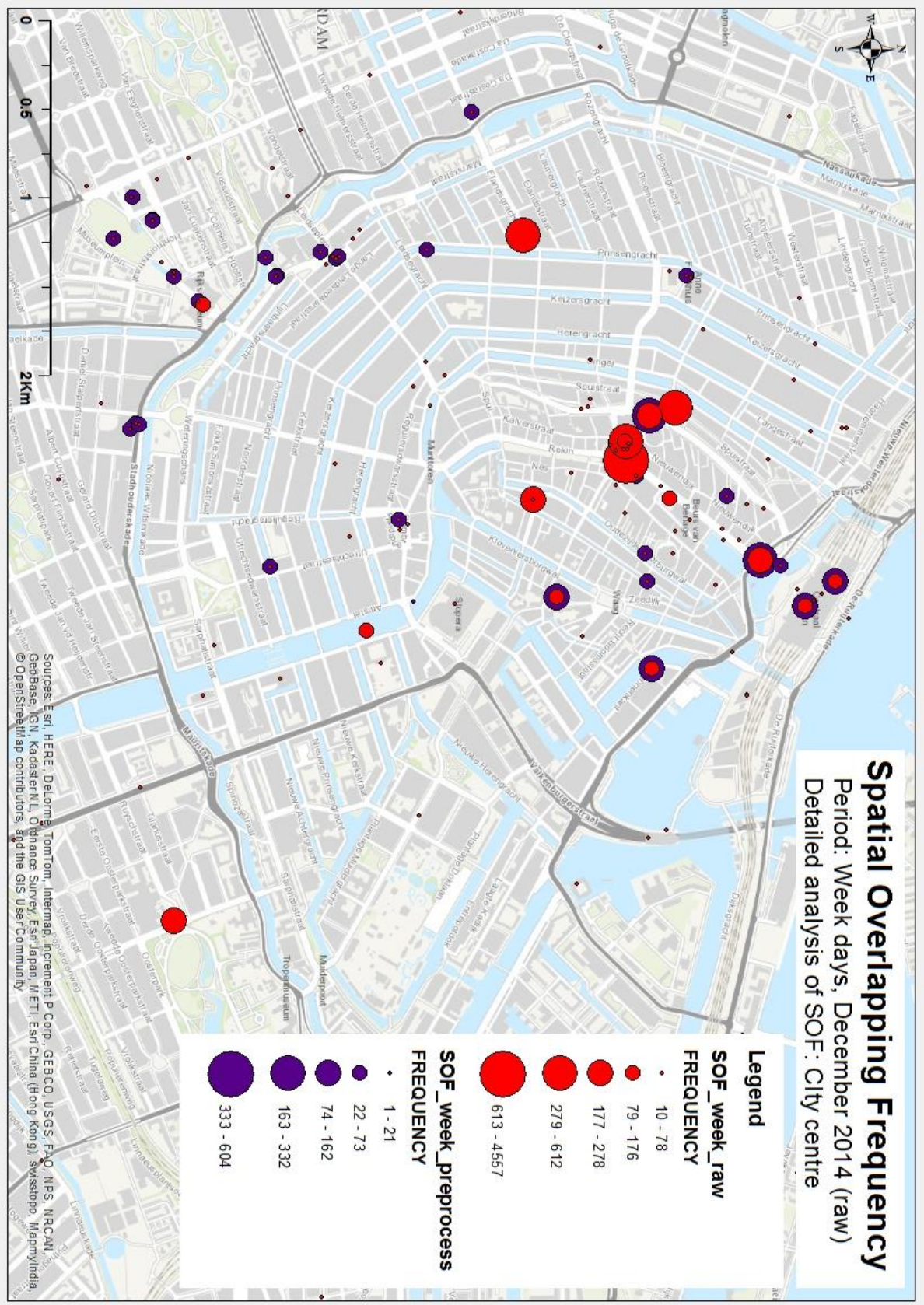
Appendix D – Transit Network



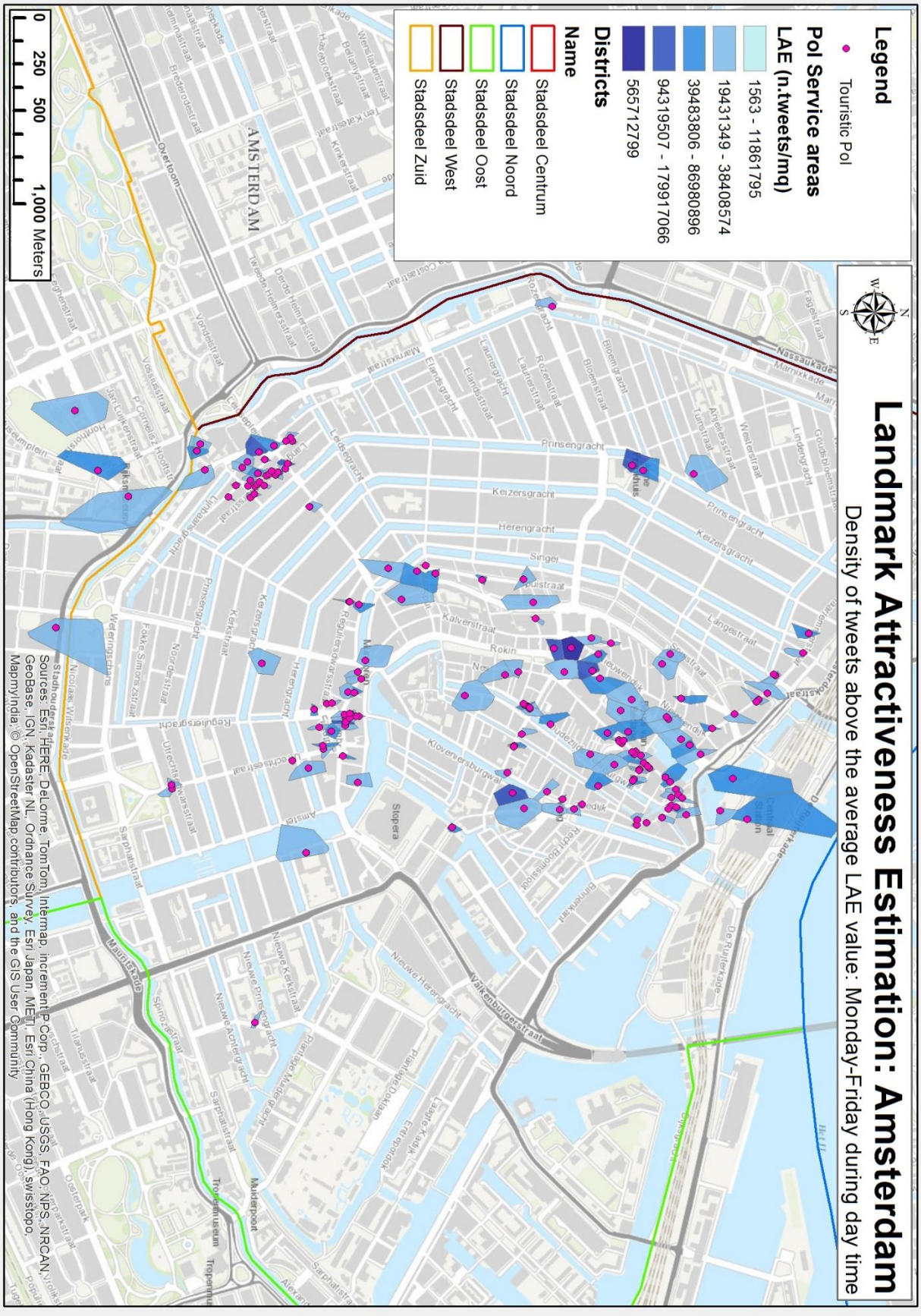
