“Should I Stay Or Should I Go”: Clash of Opinions in the Brexit Twitter Debate

Lena Mangold

Wadham College

University of Oxford

A thesis submitted for the degree of

Master of Science in Mathematical Modelling and Scientific Computing

September 2016

To Joss

Acknowledgements

I would like to thank my supervisor Professor Mason Porter for his support, his comments and his valuable criticism, and also for introducing me to the field of network science and the vastness of its applications.

I am especially grateful to Guillermo Garduño and his colleagues at Sinnia, Social Data Scientists, for collecting and providing the Twitter data set used in this thesis.

I also want to thank Dr Mariano Beguerisse for his support and advice regarding the collection and processing of Twitter data.

Finally, I want to express my gratitude to Dr Kathryn Gillow for her great commitment to running the MSc Mathematical Modelling and Scientific Computing.

Abstract

Brexit, short for "British exit", was one of the most important events to occur in recent British political and economic history. The EU referendum took place on the 23rd June, resulting in a win for the *Leave* campaign. The sentiment of both the *Leave* and the *Remain* campaigns clashed on mainstream media channels and social networking sites long before the day of the referendum; though still many are convinced that British voters were deceived — by one side or the other. With traditional opinion polls increasingly being doubted, the opportunity for alternative types of public opinion analysis arises. The micro-blogging site Twitter, on which users create 140-character messages to interact with each other, and which was a battleground for various stages of the referendum campaigning, seems to be the appropriate platform for such an analysis.

This thesis is concerned with the analysis of Twitter debates surrounding the EU referendum through methods from network science as well as text data mining. We construct *retweet* and *reply* networks, in which *nodes* act as Twitter users whose interactions connect them to others by *edges*. To shed light on the structure of Brexit conversations, we investigate the presence of groups, called *communities*, within the networks. We examine users of these networks which are more influential, or *central*, than others and are thus perhaps more likely to be spreading information. We also apply methods from the field of text data mining on different types of textual data to learn about topics discussed in conversations around Brexit. We find that the methods we suggest give preliminary insights into the structure of the EU referendum debates and set the path for possible insightful, in-depth analyses of these Twitter conversations.

Contents

List of Figures

Chapter 1

Introduction

On 23 June 2016, the people of the United Kingdom voted for “Brexit”, the withdrawal of the UK from the European Union [1]. In the weeks following the referendum, both journalists and academics attempted to analyse the causes that lead to this result [2, 3] as well as its potential implications on numerous subjects including the living standard in the UK [4], the financial market [5] and even Australian wine exports [6]. Despite the attention Brexit received from the academic world and the mainstream media even in the months before the referendum, Google’s Trend service suggested that on the morning of the day on which results were announced, the two most searched-for terms in the UK were “What does it mean to leave the EU?” and “What is the EU?” [7]. This phenomenon highlights the concern, voiced by many both before as well as after the referendum [8, 9], that members of the public were exposed to misleading or plainly incorrect messages from both sides of the referendum campaign. If these concerns are valid, were those eligible to vote able to make an informed decision on the matter at all? This question encourages an analysis of the *conversations* between members of the public to understand how information and opinions spread in the weeks before the referendum.

Unlike other political debates which are often split along regularly occuring axes such as left-wing and right-wing [10], the Brexit debate drew a different dividing line between those involved [11].

To gain a better understanding of these unusual lines of cross-party polarisation and the general spread and exchange of information and opinions regarding the EU referendum it seems natural to make use of this decade’s primary medium of communication: the Web 2.0. Recent studies have suggested that the usage of social media platforms affects not only the level of participation in political debates [12] but also our decision-making processes in general [13] and those regarding political choices [14]. The increase in online conversations, as both a generality, and in relationship to political discourse, [15], has lead to a growing awareness of the tractability of these conversations. This is encouraging scientists to quantify structures of human communicational behaviour on the web and to investigate the extent to which these findings are transferable to offline behaviour and the limits to this [16, 15]. In [16], Asur and Huberman measured the amount of attention newly released films received on a social media platform and found that this correlated with the popularity of those films based on cinema visitors. Tumasjan et al. made an attempt to predict the outcome of the German federal election in 2009 by analysing social media and they claimed that their results nearly matched those of the polls [15].

One platform that has received a particularly large amount of attention in academic spheres is the micro-blogging site Twitter. A Twitter user can create short messages of 140 characters, called "tweets", which are automatically shown to their *followers*, i.e. those that have subscribed to reading their posted content. There are several ways of reacting to another user's tweet, including *replying* or re-broadcasting (*retweeting*) it to one's own followers. Another popular feature of Twitter are *hashtags*, which can be included in tweets before words for the purpose of emphasis [17]. The short length of messages, which makes the processing of the data comparatively easy [18], and the relatively low access restrictions of content [19] make Twitter an attractive data source for researchers from various fields.

The different types of interactions between users of the social media platform have served as a basis for numerous studies that have used methods from the field of "network science" to investigate structures and characteristics of political debates [10, 20]. Other researchers have applied tools from text data mining to Twitter data and have made conclusions on demographics [21], political views of users [22] or topics discussed in social movements [23].

Inspired by some of the tools and techniques from previous work, we would like to explore Twitter conversations on the topic of Brexit by collecting tweets that were created both before and shortly after the EU referendum. We make use of various methods from network science and text data mining to understand both the structure of the conversations that took place, as well as the specific topics that were discussed. In particular, we seek to examine the extent to which participants in the debate conversed with each other with similar opinions and whether they had the chance to be informed across a broad area of topics.

1.1 Thesis outline

This thesis is divided into seven chapters. In Chapter 2 we give an overview of mathematical and methodological preliminaries, followed by a presentation of relevant work that has been done in the areas of political network analysis and text data mining on Twitter in Chapter 3. Chapter 4 explains the collection and processing of our Twitter data. In Chapters 5 and 6 we explain the methods we use and the results we obtain by implementing these methods respectively. We end this thesis with a discussion and conclusions on the insights we gained about Twitter conversations related to the EU referendum.

Chapter 2

Mathematical and methodological preliminaries

We first lay the groundwork for the methods used in this thesis by introducing the concept of a network as well as the mathematical background required for the methods that follow.

A network G is best described as a number of components, often called *nodes*, which are linked to each other in one way or another by *edges*; there are countless numbers of different types of networks, including biological networks, ecological networks, technological networks or social networks to name just a few [24]. Food-web networks are a typical sub-category of ecological networks, with the dynamics between predator and prey being a classic example [25]. A famous social network is one in which the nodes represent academics which are joined together by an edge if they have collaborated on a scientific paper [26]. Depending on the scientific questions one is asking, studying the structure of networks can give insights into patterns of interactions between nodes and can thus help explain how this affects the dynamics of the complete system [24].

2.1 Mathematics of networks

We introduce the basic mathematical representation of networks by following Newman's work in [24].

In some cases, the edges that connect two nodes point from one node to another but not vice versa; these are called *directed* edges. A common mathematical representation of a network is an *adjacency matrix*. An asymmetric adjacency matrix $\boldsymbol{A}$

of a directed network G has elements

$$A_{ij} \quad = \quad \begin{cases} 1, & \text{if there exists an edge from } j \text{ to } i \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

In some networks — such as Twitter networks, as described in Section 4.2 — pairs of nodes are likely to interact more than once during certain time frames. Such networks are called *multigraphs.* Multigraphs can be converted to *weighted* network by summing the number of times an edges is present between two nodes. Similarly to the adjacency matrix, the *weight matrix* $\boldsymbol{W}$ for a directed network has entries $W_{ij} = X$ if there is an edge of weight X from node j to node i. It is necessary to note that, despite being a common procedure, the procedure of aggregating edges to construct a weighted network can produce a bias. It assumes the unlikely case that the emerging of edges between two nodes are controlled by Poisson processes [27].

When starting to analyse a network obtained from a data set with as yet unknown structures there are a few standard diagnostics that yield initial insights into the rough structure of the network; we introduce some essential ones in this section. If not stated otherwise, we follow the work in [24] to introduce the notations in this section.

2.1.1 Degree

The *degree* k_i of a node i is defined as the sum of all edges connected to the node:

$$k_i = \sum_{j=1}^{n} A_{ij}, \tag{2.2}$$

where n is the number of nodes. In a weighted network, this concept is extended to what is called the *strength* w_i of node i, defined using the weight matrix as

$$w_i = \sum_{j=1}^{n} W_{ij}. \tag{2.3}$$

Let us denote the sum of all edge weights in a network by $\mathbf{w}$. When taking the directions of the edges into account, one differentiates between *in-degree* k^{in}, the number of edges coming into a node, and *out-degree* k^{out}, the number of edges leaving a node. Using the adjacency matrix and weight matrix respectively, these are defined as

$$k_i^{\text{in}} = \sum_{j=1}^{n} A_{ij}, \qquad k_j^{\text{out}} = \sum_{i=1}^{n} A_{ij} \tag{2.4}$$

for unweighted networks and as

$$w_i^{\text{in}} = \sum_{j=1}^{n} W_{ij}, \qquad w_j^{\text{out}} = \sum_{i=1}^{n} W_{ij} \tag{2.5}$$

for weighted networks. We can define the mean in-degree c_{in} (and the mean out-degree c_{out}) of a directed network with a total of n nodes as

$$c_{\text{in}} = \frac{1}{n}\sum_{i=1}^{n} k_i^{\text{in}} = \frac{1}{n}\sum_{j=1}^{n} k_i^{\text{out}} = c_{\text{out}}. \tag{2.6}$$

The extension to the mean strength of a directed network is straightforward.

2.1.2 Reciprocity

The *reciprocity* r of a directed network G is the fraction of edges that are reciprocated (i.e. if there is an edge from i to j and one from j to i). It is described in [24] by

$$r(G) = \frac{1}{m}\sum_{ij} A_{ij}A_{ji} = \frac{1}{m}\text{Tr}\mathbf{A}^2, \tag{2.7}$$

where m is the number of directed edges.

2.1.3 Density

The *density* of a directed network is the proportion of possible edges which are present and is described by

$$d = \frac{m}{n^2} \tag{2.8}$$

if the network has self-loops (i.e. a node can be connected by an edge to itself) [28].

2.1.4 Random walk

A *path* represents a chain of nodes such that there is an edge between each pair of successive nodes. A *random walk* describes the traversing a network by taking consecutive steps from node to node on such a path; at each node i the next node is chosen uniformly at random from i's neighbours. Assuming that at time $t-1$, the walk is at node j, then in a directed network, the probability $p_i(t)$ that the walk is at node i at time t is

$$p_i(t) = \sum_j \frac{A_{ij}}{k_j^{\text{out}}} p_j(t-1), \tag{2.9}$$

where $1/k_j^{\text{out}}$ is the probability of the walk stepping from node j along one of j's outgoing edges.

2.1.5 Components

Not every pair of nodes in a network is necessarily connected through a path. There are networks that consist of *components* in which each pair of nodes is connected through some path but which are not connected to each other. In a directed network, one needs to take into account that two nodes in a component (in the undirected sense) may not be connected through a directed edge. In a directed network, all pairs of nodes in a *weakly connected component* are connected through some path were the directions of edges can be ignored.

2.1.6 Centrality

The degree — or strength — of a node is the simplest form of measuring the node's *centrality* [24]. In social networks, centrality measures are often used to judge a person's importance or influence [29]. Clearly, the centrality of a node depends largely on the type of network one is studying [30] which has led to a vast landscape of research involving different definitions of centrality. Some assign high centrality values to nodes that are themselves connected to other influential nodes, such as *eigenvector* centrality [31] and *katz* centrality [32]. Others emphasise the extent to which a node lies on paths between other nodes (*betweenness* centrality [33]), or take into account the mean distance between a node and the nodes it is connected to (*closeness* centrality [29]). This is not an exhaustive list but is meant to give the reader an idea of the variety of existing measures.

In [34], Fowler uses an adapted version of closeness centrality to find important figures in a network of members of the US Congress; two nodes, representing Members of Congress, are connected by an edge if at some point they mutually cosponsored a bill. Fowler states that nodes he identifies as being central in this network correlate strongly with "real-life" influential politicians.

Due to the scope of this thesis and the diversity of the various methods that have been developed, we focus on a small number of centrality measures.

Eigenvector (EV) centrality is based on the idea that the importance of a node depends on the importance of its neighbours; EV centrality x_i of node i in a directed network is proportional to the sum of EV centralities of the neighbours from which i is pointed to. It is given by

$$\lambda x_i = \sum_j A_{ij} x_j, \tag{2.10}$$

which in matrix notation is $\lambda \boldsymbol{x} = \boldsymbol{A}\boldsymbol{x}$. Hence x_i is the ith element of the eigenvector of $\boldsymbol{A}$ with eigenvalue λ [35].

Equation 2.10 implies that a node with 0 in-degree will by definition have 0 EV centrality; a node that only has incoming edges from nodes with 0 EV centrality also has 0 EV centrality. It is possible for this to propagate through a network and cause a large amount of information to be lost [35]. The concept of a similar measure called Katz centrality [32] builds on the idea of EV centrality but adds a small amount of "free" centrality to each node so as to avoid the propagation of 0 centrality [24].

We introduce a third centrality measure, the *PageRank* (PR) measure which was invented and is used for Google's web rankings [36]. The PR $P(i)$ of node i is

$$P(i) = \frac{q}{n} + (1-q)\sum_j \frac{A_{ij}}{k_j^{\text{out}}} P(j), \tag{2.11}$$

which describes a random walk where the walker jumps to a random node depending on a *damping factor* q. Similarly to EV centrality, the PR of nodes is affected by the PR of the neighbours pointing to it. Unlike EV centrality, the definition of PR includes the term $1/k_j^{\text{out}}$, causing the PR of node i to increase if a neighbouring node has a high PR value but low out-degree.

We note that unless stated otherwise, the computations of network measures in this thesis are done in Python's network library `NetworkX` [37].

2.2 Communities

Another commonly studied feature of networks is that of *communities*. This is the idea that nodes of most networks are organised into groups in a way that many edges connect nodes within a group and fewer edges link nodes of different groups [38]. A common assumption is that, in real networks, nodes in the same community have common characteristics of some sort, such as, for example, pages on similar topics in a network of webpages or groups of friends in a network of members of a sports club [39]. Identifying communities is interesting as they can help us understand the general structure of a network and give insight into the importance of nodes within or between communities and the effect these have on the whole network [38].

Community detection is the idea that one can identify naturally occurring groups of nodes in a network by exploiting the topological properties of the network; however, the concept of a community is not well defined so that large numbers of methods — often tailored to specific types of networks — have been proposed in the recent decades [38]. We want to give an introduction of the general idea of a community, deliver a brief overview of existing methods and draw the reader's attention to some of the limits of these methods. In Section 5.1.1 we explain our choice of method and specify the details of it.

An intuitive description of a community, as described in [38], is a *subgraph* C (i.e. a graph made of a subset of n_C nodes and connecting edges from the original directed network G) which has higher within-community density $d_{\text{int}}(C)$ than the average density $d(G)$ of the network and a lower between-community density $d_{\text{ext}}(C)$ than the average density. Density $d_{\text{int}}(C)$ is the proportion of possible edges present within C,

$$d_{\text{int}}(C) = \frac{\#\text{ internal edges of } C}{n_C^2}, \tag{2.12}$$

where n_C is the number of nodes in community C. Similarly, $d_{\text{int}}(C)$ is the proportion of possible edges present between nodes in C and the rest of the network,

$$d_{\text{out}}(C) = \frac{\#\text{ edges between nodes in } C \text{ and } G \setminus C}{2n_C n}. \tag{2.13}$$

The idea of maximising the difference between $d_{\text{int}}(C)$ and $d_{\text{ext}}(C)$ for each community C is not a rigorous description but just a general idea behind community detection [38].

