FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

FEUP

Link Ecosystem of the Portuguese Blogosphere

José Luís Devezas

Report of Dissertation

Master in Informatics and Computing Engineering

Supervisor: Cristina Ribeiro (Aux. Professor)

Co-supervisor: Sérgio Nunes (Lecturer)

1st March, 2010

Link Ecosystem of the Portuguese Blogosphere

José Luís Devezas

Report of Dissertation
Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: Ana Cristina Ramada Paiva Pimenta (Aux. Professor)

External Examiner: Daniel Coelho Gomes (PhD in Web Research, FCCN)
Internal Examiner: Maria Cristina de Carvalho Alves Ribeiro (Aux. Professor, FEUP)

31st March, 2010

Abstract

Over the past few years, blogs have considerably grown in popularity, establishing themselves as a new form of communication media. Allowing hyperlinking and the embedding of images and videos, blogs are a large and interesting field for link analysis and can provide sociologically relevant data. By studying a large sample of the portuguese blogosphere, gathered from March 2006 to October 2009, we aim at unfolding the differences between popular and less frequently cited blogs. We examine several methodologies applied to the study of the web and the blogosphere and use them to illustrate the evolution of our sample. We analyze blog and post creation activity, the growth of the collection and the distribution of the posts in the blogs, and study the network structure of blogs.

Using an ecosystem analogy, we treat blogs as organisms, that interact or interconnect by means of hyperlinks, in the environment of the World Wide Web. We represent the portuguese blogosphere as a graph, and include some of its features as vertex and edge attributes. Data preparation involves querying the original relational database, filtering the records by hostname and date interval, parsing the post bodies and indexing the extracted linkage data using a Berkeley DB. The result is data aggregated by hostname, disregarding links that point to a host outside the collection and any invalid hostnames, that may have been extracted from malformed HTML. Based on the resulting nodes, we go through the index and generate a GraphML representation of the blog graph.

Our data set contains post data for more than 70,000 blogs, with over 400,000 links. We analyze the links between blogs in order to understand how they group and interact, to identify clusters and to characterize them. The blog graph is partitioned into several slices, according to each blog's in-degree. We then study the evolution of blog features, and observe a consistent pattern of decrease in posting frequency, number of out-links, and post length, as we move from the highly-cited blogs to the less cited ones.

This study opens a path for further analysis, performed on the same data set. It might be interesting to study the evolution of blog popularity, in order to understand what influences a blog to become a reference. Regarding the community structure, densely interconnected subgraphs could be identified and characterized, finding the common features underlying each community. Some preliminary work on community detection has already been done in this reasearch, which might serve as a starting point for future analysis.

Resumo

Ao longo dos últimos anos, os blogues têm vindo a crescer consideravelmente em popularidade, estabelecendo-se como um novo meio de comunicação. Permitindo hiperligações e a incorporação de imagens e videos, os blogues são uma extensa e interessante área para a análise de ligações, capaz de fornecer dados socialmente relevantes. Através do estudo duma ampla amostra da blogosfera portuguesa, recolhida entre Março de 2006 e Outubro de 2009, temos como objectivo revelar as diferenças entre os blogues populares e os menos citados. Examinamos várias metodologias aplicadas ao estudo da web e da blogosfera e utilizamo-las para ilustrar a evolução da nossa amostra. Analisamos a actividade de criação de blogues e entradas, o crescimento da colecção e a distribuição das entradas pelos blogues, e estudamos a estrutura de rede dos blogues.

Utilizando a analogia de ecossistema, tratamos os blogs como organismos, que interagem ou se interligam por meio de hiperligações, no ambiente da World Wide Web. Representamos o ecossistema de ligações da blogosfera portuguesa sob a forma duma estrutura de grafo, contendo vários atributos de vértice e aresta. A preparação dos dados requer a interrogação da base de dados relacional, filtrando os registos por domínio e intervalo de datas, processando o conteúdo das entradas e indexando os dados de ligação extraídos utilizando uma base de dados Berkeley. O resultado são dados agregados por domínio, desprezando ligações que apontam para um anfitrião fora da colecção e domínios inválidos, que podem ter sido extraídos de HTML mal formado. Com base nos nós resultantes, percorremos o índice e geramos uma representação GraphML do grafo de blogues.

O nosso conjunto de dados contém entradas de mais de 70,000 blogues, com mais de 400,000 ligações. Analizamos as ligações entre blogues, com o intuito de compreender como eles se agrupam e interagem, para identificar aglomerações e para os caracterizar. O grafo de blogues é particionado em várias fatias, de acordo com o número de citações. Em seguida, estudamos a evolução das características dos blogues, e observamos um padrão consistente no decréscimo da frequência de criação de entradas, número de ligações de saída, e tamanho das entradas, à medida que caminhamos dos blogues mais citados para os menos citados.

Este estudo abre caminho para outras análises, realizadas sob o mesmo conjunto de dados. Poderia ser interessante estudar a evolução da popularidade dos blogs, com o intuito de compreender o que influencia um blogue a tornar-se numa referência na blogosfera. Relativamente à estrutura de comunidade, poderiam ser identificados e caracterizados subgrafos densamente ligados, determinando a característica subjacente a cada comunidade. Algum trabalho preliminar de detecção de comunidades já foi desenvolvido nesta investigação, o que poderá servir de ponto de arranque para uma análise futura.

Acknowledgements

For all the continuous support on the advancements of this research, I would like to thank Cristina Ribeiro and Sérgio Nunes. I would also like to thank SAPO for providing the data set that served as base to the study and Telmo Couto for providing information about his work on characterizing the portuguese blogosphere. Finally, I leave some words of affection to Ana Santos and to my family, who have always supported and believed in me throughout my academic journey.

José Devezas

"Be a simple kind of man,
Be something you love and understand."

Lynyrd Skynyrd

Contents

List of Figures

List of Tables

Abbreviations

BDB	Berkeley DB
DB	Database
FEUP	Faculty of Engineering of the University of Porto
HTML	Hyper Text Markup Language
RAM	Random Access Memory
SCC	Strongly Connected Components
SQL	Structured Query Language
URL	Uniform Resource Locator
WWW	*World Wide Web*
XML	Extensible Markup Language

Chapter 1

Introduction

Blogs have grown in popularity and, more importantly, established themselves as a new form of communication media. Even though there are still many casual bloggers, prominent, committed bloggers and professional bloggers have been developing high quality content, rich in embedded resources, such as video and image, making the blogosphere one of the top sources of information of our time.

Solely by considering the links between blogs, we can determine which blogs are popular, having a high number of citations, or central to the blogosphere, serving as bridges between a great number of blogs. The blog link structure also allows for community detection, based on the premise that a community is likely to be densely interconnected, sharing content about a common interest or simply exhibiting a more general characteristic such as language.

1.1 Context

In the context of a protocol established between the Faculty of Engineering of the University of Porto and SAPO, we were given access to a recent snapshot of SAPO's collection of portuguese blogs, gathered from various blog domains, across a long time span. Following the trail of ideas left by Couto on his blogosphere characterization [Cou09], as a lead for future work, we analyze the link structure of portuguese blogs, experimenting with community detection and generating and characterizing blog clusters based on their estimated popularity.

1.2 Motivation and Goals

Our goal is to understand how portuguese blogs group and interact, identifying clusters and characterizing them. We intend to verify whether or not the model for the link ecosystem, suggested by Couto [Cou09, Figure 4.10], accurately represents the portuguese blogosphere's reality, proving the existence of an A-List, with a distinct behavior from the remaining blogs.

More specifically we intend to identify and highlight the features that distinguish intensely cited blogs from unrecognized blogs. For this analysis, we partition the blog network into several slices, using the number of in-links as a clustering criteria, and study the evolution of several characteristics, from the highly cited to the less cited slices.

1.3 Dissertation's Structure

This document is divided into five chapters, including Chapter 1, the Introduction. Chapter 2 provides an overview of third party research, on the characterization of the blogosphere, including work based on previous snapshots of our collection, and the analysis of the web and blog link structure. On Chapter 3 we present the collection used in our analysis, explain the process of data extraction and validation, and briefly characterize the blogosphere. On Chapter 4 we provide the graph theory concepts underlying our blogosphere representation, present the data structure and the technologies used to process the blog network, explain the data extraction and preparation process and do a characterization of the blog graph and an analysis of blog clusters grouped by popularity. Chapter 5 summarizes the conclusions and main contributions of our research and proposes a possible lead for future work.

Chapter 2

Researching the Blogosphere

In order to deepen our insight and to better understand the methodologies associated with the study of the blogosphere, we gathered and analyzed existing research on the characterization of the blogosphere, mainly focused on the content and the evolution of the collection, and also on link analysis, applied both to the web and the blogosphere.

2.1 Blogosphere Characterizations

The blogopshere is rich in information. Fully characterizing it means taking into consideration metadata, content, commentary and link structure. Research can branch into several different areas. The analysis of content, for instance, can involve tasks like sentiment analysis, where we use natural language processing to extract the opinions of a blogger with respect to a topic, trend detection, where we identify active topics in the blogosphere, or link polarity, where we acknowledge whether a link is positive, negative or neutral. Another important branch of blogosphere research is link analysis. We can base our study solely on the link structure, treating the blogosphere as a graph. Using the connections between blogs, we can assign popularity ranks, based on the in-degree of the nodes or more elaborate methods like PageRank or HITS, we can do community analysis by identifying densely connected subgraphs, or study the centrality of blogs.

Herring et al. [HSKW07] have done a longitudinal content analysis of blogs, studying the evolution of several characteristics over time, including the number of words and the number of links. They divided the results of their research into three categories: change, stability and variability. Characteristics showing a pattern over time were either in the category "change", in which case they consistently increased or decreased, or in the category "stability", in which case they remained fairly unaltered over time. Whenever a pattern was not identified, characteristics fell into the "variability" category.

Qazvinian et al. [QRSA07] studied the distribution of comments in persian blogs, verifying that the number of comments for a post was directly dependent on the number of comments left for that post in the first day. The researchers used a typed graph to represent the connections between bloggers. Depending on the type, each edge of the graph represented a blogroll link, a post link (found in the content of a post), a comment out-link (found in the content of a comment) or a comment in-link (footer link pointing to a resource belonging to the comment's author).