Appendix E – Amsterdam touristic venues



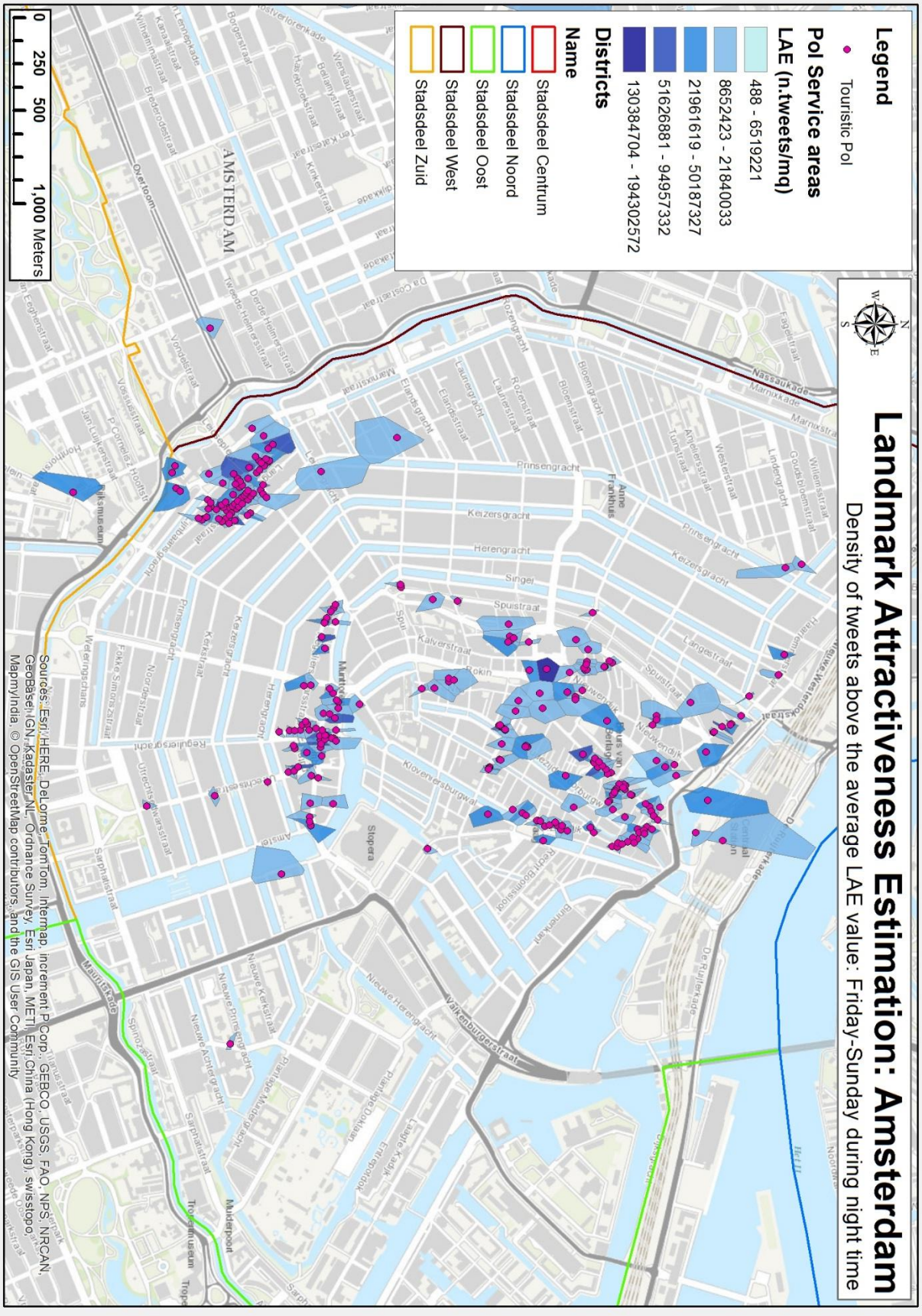