In this thesis we are interested in the detection of *partitions* which divide the network into non-overlapping groups (as opposed to ones that allow overlapping) [38]. One commonly used property of what is a "good" partition of a network is called *modularity*. The modularity Q of a partition is based on the concept that one compares the number of edges within a community to the *expected* number of edges [40],

$$Q = (\#\text{of edges in community}) - (\text{expected } \# \text{ of such edges}). \tag{2.14}$$

This requires the definition of a *null model*, which describes the number of expected edges in each community as

$$\frac{1}{2}\sum_{ij}\frac{k_i k_j}{2m}\delta(C_i, C_j), \tag{2.15}$$

where C_i is the community of node i [24]. It is based on the idea that in an undirected network with m edges there are $2m$ "ends" of edges so one expects there to be $k_i k_j/(2m)$ edges between nodes i and j. The definition of modularity in undirected networks is

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(c_i, c_j), \tag{2.16}$$

and was originally developed by Newman and Girvan [41]. It computes the difference between the edges present in each community of a partition compared to the null model. This is the definition of the undirected case; see Section 5.1.1.1 for the directed case.

Numerous other community concepts have been developed, such as those exploiting properties of random walks [42, 43] or others utilising edge centrality [39]. Countless algorithms with various different approaches have been invented to implement these methods. In this thesis, we use the locally greedy *Louvain method* that is based on the modularity of partitions; in Section we explain this choice and details on the method and its implementation.

The large amounts of data involved in recent studies of network science call for observations on *computational complexity* of the algorithms used for community detection. Some networks have millions of nodes so that algorithms with time complexity of order higher than $\mathcal{O}(n)$ or $\mathcal{O}(m)$ are not feasible for a network with n nodes and m edges [38]. Furthermore, many problems in community detection, including that of modularity optimisation as shown by [44], are *NP-hard*. Hence, one uses *heuristic* algorithms to approximate the optimal partition of a network [45].

2.3 Text mining

The process of text data mining describes the extraction of information from text bodies by applying statistical methods [46]. Common applications are the classification (supervised learning) as well as the clustering (unsupervised learning) of text documents [47]. Recent studies have used methods from text data mining to categorise demographics of social media users [21], find temporal topic structures in news articles [48] and to understand the sentiment of people's tweets [49].

Following [47], we first introduce some terms that are crucial for the understanding of text data mining. A *document* d is a section of text made up of *terms* t (i.e. words). We will call a set of u documents a *corpus* $D = \{d_1, d_2, \cdots, d_u\}$ and the collection of v unique terms in the corpus a *lexicon* $L = \{t_1, t_2, \cdots, t_v\}$. The basis of most text data mining methods is *feature extraction* which processes the corpus in a way that permits the implementation of mathematical methods. In most cases this proceeds the cleaning of the text data into processable form; we describe how this is done in Section 4.3.

2.3.1 Feature extraction

2.3.1.1 Tf-Idf

An underlying concept of feature extraction is the encoding of the terms in documents [50]. Intuitively, one seemingly should be able to express each term based on its *frequency*. However, this assigns importance to common English words, such as "a" and "the". A popular weighting method to avoid this is *term frequency inverse document frequency* (tfidf) [51]. It calculates the term frequency (TF) of term t in document d as

$$\mathrm{TF}_{td} = \frac{N_{td}}{\sum_k N_{kd}}, \tag{2.17}$$

where N_{td} is the number of times the term t occurs in document d and $\sum_k N_{kd}$ is the total number of terms in the document. The inverse document frequency (IDF) discounts terms that are very frequent but not significant. For every word t, we compute

$$\mathrm{IDF}_t = \log \frac{|D|}{|D_t|}, \tag{2.18}$$

where D_t is the subset of the corpus that contains term t. As it is just used for scaling, the log base is not important in this context [52] although most commonly base 10 is used. By definition, the IDF value of a term that is used in every text document is 0. Intuitively, tfidf gives high weightings to terms that are frequently used in a document but that do not appear frequently in the corpus [22].

2.3.1.2 Bag-of-words and vector space model

Tfidf-weighted terms can then be used to encode documents based on the *bag-of-words* model [53] in which each document is represented by an unordered collection of its weighted terms. Bag-of-word representations can then be used in a vector space model [54] to create a $v \times u$ *term-document matrix* X in which the u columns represent documents and the v rows represent all words in lexicon L. The rows of this matrix are often called a *sample* as it represents each sampled document whilst each non-zero value in these columns is called a *feature* [50]. Each element x_{ij} represents the tfidf weighting of the ith term in the jth document [47].

In this thesis we use a clustering method called *k-means* [55] on a data set from Twitter conversations to extract information on topics appearing in these conversations, similar to what was done in [56]. In Section 5.2.1 we explain the ideas of clustering text data as well as the methodology behind the k-means algorithm.

These mathematical and methodological preliminaries lay out the basis for methods and analyses to come in following sections. All computations were done in either Matlab or in Python's network package NetworkX [37].

Chapter 3

Related Work

3.1 Twitter networks

As mentioned in Section 1, Twitter provides multiple types of interactions between users; users can tweet short messages, *follow* other users to see those users' content on their timeline, *retweet* other people's content, *reply* to other user's tweets by including that user's Twitter *handle* (i.e. username) in the beginnning of a tweet, or *mention* another user by including the handle somewhere else in a tweet. Additionally, *hashtags* are used to emphasise the topic of a tweet. As a result of this versatility, researchers have created networks from Twitter data in various different ways. Possibly the most obvious choice is the "follower-network" in which nodes represent users and directed edges connect users if one user *follows* the other. An example can be found in [57], where the evolution of a follower-network is investigated to gain insight into the spread of information. A very different type of Twitter network can be found in [23], where two networks were constructed in which nodes represent hashtags that are connected by edges if they co-occurred in at least one tweet. Other types of Twitter networks make direct use of the mentioned interactions that are possible between users [10, 20, 58]. In [20], for example, nodes serve as Twitter accounts of Members of the European Parliament (MEPs) with edges connecting them representing retweets. Conover et al. created a similar network in which nodes represent Twitter users that were involved in political discussions and retweets and mentions form the joining edges [10]. A number of studies have suggested that interactions between users provide a better platform for the spread of information on Twitter than plain follower-networks [59]. Hence, in this thesis, the type of network formed will be similar to the last example; we will explain the exact formation of this network in Section 4.2.

Different possibilities of creating Twitter networks imply different methods to examine them. Centrality measures, for example, can provide insight into completely different matters depending on the type of Twitter network. Clearly, nodes that are found to be central in a hashtag network as created in [23], for example, give insight into topics of Twitter conversations. In [58], however, important nodes represent users that are central to the spread of messages. The latter type of centrality is of particularly interest for the analysis of political Twitter conversation as influential users have the power to spread opinions and ideas as well as potentially stop the spread of rumours or other pieces of information [60]. Accordingly, we investigate the latter type of centrality measure in this thesis. In Section 5.1.2 we explain the choice of centrality measure for our network as well as the implementation. We hope that finding individuals with influential roles in the Brexit debate could shed light on the way information on the EU referendum was spread through the Twitter network.

As mentioned in Section 2.2, the detection of communities in a network can give insight into general structures [38]. The study in [10] examined Twitter users that created political tweets in the weeks before a midterm election in the US congress. Using an algorithm that finds exactly two communities, Conover et al. investigated whether or not these communities of users corresponded to groups with shared political inclinations. Other studies have implemented algorithms that do not require a predefined number of communities for the network to be partitioned into [20, 61]. In [20], Cherepnalkoski and Mozetič identify communities within the retweet network of the MEPs by using the locally greedy Louvain method which approximately finds the partition with highest modularity (see Section 5.1.1.3). The findings in this paper suggest that the detected communities correspond well with node labels such as party membership or country of origin.

Various different types of studies have been done on examining identified communities. Whilst Cherepnalkoski and Mozetič [20] use what they call the B^3 measure to asses how well their communities corresond to metadata of nodes, Traud et al. [62] use the Rand similarity to determine whether communities amongst Facebook friendships between US college students correlate with information on students programme of study, year of graduation and dormitory.

A lack of this type of metadata requires other techniques to identify characteristics of communities. Analysing unknown Twitter users — as is the case in this thesis (see Chapter 4 for a description of our data) — motivates some sort of "artificial" labelling of nodes. This is where the analysis of text documents, such as tweets, can be of aid. An example for this is the work in [22], where text data mining and machine learning methods are used to identify the political opinion of users. The authors claim a correlation of the political alignment of users (separated into left- and right-wing users) with the two communities detected in the network. As suggested in [63], another way of labelling nodes to find themes in communities is by utilising the text blocks provided by *Twitter biographies* — the small descriptions every Twitter user can create to inform about themselves.

3.2 Twitter data mining

Using tools from network science is only one approach of analysing Twitter data.

In the second quarter of 2016, Twitter had 313 million users were active on Twitter each month as reported by the statistics portal Statista [64]. And, according to the site ranking service Alexa[1], Twitter has continuously been amongst the top ten most popular websites worldwide. These statistics, as well as the nature of the platform, suggest an exceptional opportunity for researchers with regards to analysing the content of what people debate online. The high volume and heterogeneity of the text content as well as noisiness of the data require advanced tools from text data mining as well as language processing so that one can make sense of the information obtained [65].

Several studies have used these types of tools to understand Twitter conversations, the topics therein, as well as unexpected events [16]. Other studies, such as [66], attempted to identify the sentiment of tweets by classifying words based on the famous "hedonometer", a mixture of machine learning techniques and human assessment [49]. With the British Polling Council investigating in the failure of the prediction of the 2015 UK General Election [67] and many claiming that traditional polls in general are increasingly failing [68], researchers are looking for alternative predictive methods and are repeatedly using Twitter for this purpose [16]. Other studies have tried to find general topic areas from tweets. In [56], Villiers et al. implement machine learning methods to create Twitter topic "lists" by comparing different techniques of document representation as well as various methods to cluster topics.

Mentioned related work on Twitter data using network science, text data mining or the combination of both motivated us to apply similar methods to a collection of tweets related to the EU referendum. We investigate both the structure of networks based on Twitter data and examine actual topics of conversations discussed in the

[1] ©2016, Alexa Internet (`www.alexa.com`)

tweets. We quantify themes evolving in these debates by attempting to detect patterns in the textual data by using a clustering algorithm similar to the work done in [56].

3.3 Limitations

When analysing Twitter data — and especially when trying to draw conclusions from Twitter debates to communicational behaviour offline — one needs to be careful about a number of issues. First, common knowledge tells us that the nature of Twitter conversations, e.g. the 140 character limit and the fast pace that goes along with it, compared to other social media platforms may cause a bias in itself as it stimulates a certain type of debate. Second, one needs to note the sampling bias induced by trying to represent the "real world" by a Twitter population, which was studied in [69]. For example, the authors stated that Twitter samples over-represented male users from populous areas at the time of data collection (2011). Thirdly, there is a considerable number of bots that create tweets and retweets and that are thus somewhat involved in the shaping of public opinion through Twiter, as suggested in the paper by Howard and Kollanyi [70]. They warned that this type of automation occurs mainly through retweets, rather than replies.

Additionally, it is important to draw the reader's attention to an important message from Cihon and Yasseri [71] on network science in general. The immersion of mathematicians, computer scientists and physicists into the world of social science through the connection of statistics and the study of complex networks is a fairly recent thing. Explaining social phenomena requires a solid background in not only handling and analysing data but also theories and methodologies from the areas of social sciences. Thus, we point out that we are careful when interpreting and drawing conclusions from our results.

Chapter 4

Data

4.1 Data collection

To analyse Twitter Referendum conversations and to form a network we first collect appropriate data to create a representative sample. Unlike other social media platforms, Twitter allows comparatively unrestricted access to their data; one of their services is a free and open-to-all but limited Streaming API [72]. The API caps the data collected according to some search terms when it reaches 1% of the total tweets on Twitter and then samples those tweets. To avoid this limitation one can use the Twitter Firehose, which is an unlimited but paid-for service [73]. Morstatter et al. [74] investigated the extent to which data collected through the Streaming API is able to represent a complete data set obtained by the Twitter Firehose. They suggested that it depends on the search terms that are fed into the Streaming API, leading to different amounts of coverage. Additionally, Twitter's sampling on the API is unknown. Morstatter et al. claimed that it performs worse at representing the full data set than types of sampling that they tried themselves [74]. The data set for this thesis, accessed by the Twitter Firehose, was generously provided by our collaborator from the data science company Sinnia[1] Guillermo Garduño who we are most grateful to.

[1] Sinnia, Social Data Scientists (`www.sinnia.com`)

Hashtags			
Brexit	Remain	Imleavebecause	Incampaign
Euref	BetterOffOut	BeLeave	NotoEU
VoteLeave	UKinEU	LabourLeave	VoteIn
LeaveEU	LabourInForBritain	RemainInEU	LoveEuropeLeaveEU
Eureferendum	VoteOut	LabourGo	UKinEurope
StrongerIn	VoteRemain	StayinEU	YestoEU
No2EU	StopTheEU	VoteStay	LeaveChaos
TakeControl	Bremain	Yes2EU	BritainOut

Table 4.1: List of hashtags used to collect data.

Crucial to any form of data collection from Twitter is the provision of suitable search terms, such as hashtags, users or locations [74]. Because hashtags are often used to assign tweets to certains topic [75] we create a list of hashtags that were commonly attributed to tweets discussing the EU referendum. We compiled the hashtag list based on ideas inspired by [76] as follows: 1. We feed the accounts of the official Leave and Remain campaigns into a website called `www.twitonomy.com`, which displays the top-10 hashtag used by the specified Twitter account at the given time. 2. We run the output from Step 1 through `hashtagify.me`, a website that shows the all-time (up to the given time of input) top-10 hashtags that are related to the input hashtag by appearing in the same tweets. Taking the above steps and removing hashtags that are too general (such as #UK, #EU or #BBC) or off-topic (such as #exeter or #legendary) we compiled a list of 32 hashtags (see Table 4.1) to be used as search terms for our data collection.

The data was collected for five weeks from 27/05/2016 (four weeks before the referendum) to 30/06/2016 (one week after the referendum). In these five weeks we received a total number of $10,535,674$ tweets which included at least one of the hashtags from the list in Table [17].

Table A.1 in Appendix A shows the numbers of tweets created in our five-week time window separated by the hashtags from Table 4.1. As people often use multiple hashtags a significant overlap is likely. To reduce the size of our data set, we focus on the five hashtags which produced the largest amounts of tweets, namely *#EUref*, *#Brexit*, *#VoteLeave*, *#VoteRemain* and *#EUreferendum*. The numbers of tweets that included these hashtags are displayed in Table 4.2.

Hashtags	Number of tweets
EUref	3620465
Brexit	2787365
VoteLeave	1805621
VoteRemain	608176
EUreferendum	544908

Table 4.2: Hashtags used in the largest number of tweets.

Due to the large amount of data — and to investigate a change over time — we look at the tweets split into five weeks. The names we give these weeks and the exact time spans are shown in Table 4.3.

Week 1	**Week** 2	**Week** 3	**Week** 4	**Week** 5
27/05/2016 – 02/06/2016	03/06/2016 – 09/06/2016	10/06/2016 – 16/06/2016	17/06/2016 – 23/06/2016	24/06/2016 – 30/06/2016

Table 4.3: Weeks in which Twitter data was collected.

4.1.1 Size of data set

We note at this point that the large size of the obtained data set meant that a large amount of time was spent on preprocessing the data. Both computations to form networks from these data sets (see Section 4.2) as well as the text processing (see Section 4.3) required a considerable amount of effort and time. Additionally, despite our efforts in choosing algorithms with appropriate time complexity, a lot of the computations that we introduce in Chapter 5 have long running times. Accordingly, the results in Chapter 6 are the outcome of multiple trials and obtaining the final results thus took up a large proportion of the time we spent on this thesis.