Given the protocol established with SAPO, the Internet service provider granted portuguese scholars access to a collection of blogs, both obtained from popular blogging services, like Blogger, and from their own SAPO Blogs service. This resource has been used for several studies surrounding blogs, their content and their link structure.

In 2008, Pinto and Branco collaborated on the characterization of SAPO's blog collection. Pinto researched "Detection Methods for Blog Trends" [Pin08], introducing a variation of the Frequency Segments algorithm, that was used to extract the most relevant topics for a month. Branco applied the h-index, commonly used as a measure for the quality of a scientist's work based on the citation counts for his papers, to analogously classify and order blogs [Bra08].

In 2009, Couto studied the representativity of SAPO's blog collection concerning the portuguese blogosphere. In "Characterizing the Portuguese Blogosphere" [Cou09], Couto presented multiple statistics illustrating the evolution of the blogging activity over the years, verifying it increased over time and identifying blogging habits. He observed a growth in the link usage and suggested a possible model for the link ecosystem of the portuguese blogosphere.

2.2 Link Analysis

The web and the blogosphere have been subject to several studies focused on the link structure. Based on the methodologies used in these studies, we intend to explore the portuguese blogosphere, identifying and characterizing several blog clusters, in order to determine the existence of an A-List, a group of influential blogs that share distinct features from the remaining blogs.

Studying the Web and the Blogosphere as a Graph

In the year 2000, Broder et al. [BKM+00] studied the web as a graph, using data provided by the AltaVista search engine. This collection compiled more than 200 million pages and 1.5 billion links from a total of two crawls. Using the Connectivity Server 2 software, at the Compaq Research Center, which provided fast access to the linkage data, they made a series of studies based on both direct and undirected versions of the Web graph. They

determined the graph diameter – defined as the length of the shortest path from u to v, averaged over all ordered pairs (u, v) such that there is a path from u to v –, studied degree distributions – number of in-links and out-links per page –, the connected components and the macroscopic structure. They found that the in-degree and the out-degree distributions follow a power law. This behavior is compared to a fractal-like quality, in the sense that it is verified on a macroscopic level – the entire Web –, a microscopic level – a single Website – and the levels in-between.

Using a breadth-first graph traversal algorithm, they chose 570 random nodes to start from and ran the algorithm both forward and backward, verifying that it would either "die out", covering a rather small portion of nodes, or "explode", covering an extremely big group of nodes, but not all of them. They presented the theory that the connectivity on the Web followed a "bow-tie" model, made of three main clusters: in the center, the Strongly Connected Components (SCC); connecting to the SCC, a group of nodes denominated IN, start nodes for the forward breadth-first search that resulted on an "explosion"; out-linked from the SCC, a group of nodes called OUT, that resulted from the symmetrical experience – the backward breadth-first search "explosion". Some nodes that could be reached from portions of IN and others that could reach portions of OUT were called TENDRILS. A tendril from IN directly connecting to a tendril from OUT was called a TUBE. Both tendrils and tubes aren't part of the IN, SCC and OUT clusters. Given the blogosphere can also be represented by a graph structure, using posts or blogs as nodes, we found this study relevant to our research, since it shares the concept of network.

In 2005, Kumar et al. [KNRT05] presented two reasons for the systematic study of the blogosphere: social reasons – community interactions – and technical reasons – the organization of the blogosphere allows for a time-oriented analysis. They introduced and focused on a time-driven version of the blog graph, called the blog time graph, studying various prefix graphs – sets of nodes from the time axis origin to a certain point in time. They developed the notion of bursty communities of blogs, topically and temporally focused, and introduced the concept of randomized blogspace. All data was crawled from seven popular blogging services, resulting on $21,109$ blog URLs and a blog graph with $22,299$ nodes, $70,472$ unique edges and $777,653$ total edges. Time information was later associated with the graph.

As presented in "Trawling the Web for emerging cyber-communities" [KRRT99], Kumar et al. detected communities by identifying bipartite cliques, meaning co-citing parties were considered to share an interest or common characteristic. Whenever those connections were dense enough, the node set was marked as a community. Burst detection was an expansion of Kleinberg's work [Kle02], who detected bursts using a high and low state automaton. Whenever an event was on a low state for a long period of time and then suddenly erupted into a high state, during a short time interval, a burst was detected. These bursts of activity were characterized by a high rate of linking between the involved

entities.

Several prefix graphs were analyzed, taking into consideration macroscopic and microscopic phenomena. On a macroscopic level, Kumar et al. studied the evolution of the Strongly Connected Components (SCC), which exhibited a slow growth until 2001, at which time 3% of all nodes were contained in the SCC; it then expanded faster, reaching a maximum of 20% to the date of the article's writing. On a microscopic level, they presented the fraction of nodes participating in a community and the evolution of the communities, regarding the number of communities in the blogspace and the number of nodes that were part of a community.

The blog graph was compared to its randomized version, verifying that the SCC had a similar growth to its randomized blogspace version, even though the randomized SCC attained higher values sooner. On the other hand, Kumar et al. concluded that community formation was not solely related to the graph expansion, since the randomized blogspace community size and number of intervening nodes were an order of magnitude smaller than the real study case, making it clear that community structure is a human and social factor.

In 2009, Cha et al. [CPH09] studied the media content spreading across a network of blogs from 15 distinct blog hosting sites – totaling 8.7 million posts and 1.1 million blogs, aggregating multiple domains and language groups. Analyzing the blog graph, they verified that the degree distribution was heavy-tailed, as the degree value increases, but there is a significant number of blogs spread over the low degree values. They also characterized the blogosphere has having a low reciprocity – bidirectional connections between blogs are scarce – and a low density – the number of actual connections versus the maximum number of possible connections is small. Links happen between domains, but are seldom used to connect blogs written in different languages, with the exception of non-English to English-written blogs.

Verifying that YouTube was the number one cited resource on their data set, Cha et al. decided to deepen their research by gathering data about the cited YouTube videos – category, age and spreading speed. Interestingly, they concluded that media content spread according to two distinct patterns – flash floods and ripples. Flash floods usually occur with news, political commentary and opinion contents. These types of contents quickly propagate and then disappear. Ripples, on the other hand, are usually associated with more timeless content, like music and other forms of entertainment. This media category tends to spread a lot slower and through a long period of time.

Community Detection

A community is a group of individual entities that share a common characteristic, be it language or a mutual interest. In graph theory, the concept of community is a synonym

of a dense subgraph. Physicists have developed methods to identify and extract dense portions of a graph. Most of these methods are based on the function of modularity, which quantifies the quality of a division of a network into modules. Modularity results in a high value when the network division proposal is made of modules that are internally dense and sparsely connected to each other.

Some of the research on community detection includes "Finding community structure in very large networks", by Clauset et al. [CNM04], who proposed a low computational cost algorithm based on a fast greedy optimization of the modularity score, "Computing communities in large networks using random walks", by Pons & Latapy [PL05], who developed a *Walktrap* algorithm based on the intuition that random walks on a graph tend to get "trapped" into densely connected parts corresponding to communities – they only use the modularity measure as term of comparison to other methods – and "Finding community structure in networks using the eigenvectors of matrices", by Newman [New06], who proposed a new method for maximizing the modularity function in terms of the eigenspectrum of the modularity matrix.

Other methods for community detection include "Statistical Mechanics of Community Detection", by Reichardt & Bornholdt [RB06], who find communities in graphs via a spin-glass model and simulated annealing, and the detection of communities based on the successive removal of edges with a high betweenness value – this method is based on the work of Freeman about centrality in social networks [Fre79].

2.3 Summary

In this chapter we provided an overview of the research on the characterization of the blogosphere, including work based on previous snapshots of our collection, and the analysis of the web and blog link structure. These studies, dated from 1999 to 2009, served as base to our research, mainly regarding methodology and data structure. They inspired the association of content analysis with link analysis, by means of a graph structure with vertex and edge attributes. Similarly to Qazvinian's typed graph, we store metadata and content-related characteristics. Using some of the techniques available in the reviewed research documents, we present an organized method for studying the differences in behavior between popular and unpopular blogs regarding their features.

Chapter 3

Characterizing the Blogosphere

We explain the fundamental concepts regarding the blogosphere and its network structure and introduce the blog collection subject to our analysis. We present the technologies that support our research and describe the process of data extraction and validation, briefly characterizing a sample of the portuguese blogosphere.

3.1 Blog Terminology

In this section, we define the concepts inherent to our research, making an overview of the structure of the blogosphere and explaining how some of these concepts apply to the context of our work. We define "blogosphere" and the activity of blogging and describe the network structure found in blogs.

What is a Blog?

A blog or web log consists of a set of entries, called posts, that are organized in a timely manner, from the most recent to the oldest, using the web as a presentation medium. Essentially, blogs are websites with a special structure. Blogs can be collaborative, having more than one author, usually associated to particular posts, and they can link to resources around the web, either explicitly or by embedding them. Connections to other blogs are commonly gathered in a blogroll, displayed in a sidebar list, and are global to the blog. Citations to other blogs and posts can also appear in the actual content of the blog's posts, as HTML anchors. Depending on the service used or the permissions set by the blog owner, posts can be commented by the readers. A person who writes for a blog is called a blogger and the act of doing so is called blogging.

Each post has several data fields: a title, representative of the content, an author, the writer of the entry, and a creation date. Some blog services allow tagging or categorization of posts, having extra fields for that purpose.

Defining the Blogosphere: The Network Structure of Blogs

The blogosphere, also called blogspace [KNRT05] or blogistan [CK06], is the set of all blogs and the connections between them. Even though the blogosphere doesn't have an explicit link structure, like the one that exists in a social network, it is possible to infer that structure from the citations present in the content of the posts. We consider all the URLs belonging to an HTML anchor, embedded resource or image. By filtering those URLs by hostname and restricting them to blog links, we can build the blog graph, a directed graph where each vertex represents a blog and each edge a link between blogs.

Using the blog graph as a representation of the network structure of the blogosphere, we can apply known social network analysis algorithms to explore and study several aspects of the blogosphere. We can identify communities, represented by densely interconnected subgraphs, and central blogs, that serve as bridges between a great number of blogs, or assign popularity ranks to blogs based on the number of in-links.