Appendix F - Spatial Overlapping Frequencies (raw vs clean)



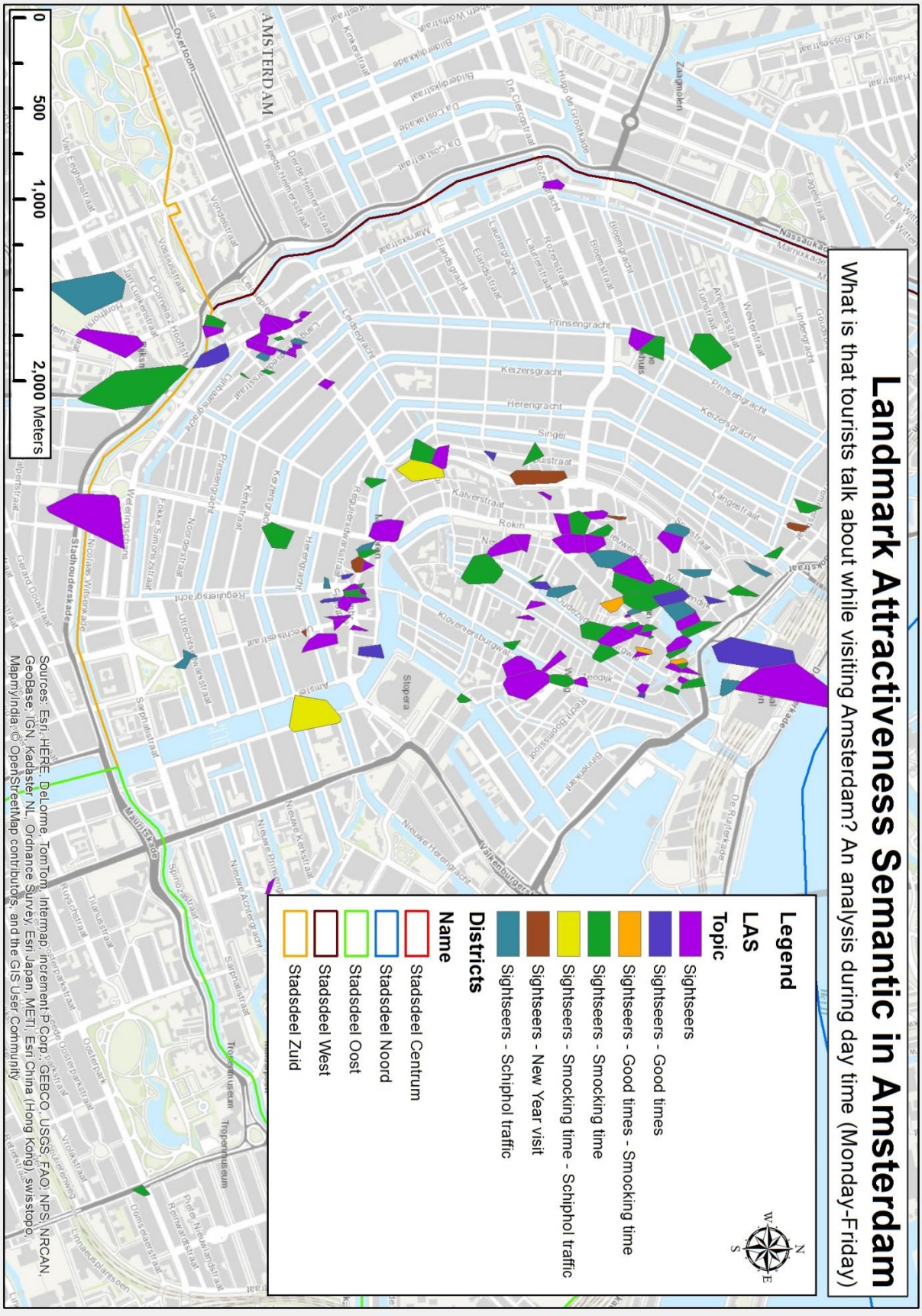