4.2 Network formation

We construct the networks from the tweets containing any of the top-5 hashtags as follows: For each week we create two separate networks in which nodes represent Twitter users. In the *retweet networks*, there is an edge from user i to j for each time user i has retweeted content created by user j at least once in the specific week. Similarly, in the *reply networks*, edges point from user i to j for each time user i has replied to or mentioned user j at least once in the specific week by including user j's user handle (username) anywhere in the tweet. Following [10], we include both mentions and replies in the *reply network* based on interpreting the "@" symbol as an indicator of addressivity [77].

Recall from Section 2.1 that this type of network is called a multigraph as several retweets or replies from one user to another induce multiple edges between those users. By summing up the edges between each pair of nodes we produce a weighted graph. It is important to note at this point that intuitively networks constructed from tweets are not in fact *static* networks but rather *dynamic* networks because tweets are created at different points in time which can cause the bias [27] mentioned in Section 2.1. It is possible to represent these types of networks as *temporal* networks [78] in which the timings of the forming of edges can yield interesting insights. In some cases, projecting a temporal network onto a static network (by summing up edges) is an ineffective representation of the topology of a temporal network, such as when modelling the spread of diseases and immunisation [79]. On the other hand, Twitter networks in particular have been represented by static networks numerous times and have been the basis for interesting insights into the Twitter communities [10, 20, 58]. For reasons of simplicity, we thus go ahead with this representation.

From these weighted networks, we find the largest weakly connected component (see Section 2.1.5) and find that in all cases it heavily dominates other weakly connected components. For simplicity, every network we mention in the rest of this thesis is actually the largest weakly connected component of the same network. Table 4.4 shows the number of nodes and edges in the largest weakly connected component of the weekly reply and retweet networks.

	Week 1		**Week** 2		**Week** 3		**Week** 4		**Week** 5	
	Nodes	Edges	Nodes	Edges	Nodes	Edges	Nodes	Edges	Nodes	Edges
Retweet	121942	327465	182921	519436	230090	693492	729852	1886352	403794	554344
Reply	5856	9097	8227	13269	10998	18413	28520	38807	345	363

Table 4.4: Number of nodes and edges in the largest weakly connected component of each weekly retweet and reply network.

Additionally to the weekly networks, we construct separate retweet networks and reply networks for each of our top-5 hashtags. We create these similarly to above but by only including edges which contain the specific hashtag for each of the networks. The number of nodes and edges in these hashtag networks can be seen in Tables B.1 and B.2 in Appendix B. Generally, the reply networks are much smaller than the retweet networks. Also, both the retweet and the reply networks grow in size from weeks 1 to 4 and decline in size in week 5. Intuitively, we expect that different hashtags might be used *after* the announcement of the referendum result, which is a

Data	**Action**
Punctuation	REMOVE
# word	REMOVE
@ username	REMOVE
RT	REMOVE
URLs	REMOVE
uppercase letters	LOWERCASE

Table 4.5: Cleaning of unwanted data.

possible explanation for the decline in week 5. The reply networks experience a much larger decrease in size between weeks 4 and 5 than the retweet network. See Section 6.1 for details on the network growth.

The most striking difference between the network sizes of the retweet networks and those of the reply networks can be seen in week 5, where the reply debate decreases heavily. Keeping in mind that one might expect a general decline in conversations (as different hashtags may have been used after the results of the referendum), it is interesting to see that this decline.

4.3 Text processing

As mentioned in Section 3.2 we are interested in not only the network structures but also textual actual content of our data. Before applying the feature extraction techniques introduced in Section 2.3, we “clean” text documents from any unwanted data. As seen in [80], we write a Python script which processes our raw data according to Table 4.5. Other pieces of data that are intuitively not of interest are common English words such as “the”, “and” and “is”. We remove these words by using the Python package NLTK (Natural Language Toolkit) [81], which removes a pre-installed list of words.

We use the cleaned text data to further process documents into a form that can be used by text mining methods. We are interested in the following two types of documents: Tweets which were posted by users in our Twitter networks and which are either replies or retweets as well as Twitter biographies created by these users. Following the approach described in Section 2.3 we create bags-of-words using tfidf weighting for the documents in the text corpora of twitter biographies D_B and tweets D_T made of words from lexica L_B and L_T. Each document d_{Ti} in $D_T = \{d_{T1}, d_{T2}, \cdots, d_{Tu}$ contains all words contained in any of the tweets posted by user i. Based on this, we create term-document matrices X_T and X_B for the tweets and Twitter biographies respectively, so that rows are documents (i.e. tweets and biographies respectively) and columns are the terms in the lexica. Element x_{Bij} in X_B represents the tfidf weighting of the ith term in the Twitter biography of user j. Similarly, element x_{Tij} in X_T represents the tfidf weighting of term i in Tweets by user j. We construct these matrices, separately for each week, by feeding the cleaned text documents into the `TfidfVectorizer`, which is built into Python's machine learning toolbox `scikit-learn` [82].

Chapter 5

Methods

5.1 Network-based methods

5.1.1 Community detection

As illustrated in Section 3.1, identifying communities of Twitter users has been shown to yield interesting results in past studies. Twitter communities of Members of the European Parliament (MEPs) [20] and of US college students [62] showed remarkable correlations with metadata. For example, communities of MEPs matched their party membership and college students were likely to be in the same communities with others from their graduation year. These findings suggest that detecting communities in our Twitter networks may give interesting results about the structure of Brexit conversations. We hope to gain insight into the extent to which Twitter users involved in the debate primarily communicate with users of similar characteristics or within certain topic areas. We thus try to identify communities of Twitter users within both the reply and retweet networks we created from our Twitter data; we later use text analysis to understand user characteristics and topics based on tweets and Twitter biographies respectively.

5.1.1.1 Directed and weighted modularity

In Section 2.2 we described briefly the issues and limitations of community detection as well as the choice of different approaches. A great number of scientists have shown that methods utilising the modularity [41] of partitions outperformed that of other measures. The results in [83], for example, approaches utilising modularity optimisation outperform 14 other methods of community detection. Due to the popularity of this approach approach and the vast number of applications that have illustrated its success, we, too, choose a community detection method that is based on modularity.

We recall that the original definition [41] of the modularity of a partition (see Section 2.2) for an undirected, unweighted network can be described as

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j). \tag{5.1}$$

I.e. a "good" partition of network is one in which communities have more internal edges than we would randomly expect.

Equation 5.1 needs to be adjusted for our directed and weighted networks. The direction of an edge changes the null model, because the probability of an edge going from a node with high out-degree but low in-degree to one with the opposing properties is different than one running the other way around. In the directed case, one can describe the expected number of edges from node j to node i as $k_i^{\mathrm{in}} k_j^{\mathrm{out}}/m$ [84]. Newman and Leicht used this definition to introduce the modularity Q_d for a directed network as

$$Q_d = \frac{1}{m} \sum_{ij} \left(A_{ij} - \frac{k_i^{in} k_j^{out}}{m} \right) \delta(c_i, c_j), \tag{5.2}$$

which we can rewrite for weighted networks [85] as

$$Q_{\mathrm{dw}} = \frac{1}{\mathbf{w}} \sum_{ij} \left(W_{ij} - \frac{w_i^{in} w_j^{out}}{\mathbf{w}} \right) \delta(c_i, c_j). \tag{5.3}$$

5.1.1.2 Modularity optimisation

From here, any algorithm using modularity optimisation can use this adapted definition of modularity instead of the classic undirected version. This is generally done by assigning a modularity Q score to each partition and identifying the "optimal" way of decomposing the network that has the maximum modularity score. Despite the fairly simple description of the concept behind community detection, finding the global maximum modularity over all possible partitions has been shown to be an *NP-hard* problem [44]. It is beyond the scope of this thesis to give an indepth review of the issue of NP-hardness in this context. A brief summary is given by Duch and Arenas [86]; they explain that with increasing network size, the number of partitions one can divide the network into grows faster than by any power of the network size. Many studies, such as [87], have achieved reasonable results by using approximate techniques.

It is important to remark that some studies have expressed warnings on approximating the highest modularity partition of a network [88, 89, 90]. Good et al. [90] conducted a thorough examination of several properties of the modularity function and uncovered near degeneracy issues — they demonstrated that the optimal modularity decomposition of a network is one amongst exponentially many partitions of very similar modularity values. This and other issues prompt us to be cautious when interpreting any results that we obtain from algorithms the maximum modularity of a partition.

5.1.1.3 Louvain method

Heuristic algorithms based on modularity optimisation include spectral optimisation [91], simulated annealing [92] and greedy techniques [87, 85]. We use the popular locally greedy *Louvain* method [85], which does not required a pre-defined number of communities and is known to yield good results as well as to be computationally faster than most other algorithms [38, 93, 94].

The basic idea behind the Louvain method is split into two phases. In phase 1, each node is assigned its own unique community. One then considers each of the n nodes in a random order, moves it to the communities of each of its neighbours and computes the change in modularity resulting from these moves. The node is then moved to the community for which the modularity gain was largest, unless the modularity increases for none of the options, in which case the node stays in its original community. One then repeats this process for all nodes up to the point at which there is no more gain in modularity; one obtains a local maximum is reached and phase 1 is finished. This is the first level partition. In phase 2, one constructs a reduced network in which the communities detected in phase 1 serve as the nodes of this new network, with the weight of the edges between the communities of level one summing up as the edges between the new nodes. One applies phase 1 on the new nodes until a second level partition is reached. One repeats phases 1 and 2 until a partition is obtained for which moving nodes to different communities does not induce a positive modularity change. For every partition one obtains at each level, one computes the overall modularity of this partition so that the one with the highest overall modularity value can be chosen [85].

Blondel et al. designed the Louvain method with the aim of creating an algorithm efficient enough for the handling of large networks. As has been confirmed repeatedly [38, 93, 94], the computational complexity of the Louvain method is essentially of order $\mathcal{O}(m)$, where m is the number of edges of the network. This linearity of computational complexity is based on the calculation of the modularity gain ΔQ resulting from removing node i from its old community to community C. For a weighted network, it is described by

$$\Delta Q = \left[\frac{\sum_{\text{in}} + w_{i,\text{in}}}{2\mathbf{w}} - \left(\frac{\sum_{\text{tot}} + w_i}{2\mathbf{w}}\right)^2\right] - \left[\frac{\sum_{\text{in}}}{2\mathbf{w}} - \left(\frac{\sum_{\text{tot}}}{2\mathbf{w}}\right)^2 - \left(\frac{w_i}{2\mathbf{w}}\right)^2\right]. \tag{5.4}$$

Here, $\mathbf{w}$ is the sum of all edge weights in the network whilst the sum of weights of edges inside community C is $\sum_{\text{in}}$. w_i is the strength of node i, $\sum_{\text{tot}}$ is the sum of the weight of all edges connected to nodes in C and $w_{i,\text{in}}$ is the total weight of edges between node i and other nodes in C. Equation (5.4) can be rewritten [95] as

$$\Delta Q = \frac{w_{i,\text{in}}}{2\mathbf{w}} - \frac{\sum_{\text{tot}} \cdot w_i}{2\mathbf{w}^2}. \tag{5.5}$$

For the directed version, Dugué and Perez [95] suggested changing Equation (5.5) to

$$\Delta Q = \frac{w_{i,\text{in}}}{2\mathbf{w}} - \frac{w_i^{\text{out}} \cdot \sum_{\text{tot}}^{\text{in}} + w_i^{\text{in}} \cdot \sum_{\text{tot}}^{\text{out}}}{\mathbf{w}^2}, \tag{5.6}$$

where $\sum_{\text{tot}}^{\text{out}}$ is the sum of the weight of all out-going edges from nodes in community C.

We use a Matlab implementation of the Louvain method for weighted and directed networks by Antoine Scherrer [96].

5.1.2 Centralities

We want to find central nodes according to PageRank centrality to describe important nodes in the network as a whole as well as in communities. We recall from Section 2.1.6 that the PageRank of a node i is computed by

$$P(i) = \frac{q}{n} + (1-q)\sum_j \frac{A_{ij}}{k_j^{\text{out}}} P(j). \tag{5.7}$$

Finding the PageRank for all nodes $i = 1, 2, \cdots, n$ is an eigenvalue problem for the transition matrix $\mathcal{P}$ [35] with entries

$$\mathcal{P}_{ij} = \frac{q}{n}(1-q)\frac{1}{k_j^{\text{out}}} A_{ij}. \tag{5.8}$$

The vector with entries being PageRank values for each node in the network is then the leading eigenvector of $\mathcal{P}$. The most common way of computing the leading eigenvector of a matrix $\mathcal{P}$ is the *power method* [97]. The basic idea behind the power method is that, starting with an initial non-zero guess for an eigenvector $x^{(0)}$, one computes

$$x^{(k+1)} = \frac{\mathcal{P}x^{(k)}}{\|\mathcal{P}x^{(k)}\|} = \frac{\mathcal{P}^k x^{(0)}}{\|\mathcal{P}^k x^{(0)}\|}. \tag{5.9}$$

We refer to [98] for a general introduction to the power method and to [99] for details on its application for the PageRank. To compute PageRank values we use a built-in PageRank function in Python's NetworkX package which utilises the power method.

5.2 Text-analysis-based methods

As mentioned in Section 3.2, we are interested in extracting information about different topics in our Twitter data set. Due to the lack of metadata on the Twitter users in our data set, we expect that it is appropriate to use methods from *unsupervised classification*, also called *clustering.* The general idea of clustering is to divide a data set into a collection of groups based on some sort of similarity or distance measure; data points within groups should be more "similar" or "closer" to each other than to data points in other groups (similarly to the problem of community detection, there is no fixed definition) [100]. Various clustering algorithms have been successful in uncovering structures in Twitter data sets [56, 101, 102, 103]. In [102], for example, Vicente et al. use different clustering algorithms to detect the gender of Twitter users. The work in [103], suggests that it is possible find clusters of different types of customers on Twitter. More generally, clustering algorithms have been used numerous times to find general topic clusters in Twitter conversations [56, 101].

5.2.1 K-means clustering

Two different types of clustering are *partitional* clustering, the partition of the data set into non-overlapping groups, and *hierarchical* clustering, the division of the data into nested tree-like partitions [100]. We use a partitional clustering approach because algorithms which detect hierarchical partitions are reportedly slower [104]. One of the oldest and most popular partitional clustering algorithm is the *K-means algorithm* [55]. We choose a version of K-means called *Lloyds algorithm* because of its simplicity and suitability for large data sets [100]. This is due to its time complexity of $\mathcal{O}(vKu)$, where the number of clusters K and the number of documents u are commonly much smaller than the number of terms v. It has delivered reasonable results for the clustering of documents in the past [104].

Recall that our data points are u v-dimensional documents in corpus $D = \{d_1, d_2, \cdots, d_u\}$. Applying clustering to a data set requires a choice of measurement for the distance or similarity between any pair of data points [100]. Famous measures are Euclidean distance, cosine similarity, Pearson correlation coefficient and the Kullback–Leibler Divergence to name just a few [56]. We choose the Euclidean distance measure, which is most commonly used for for the K-means algorithm [55]. The distance between two documents d_i and d_j is then

$$\delta(d_i, d_j) = \sqrt{\sum_{h=1}^{v} (d_{ih} - d_{jh})^2}, \tag{5.10}$$

, where v is the number of words corpus D and therefore the length of vectors representing each document [56].

We want to assign each of our documents to one of K clusters $\mathcal{C}_\kappa$, $\kappa = 1, \cdots, K$. The idea of the K-means algorithm is to find clusters so that the sum of the squared distances δ between the points (i.e. documents) and the centroid μ_κ of each cluster $\mathcal{C}$ is minimised. Minimising this objective function,

$$J(\mathcal{C}) = \sum_{\kappa=1}^{K} \sum_{d_i \in C_\kappa} \delta(d_i, \mu_\kappa)^2, \tag{5.11}$$

is an NP-hard problem, as shown in [105]. The Lloyds algorithm is a popular heuristic algorithm that tries to approximate the K-means clustering problem by finding a local minimum of the objective function in Equation 5.11 [106]. The idea of this algorithm is the following [55]: Initially, K random cluster centroids are selected, and then each data point is assigned to its nearest cluster centroid. The new centroid is then calculated for each cluster, points are assigned to those new centroids, and the process is repeated until the cluster membership is stable, i.e. computing new centroids of each cluster results in already existing centroids.