3.2 The Collection

The collection we use for our analysis was provided by SAPO, a portuguese Internet service provider and owner of SAPO Blogs [SAP09], a popular portuguese blog hosting service. This collection is made of a group of posts, written in portuguese, from various blogging services, mostly SAPO Blogs and Blogger.

Since we are unaware of the criteria used to select portuguese blogs outside of SAPO's domain and cannot ensure the thoroughness of that data, we decide to focus our study on SAPO's blogs, which have previously been determined, by Couto [Cou09], as representative of the portuguese blogosphere. The data set we use compiles more than 96,000 blogs, with over 2,247,200 posts and 459,700 links, extracted from a table with approximately 17 GB. Each blog is hosted under the "blogs.sapo.pt" domain using a user-defined subdomain, in the format "blogname.blogs.sapo.pt".

We have access to posts with dates ranging from March 1st 2006 to October 1st 2009. This means that the data set compiles post data starting from the latest release of SAPO Blogs until the month we start our analysis. We do not, however, consider October 2009, because we only have posts for the first few days of that month. All the data is stored in a MySQL relational database management system and, for each post, we have access to a series of fields, from which we only use the ID, the URL, the creation date and the actual content of the post.

3.3 Technologies

The provided data set was stored in a MySQL relational database management system, making it imperative for us to use this technology, in order to access the post table. We adopted the Perl language to retrieve and organize the information available in the database. Perl is a powerful scripting language, with an extended collection of comprehensively organized modules. This facilitated the task of accessing the SQL database and the task of parsing the content, which was either done by using regular expressions or by accessing libraries designed specifically for HTML processing. Perl also provides modules to create and manage Berkeley DBs, which we use for indexing blog and linkage data, and to build XML documents, which we need for the link analysis on Chapter 4.

At this stage, we only use the `DBI` [DB00] module to query the database and the `BerkeleyDB` [OBS99] module to write a blog index, using the hostname of the blog as key and its posts and respective creation dates as value. This process is described in Section 3.4.

Since our current goal is to characterize the data set, based on statistics that illustrate the evolution of the corpus, we also need a tool that will allow us to support that activity, providing the data structures and the statistical methods we require and the ability to plot charts, depicting the obtained results. Spreadsheets have most of the features we need, except they have a limited number of rows we can work with and usually aren't very extendable, so we adopt a tool called R [R D09]. R is a free software environment for statistical computing and graphics, with its own scripting language and library repository. R natively supports chart plotting. However, since it requires a greater deal of effort regarding plot details, like captions or multi-layering, we decide to use the ggplot2 [Wic09] library, which makes this task a lot easier and results in more elegant charts.

3.4 Data Extraction

Given the dimension of the database where the collection is stored, we need to acquire and pre-process the data specific to our characterization, optimizing it for quick access. Since there are blogs from other domains, stored in the relational database, that we won't be considering for our study, we need to do a query where we extract only the blogs that contain "blogs.sapo.pt" on the hostname. Also, because we are doing an analysis over time, we choose a date range that includes all the posts from the beginning of the collection to the end of the most recent month we have full information for. In order to avoid thrashing, the queries are ordered by ID and limited to groups of 5,000 records, clearing the memory after we're done processing each batch. This way we can restart the process, querying the table for the next 5,000 records, with an ID larger than the ID of the last processed post.

Date	Blogs	Posts
2006-03-01	13	20
2006-03-02	16	44
2006-03-03	6	23
2006-03-04	10	28
2006-03-05	15	31
2006-03-06	5	24
...	...	...

Table 3.1: Number of new blogs and posts created for each day.

At first, we are only interested in extracting a list of posts that will help us establish our data set. This way, if we query the database later, we will only consider data regarding this previous selection. By parsing the list of posts, we extract the corresponding list of blogs. At this point, we've determined which posts and blogs are to be characterized and, further on, considered in our network analysis.

Next, we need to assign a creation date to each post and blog. Once again we query the database, retrieving the date field for each post and generating a new text file with a post URL and respective creation date per line. As for the blog creation date, since it's not specifically stored in the database, we consider it to be the same as the creation date for the blog's first post, as done by Couto [Cou09].

Using the post URL and creation date list to extract the earliest dated post for each blog proved to be ineffective, producing a hard drive thrashing effect. Since we are sequentially reading an unordered, ungrouped post/date list, we always add the blog correspondent to each post to a hash, if it doesn't exist, or compare the date of the post with the current one to see if it's an earlier date. The issue with this method is that the hash grows too big in dimension, making the process progressively slower while generating a hard drive thrashing effect.

A possible solution for this problem is grouping the posts by blog. With that goal in mind, we generate a Berkeley DB key/value database containing the blog hostname as key and a list of posts and respective dates as value, for each key. Using this database we find the earliest post date for each blog, define it as the creation date for the blog and save these results into a text file. As a verification, we compare the new blog list with the one previously extracted from the post list, confirming that the blog names are the same.

To complete the process, we generate a text file, formatted as a table with blank space as a column separator, that contains the number of new blogs and new posts per day. We load this data (illustrated in Table 3.1) into R, as a `data.frame` structure, for it to be analyzed and depicted, by plotting the corresponding chart with ggplot2.

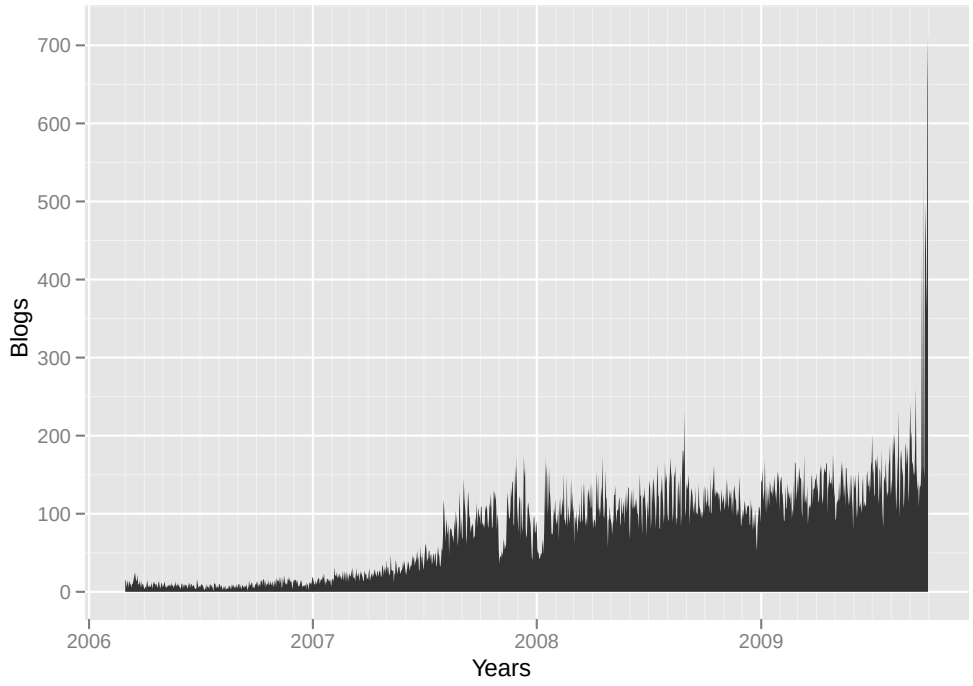

Figure 3.1: Newly created blogs, per day, over the years (including September 2009).

3.5 Data Validation

We depict the newly created posts and blogs, per day, over the years. The results prove to be less straightforward than we expected. Figure 3.1 shows the number of new blogs, created per day, over the years. During the last month – September 2009 –, there is a peak that stands out. Using R, we determine that the average value for newly created blogs per day is 79.09 and the median value is 88. Having a value of 715 for the newly created blogs on the last day of September 2009 appears to be an irregularity.

In an attempt to understand and explain this pike, we manually browse through some of the blogs created on 30 September 2009 and verify that many don't exist anymore. Since this verification is being made less than one month later, it all indicates that those blogs have been deleted by SAPO for being spam blogs (splogs). We create a script that, for a given day or month, returns a list of blogs that don't exist anymore – this means the HTTP request either returns a "404 Not Found" error or a SAPO web page with the information that the user is unknown.

We run the script for September 2009 and get a list of 3,187 bogus blogs – 42% of September 2009 blogs don't exist on 21st October 2009. Even though some of the blogs could have been deleted by their owners, the percentage of bogus blogs represents almost half of the blogs for that month and most of the usernames for those blogs seem to be computer generated, with very few exceptions. By running the script for the rest of the months of 2009 — January to August — we verify that, in average, 22% of the blogs

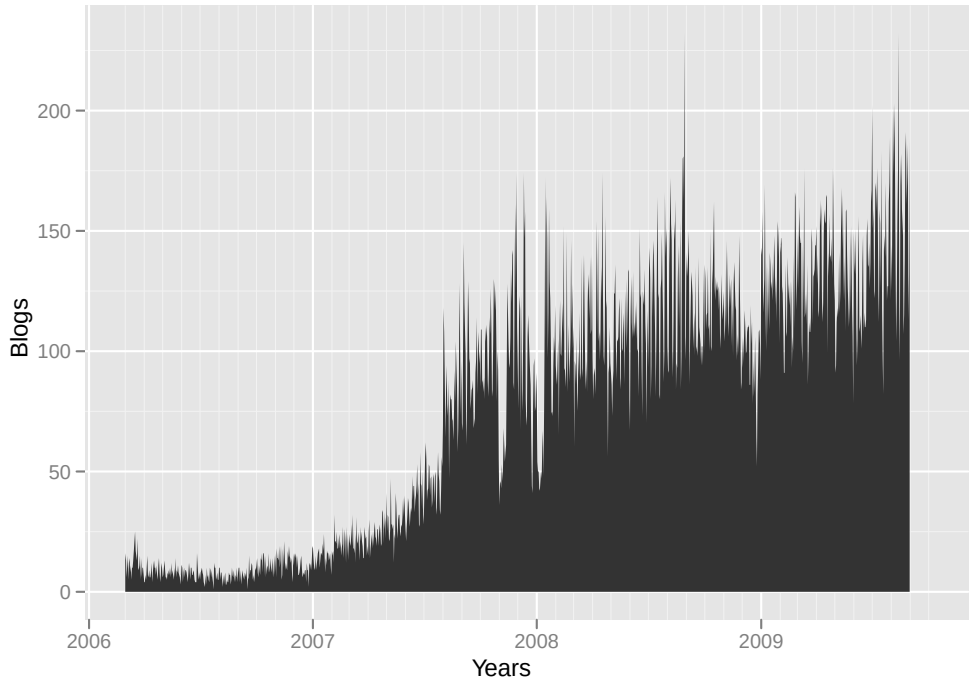

Figure 3.2: Newly created blogs, per day, over the years.

don't presently exist on the web. Though September 2009 is the most recent month of the collection, it already has twice as much bogus blogs than the average for the previous months of the same year.