Appendix G – Attractiveness (day)



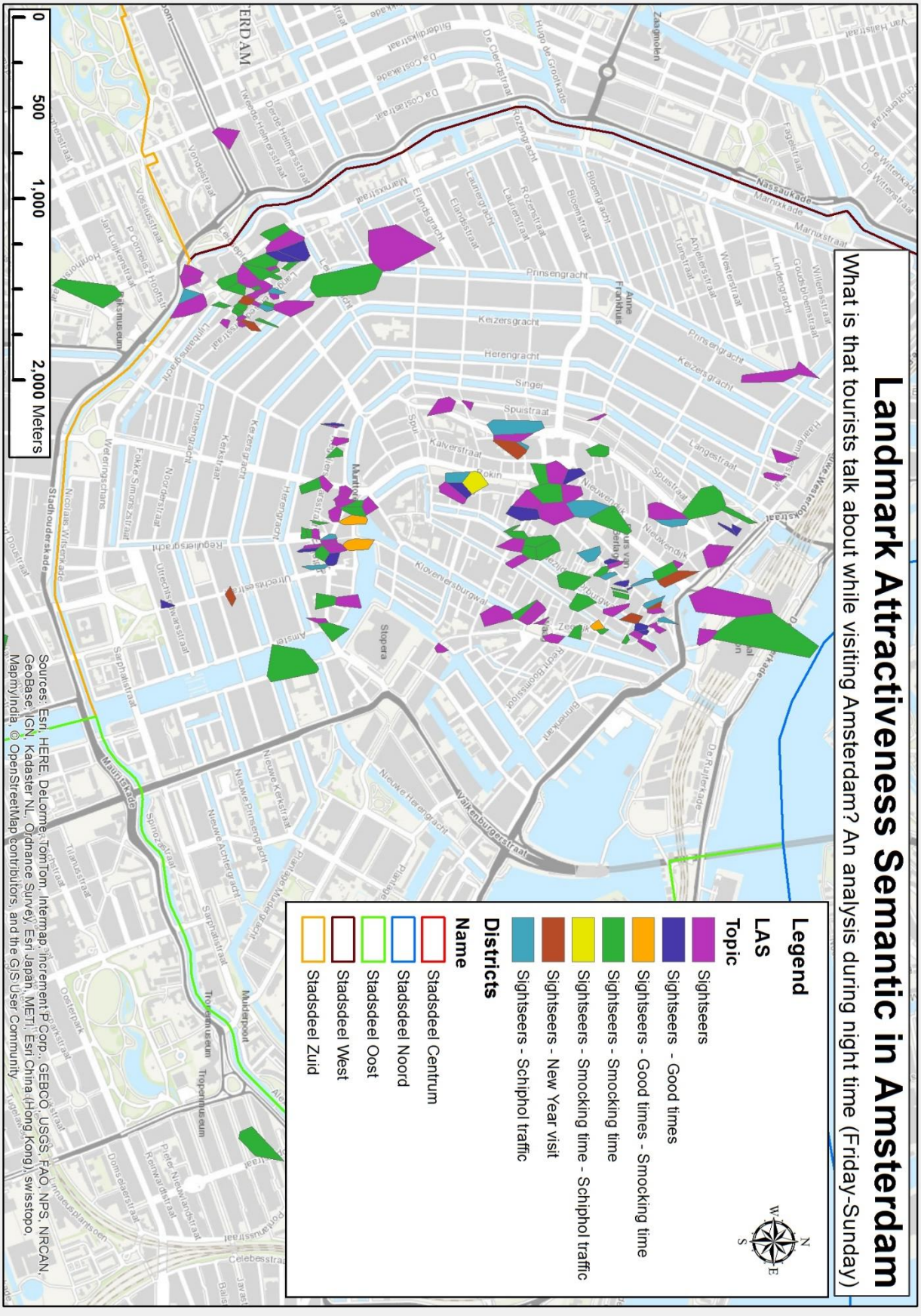
Appendix H – Attractiveness (Night)



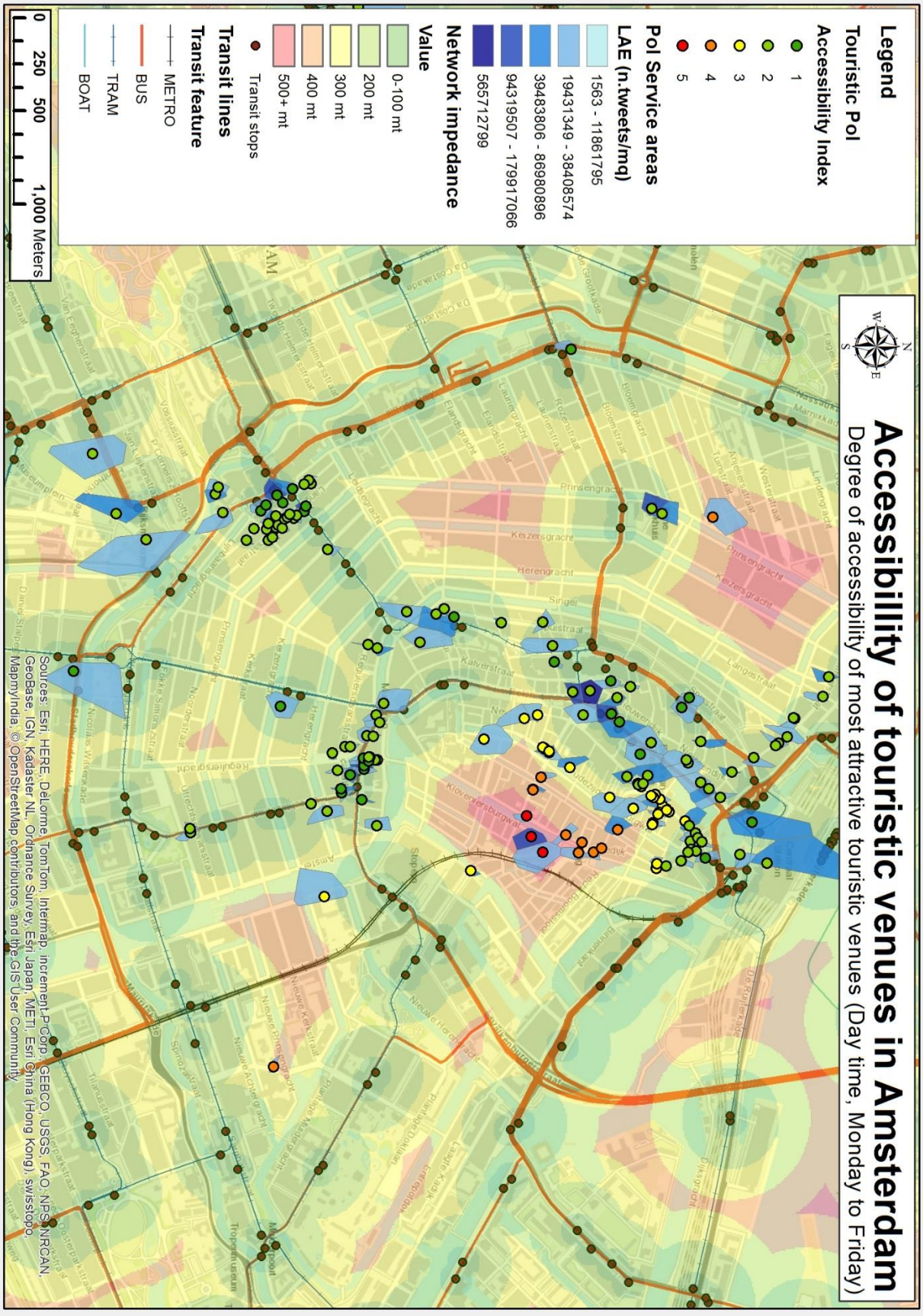
Appendix I – Semantic (day)



Appendix J – Semantic (Night)



Appendix K – Accessibility (Day)



Appendix L – Accessibility (Night)