Because the heuristic Lloyds algorithm converges to a local minimum of the objective function, different initial starting centroids can lead to different local minima. Jain [100] suggested running the algorithm repeatedly and choosing the result that gives the smallest squared error partition for optimal results. In [107], Arthur and Vassilvitskii formulated a randomised technique for the initial choice of centroids to improve the accuracy and speed of the Lloyds algorithm. This adaptation, called *K-means++* chooses the first centroid uniformly at random from all data points; it then samples the other starting centroids with probability proportional to their distance from the nearest centroid to avoid the inital centroids from being too close to each other [108].

5.2.2 Silhouette scores

The K-means algorithm requires as an input the number of clusters K it is meant to find in the data. However, we do not know this number in advance for our Twitter data set. The challenge of finding the optimal way to determine K has been of considerable interest to researchers for many years [109, 110, 111]. We use a measure named *Silhouette* values, which was designed to determine the distances of data points to other points in the same cluster compared to those to points in other clusters [112].

Rousseeuw [112] describes the Silhouette measure as follows. Assume that a cluster $\mathcal{C}$ contains the data point d (i.e. one of our text documents). Now define $\alpha(d)$ to be the mean distance between point d and all other points in $\mathcal{C}_\lceil$ and let $\gamma(d, \mathcal{C}_\kappa)$ be the mean distance between d and the points in another cluster C_κ. We also define $\beta(d) = \min_\kappa\{\gamma(d, \mathcal{C}_\kappa)\}$ to be the distance between point d and the cluster that is nearest to d's cluster. One can then define the Silhouette value $s(d)$ of point d as

$$s(d) = \frac{\beta(d) - \alpha(d)}{\max\{\alpha(d), \beta(d)\}}. \tag{5.12}$$

The quantity $s(d)$ can take values between -1 and $+1$, where -1 indicates that d was wrongly assigned to its cluster and $+1$ suggests that d is very far away from neighbouring clusters. A Silhouette value of 0 indicates is on the "boundary" between two clusters [113]. Rousseuw [112] suggested that one can compute the mean of the silhouette scores of one partition of K clusters and compare it to the mean silhouette scores of other partitions.

5.2.3 A dimensionality problem

Because we are working with data in a high-dimensional space — induced by the number of different terms in the corpora of tweets — the distances between data points are large. This raises two central issues when implementing a clustering algorithm such as K-means. Both of these issues are linked to what Richard Bellman first introduced as the "curse of dimensionality" [114], which stems from the fact that higher dimensionality of a certain unchanging number of data points causes an increase in sparcity of the data set [115].

First, the "curse of dimensionality" can have an effect on the K-means algorithm, as it is possible for high dimensionality to influence the distance measures on which clustering algorithms rely [116]. To evade this problem and improve the results of clustering algorithms on high-dimensional data, Steinbach et al. [115] explained the approach of dimensionality reduction such as principal component analysis (PCA) (see Section 5.2.4 for a definition of PCA). On the other hand, one can find studies whose results yield doubts on whether or not a reduction of dimensionality would actually improve the outcome of the clustering of data such as our tweets and Twitter biographies. Yeung and Ruzzo [117], for example, studied five large data sets and found that using clustering algorithms on dimensionality-reduced data by using PCA often does not change or may in fact lower the quality of the clusters. It thus seems reasonable to attempt a dimensionality reduction on our Twitter data, investigate the outcome, and then implement the K-means clustering algorithm on both the reduced and the high-dimensional dataset to find out how beneficial PCA is to our data set and to the quality of the clusters.

Second, the generally large distances in our high dimensional data are likely to lead to very low silhouette scores, making it difficult to interpret the mean silhouette scores of each partition as well as the silhouette plots (see Section 6.5). To have a second point of reference, we thus follow a suggestion in [118] and visualise our data clusters by using principal components to project the data on a lower dimensional space.

5.2.4 Principal component analysis

Bishop [119] gave a comprehensive definition of principal components. Note that we use bold symbols for vectors in the equations in this section for clarity. The aim of PCA is to highlight the prominent directions of the data by mapping the v-dimensional vectors in our text corpus D, representing the bags-of-words of our documents, to a lower-dimensional space F by performing a linear transformation. More precisely, we want to map a v-dimensional vector $\mathbf{d}$, onto an F-dimensional vector $\mathbf{z}$, whose elements are $(z_1, \cdots, z_F)$.

We can write vector $\mathbf{d}$ as a linear combination of v orthonormal basis vectors $\mathbf{b}_i$, such that $\mathbf{d} = \sum_{i=1}^{v} z_i \mathbf{b}_i$. The idea is that we can then approximate vector $\mathbf{d}$ by vector $\tilde{\mathbf{d}}$, which can be described by

$$\tilde{\mathbf{d}} = \sum_{i=1}^{F} z_i \mathbf{b}_i + \sum_{i=F+1}^{v} \xi_i \mathbf{b}_i. \tag{5.13}$$

The constants ξ_i replace the coefficients of the basis vectors $\mathbf{b}_i$ which are not in the subset of F basis vectors $\mathbf{d}_i$ included in the first part of the right-hand side of Equation (5.13).

The squared error which is caused by the dimensionality reduction to $F < v$ dimensions, is described in [119] as

$$E = \frac{1}{2} \sum_{\mathbf{d} \in D} \|\boldsymbol{d} - \tilde{\boldsymbol{d}}\|^2 = \frac{1}{2} \sum_{\mathbf{d} \in D} \sum_{i=F+1}^{v} (z_i^{\mathbf{d}} - \xi_i)^2, \tag{5.14}$$

where $z_i^{\mathbf{d}}$ is the ith element of the vector $\mathbf{z}$ that vector $\mathbf{d}$ is mapped onto. The error E in Equation (5.14) can be rewritten as

$$E = \frac{1}{2} \sum_{i=F+1}^{v} \mathbf{b}_i^T \Sigma \mathbf{b}_i, \tag{5.15}$$

where $\Sigma = (\boldsymbol{d} - \bar{\boldsymbol{d}})(\boldsymbol{d} - \bar{\boldsymbol{d}})^T$ is the covariance matrix of $\boldsymbol{d}$.

It is beyond the scope of this thesis to show that E is minimised with respect to basis vectors $\boldsymbol{b}_i$ if

$$\Sigma \boldsymbol{b}_i = \lambda_i \boldsymbol{b}_i. \tag{5.16}$$

Hence, the basis vectors $\boldsymbol{b}_i$ — called the *principal components* — are the eigenvectors of the covariance matrix. From Equation (5.15) we can now write error E as $E = \frac{1}{2} \sum_{i=F+1}^{v} \lambda_i$.

Based on the above derivation, an algorithm performing a PCA computes the covariance matrix as described above and find its eigenvectors and eigenvalues. Our document vectors $\mathbf{d}$ are then projected onto the eigenvectors corresponding to the F largest eigenvalues, creating new F-dimensional vectors $\mathbf{z}$.

5.2.5 Twitter topics

Python's machine learning toolbox `scikit-learn` provides built-in implementations of the K-means algorithm (Lloyds algorithm version), of the computation of Silhouette values and of PCA. We use these tools to apply the methods described in Sections 5.2.1 to 5.2.4 to our data set as follows:

We recall that from Section 4.3 we obtained the tfidf-weighted term-document matrices representing both tweets as well as Twitter biographies. For each week, we remove all users (i.e. columns) from the term-document matrices which are *not* in the set of nodes of the Twitter network of that week. We first perform a PCA on each data set and evaluate whether or not a reduction of dimensionality is beneficial. For each week separately, we then apply the K-means algorithm with $K = 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30$ to identify clusters of users in both representations based on Euclidean distances. We choose the K-means++ approach and set the number of times the algorithm should run with different initial centroids to be 20. We then compute the Silhouette values of the data points in all clusters for each of the partition caused by the different number of clusters K. We use these Silhouette values, alongside a visualisation of the first two principal components, to identify which K is most appropriate. With these steps, we aim to detect user clusters based on tweets created by these users and based on the users' Twitter biography; from now on, we will refer to these as *tweet clusters* and *bio clusters* respectively.

5.3 Comparative methods

5.3.1 Cluster similarity

The methods in Sections 5.1.1 and 5.2.1 provide the basis for two different types of partitions of our Twitter data, one of network communities and one based on text clusters. To emphasise this, we assign three different labels to each node in our Twitter networks. We designate one label according to the network community; a second label describes which tweet cluster the node is part of; and the third label is assigned based on the node's bio cluster.

We are interested in the extent to which these partitions correspond. The work done by Traud et al. in [62], suggested a method of finding the correlation between a network partition induced by community detection and a second partition based on node labels from meta data. More precisely, they try to determine the similarity of two partitions of a network of US college students. One partition is induced by network communities identified by the Louvain method; the other partition is caused by meta data of the nodes (i.e. students) such as dormitory residence or graduation year.

Traud et al. illustrated the *standardized pair counting* method to perform a quantitative analysis of the two partitions. To simplify this explanation, we will say *group* when we talk about communities *and* text clusters in the following paragraphs. The underlying idea is the following: Out of all nodes, one draws pair of nodes and checks in both partitions whether or not the two nodes are members of the same group or not. Each node pair is then counted as part of either w_{11} (pair in the same group in both partitions), w_{10} or w_{01} (pair in the same group in one partition and in different groups in the other partition), or w_{00} (pair in different groups in both partitions). The total number, $\mathcal{N}$, of node pairs is then $\mathcal{N} = w_{11} + w_{10} + w_{01} + w_{00} = \binom{n}{2} = n(n-1)/2$; n is the total number of nodes in the network. They use these counts to create similarity scores of the two network partitions as well as z-scores to identify the similarity of two partitions relative to what one would expect at random. Traud et al. examine a variety of similarity measures, such as the Rand similarity coefficient $S_R = (w_{11}+w_{00})/\mathcal{N}$ [120], and find that most of them are linear functions of $w = w_{11}$. Based on this, they identify an analytical formula for z-score z_R [121] for all linear-in-w similarity measures. The formulas can be found in Appendix C.

It is important to note that due to different network sizes, one needs to be careful not to over-interpret the meaning of z_R when directly comparing it between different networks. Traud et al. suggested them as quantitative advice on the statistical significance of how well network communities and those groups created from node labels correspond.

We are grateful to Mason Porter, who kindly provided the code for computing the z_R scores in Section 6.6.

5.3.2 Jensen–Shannon divergence

We introduce another measure which we use to gain insight into our network communities. We are interested in the way our communities differ with regards to the words that were used in tweets by users within the communities.

In [23], Galagher et al. use a method from information theory to quantify the divergence between a set of tweets containing the hashtag *#BlackLivesMatter* and another set of tweets containing *#AllLivesMatter*. We want to extend this idea so that we can compare not just two, but multiple text bodies.

We introduce a method whose procedure is based on Claude Shannon's work of using the notion of entropy as a measure of unpredictability of a distribution [122]. Kullback and Leibler [123] extended this to a statistic which measures the difference between two probability distributions. Given two probability distributions P_1 and P_2, the Kullback–Leibler (KL) divergence is defined by

$$D_{\mathrm{KL}}(P_1||P_2) = \sum_{i=1}^{l} p_i \log_2 \frac{p_{1_i}}{p_{2_i}}, \tag{5.17}$$

where l is the size of the sample space. Because the logarithm has base 2, Equation 5.17 can be interpreted as the number of extra number of bits that need to be used to encode values from distribution P_1 with a code that is based on distribution P_2. Thus, we give the results in Section 6.7 in units of bits [124].

As highlighted in [23], applying KL divergence directly to two text bodies made of tweets is likely to raise issues. This is due to the logarithm in the definition causing the divergence measure to be undefined if $p_{2_i} = 0$ for any i (i.e. if word i is present in word distribution P_1 but not in P_2 [23]). To avoid this, Gallagher et al. suggested implementing the Jensen–Shannon (JS) divergence instead, which is an adaptation of the KL divergence. It finds a way around the problem arising due to the risk of the undefined logarithm by introducing the mixed distribution $\mathcal{M} = \pi_1 P_1 + \pi_2 P_2$, where π_1 and π_2 are the weights of the distributions P_1 and P_2, respectively, with $\pi_1 + \pi_2 = 1$. The JS divergence was first defined by Lin [125] as

$$D_{\mathrm{JS}}(P_1||P_2) = H(\mathcal{M}) - \pi_1 H(P_1) - \pi_2 H(P_2), \tag{5.18}$$

where $H(p) = -\sum_{i=1}^{n} p \log_2 p$ is Shannon's entropy [122]. Gallagher et al. point out that Equation (5.18) can be rewritten in terms of the KL divergence follows:

$$D_{\mathrm{JS}}(P_1||P_2) = \pi_1 D_{\mathrm{KL}}(P_1||\mathcal{M}) + \pi_2 D_{\mathrm{KL}}(P_2||\mathcal{M}). \tag{5.19}$$

As shown in [125], the JSD is bounded between 0 and 1. When comparing two text bodies, this can be interpreted as follows: A JSD of 0 implies that word probability distributions in both text bodies are equal, and a JSD of 1 suggests that there is not a single word that appears in both distributions [23]. Having found a measure for the difference of two text bodies, an interesting question to ask is the contribution of individual words to this distance. Looking at the original formulation of the JS divergence in Equation (5.18), one can see that the contribution of each word i is found by calculating [23]

$$D_{\mathrm{JS},i}(P_1||P_2) = -m_i \log_2 m_i + \pi_1 p_{1_i} \log_2 p_{1_i} + \pi_2 p_{2_i} \log_2 p_{2_i}. \tag{5.20}$$

For our purposes — as we generally are working with more than two text bodies — it is useful to describe how the JS divergence can be extended to the case of more than two probability distributions that are compared to each other. The extension of the above definition to q distributions is [125]

$$D_{\mathrm{JS}}(P_1||P_2||\cdots||P_q) = H(\mathcal{M}) - \sum_{j=1}^{q} \pi_j H(P_j), \tag{5.21}$$

where, again, $\mathcal{M}$ is the mixed distribution

$$\mathcal{M} = \sum_{j=1}^{d} \pi_j P_j. \tag{5.22}$$

Similar to the two-distribution case, one can find the upper bound $D_{\mathrm{JS}}(P_1||P_2||\cdots||P_q) \leq \log_2(q)$ for q distributions [125]. One can compute the individual word contributions as

$$D_{\mathrm{JS},i}(P_1||P_2||\cdots||P_q) = -m_i \log_2 m_i + \sum_{j=1}^{q} \pi_j p_{j_i} \log_2 p_{j_i}. \tag{5.23}$$

5.3.2.1 Preprocessing of document probabilities

In practice, the word distributions that we are comparing (for example all words from tweets created by nodes in one community compared to those in other communities) are likely going to have zero entries for some words when these words only appear in some of the distributions. If one of the word probabilities p_{j_i} in Equation 5.23

is 0, then the term including the logarithm becomes undefined. To avoid this, we follow the suggestions of [124] and explain how to preprocess the word distributions by assigning every word with probability 0 a small amount of "free" probability. We define the document probability $P(t, d_j)$ of document d_j as

$$P(t, d_j) = \ = \begin{cases} \psi P(t_i|d_j) & \text{if word } t_i \text{ is in document } d_j \\ \epsilon & \text{otherwise,} \end{cases} \tag{5.24}$$

where t_i, $i = 1, \cdots, v$, are unique words in the document corpus D, $P(t_i|d_j)$ is the probability of word t_i being present in d_j, ϵ is the "free" probability assigned to words of originally zero probability and ψ is a normalisation coefficient. Clearly, we must choose ψ and ϵ so that the document probability sums to 1. Hence,

$$\sum_{t_i \in d_j} \psi P(t_i|d_j) + \sum_{t_i \notin d_j, t_i \in D} \epsilon = 1, \tag{5.25}$$

. We first choose ϵ smaller than the smallest probability of any word t_i in any of the documents d_j. We then rearrange Equation (5.25) to

$$\psi = 1 - \sum_{i \notin d_j, i \in D} \epsilon, \tag{5.26}$$

which will enable us to compute ψ.