We can then speculate that SAPO frequently removes splogs from the collection and that the pike on the chart is associated with a time frame when the cleaning process hasn't yet been applied. Being unsure of how representative that month really is and given that it is out of our scope to study spam blogs, their detection and removal, we decide to simply leave September 2009 out of our analysis.

3.6 Result Analysis

At this point, we are using a data set that compiles all SAPO blogs from March 1st 2006 to August 31st 2009. We are interested in doing a brief characterization of the evolution of the collection, regarding blog creation and posting activity, dimension of the sample and post distribution. With the month of September 2009 purged from our collection, we plot the charts depicting these statistics.

Figure 3.2 shows the number of newly created blogs, per day, over the years. Removing September 2009 results in a more homogeneous distribution. We verify that around the second half of 2007 there was a significative growth; people create more blogs after that date than they did before – this might be indicative of a marketing campaign carried

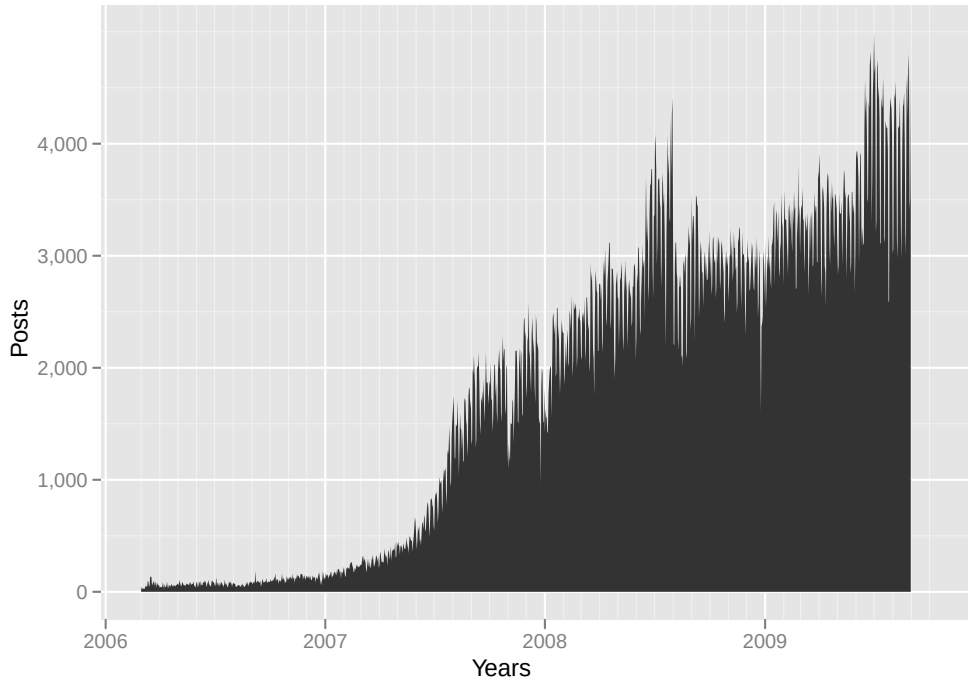

Figure 3.3: Newly created posts, per day, over the years.

on by SAPO, at the time. Before that date, the number of new blogs per day ranges from about 10 to 50 and, after that date, it ranges from about 100 to 150.

Figure 3.3 depicts the number of newly created posts, per day, over the years, presenting a behavior similar to Figure 3.2. Before the second half of 2007, the number of newly created posts per day ranges from approximately 0 to 1,000 and, after the first half of 2007, it ranges from about 2,000 to 4,000, clearly showing a more intensive posting activity.

We illustrate the growth of the collection, over the years, regarding the number of blogs (Figure 3.4) and the number of posts (Figure 3.5). The average monthly growth rate for the number of blogs is 14.96% and 19.53% for the number of posts. Again, there is a rapid growth, after the first half of 2007, for both values.

In order to understand how posts are distributed inside blogs, we depict the ratio between total number of posts and total number of blogs, for a given day, over the years (Figure 3.6). At the beginning, the ratio of posts per blog shows low values, starting at 1.5. It then grows to 16.6, halfway the time frame, reaching the value of 23.4, for the current state of the blog collection.

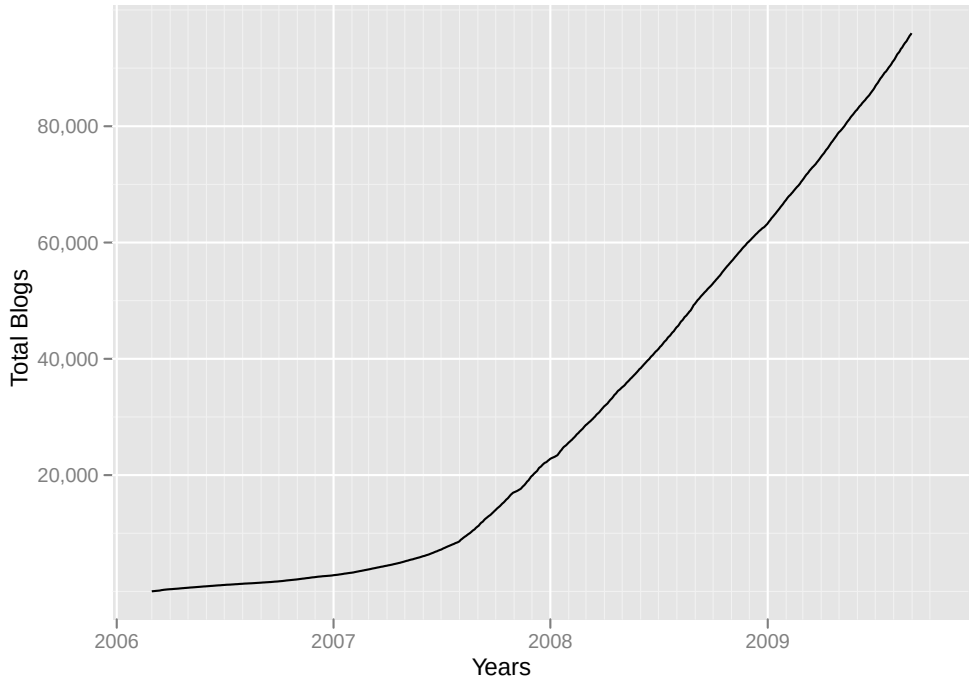

Figure 3.4: Total number of blogs, per day, over the years.

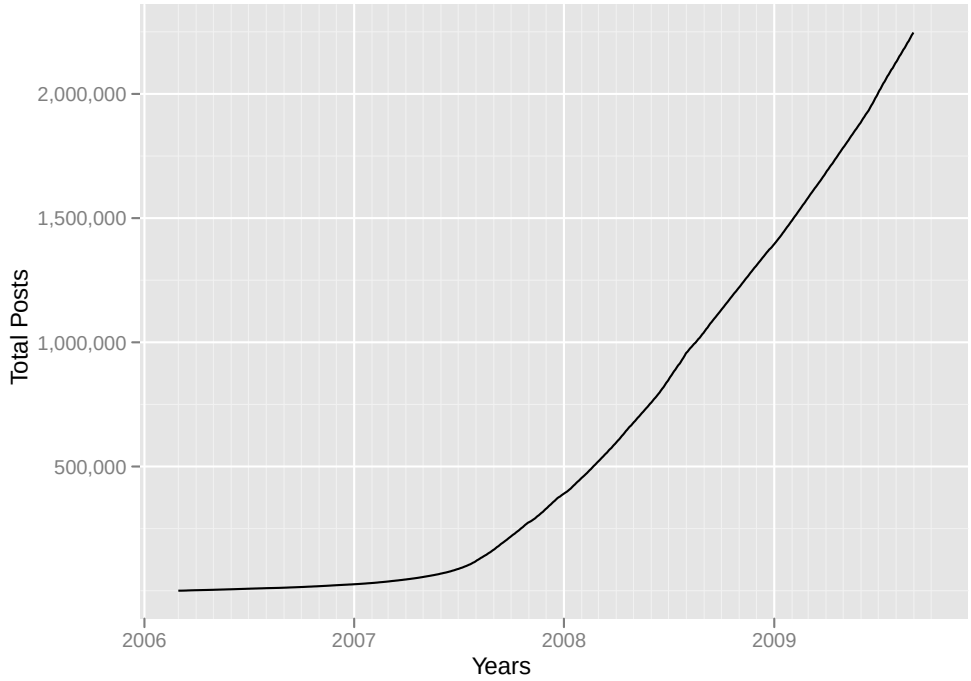

Figure 3.5: Total number of posts, per day, over the years.