To compute the JS divergence between multiple documents (such as tweets created by users in different communities), as well as word contributions, we first use `scikit-learn`'s function `CountVectorizer`. This returns vectors representing documents. Each element of such a vector represents the frequency of a word from the lexicon L in this document. We write a Python script in which we then compute the word probabilities and proceed as described in the last paragraph to assign probability ψ to words which have 0 probability. We then use Equations 5.21 and 5.23 to compute the JS divergences and word contributions.

Chapter 6

Results

6.1 Network growth

We first demonstrate results obtained from the network-based methods in Chapter 5. To introduce our Twitter networks, we show a series of longitudinal snapshots of the complete reply network in Figure 6.1. We measure the number of new nodes and tweets every 24 hours ($\Delta t = 24\text{h}$) between the the start of our data collection (27/05/2016) and the end of it (30/06/2016).

Generally, as expected, the growth of the network itself (i.e. the number of nodes) seems to correspond with the increase of daily tweets. Investigating the net daily change in both the number of new nodes and the number of tweets reveals that the spikes in the two red graphs seem to correspond with the following events. The spike at 7 June marks the date of the official deadline to register to vote in the EU referendum. On this day, there is a distinct spike in the new number of nodes, suggesting that the deadline lead to a number of people joining the Brexit Twitter debate. Another interesting increase in both number of tweets and nodes occurs on 17 June, just after Labour MP Jo Cox was murdered on 16 June [126] and a Leave versus Remain "boat-off" on the river Thames [127]. The final spike coincides with the day of the EU referendum and the few days running up to it [128]. The network size decreases rapidly from the day after the EU referendum. The correlation of crucial events in the timeline before the referendum with spikes in the Twitter debates demonstrate that, although a social media conversation is at best an imperfect representation for general public opinion, the social platform nevertheless provides a mouthpiece for real-life political incidences.

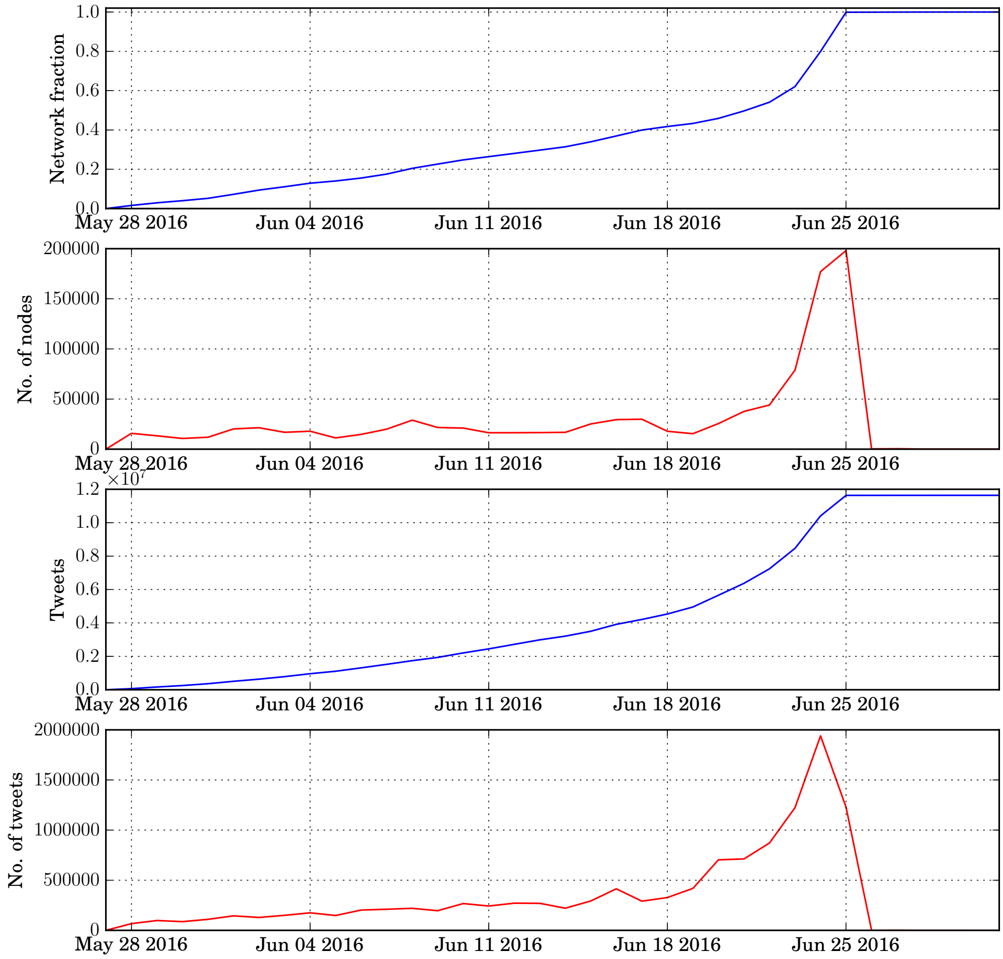

Figure 6.1: Network growth ($\Delta t = 24$h). First graph: Total number of nodes in network by day, normalised by the final network size. Second graph: Net growth of network by day. Third graph: Total number of tweets in network by day. Fourth graph: New tweets in network by day.

6.2 Standard network diagnostics

We calculate a few standard network diagnostics and compare the reply and retweet networks as well as the networks from consecutive weeks. We compute the number of nodes, the number of edges, and the number of actual tweets to get a general idea of sizes. We also calculate some standard measures such as the mean in-coming strength (in-strength), reciprocity, and density (as defined in Section 2.1). We display the results in Tables 6.1 and 6.2.

	Week 1	**Week** 2	**Week** 3	**Week** 4	**Week** 5	**Total**
# nodes	5856	8227	10998	28520	345	44557
# edges	9097	13269	18413	38807	363	77583
# tweets	25860	36786	51296	127347	732	225946
Mean in-strength	4.416	4.471	4.664	4.465	2.122	5.071
Reciprocity	0.066	0.055	0.055	0.030	0.010	0.047
Density	0.0008	0.0005	0.0004	0.0002	0.006	0.0001

Table 6.1: Summary statistics for the largest connected component of the weekly and total reply networks.

Similarly to what we saw in Figure 6.1, we can observe that the network grows — with respect to both nodes and edges — from weeks 1 to 4 and then decreases in size in the week after the referendum. It is worth noting that for both the reply network, and the retweet network the measures for week 5 differ considerably from the four weeks before. It seems that this could be due to the difference in network sizes.

In the four weeks before the referendum in the case of the reply networks, one can observe that the mean in-strength stays fairly constant. The retweet network has mean in-degrees that are on average 2.5 times higher than those of the reply network. One possible explanation is that rebroadcasting another user's message on Twitter is generally done in one click whereas the drafting of a reply to someone requires more time and effort. Additionally, the mean in-strength in the retweet network is considerably lower in week 4.

	Week 1	**Week 2**	**Week 3**	**Week 4**	**Week 5**	**Total**
# nodes	121942	182921	230090	729852	403794	944821
# edges	327465	519436	693492	1886352	554344	3136588
# tweets	1431102	2178851	2768015	6952417	1216225	11318026
Mean in-strength	11.736	11.911	12.030	9.526	3.0120	11.979
Reciprocity	0.009	0.010	0.009	0.007	0.001	0.0118
Density	9.62×10^{-5}	6.51×10^{-5}	5.23×10^{-5}	1.31×10^{-5}	7.46×10^{-6}	1.27×10^{-5}

Table 6.2: Summary statistics for the largest connected component of the weekly and total retweet networks.

Reciprocity values indicate how often nodes have in-coming edges from nodes that they point to. These values are considerably higher in the reply networks than the retweet networks. Intuitively, this lies in the nature of replies and mentions (recalling that the latter are included in our reply networks) compared to that of a retweet; replying to or mentioning a user's tweet indicates the beginning of a conversation between the two or more users. Retweets, however, are often used to propagate content from other users to one's follower audience [129]. Hence, by definition one can expect that the acts of replying and mentioning induce higher reciprocity values than that of a retweet.

We also observe that the reply networks for all weeks are considerably denser than the retweet networks; i.e. any two nodes in the reply network are more likely to be connected by an edge than any two nodes in the retweet network.

Another interesting measure of properties in a directed network is the distribution of in- and out-strengths. Recall that the mean in-strengths (which equals the mean out-strength [24]) of the total retweet and reply networks are 11.9 and 5.1, respectively. Let us consider the standard deviation of the in- and out-strength distributions of the total (i.e. not divided by weeks) reply and retweet networks. The in-strength distribution of the retweet network has a standard deviation of 897 whilst that for the out-strength distribution is 177. This result is aligned with the findings by Son et al. [130] who demonstrate that for a set of networks created from web pages linking to each other, the in-degree distribution is much broader than the out-degree one. This has also been found for Twitter networks [131]. With regards to our retweet network, this implies that there is a larger variance in the number of Twitter users who retweeted any individual user (called *popularity* [131]) than in the number of other users each individual user retweeted (*prolificity*). In the reply network, however, we observe experience the opposite behaviour: the standard deviation of the in-strength distribution (43) is a lot smaller than that of the out-strength distribution (242), implying that in this case popularity is less variable than prolificity [131].

We recall from Section 3.3 that bots contributed considerably to retweets on topics around Brexit [70]. Generally, we must thus be careful when interpreting the properties of our Twitter networks.

6.3 Network communities

We apply a directed and weighted version of the Louvain algorithm as described in Section 5.1.1.3 to our retweet and reply networks for every week. For each network, we choose the partition with the highest modularity value. In Tables 6.3 and 6.4 we display the number of identified communities as well as the modularity value of the partition. Note that due to the large network sizes our Matlab computations for the retweet networks in weeks 3 to 5 did not finish in time to be included in follow-up computations. For this reason we restrict our computations in the remaining sections

	Week 1	**Week** 2	**Week** 3	**Week** 4	**Week** 5
# communities	163	203	269	423	21
Modularity	0.7703	0.7527	0.7521	0.6928	0.8481

Table 6.3: Communities in weekly and total reply networks and modularity values for each partition.

	Week 1	**Week** 2	**Week** 3	**Week** 4	**Week** 5
# communities	668	1140	–	–	–
Modularity	0.5155	0.5278	–	–	–

Table 6.4: Communities in weekly and total retweet networks and modularity values for each.

of this thesis on the reply networks. From now on, we refer to the reply networks as just networks.

We investigate the size distributions of each community and we find that in week 4, the largest five communities include about 64% of the network nodes. For weeks 1, 2, and 3, these percentages are 31%, 27%, and 20%, respectively. For week 5, we find that the top five communities make up 50%, though this seems to be less interesting due to the comparatively small network size. Because the largest five communities include a relatively large percentage of all nodes in all weeks, we focus on these top five communities in later sections.

6.4 Community centralities

We were motivated by the work done in [58], whose findings suggested that those users that are responsible for spreading information in a network are also central users (according to a certain centrality measure). We use PageRank centrality on our Twitter network to examine the importance of users who might have been likely the source of information spread across the network. We compute PageRank centralities for all users in each of our weekly networks. We then find the user with the highest PageRank centrality in each community. Because we are interested in finding a “label” for each of our communities (i.e. some sort of characterisation), we choose the node with the largest PageRank centrality in each community as the “representative” node for this community. We understand that this is not rigorous but it serves as an indication for the type of community. In Table 6.5 we display the top-5 largest communities for each week, each represented by the node with the

	Week 1	**Week** 2	**Week** 3	**Week** 4	**Week** 5
1	StrongerIn	theordinaryman2	The_TUC	StrongerIn	stephbreakfast
2	BBC_HaveYourSay	BorisJohnson	StrongerInPress	business	guardian
3	David_Cameron	George_Osborne	Patriotic_Brit	BorisJohnson	Channel4News
4	BBCNews	Nigel_Farage	BBCr4today	UKLabour	BBCBreaking
5	A_Liberty_Rebel	SolWielkopolski	DanKennett	SkyNews	rachy_babyx

Table 6.5: The five largest communities in each weekly network represented by the User IDs of the most central nodes based on PageRank centrality.

highest PageRank value. See Appendix D for a table showing the nodes representing the top-20 communities. We recognise a large amount of the User IDs in Table 6.5 as figures that were clearly involved in public debates around the EU referendum. These include David Cameron, the then Prime Minister of the UK, Boris Johnson, the former Mayor of London and several accounts connected to BBC news. Based on the definition of the PageRank centrality [36] and following the findings in [58], we expect that the users in Table 6.5 are at least somewhat responsible for spreading information within the community. Based on the general public Brexit debate in the UK, one can clearly recognise some of these User IDs as clear supporters for either the Leave- or the Remain-campaign. Nigel Farage, for example, the former leader of the UK Independence Party (UKIP) as well as Boris Johnson were strong representatives of the Leave-campaign. David Cameron and George Osborne, the former Chancellor of the Exchequer, as well as the StrongerIn Twitter account (which was the official account run by the Remain-campaign) all lobbied for the UK to stay in the EU. For future work, it would be worth investigating the extend to which these or other central Twitter users spread information and dominate debates within communities and thereby somewhat influence the political tone within a group of Twitter users.

6.5 Text clusters

To identify clusters of users based on our textual data (i.e. tweets and Twitter biographies), we apply the steps described in Section 5.2.5. For each week we perform a PCA on the data points which represent each users tweets and each users Twitter biographies. We find that for the tweet data sets in weeks 1 to 4, more than the first 500 principal components are needed to explain 50% of the variance of the unreduced data. To explain the same percentage of variance in the tweet data set in week 5, which is smaller, we still need more than the first 80 principal components. Similarly, to explain half of the variance of the Twitter biographies data sets of any of the four weeks, we need between the first 100 and 500 principal components. As this may indicate that the first few principal components do not provide sufficient information required for a successful clustering of the data, we apply the K-means algorithm on the original, unreduced data set. We recall that these findings are in line with warnings originally made by Chang [132] as well as with the suggestions in [117] to analyse the effectiveness of the PCA before attempting to detect clusters.

We each $K = 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30$ we compute Silhouette scores of all data points. We use these scores to compoute the mean Silhouette score for every K. As predicted in Section 5.2.3 by the high dimensionality of our data, the mean Silhouette scores for each of our data sets are very low (between 0.003 and 0.015) and very similar. It therefore does not seem a sufficient basis for a choice of K. We thus draw a Silhouette plot for each partition, as suggested in [112]. To have another benchmark for judging a suitable number of clusters, we visualise the results for each K by plotting a PCA projection of the first two principal components of our data.

Choosing an appropriate K on the basis of a visualisation of Silhouette plots and PCA projections is by no means a systematic or rigorous way of finding an "optimal" number of clusters. Having found that average Silhouette scores are not sufficient for this task, future work on this topic should include a more thorough investigation of different methods on how to choose K.

Table 6.6 shows the values for K, i.e. the number of clusters we choose for each week based on a visualisation of Silhouette scores and PCA projections.

	Week 1	**Week** 2	**Week** 3	**Week** 4	**Week** 5
Tweets	4	3	5	4	2
Twitter bios	4	3	2	3	2

Table 6.6: First row: Suitable number of clusters k for each week when analysing text clusters induced by tweets. Second row: Same for Twitter biographies.