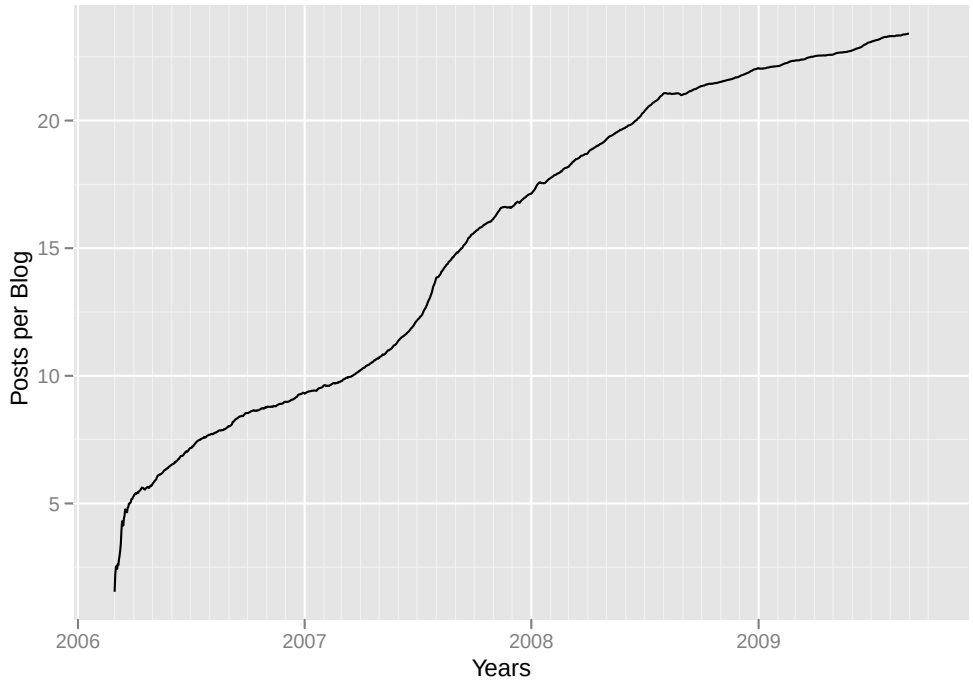

Figure 3.6: Daily posts per blog over the years.

3.7 Summary

In this chapter, we described some blog concepts, exposing the network structure of blogs, and introduced the collection that will serve as base to the link analysis carried in Chapter 4. We provided an overview of the technologies used for the characterization and described the data extraction and validation process. We illustrated the evolution of the blogosphere, regarding blog and post creation activity, growth of the collection, and posts per blog ratio. We concluded there was an accentuated growth of activity, after the first half of 2007, which is consistent with the behavior presented in previous characterizations of older snapshots of the same collection.

Chapter 4

Link Ecosystem Analysis

An ecosystem consists of organisms that live and interact within an environment. In the context of our work, link ecosystem refers to the blogs as organisms, with specific characteristics and behaviors, that interact with each other, interconnecting by means of hyperlinks, in the World Wide Web environment. We represent the blog link structure using a graph with vertex and edge attributes, that we explore using a network analysis tool. We develop a series of programmatic methods to help us characterize the collection and partition it into slices, using the number of citations as criteria, and study the behavior of the set of parts.

4.1 The Blog Graph

We present a brief review of graph theory, defining "graph" and its properties, and describe the data structure used to represent the sample of the blogosphere used in this work.

4.1.1 Graph Structure

We use a graph data structure to represent the blogs and their interconnections. A graph $G = (V,E)$ is made of a set of vertices V, that represent objects or entities, and a set of edges E, each representing a link between two vertices in V. A graph can be directed, in which case an edge corresponds to an ordered pair of vertices, or undirected, when the pair of vertices is unordered, merely symbolizing a bilateral connection between two vertices.

The blog graph used to represent our collection is a directed graph. Each vertex represents a blog and each edge acts as a directed link between two blogs. A blog can link to itself, for instance citing one of its posts, and more than one link may exist for a given

	Attribute	Description
Blogs	`name`	Blog hostname.
	`date`	Creation date of the oldest post.
	`hostgraph.outdegree`	Total number of links extracted from the blog's content.
Posts	`post.url`	Complete URL for the link source
	`post.date`	Creation date of the post.
	`post.wordcount`	Number of words of the posts's content.
	`post.charcount`	Number of characters of the post's content.
Links	`name`	Complete URL for the link target.
	`source`	Link source blog node.
	`target`	Link target blog node.

Table 4.1: Summary of the information stored in the blog graph.

pair of blogs. Post information is stored as edge attributes, meaning that we only save data for posts with links in their content.

Attributes are stored for each vertex and edge, representing post data, including computed characteristics. The vertices have the attributes `name`, `date` and `hostgraph.outdegree` — `name` stores the hostname of the blog and `date` stores the creation date of the blog; `hostgraph.outdegree` stores the actual number of out-links for the blog, including those for hosts outside of SAPO Blog's domain. The edges have the attributes `name`, `post.url`, `post.date`, `post.wordcount` and `post.charcount` — `name` stores the complete URL for the link target, `post.url` stores the permalink for the post where the link was extracted from, `post.date` the respective creation date and `post.wordcound` and `post.charcount` the number of words and characters (including white space) for the post. Table 4.1 summarizes the information captured in the blog graph and Table 4.2 provides examples for that data. Both the post and link information is stored in the graph edges.

	Attribute	Example
Blogs	`name`	`blog.blogs.sapo.pt`
	`date`	`2007-10-11 16:22:57`
	`hostgraph.outdegree`	`50,077`
Posts	`post.url`	`http://blog.blogs.sapo.pt/1046448.html`
	`post.date`	`2008-09-09 19:14:49`
	`post.wordcount`	`25`
	`post.charcount`	`216`
Links	`name`	`http://outro.blogs.sapo.pt/25856.html`
	`source`	`blog.blogs.sapo.pt`
	`target`	`outro.blogs.sapo.pt`

Table 4.2: Example of the information stored in the blog graph.

4.1.2 Properties and Concepts

When doing link analysis, there are some graph properties and concepts whose meanings must be made clear, in order to be able to create a correspondence between the graph structure and the blogosphere's reality:

- **Degree** — The degree measures the number of edges associated with a node. For a directed graph, we can divide the degree into two categories: in and out. The in-degree of a vertex v is the number of edges that target v, or have v as the destiny vertex. The out-degree, on the other hand, is the number of edges that have their source in v, that is, the number of edges going out from v.

- **Density** — The density of a graph is the ratio between the number of existing edges and the total possible number of edges. For instance, a complete graph is 100% dense, since its definition states that any pair of distinct vertices is connected by an edge.

- **Reciprocity** — The reciprocity of a directed graph is the percentage of double links for the edge set. A graph is 100% reciprocal if, for each directed edge, there is an inversely directed edge, connecting the same vertices. The reciprocity is calculated by the ratio between the number of bidirected links and the total number of existing links.

- **Diameter** — The diameter of a graph is the length of the longest path between any pair of vertices. It can be computed by finding the shortest paths between all pairs of vertices and identifying the longest of these. Broder et al. [BKM+00] have used the average value of the path length over all pairs of vertices. We do, however, use the traditional definition of diameter.

- **Connected Component** — A connected component of an undirected graph is a subgraph consisting of the maximal set of vertices and edges where, for any two vertices, there is a path connecting them. On directed graphs, connected components can either be strong or weak.

- **Strongly Connected Component** — A strongly connected component of a directed graph is a connected component that takes into consideration the direction of the edges. This means that, for every two vertices, there is a path, that respects edge directions, connecting them.

- **Weakly Connected Component** — A weakly connected component of a directed graph is a connected component that does not take into consideration the direction of the edges. The weakly connected components of a directed graph can be determined by identifying the connected components of its underlying undirected graph.

4.2 Technologies

Our main goal is to analyze the blog network. With that in mind, we need to choose a representation method that provides the opportunity to compute several elemental network properties and facilitates any further processing, like community detection or subnetwork extraction.

Network analysis is often based on the graph structure. There are several libraries dedicated to graph processing and analysis, like JUNG [OFS+05], the Java Universal Network/Graph Framework, or statnet [HHB+03], a set of software tools for the analysis, simulation and visualization of network data. We opt, however, to use the igraph [CN06] library as our main analysis tool for the blog graph, since it's known to deal well with high volumes of data and can be used either in R [R D09] as a package, in Python as an extension, or in C as a library, running in Windows, Mac OS X and Linux.

Using igraph we are able to manipulate the blog graph, extracting sets of vertices or edges according to a selected criterium or assigning them attributes. We can also calculate structural properties, like the in-degree and the out-degree, identify the connected components of the graph and detect communities using several different heuristics. When using the R or Python interfaces, it is also possible to generate a visualization of the graph, manipulating several display properties and applying various layout algorithms.

In order to load the graph structure into the network analysis tool, we need to choose one of the many file formats supported by igraph. GraphML [BEL05] is a widely used format, that allows the representation of attributes for the vertices and nodes of the graph. It's an XML dialect, so it is easy to generate a GraphML document that represents the blog network. With Perl, we extract the links from the post content using `HTML::LinkExtor`, compute characteristics like number of words and characters for a post, applying the `HTML::FormatText` module to convert the HTML to plain text, and write the graph data using `XML::Writer`.

4.3 Data Extraction

Given the dimension of our data set, which is over 17 GB, the data extraction process is a fundamental stage of our analysis. We go from the relational database to the blog graph in five steps, saving the results for each step, and trying to discard as little information as possible.

Extracting the Links

The first step is to go through each record of the post table, on the relational database, and extract the HTTP URLs found on the post content. Each URL must be either acquired from the value of an `href` attribute of an HTML anchor `a`, or from the value of

a `src` attribute of an `img` or `embed` tag. This means we only consider explicit links and embedded resources, like images or videos.

In order to avoid thrashing, we process only enough records to fit in the 2 GB RAM memory, choosing a rather conservative limit of 5,000 records, since we also need to allocate memory for the local variables associated with the processing of each post. To allow this segmentation, we query the data ordered by ID and save the last processed ID to a temporary text file. We access this file to continue the process, after clearing the memory for the previously processed batch of records.

We query the SQL database for all the posts that meet the following conditions: the URL field contains the string ".sapo.pt" and the post date field ranges from "2006-03-01 00:00:00" to "2009-08-31 23:59:59". Using the `HTML::FormatText` Perl module, we convert the post content into plain text and compute its number of words and characters.

For each post, we build a string concatenating the post URL, the post date and the number of words and characters on the post content. Next, using the module `HTML::LinkExtor`, we extract the links from the post content, storing them into an array and discarding the ones that don't begin with `http://` or that are more than 256 characters long. We store each link in a Berkeley DB as a key, with the respective value corresponding to the list of posts where the link is cited — each element of this list is the string we previously built. The end result for an entry should be structured as follows:

```
http://bit.ly/3d4a4 ->
    http://someblog/3222.html|2006-05-02 05:43:54|50|200\t
    http://otherblog/232.html|2006-04-23 12:23:34|19|101\t
    http://someblog/1223.html|2008-06-21 13:32:43|7|32
```

This concludes the indexing process of all the linkage data that we will use to build the blog graph, the main object of our study.

Aggregating by Hostname

The second step is to go through all of the indexed data and do an aggregation by hostname. Initially, we wanted to be able to prune low degree nodes previous to generating the blog graph, so we calculated the in-degree and out-degree for each host and stored it in a text file, formatted as a table, which could then be read and stored in a `data.frame` R structure. Table 4.3 shows an excerpt of the resulting structure. What we do, in fact, is to leave the pruning process as an operation we execute in R, prior to loading the graph. The calculated host graph out-degree value is kept, so we can later access that information for each node on our blog graph.

We use Perl regular expressions to extract the hostnames from the link target (the Berkeley DB key) and the link source (the Berkeley DB value) and, using a hash with the hostname as key, we increment a variable for the in-degree of each target and a variable

Hostname	In-Degree	Out-Degree
modacao.blogspot.com	1	0
1415139306802.usercash.com	1	0
summeromance.blogs.sapo.pt	1	0
watch-star.gnqmx.myip.org	99	0
kogler.net	2	0
aquelaopiniao.blogs.sapo.pt	7	172
...	...	...

Table 4.3: Host graph degree.

for the out-degree of each source. Note that a target can be a blog, so the only value we aren't accounting for is the out-degree of hosts outside of SAPO Blog's domain — this is data we don't have access to and will not be considered in our analysis.

As we generate the degree table (Table 4.3), we also filter out data that may have been extracted from malformed HTML, ensuring that the table only contains valid URL strings. At this point, we have a list of all the hostnames that can be used to build a host graph, together with their in-degree and out-degree, and an index with the link structure necessary to define the edges of the graph. Note that the resulting data for the host graph IS incomplete, as it includes all the in-links and out-links for SAPO blogs, but none of the out-links and only some of the in-links for other domains, which illustrates the web citations found in post content.

Extracting Blogs and Associating the Creation Date

Since our purpose is to study the link ecosystem of the blogosphere, prior to generating the blog graph we have to remove any hostname that isn't part of our blog network, specifically any domains outside of SAPO Blogs. By now, we have Table 4.3 loaded as a `data.frame` in R, so we apply a filter to select all the table lines containing a hostname with the string "blogs.sapo.pt", the domain for our sample.

This operation is momentary, so we also append the creation date for each blog, that we have previously determined in Section 3.4, to the filtered table. We merge the blog degree table (resulting from the filter applied to Table 4.3) with the blog creation date table, using the hostname as the intersection between both tables. For every blog creation date not matched, we define the date as `NA` (R's built value for "Not Available"). We then save the resulting table (depicted in Table 4.4) into a text file.

Generating the GraphML

Using the Berkeley DB link index and Table 4.4, we generate the GraphML file for our network of blogs. A GraphML document is made of a sequence of nodes, each with a distinct ID, and a set of edges, with source and target attributes, that contain an ID

Hostname	In-Degree	Out-Degree	Creation Date
000333.blogs.sapo.pt	0	6	2009-01-17 19:37:28
001972.blogs.sapo.pt	0	7	2008-02-15 16:08:25
001warez-bb.blogs.sapo.pt	0	3	2009-08-31 21:50:07
0023missaoaniversario.blogs.sapo.pt	2	0	2009-04-17 16:35:16
002ordemparacriar.blogs.sapo.pt	1	6	2007-10-24 17:33:25
004.blogs.sapo.pt	0	1	2009-05-23 15:07:30
...	...	...	...

Table 4.4: Blog graph degree and blog creation date.

of a defined node. Our network analysis package — igraph — can read and store user defined attributes for nodes and edges. The `name` attribute of each node is used for the display of the vertex and edge lists in igraph, so we opt to use this attribute to store the hostname of each node and the URL for each edge. We also add the attributes `date` and `hostname.outdegree` to each node and the attributes `post.url`, `post.date`, `post.wordcount` and `post.charcount` to each edge, as discussed in Section 4.1.1. We don't store an explicit date attribute for the links, since we consider the post date to be the same as the creation date of each link.

To generate the GraphML, we use the Perl `XML::Writer` module, which merely facilitates writing well-formed XML, by supporting tag closing and attribute writing. A sample of the resulting GraphML is presented:

```
<?xml version="1.0" encoding="UTF-8"?>

<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
 http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">

 <key id="na0" for="node" attr.name="name" attr.type="string" />
 ...
 <key id="ea0" for="edge" attr.name="name" attr.type="string" />
 ...
 <graph id="G" edgedefault="directed">
  <node id="n1">
   <data key="na0">000333.blogs.sapo.pt</data>
   <data key="na1">2009-01-17 19:37:28</data>
   <data key="na2">6</data>
  </node>
  ...
  <edge source="n61632" target="n6178">
   <data key="ea0">http://arrotoazul.blogs.sapo.pt/28743.html</data>
   <data key="ea1">http://simplesmentemeu.blogs.sapo.pt/65917.html</data>
```

```
    <data key="ea2">2009-02-02 00:00:00</data>
    <data key="ea3">179</data>
    <data key="ea4">1036</data>
   </edge>
   ...
  </graph>
</graphml>
```

The `key` tag is used to define the attributes for the nodes and edges, assigning a name and a data type to each of them. Since the document only describes one graph, there is one `graph` tag, with the `edgedefault` attribute set to "directed", indicating that the source → target direction of edges is to be taken into consideration. In the sample, the node with identifier "n1" is represented by a `node` element with three sub-elements `data`, whose values represent the attributes of the respective blog. The first element, "na0", represents the node's name — "000333.blogs.sapo.pt" — of type string, as defined in the `key` with *id* "na0". The element "na1" represents the creation date of the blog — "2009-01-17 19:37:28" — and "na2" represents the number of out-links for the blog, including hosts outside of SAPO Blog's domain — a total of 6 links were extracted from this specific blog. The `edge` tag establishes a link between two previously defined nodes, `source` and `target`. Each edge has five attributes. The first one, represented by "ea0", is the URL of the link described by the edge and the remaining attributes are data about the post where the link was extracted from, including its permalink, creation date, number of words and number of characters.

The fifth and final step of the data extraction process is to load the GraphML document into R, in order to begin the blog network analysis. This is done by using the function `read.graph` from the igraph library. Together with the link indexing process, loading the GraphML file into R was one of the most time consuming tasks — it ran and completed over night.

4.4 Data Preparation

During our initial exploration of the graph, we noticed that the node "blogs.sapo.pt" was part of the graph. Since it is not a blog, but the blog service's homepage, we immediately removed this node.

The currently loaded graph, with 72,591 nodes and 459,737 edges, now accurately reflects the blogosphere sample's reality. This means that if there are N links from blog A to blog B, there will also be N edges from nodes A to B, one for each link. The same happens whenever a blog links to itself.

Rank	Blog
1	classificados.blogs.sapo.pt
2	ad-online.blogs.sapo.pt
3	adsfree.blogs.sapo.pt
4	adslist.blogs.sapo.pt
5	anuncie-gratis.blogs.sapo.pt
6	meus-anuncios.blogs.sapo.pt
7	weblist.blogs.sapo.pt
8	ichliebetokiohotel.blogs.sapo.pt
9	tokygirl.blogs.sapo.pt
10	abruxinhadiz.blogs.sapo.pt

Table 4.5: Top cited blogs for the original graph.

We extract the top cited blogs from the current blog network and verify that it results in a set populated with ad blogs (see Table 4.5), indicating that a cleaning process must occur. This probably happens either because these blogs link intensely to each other or self-cite their own posts frequently, with the objective of self-promoting, in an attempt to increase their popularity in search engines.

Our interest is to study the connections between bloggers who develop content about a subject, not bloggers who use their blogs to advertise products or services, so we generate a new version of the graph without edge multiplicity or loops (self-citations). At this point, our blog network is composed of 72,591 blogs and 57,433 links, which means that 402,304 links have been removed. Considering there are attributes stored in each edge, we cannot simply discard the extra edges, however we maintain access to all the attributes, by accessing the matching vertices and edges in the original graph. All of the previously computed attributes are not recalculated.

Our analysis focuses on the connections, so we decide to remove any unconnected nodes and nodes with either only one in-link or one out-link — this means we remove nodes with degree 0 and 1. Removing nodes with degree 1, originates new unconnected nodes, that, once again we remove. We now have a new version of the blog network — a simple graph, pruned of nodes with degree lower than 2 and clean of unconnected nodes, counting 10,937 nodes and 48,399 edges; 61,654 nodes and 9,034 edges have been removed.

Again we extract the top cited blogs and verify that the list is now clean from ad blogs (a sample in Table 4.6). Nonetheless, we question whether the pruned version of the blog network can be used to represent the blogosphere's reality — nearly 85% of the vertices and 90% of the edges were removed. Indeed, the cleaned version of the graph focuses on the connectivity and the diversity of links for each node — by ordering the nodes by in-degree, we obtain a list of the blogs that are cited by a broader set of bloggers. On the other hand, since our method wasn't built for spam detection, we cannot ensure that only

Rank	Blog
1	blogs.blogs.sapo.pt
2	31daarmada.blogs.sapo.pt
3	havidaemmarkl.blogs.sapo.pt
4	tokiohotelfans.blogs.sapo.pt
5	ichliebetokiohotel.blogs.sapo.pt
6	tokygirl.blogs.sapo.pt
7	corta-fitas.blogs.sapo.pt
8	origemdasespecies.blogs.sapo.pt
9	jugular.blogs.sapo.pt
10	tibeu.blogs.sapo.pt

Table 4.6: Top cited blogs for the pruned graph.

the ad blogs were affected by our pruning process. Considering the facts, we decide to explore and characterize both versions of the graph.

4.4.1 R Functions for Cluster Analysis

Considering that our main goal is to study the evolution of several features for progressively less cited slices, partitioning the collection into slices, ordered by in-degree, we develop a set of programatic methods, using the R scripting language, to aid us in that task.

The programmed functions can be divided into two main categories: graph slicing and cluster analysis. The graph slicing methods are responsible for the generation of several blog clusters, depending on the in-degree rank, and the cluster analysis methods allow the extraction of attributes for a group of blogs that belong to a specific cluster. The function header and description for each method can be found in Appendix A.