In Figures 6.2 and 6.3, we show examples of the visualisations of Silhouette scores and PCA projections.

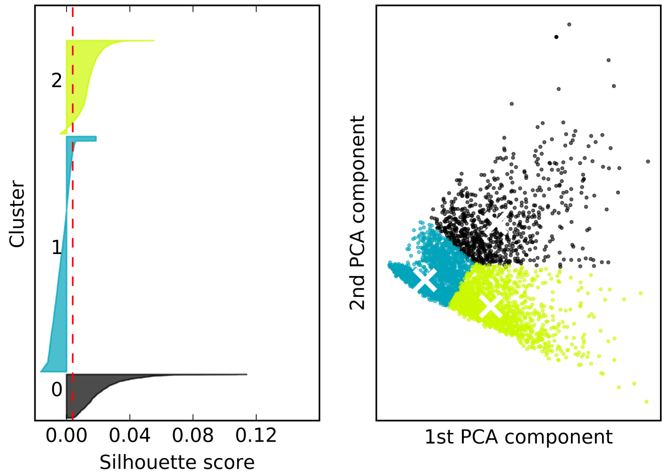

Figure 6.2: Silhouette plot and PCA projection for topic clusters in tweets found by K-means algorithm; the red dashed line in the Silhouette plot is the mean silhouette score. The design of these plots was inspired by Python's `scikit-learn` documentation.

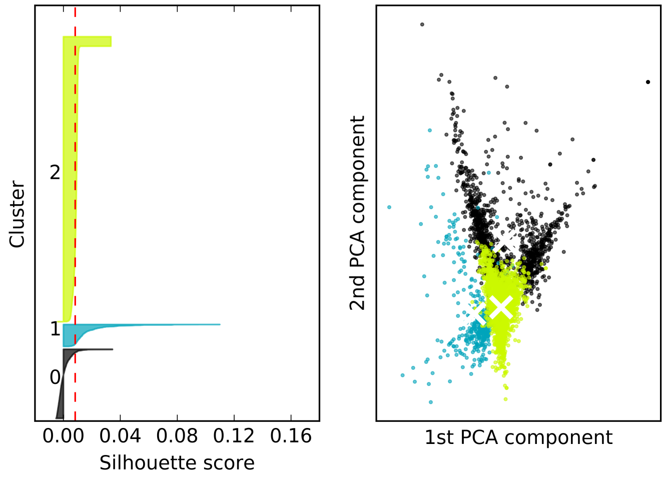

Figure 6.3: Silhouette plot and PCA projection for topic clusters in Twitter bios found by K-means algorithm; the red dashed line in the Silhouette plot is the mean silhouette score. The design of these plots was inspired by Python's `scikit-learn` documentation.

We start by an examination of the example Silhouette plot and PCA projection for our tweet dataset. Observing the silhouette plot in Figure 6.2, one can see that out of the three clusters that we detect, the data points in two of them — in cluster 0 and cluster 2 — have predominantly positive silhouette scores. This implies that the data points in these clusters have likely been assigned to the correct cluster. Cluster 1 has a large proportion of its data points on the negative silhouette axis, so we expect this cluster to overlap with the other two groups. To look at this further, we investigate the clusters by visualising them as wordclouds[1]. The sizing of the words in the wordclouds in Figure 6.4 is proportional to the tfidf weighting of the word.

In line with our interpretation of the silhouette plot and PCA projection, two of the clusters (cluster 0 and 2) have fairly distinct topics. Cluster 0 includes conversations about the deadline for the voting registration, which took place in week 2 (on 9 June. Prominent features in cluster 2 suggest debates on immigration in particular and perhaps economics more generally. It is difficult to discern a clear topic from cluster 1, which is in line with our expectations from Figure 6.2.

We proceed to examine the example Silhouette plot and PCA projection for Twitter biography data set in Figure 6.3. The Sihouette plot shows a smaller number of negative Silhouette scores as well as a larger mean Silhouette score which suggests that clusters of users based on Twitter biographies are more distinct than those based on what the same people tweeted (in week 2).

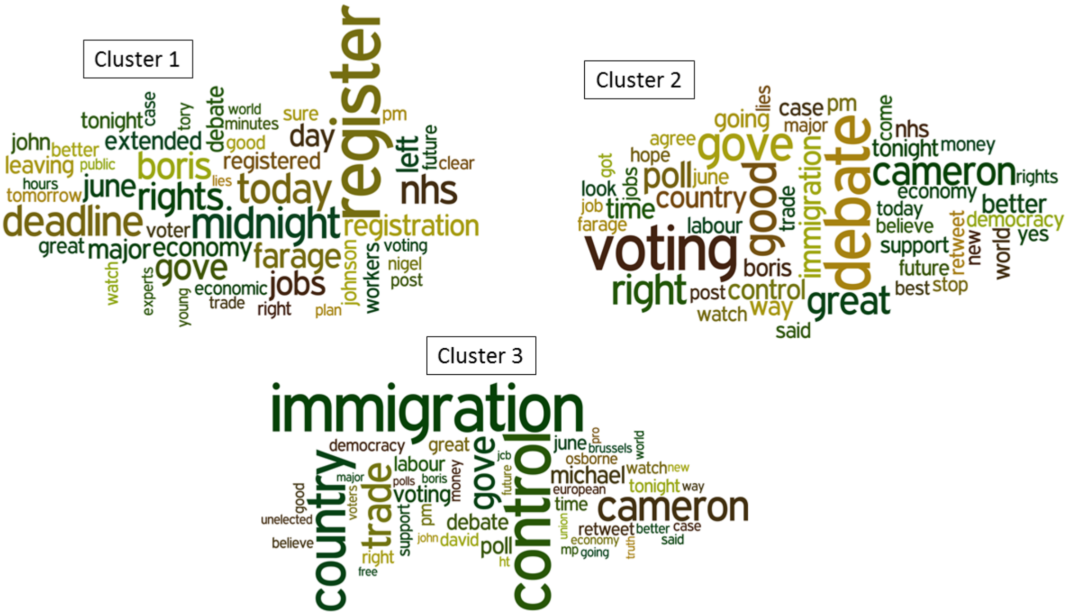

Figure 6.4: Wordclouds for the three tweet clusters found by the K-means algorithm in week 2.

[1] Wordclouds created with `www.wordle.net`

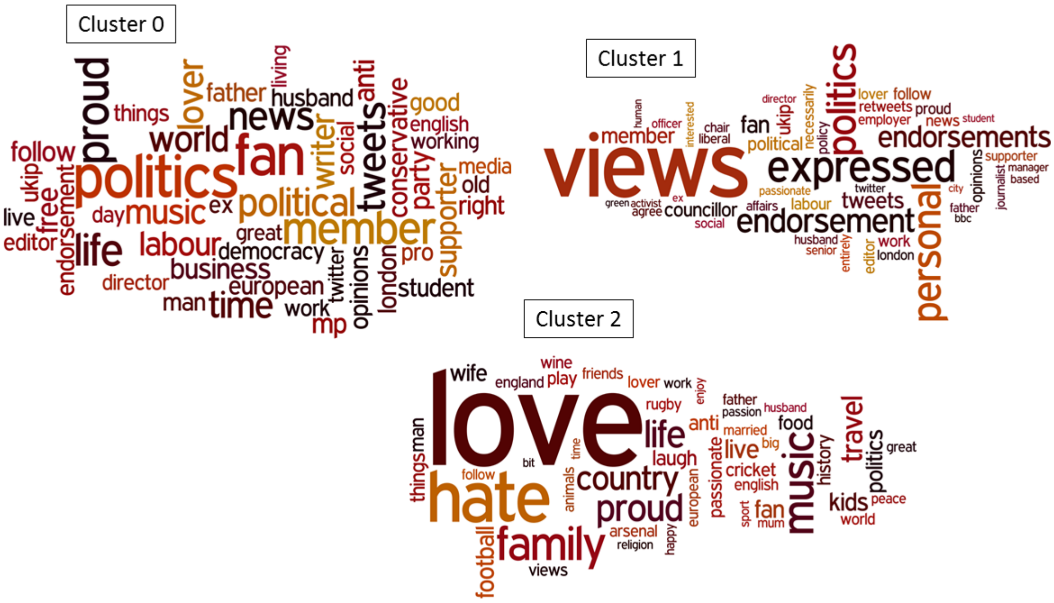

Figure 6.5: Wordclouds for the three bio clusters found by the K-means algorithm in week 2.

The wordcloud in Figure 6.5 shows the highest tfidf weighted terms in the clusters of nodes found based on their Twitter biographies. Out of the three clusters, cluster 0 is the most varied in topics and has some overlap with both of the other clusters. This is confirmed by the Silhouette plot in Figure 6.3, which shows that some data points in cluster 0 have negative Silhouette scores.

Keeping in mind the warnings on high-dimensionality and the unsystematic method of choosing K, we must be careful to over-interpret the clusters we identified. However, in both the tweet and the Twitter biographies data set we observe some distinction between the detected clusters.

6.6 Partition similarity

According to the suggestions in Section 5.3.1, we assign labels to each node in our Twitter networks depending on the network community, tweet cluster and bio cluster of which the node is a member. To investigate the extent to which the text clusters correlate with the network communities, we apply the z_R-score method [62] introduced in Section 5.3.1. Table 6.7 shows the result for applying the method by Traud et al. [62] introduced above. The first column shows the z-score values when taking the two different partitions to be the communities and the bio clusters. The second column shows the result for the same computation but replacing the partition induced by the bio clusters by the one obtained from tweets.

	Communities/ **Twitter bio clusters**	**Communities/** **Tweet clusters**
Week 1	1.0593	−0.7887
Week 2	−2.6414	−1.1235
Week 3	−0.6506	−0.2758
Week 4	0.8351	−0.2003
Week 5	−0.8225	−0.2037

Table 6.7: Z-scores for Twitter bio clusters and tweet clusters compared to the communities found by the Louvain method.

Because the networks in different weeks have different sizes (see Figure 4.4 in Section 4.2) we cannot directly compare the z-score values across the consecutive weeks but we can only make conclusions about the two different text analysis partitions and how well they correlate with the Louvain communities. One can observe that for all weeks and for both types of text clusters, there is no statistically relevant correlation between the partition based on network communities and that induced by tweets or Twitter bios. Only the z-score in the first column of week 1 suggests that there might be a marginally statistically relevant correlation between the partition based on Louvain communities and that induced by the bio clusters.

6.7 JS Divergence of documents

We apply the methods suggested in Section 5.3.2 to compare different text bodies. We are interested in the divergence between the words in tweets created by nodes within the five largest network communities. We also want to find out the extent to which the conversations within different hashtag subgraphs differ from each other. Recall from Section 4.2 the construction of network hashtag subgraphs (see also Appendix B), for the hashtags #EUref, #Brexit, #VoteLeave, #VoteRemain and #EUreferendum. We thereby hope to discern which of the two partitions of the complete network comprises of more distinct conversations.

We apply the extended version of the Jensen–Shannon divergence, which we introduced in Equation (5.21), to word distributions in two sets of text bodies.

The first set, which describes the five largest network communities, is constructed as follows: 1. For each of the five largest communities (in each week), we merge all tweets created by the nodes within the community into a large document. This gives five documents containing non-unique terms from all tweets. 2. We count the frequencies of each word and vectorise the documents by using `scikit-learn`'s `CountVectorizer`. 3. We then compute the document probability for each document as described in Section 5.3.2.1.

We take the same steps to calculate document probabilities for the words contained in tweets which were created by users in hashtag subgraphs. However, some nodes might be in more than one of the hashtag subgraphs — as some users may have used more than one of the hashtags — which might lead to overlapping text documents. This causes word distributions to be more similar to each other than if there was no overlap. Because the Louvain communities do not overlap, this issue makes it difficult to compare the JS divergence between these two partitions. We thus remove those nodes that are part of more than one of the hashtag subgraphs before computing the divergences.

	Hashtags	**Communities**
Week 1	0.1693 bits	0.4155 bits
Week 2	0.1778 bits	0.3618 bits
Week 3	0.1861 bits	0.3747 bits
Week 4	0.3441 bits	0.3533 bits
Week 5	0.2997 bits	1.4713 bits

Table 6.8: JSD values by weeks for the collection of tweets in (first column) each hashtag subgraph and (second column) the collection of tweets in each of the 5 largest communities.

The resulting JS divergence for each of the five hashtags and communities in each week is displayed in Table 6.8. We can see that the communities in week 5 seem to differ from each other more than in other weeks. Interestingly, the network partition in week 5 is also the one with the largest maximum modularity value compared to the other weeks (see Table 6.3). Comparing the bit scores of the hashtag subgraphs to those of the communities yields that conversations within communities differ more from each other than those between the nodes in the hashtag subgraphs. This suggests that nodes that are part of different communities do in fact debate different topics to a certain extent.

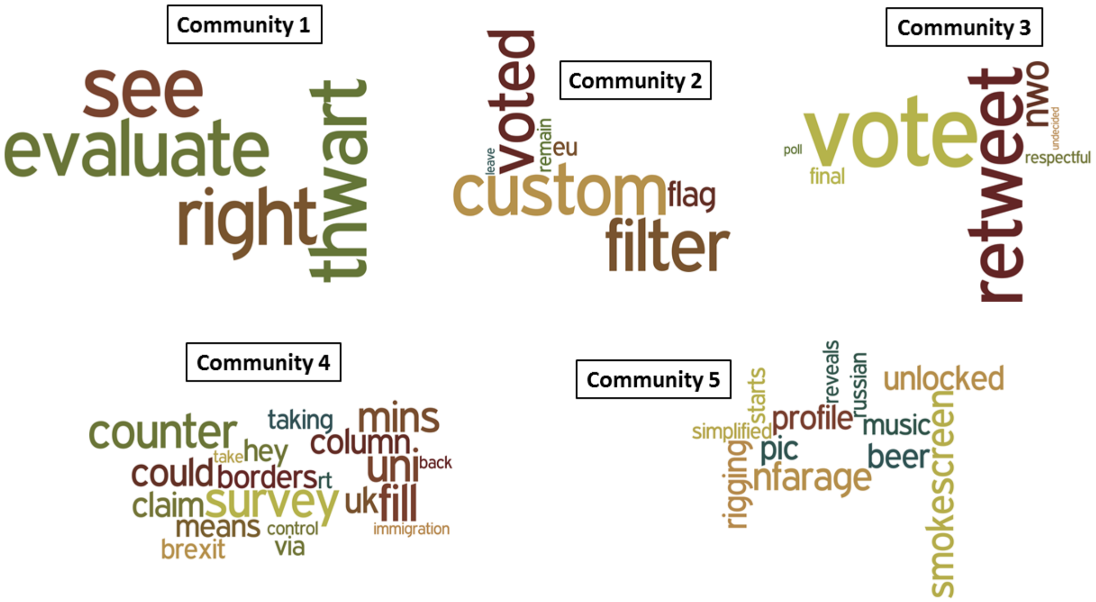

Figure 6.6: Word contributions to the JSD between the five largest communities in week 4.

This motivates the question of the type of conversations that distinguish the different communities. We thus compute the contributions of individual words in the different text bodies to the overall JS divergence. We use Equation (5.23) to compute the top 50 words with the highest contribution to the total JSD value. For each word, we then calculate the probability at which it occurs in each of the distributions, (i.e. in each of the text bodies). The text body for which the probability of each of the top 50 words is highest is the one that is most "responsible" for the word's contribution to the total JS divergence. In other words, the contribution to the overall JSD from word i is caused by the text body P in which this word has the highest probability. We find the 50 words that contribute most to the JS divergence. We then visualise these word contributions by creating the wordclouds in Figures 6.6 and 6.7 for the JS divergence between the communities and the hasthag subgraphs, respectively.