4.5 Blog Graph Analysis

We compare the original and pruned versions of the blog graph, considering characteristics like density and reciprocity. Table 4.7 summarizes the general properties of each version — the mean and median values are for the out-links of each blog. Analyzing the values for the original blog graph, we verify that it has a very low density, meaning that the studied sample isn't very connected, a reciprocity of 31%, indicating that approximately $1/3$ of the connections are bilateral, and a diameter of 20, symbolizing the length of the path that connects the pair of blogs that are most further apart.

We examine the original blog graph regarding link activity. Figure 4.1 depicts the daily link usage for the blog collection. On average, bloggers create 363 links per day and 10,950 links per month. Looking at the median daily value of 190, we verify that, for the majority of the days, bloggers create less links than average. There is at minimum 1 link

	Original	Pruned
Density	8.72e-5	4.05e-4
Reciprocity	0.31	0.15
Diameter	20	19
Edges	459,737	48,399
Vertices	72,591	10,937
Mean Links per Blog	6.33	4.43
Median Links per Blog	0	2

Table 4.7: Properties of the original and pruned blog graphs.

per day and 184 links per month, reaching a maximum of 9,195 links created in a single day — corresponding to the peak for July 30th 2008 — and 157,800 in a single month, for the whole set of blogs.

Figure 4.2 illustrates the growth of the sample's link set, over the years. The average monthly growth rate of the total number of links is 17.88%. After the first half of 2008, the link number shows a more accentuated growth, because of a link usage burst that occurs in the middle of 2008, during the months of June and July. This means that link usage only becomes more prominent one year after the burst in blog and post creation.

We also depict the in-degree and out-degree distributions of the original blog graph in Figures 4.3 and 4.4, respectively. It illustrates the quantity of blogs with each number of in-links and out-links. The amount of blogs with in-degree and out-degree 0 are not depicted and have the values of 52,978 and 57,559, meaning that 73% of the blogs have no in-links and 79% have no out-links. In fact, 95% of the blogs have an in-degree between 0 and 10 and 96% have an out-degree in the same interval.

4.6 Identifying Blog Communities

During an initial stage, we conduct an exploration of the blog graph supported by igraph. The goal is to determine a research path to follow, defining the criteria for blog clustering. We begin by experimenting with several community detection methods, implemented in the igraph library. Community detection is based on the premise that a community is likely to be densely interconnected, sharing content about a common interest or simply exhibiting a more general characteristic such as language. Applying the algorithms to the pruned graph for testing purposes isn't viable given the size of the structure. We extract a subgraph of the top 1,000 most cited blogs and run the detection methods for this smaller sample.

Using the `walktrap.community` function, an igraph implementation of Pons & Latapy's [PL05] idea that short random walks tend to stay, or get "trapped", in the same community, we obtain the first results for community structure (illustrated in Figure 4.5). Next, we use `fastgreedy.community`, an implementation of Clauset et al.'s [CNM04]

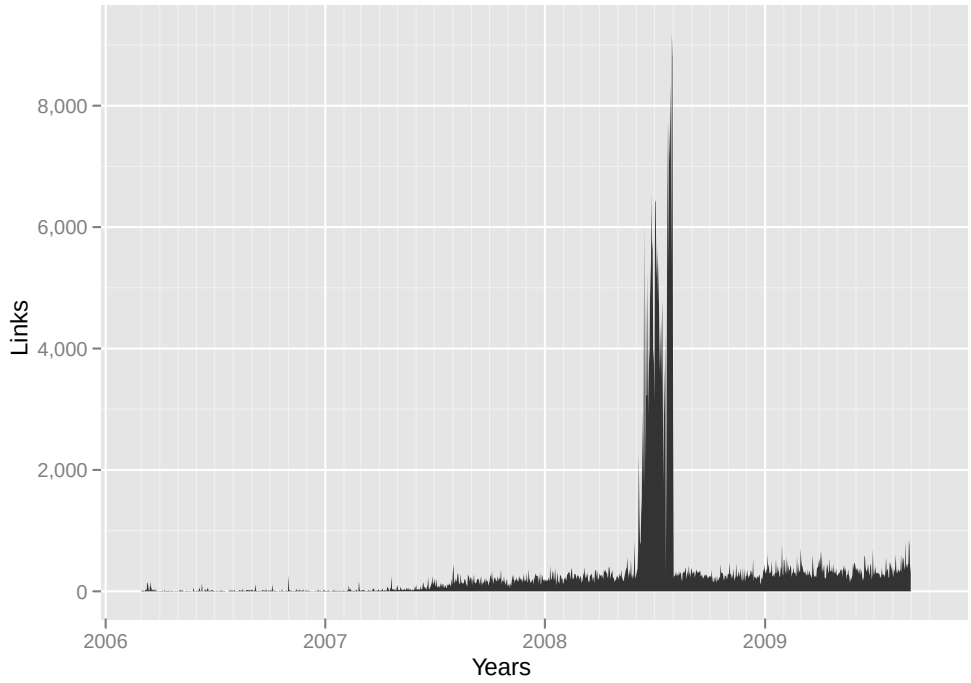

Figure 4.1: Number of links, per day, over the years.

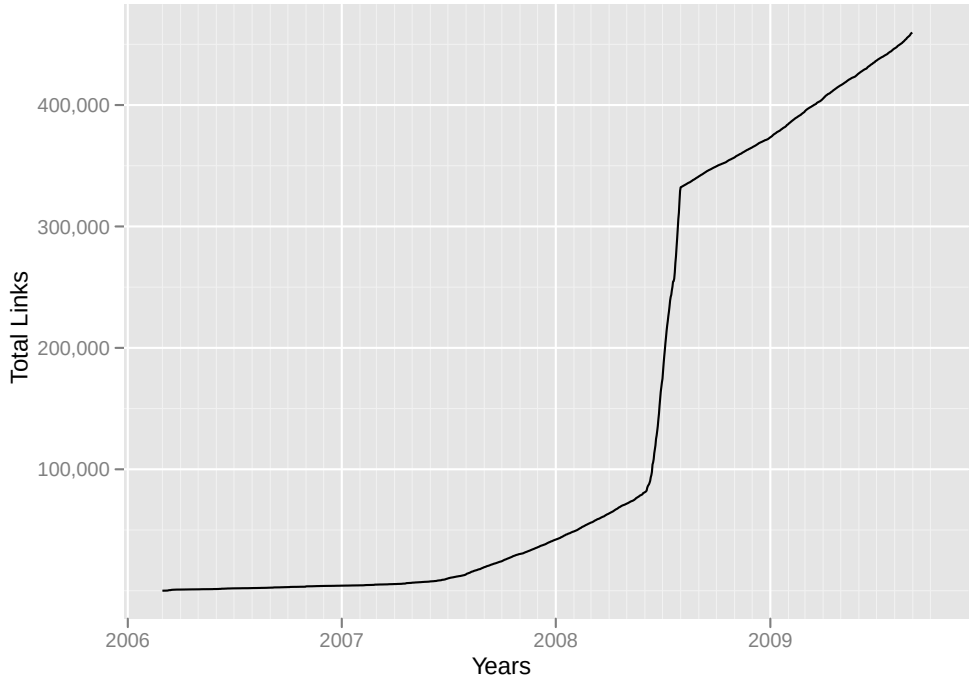

Figure 4.2: Total number of links, for a given day, over the years.

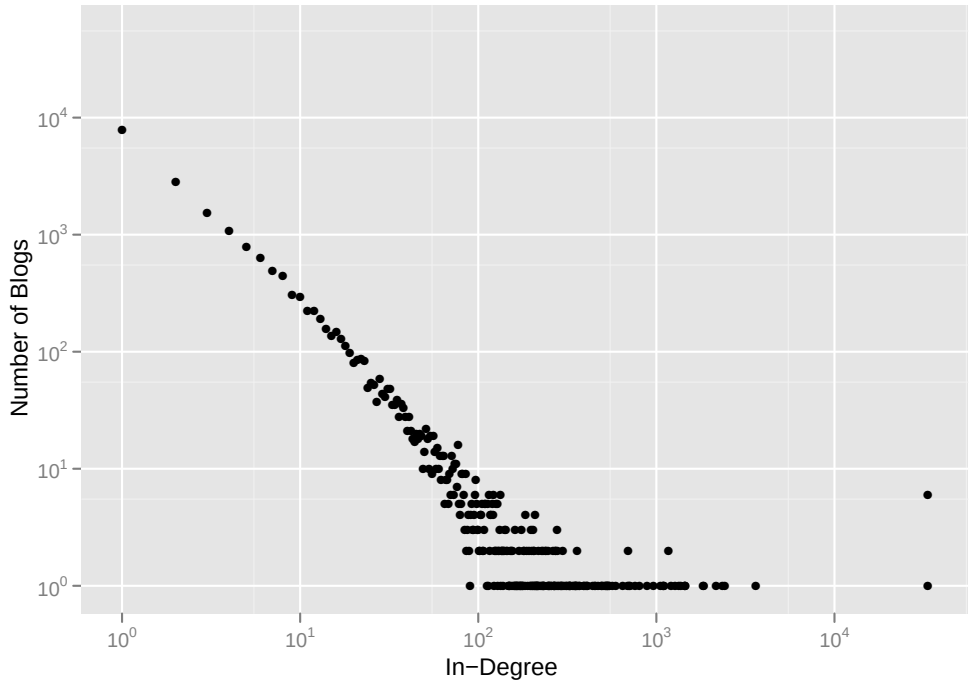

Figure 4.3: Blog graph in-degree distribution.