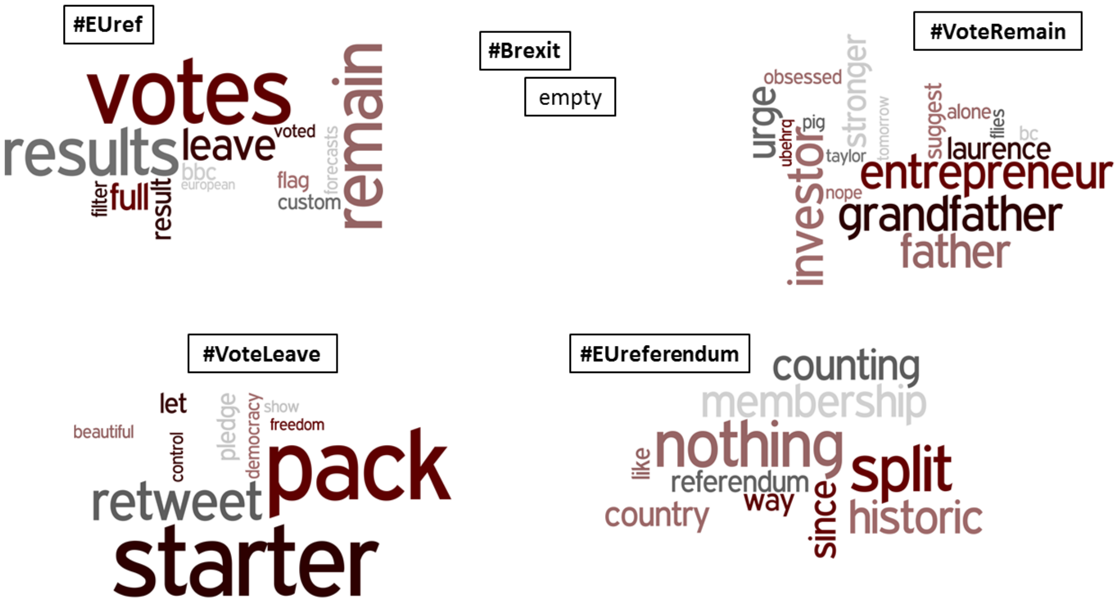

Figure 6.7: Word contributions to the JSD between the five hashtag subgraphs in week 4.

The wordclouds in Figure 6.6 show the 50 most-contributing words split by the communities in which they appeared most. Similarly, the wordclouds in Figure 6.7 visualise the word contributions for each hashtag subgraph. The size of words in the wordclouds is proportional to the contribution they made to the total JS divergence.

Gallagher et al. used a similar procedure to compute the divergence between words used in tweets alongside two different hashtags. In particular, they investigated how tweets including the hashtag #BlackLivesMatter differ from those containing the hashtag #AllLivesMatter. They had the advantage of comparing only two text bodies to each other. Additionally, they have more prior knowledge on the text bodies, as it is known that the AllLivesMatter movement was created as a protest against the BlackLivesMatter movement. By investigating the JS divergence of the two text bodies, Gallagher et al. make conclusions about different styles of language that people in the two movements use, as well as the events they talk about. Investigating our Twitter data is more challenging as we have a much smaller amount of background knowledge on the text bodies which we are examining.

However, we have reasons to believe that the word distributions for both the communities and the hashtag subgraphs do indeed differ from each. This in itself is a motivation to further investigate the topics in each of the text bodies in possible future work.

Chapter 7

Conclusions

In this thesis, we used methods from network science as well as text data mining to investigate the structure of the Twitter debate on the EU referendum.

For each of five consecutive weeks of the period studied, we constructed a directed, weighted retweet as well as reply network. Additionally, we constructed subgraphs of these networks, which were restricted to nodes using specific hashtags. We employed a directed version of the Louvain modularity optimisation method to detect communities in each of the networks. One of the aims of this thesis was to understand how these communities were different from each other, who influenced the debates within these communities and whether we can find distinct content that was between the members of the communities.

To identify influential users in the twitter networks we used the well-known PageRank centrality measure. In each network community, we determined the most central node (i.e. Twitter user) and assigned this user to be a "representative" for this community. We found that most of the representatives of the five largest communities were commonly known public figures which were influential in the Brexit debate in the UK media.

To get insights on contents of Brexit Twitter debates, we examined tweets that were created by users (i.e. nodes) in Twitter networks. We also analysed Twitter biographies by which the same users described themselves. We first used feature extraction to represent cleaned text documents as vectors indicating weighted frequencies of each term in the documents. We applied the popular K-means algorithm on these two different types of textual data to find groups of users which talk about similar topics and describe themselves similarly in their Twitter biographies. Despite issues regarding the high dimensionality of our data sets, we identified clusters in both the sets of tweets and Twitter biographies which included somewhat distinct set of words.

Having partitioned the nodes based on different criteria — i.e. network communities and text clusters — we were interested in the extent to which these partition correlate with each other. Computing the z-scores of a partition similarity coefficient yielded no statistically relevant correlation.

Next, we applied a method from information theory to compare the word distributions within communities and within the different hashtag subgraphs. We found that the conversations within different communities differed more from each other than those between people using the hashtags.

Although the K-means algorithm identified clusters in our textual data with reasonable success, we also understand that our methods of choosing the number of clusters K is unsystematic and must be improved. We are also aware that many more recently invented algorithms were designed with the purpose of tackling high-dimensional data (see, for example, the ones suggested in [100]). For future work, we suggest investigations into different algorithms with the aim of identifying clusters more distinct than the ones we found in this thesis.

One of the major drawbacks of the nature of our Twitter data set was the lack of metadata on users. Studies which use network science methods to analyse the dynamics of social media platforms often possess prior knowledge about the network nodes [20, 62]. This knowledge can be used to validate whether results obtained from network measures correspond to “real-world” scenarios. In this thesis, we attempted to artificially assign labels to nodes based on their text cluster membership. Finding more distinct clusters is a first step of improvement on this level. We suggest another, perhaps more insightful method of assigning labels to nodes. The divided nature of the Brexit debate [3] suggests that the majority of participants, or at least a significant percentage, took sides when they debated the topic on social media. Because we found it to be challenging to make conclusions about the structure of our networks without prior knowledge on the nodes, we would have utilised this polarised character of the Brexit debate if time allowed it. We suggest that it would be a suitable basis for using a different type of machine learning, namely that of content-based classification [22]. Similar to the work by Conover et al. [22], we suggest that one could train a classification algorithm to identify for each user whether they support the Leave- or the Remain-campaign. We expect that labelling all nodes based on which side of the campaign they supported might be a useful tool when attempting to understand the communities of the network.

By computing the PageRank centralities of the nodes in our networks, we gained insight into influential figures in the Brexit Twitter debate. However, the PageRank measure is only one of many types of centralities which we could have used. PageRank, as well as many other standard centrality measures, work on nodes that are fixed in time. A piece of information that is lost when using static centrality measures is the *ordering* of the interactions of Twitter users, as emphasised in by Grindrod and Parsons[133]. For future work, we suggest investigating centrality measures which can be computed at different points in time and thus capture the pathway of information more correctly.

Overall, we would like to state that the large size of our data set and the vast amounts of technical detail that accompany the processing of such data sets have prevented us from taking our analysis to the next level. We were not able to draw any conclusions from our work onto whether or not the British public was well-informed on Brexit or on British politics in general. However, we did find preliminary results which will certainly lay the groundwork for further investigations on the matter.

Appendix A

Data collection

Hashtag	**Number of tweets**
Euref	3620465
Brexit	2787365
VoteLeave	1805621
VoteRemain	608176
Eureferendum	544908
StrongerIn	507174
Remain	415326
LeaveEU	239808
TakeControl	184304
LabourInForBritain	124182
Bremain	85898
VoteIn	85129
VoteOut	45092
BetterOffOut	39508
RemainInEU	36855
No2EU	22609
LabourLeave	18452
BeLeave	15764
VoteStay	14217
UKinEU	5716
Imleavebecause	5441
StayinEU	2706
LoveEuropeLeaveEU	2336
Yes2EU	1875
BritainOut	1176
NotoEU	955
YestoEU	942
Incampaign	620
LabourGo	379
StopTheEU	268
LeaveChaos	61
UKinEurope	30

Table A.1: Number of tweets separated by hashtags.

Appendix B

Hashtag networks

	Week 1		Week 2		Week 3		Week 4		Week 5	
	Nodes	Edges	Nodes	Edges	Nodes	Edges	Nodes	Edges	Nodes	Edges
EUref	672	834	1140	1448	1869	2406	16812	18397	253	266
Brexit	3934	5721	5057	7572	6867	10392	6043	8653	76	79
VoteLeave	2199	2852	3237	4505	4442	6467	8156	12489	10	9
VoteRemain	248	262	368	394	724	810	1867	2176	4	3
EUreferendum	65	67	48	50	45	45	350	386	11	10

Table B.1: Number of nodes and edges for hashtag reply networks in each week.

	Week 1		Week 2		Week 3		Week 4		Week 5	
	Nodes	Edges	Nodes	Edges	Nodes	Edges	Nodes	Edges	Nodes	Edges
EUref	37420	71114	74726	146835	91206	199557	515928	1075407	297347	395480
Brexit	82738	185789	91850	228089	145470	358557	219397	469539	140721	169888
VoteLeave	34729	111894	47504	171717	57720	217815	122522	460362	7872	9654
VoteRemain	13548	21514	32205	51100	39170	67187	141159	235533	2911	3433
EUreferendum	17284	24139	29135	39952	17210	22916	89898	112965	47032	49748

Table B.2: Number of nodes and edges for hashtag retweet networks in each week.

Appendix C

Analytical z_R-score

Following [62], let $\mathcal{N}$ be the total number of pairs of nodes in a network, $\mathcal{N}_1$ be the number of pairs that are assigned to the same group in one partition, $\mathcal{N}_2$ is the number of pairs that are part of the same group in the other partition and $w = w_{11}$ is the number of pairs of nodes for which both nodes are in the same group in both partitions. Additionally, assuming a contingency table in which element n_{ij} gives the number of nodes which are in group i in one partition and in group j in the other partition. Also note that $n_{i.} = \sum_j n_{ij}$ and $n_{.j} = \sum_i n_{ij}$. The z-score z_R is then computed by the following formulas:

$$z_R = \frac{1}{\sigma_w}\left(w - \frac{\mathcal{N}_1\mathcal{N}_2}{\mathcal{N}}\right), \tag{C.1}$$

$$\begin{aligned} \sigma_w^2 = \frac{\mathcal{N}}{16} - \frac{(4\mathcal{N}_1 - 2\mathcal{N})^2(4\mathcal{N}_2 - 2\mathcal{N})^2}{256\mathcal{N}^2} + \frac{C_1C_2}{16n(n-1)(n-2)} \\ + \frac{[(4\mathcal{N}_1 - 2\mathcal{N})^2 - 4C_1 - 4\mathcal{N}][(4\mathcal{N}_2 - 2\mathcal{N})^2 - 4C_2 - 4\mathcal{N}]}{64n(n-1)(n-2)(n-3)}, \end{aligned} \tag{C.2}$$

$$C_1 = n(n^2 - 3n - 2) - 8(n+1)\mathcal{N}_1 + 4\sum_i n_{i.}^3, \tag{C.3}$$

$$C_2 = n(n^2 - 3n - 2) - 8(n+1)\mathcal{N}_2 + 4\sum_j n_{.j}^3. \tag{C.4}$$

Appendix D

Top PageRank in communities

	Week 1	**Week** 2	**Week** 3	**Week** 4	**Week** 5
1	StrongerIn	theordinaryman2	The_TUC	StrongerIn	stephbreakfast
2	BBC_HaveYourSay	BorisJohnson	StrongerInPress	business	guardian
3	David_Cameron	George_Osborne	Patriotic_Brit	BorisJohnson	Channel4News
4	BBCNews	Nigel_Farage	BBCr4today	UKLabour	BBCBreaking
5	A_Liberty_Rebel	SolWielkopolski	DanKennett	SkyNews	rachy_babyx
6	UKLabour	PrisonPlanet	Telegraph	George_Osborne	Telegraph
7	ConversationUK	SkyNews	PrisonPlanet	bbclysedoucet	BBCPolitics
8	chalkeblue	bradclockwork	lucycthomas	David_Cameron	business
9	theordinaryman2	TheFogeys	PA	br_uk	DailyMailUK
10	gsoh31	EmmaReynoldsMP	Australiaunwra6	DanielJHannan	Irish_Belfast
11	Joel_E928	RegenerationEX	BritsLovePolls	RT_com	euronews
12	RJohnDickinson	UKLabour	David_Cameron	Sadgrovem	stefandijkstra9
13	drjennings	David_Cameron	SamuelsKaty	MrRhysBenjamin	JonathanOPrice
14	chrisg0000	vote_leave	stardust193	HackneyAbbott	FT
15	vote_leave	labourleave	jpublik	VoteLeaveUKIP	NotUnderdog
16	NicolaWebb17	holland_tom	theordinaryman2	Zoella	PunyaBhandari1
17	petercoles44	booshyharris	NicolaSturgeon	PoliticoRyan	stiobhart
18	GeorgeKyris	ohboywhatashot	Sargon_of_Akkad	Another_Europe	NicolaSturgeon
19	PrisonPlanet	DavidLenigas	marcuschown	benrileysmith	BooykaVideos
20	sqlblues	Vote_LeaveMedia	WillBlackWriter	599bt	newyearsdayboy

Table D.1: The top-50 communities for networks in each week, represented by the Twitter user with the highest PageRank centrality within the community.

Bibliography

[1] Wikipedia. United kingdom european union membership referendum, 2016. https://en.wikipedia.org/wiki/United_Kingdom_European_Union_membership_referendum,_2016. Accessed: 27/08/2016.

[2] Agust Arnorsson, Gylfi Zoega, et al. On the causes of brexit. Technical report, Birkbeck, Department of Economics, Mathematics & Statistics, 2016.

[3] Danny Dorling. Brexit: the decision of a divided country, 2016.

[4] Swati Dhingra, Gianmarco Ottaviano, Thomas Sampson, and John Van Reenen. Brexit: the impact on uk trade and living standards. Technical report, Centre for Economic Performance, LSE, 2016.

[5] Niamh Moloney. Financial services, the eu, and brexit: an uncertain future for the city? *German Law Journal*, 17:75–82, 2016.

[6] Angelica Crabb. Wine export: Impact of the'brexit'on australian wine exports. *Wine & Viticulture Journal*, 31(3):73, 2016.

[7] Jeff John Roberts. Brits scramble to google "what is the eu?" hours after voting to leave it. http://fortune.com/2016/06/24/brexit-google-trends/. Accessed: 27/08/2016.

[8] BBC. Reality check: Have leave campaigners changed their tune? http://www.bbc.co.uk/news/uk-politics-eu-referendum-36641390. Accessed: 27/08/2016.

[9] Chris Roycroft-Davis. Remain campaign peddles outright lies at our expense, blasts chris roycroft-davis. http://www.express.co.uk/comment/expresscomment/659189/Remain-campaign-peddles-outright-lies-our-expense. Accessed: 27/08/2016.

[10] Michael Conover, Jacob Ratkiewicz, Matthew R Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. Political polarization on twitter. *ICWSM*, 133:89–96, 2011.

[11] George Parker and Federica Cocco. How battle over brexit crosses traditional party lines. `http://www.ft.com/cms/s/2/32414e3e-2804-11e6-8b18-91555f2f4fde.html`. Accessed: 27/08/2016.

[12] Homero Gil de Zúñiga, Nakwon Jung, and Sebastián Valenzuela. Social media use for news and individuals' social capital, civic engagement and political participation. *Journal of Computer-Mediated Communication*, 17(3):319–336, 2012.

[13] Daniel J Power and Gloria Phillips-Wren. Impact of social media and web 2.0 on decision-making. *Journal of Decision Systems*, 20(3):249–261, 2011.

[14] Robin Effing, Jos van Hillegersberg, and Theo Huibers. Social media and political participation: are facebook, twitter and youtube democratizing our political systems? In *International Conference on Electronic Participation*, pages 25–35. Springer, 2011.

[15] Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. *ICWSM*, 10:178–185, 2010.

[16] Sitaram Asur and Bernardo A Huberman. Predicting the future with social media. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 1, pages 492–499. IEEE, 2010.

[17] Twitter. Using hashtags on twitter. `https://support.twitter.com/articles/49309`. Accessed: 29/08/2016.

[18] Axel Bruns, Dr Dr Katrin Weller, Michael Zimmer, and Nicholas John Proferes. A topology of twitter research: Disciplines, methods, and ethics. *Aslib Journal of Information Management*, 66(3):250–261, 2014.

[19] Brendan Meeder, Jennifer Tam, Patrick Gage Kelley, and Lorrie Faith Cranor. Rt@ iwantprivacy: Widespread violation of privacy settings in the twitter social network. In *Proceedings of the Web*, volume 2, pages 1–2, 2010.

[20] Darko Cherepnalkoski and Igor Mozetič. Retweet networks of the european parliament: evaluation of the community structure. *Applied Network Science*, 1(1):1–20, 2016.

[21] Eszter Bokányi, Dániel Kondor, László Dobos, Tamás Sebők, József Stéger, István Csabai, and Gábor Vattay. Race, religion and the city: twitter word frequency patterns reveal dominant demographic dimensions in the united states. *Palgrave Communications*, 2(16010), 2016.

[22] Michael D Conover, Bruno Gonçalves, Jacob Ratkiewicz, Alessandro Flammini, and Filippo Menczer. Predicting the political alignment of twitter users. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom)*, pages 192–199. IEEE, 2011.

[23] Ryan Gallagher, Andrew Reagan, Christopher M Danforth, and Peter Sheridan Dodds. Divergent discourse between protests and counter-protests: #blacklivesmatter and #alllivesmatter. *arXiv preprint arXiv:1606.06820*, 2016.

[24] Mark Newman. *Networks: an introduction*. Oxford university press, 2010.

[25] Stefano Allesina and Mercedes Pascual. Network structure, predator–prey modules, and stability in large food webs. *Theoretical Ecology*, 1(1):55–64, 2008.

[26] Mark EJ Newman. Scientific collaboration networks. i. network construction and fundamental results. *Physical Review E*, 64(1):016131, 2001.

[27] Till Hoffmann, Mason A Porter, and Renaud Lambiotte. Generalized master equations for non-poisson dynamics on networks. *Physical Review E*, 86(4):046102, 2012.

[28] Stanley Wasserman and Katherine Faust. *Social Network Nnalysis: Methods and Applications*, volume 8. Cambridge University Press, 1994.

[29] Francis Bloch, Matthew O Jackson, and Pietro Tebaldi. Centrality measures in networks. *Available at SSRN 2749124*, 2016.

[30] Phillip Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92(5):1170–1182, 1987.

[31] Phillip Bonacich and Paulette Lloyd. Eigenvector-like measures of centrality for asymmetric relations. *Social Networks*, 23(3):191–201, 2001.

[32] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[33] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[34] James H Fowler. Connecting the congress: A study of cosponsorship networks. *Political Analysis*, 14(4):456–487, 2006.

[35] Nicola Perra and Santo Fortunato. Spectral centrality measures in complex networks. *Physical Review E*, 78(3):036107, 2008.

[36] Sergey Brin and Lawrence Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 56(18):3825–3833, 2012.

[37] Dan Schult Aric Hagberg and Pieter Swart. Networkx. `http://networkx.readthedocs.io/en/networkx-1.10/`. Accessed: 31/08/2016.

[38] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

[39] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[40] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.

[41] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

[42] Haijun Zhou. Distance, dissimilarity index, and network community structure. *Physical Review E*, 67(6):061901, 2003.

[43] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *International Symposium on Computer and Information Sciences*, pages 284–293. Springer, 2005.

[44] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, 2008.

[45] Jure Leskovec, Kevin J Lang, and Michael Mahoney. Empirical comparison of [115]network community detection. In *Proceedings of the 19th International Conference on World wide web*, pages 631–640. ACM, 2010.

[46] Dieter Merkl. Text data mining. In *In A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*, pages 269–276. Marcel Dekker, 1998.

[47] Jeffrey L Solka et al. Text data mining: theory and methods. *Statistics Surveys*, 2:94–112, 2008.

[48] Qiaozhu Mei and ChengXiang Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 198–207. ACM, 2005.

[49] Peter Sheridan Dodds and Christopher M Danforth. Measuring the happiness of large-scale written expression: Songs, blogs, and presidents. *Journal of Happiness Studies*, 11(4):441–456, 2010.

[50] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2014.

[51] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & Management*, 24(5):513–523, 1988.

[52] Stephen Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of Documentation*, 60(5):503–520, 2004.

[53] Gerard Salton and Michael J McGill. Introduction to modern information retrieval. 1986.

[54] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[55] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[56] Francois de Villiers, McElory Hoffmann, and Steve Kroon. Unsupervised construction of topic-based twitter lists. In *2012 International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2012 International Conference on Social Computing (SocialCom)*, pages 283–292. IEEE, 2012.

[57] Seth A Myers and Jure Leskovec. The bursty dynamics of the twitter information network. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 913–924. ACM, 2014.

[58] Sandra González-Bailón, Javier Borge-Holthoefer, Alejandro Rivero, and Yamir Moreno. The dynamics of protest recruitment through an online network. *Scientific reports*, 1(197), 2011.

[59] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and P Krishna Gummadi. Measuring user influence in twitter: The million follower fallacy. *ICWSM*, 10(10-17):30, 2010.

[60] Kristina Lerman, Prachi Jain, Rumi Ghosh, Jeon-Hyung Kang, and Ponnurangam Kumaraguru. Limited attention and centrality in social networks. In *2013 International Conference on Social Intelligence and Technology (SOCIETY)*, pages 80–89. IEEE, 2013.

[61] Warren Pearce, Kim Holmberg, Iina Hellsten, and Brigitte Nerlich. Climate change on twitter: Topics, communities and conversations about the 2013 ipcc working group 1 report. *PloS one*, 9(4):e94785, 2014.

[62] Amanda L Traud, Eric D Kelsic, Peter J Mucha, and Mason A Porter. Comparing community structure to characteristics in online collegiate social networks. *SIAM Review*, 53(3):526–543, 2011.

[63] Mariano Beguerisse-Díaz, Guillermo Garduno-Hernández, Borislav Vangelov, Sophia N Yaliraki, and Mauricio Barahona. Interest communities and flow roles in directed networks: the twitter network of the uk riots. *Journal of The Royal Society Interface*, 11(101):20140940, 2014.

[64] Statista. Number of monthly active twitter users worldwide from 1st quarter 2010 to 2nd quarter 2016 (in millions). `http://www.statista.`

com/statistics/282087/number-of-monthly-active-twitter-users/. Accessed: 29/08/2016.

[65] Prashanth Vijayaraghavan, Soroush Vosoughi, and Deb Roy. Automatic detection and categorization of election-related tweets. *arXiv preprint arXiv:1605.05150*, 2016.

[66] Emily M Cody, Andrew J Reagan, Peter Sheridan Dodds, and Christopher M Danforth. Public opinion polling with twitter. *arXiv preprint arXiv:1608.02024*, 2016.

[67] British Polling Council. British polling council welcomes unveiling of the provisional findings of polling inquiry. http://www.britishpollingcouncil.org/tag/inquiry/. Accessed: 29/08/2016.

[68] The NY Times Nate Cohn. Why polls have been wrong recently. http://www.nytimes.com/2016/01/08/upshot/why-polls-have-been-wrong-recently.html. Accessed: 29/08/2016.

[69] Alan Mislove, Sune Lehmann, Yong-Yeol Ahn, Jukka-Pekka Onnela, and J Niels Rosenquist. Understanding the demographics of twitter users. *ICWSM*, 11:5, 2011.

[70] Philip N Howard and Bence Kollanyi. Bots, #strongerin, and #brexit: Computational propaganda during the uk-eu referendum. *arXiv preprint arXiv:1606.06356*, 2016.

[71] Peter Cihon and Taha Yasseri. A biased review of biases in twitter studies on political collective action. *arXiv preprint arXiv:1605.04774*, 2016.

[72] Twitter. The streaming apis. https://dev.twitter.com/streaming/overview. Accessed: 29/08/2016.

[73] Twitter. Firehose. https://dev.twitter.com/streaming/firehose. Accessed: 29/08/2016.

[74] Fred Morstatter, Jürgen Pfeffer, Huan Liu, and Kathleen M Carley. Is the sample good enough? comparing data from twitter's streaming api with twitter's firehose. *arXiv preprint arXiv:1306.5204*, 2013.

[75] Oren Tsur and Ari Rappoport. What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, pages 643–652. ACM, 2012.

[76] Bin Teo. How to identify the relevant hashtags on twitter for your business. `http://hengbinteo.com/identify-relevant-hashtags-twitter-business/`. Accessed: 29/08/2016.

[77] Courtenay Honeycutt and Susan C Herring. Beyond microblogging: Conversation and collaboration via twitter. In *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*, pages 1–10. IEEE, 2009.

[78] Petter Holme and Jari Saramäki. Temporal networks. *Physics Reports*, 519(3):97–125, 2012.

[79] Sungmin Lee, Luis EC Rocha, Fredrik Liljeros, and Petter Holme. Exploiting temporal network structures of human interaction to effectively immunize populations. *PloS one*, 7(5):e36439, 2012.

[80] John Dodd. *Twitter sentiment analysis*. PhD thesis, Dublin, National College of Ireland, 2014.

[81] Ewan Klein Steven Bird, Edward Loper. NLTK, Natural Language Toolkit. `http://www.nltk.org/`.

[82] David Cournapeau. scikit-learn, Machine Learning in Python. `http://scikit-learn.org/`.

[83] Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005.

[84] Elizabeth A Leicht and Mark EJ Newman. Community structure in directed networks. *Physical Review Letters*, 100(11):118703, 2008.

[85] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: theory and experiment*, 2008(10):P10008, 2008.

[86] Jordi Duch and Alex Arenas. Community detection in complex networks using extremal optimization. *Physical Review E*, 72(2):027104, 2005.

[87] Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6):066133, 2004.

[88] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[89] Marta Sales-Pardo, Roger Guimera, André A Moreira, and Luís A Nunes Amaral. Extracting the hierarchical organization of complex systems. *Proceedings of the National Academy of Sciences*, 104(39):15224–15229, 2007.

[90] Benjamin H Good, Yves-Alexandre de Montjoye, and Aaron Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106, 2010.

[91] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104, 2006.

[92] Roger Guimera and Luis A Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, 2005.

[93] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical Review E*, 80(5):056117, 2009.

[94] Jukka-Pekka Onnela, Daniel J Fenn, Stephen Reid, Mason A Porter, Peter J Mucha, Mark D Fricker, and Nick S Jones. Taxonomies of networks from community structure. *Physical Review E*, 86(3):036104, 2012.

[95] Nicolas Dugué and Anthony Perez. *Directed Louvain: maximizing modularity in directed networks*. PhD thesis, Université d'Orléans, 2015.

[96] Antoine Scherrer. The louvain method for community detection in large networks. `https://perso.uclouvain.be/vincent.blondel/research/louvain.html`. Accessed: 31/08/2016.

[97] Sepandar D Kamvar, Taher H Haveliwala, Christopher D Manning, and Gene H Golub. Extrapolation methods for accelerating pagerank computations. In *Proceedings of the 12th International Conference on World Wide Web*, pages 261–270. ACM, 2003.

[98] Youcef Saad. *Numerical methods for large eigenvalue problems*, volume 158. SIAM, 1992.

[99] Ilse CF Ipsen and Rebecca S Wills. Mathematical properties and analysis of google's pagerank. *Bol. Soc. Esp. Mat. Apl*, 34:191–196, 2006.

[100] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.

[101] Alan Ritter, Colin Cherry, and Bill Dolan. Unsupervised modeling of twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180. Association for Computational Linguistics, 2010.

[102] Marco Vicente, Fernando Batista, and Joao Paulo Carvalho. Twitter gender classification using user unstructured information. In *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–7. IEEE, 2015.

[103] Vanessa Friedemann. Clustering a customer base using twitter data. 2015.

[104] Michael Steinbach, George Karypis, Vipin Kumar, et al. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, volume 400, pages 525–526, 2000.

[105] Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan. The planar k-means problem is np-hard. In *International Workshop on Algorithms and Computation*, pages 274–285. Springer, 2009.

[106] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. A local search approximation algorithm for k-means clustering. In *Proceedings of the Eighteenth Annual Symposium on Computational Geometry*, pages 10–18. ACM, 2002.

[107] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[108] M Emre Celebi, Hassan A Kingravi, and Patricio A Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, 40(1):200–210, 2013.

[109] Mark Ming-Tso Chiang and Boris Mirkin. Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads. *Journal of Classification*, 27(1):3–40, 2010.

[110] R Letı, M Cruz Ortiz, Luis A Sarabia, and M Sagrario Sánchez. Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes. *Analytica Chimica Acta*, 515(1):87–100, 2004.

[111] Robert Tibshirani, Guenther Walther, and Trevor Hastie. 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, 2001.

[112] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[113] R Letı, M Cruz Ortiz, Luis A Sarabia, and M Sagrario Sánchez. Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes. *Analytica Chimica Acta*, 515(1):87–100, 2004.

[114] Richard E Bellman. *Adaptive Control Processes: A Guided Tour.* Princeton University Press, 2015.

[115] Michael Steinbach, Levent Ertöz, and Vipin Kumar. The challenges of clustering high dimensional data. In *New Directions in Statistical Physics*, pages 273–309. Springer, 2004.

[116] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is “nearest neighbor” meaningful? In *International Conference on Database Theory*, pages 217–235. Springer, 1999.

[117] Ka Yee Yeung and Walter L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001.

[118] Markus Ringnér. What is principal component analysis? *Nature Biotechnology*, 26(3):303–304, 2008.

[119] Christopher M Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[120] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[121] Lawrence Hubert. Nominal scale response agreement as a generalized correlation. *British Journal of Mathematical and Statistical Psychology*, 30(1):98–103, 1977.

[122] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, 2001.

[123] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[124] Brigitte Bigi. Using kullback-leibler distance for text categorization. In *European Conference on Information Retrieval*, pages 305–319. Springer, 2003.

[125] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

[126] BBC. Jo cox mp dead after shooting attack. `http://www.bbc.co.uk/news/uk-england-36550304`. Accessed: 31/08/2016.

[127] BBC. Thames: Nigel farage and bob geldof fishing flotilla clash. `http://www.bbc.co.uk/news/uk-politics-eu-referendum-36537180`. Accessed: 31/08/2016.

[128] Newstalk. `http://www.newstalk.com/Brexit:-How-did-it-come-to-this`. Accessed: 17/08/2016.

[129] Daniel M Romero, Wojciech Galuba, Sitaram Asur, and Bernardo A Huberman. Influence and passivity in social media. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 18–33. Springer, 2011.

[130] S-W Son, Claire Christensen, Golnoosh Bizhani, David V Foster, Peter Grassberger, and Maya Paczuski. Sampling properties of directed networks. *Physical Review E*, 86(4):046104, 2012.

[131] David R Bild, Yue Liu, Robert P Dick, Z Morley Mao, and Dan S Wallach. Aggregate characterization of user behavior in twitter and analysis of the retweet graph. *ACM Transactions on Internet Technology (TOIT)*, 15(1):4, 2015.

[132] Wei-Chien Chang. On using principal components before separating a mixture of two multivariate normal distributions. *Applied Statistics*, 32(3):267–275, 1983.

[133] Peter Grindrod, Mark C Parsons, Desmond J Higham, and Ernesto Estrada. Communicability across evolving networks. *Physical Review E*, 83(4):046120, 2011.