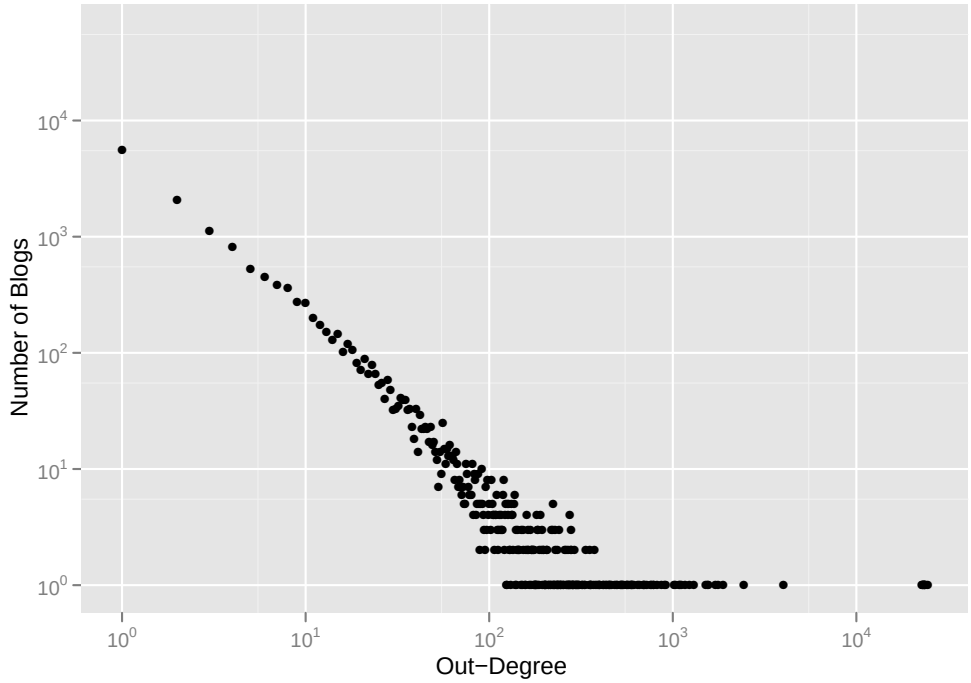

Figure 4.4: Blog graph out-degree distribution.

Algorithm	Running Time
walktrap.community	$O(n^2 \log n)$
fastgreedy.community	$O(n \log^2 n)$
leading.eigenvector.community	$O(n^2 \log n)$
edge.betweenness.community	$O(n^3)$

Table 4.8: Community detection algorithms efficiency.

low computational cost algorithm for a fast greedy direct optimization of the modularity score. This method's application is restricted to simple undirected graphs, which means we ignore the blog graph's edge directions. Also, it cannot be used on the original blog graph, since it contains multiple edges and loops. Another algorithm we experimented with was igraph's implementation of Newman's [New06] method for maximizing the modularity function in terms of the eigenspectrum of the modularity matrix. Running `leading.eigenvector.community` resulted in a community structure similar to the one obtained with the walktrap algorithm, only a little more subdivided (observe Figure 4.6).

All three methods ran instantly for our sample. Some of them were prepared to use the `weight` attribute of the edges, which could have been set to represent the number of links between two blog nodes prior to the pruning process. To facilitate the visualization of the two resulting community structures, we used Fruchterman & Reingold's [FR91] layout method, which functions as a repulsion system, with the vertices pushing each other away and the edges trying to keep them together. The result is vertices evenly distributed in the frame, minimizing edge crossings and keeping the nodes of dense subgraphs together.

We also considered using the `spinglass.community` and `edge.betweenness.community` methods. However, the spin glass function only works with undirected connected graphs, which doesn't apply to the blog graph case. On the other hand, the edge betweenness function uses a clever heuristic of progressively removing edges with a high betweenness measure, which could be applied to our blog graph, if not for the fact that it is a very demanding method regarding running time. The efficiency for the community detection methods we tested can be consulted in Table 4.8. The displayed expressions correspond to the effiency for most real world applications of the algorithms and n represents the number of vertices in the graph.

Parallel to the exploration of the community detection methods, we do different studies that eventually lead into the slice analysis of the blog graph, where we partition the blog graph into several in-degree dependent clusters. This is described in the section that follows.

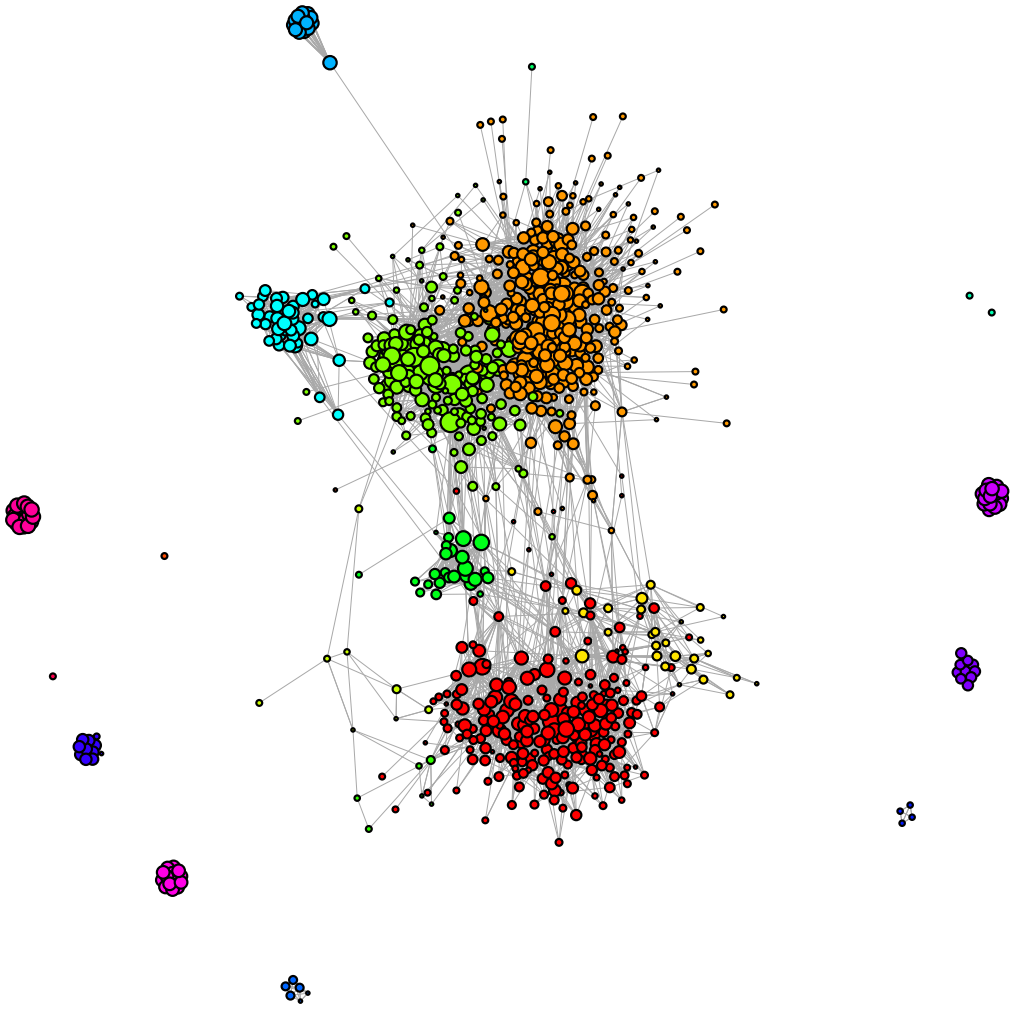

Figure 4.5: Communities on the pruned graph sample using the walktrap algorithm.

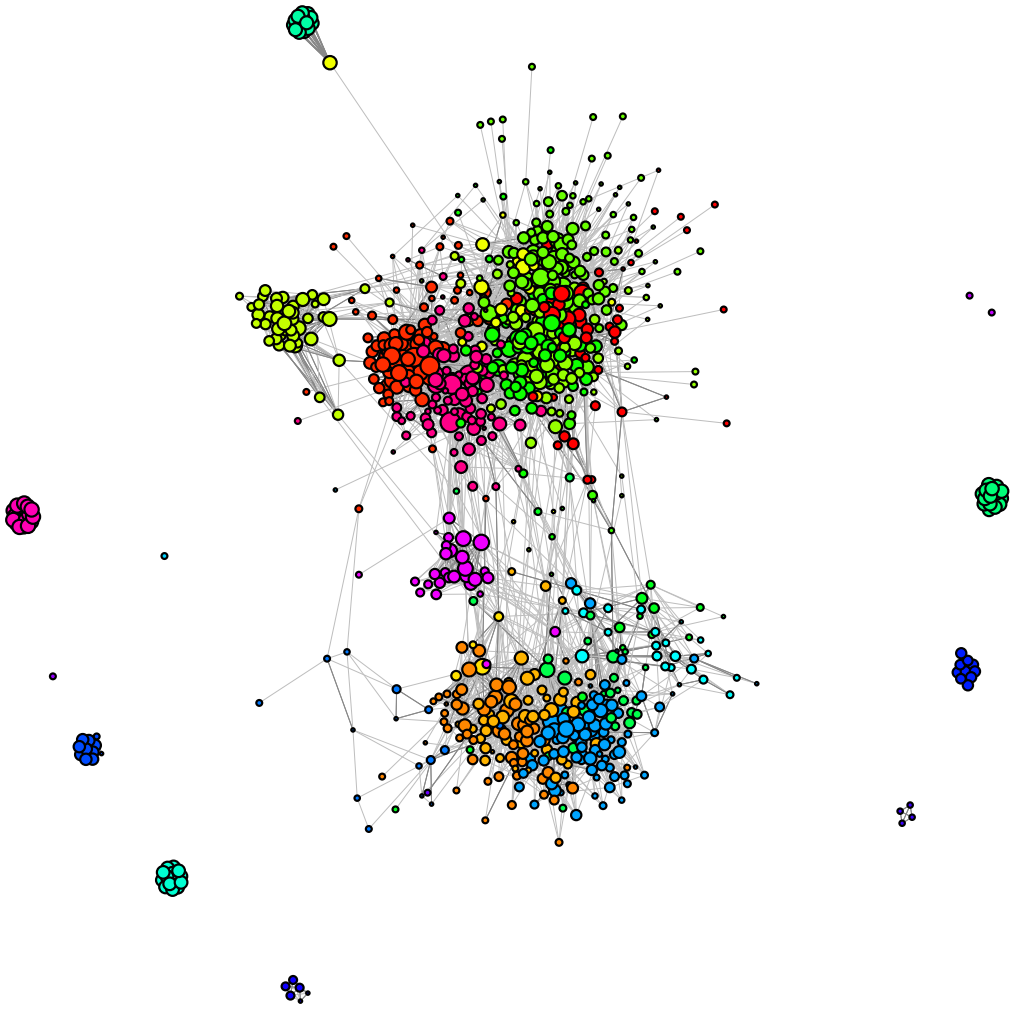

Figure 4.6: Communities on the pruned graph sample using the leading eigenvector algorithm.

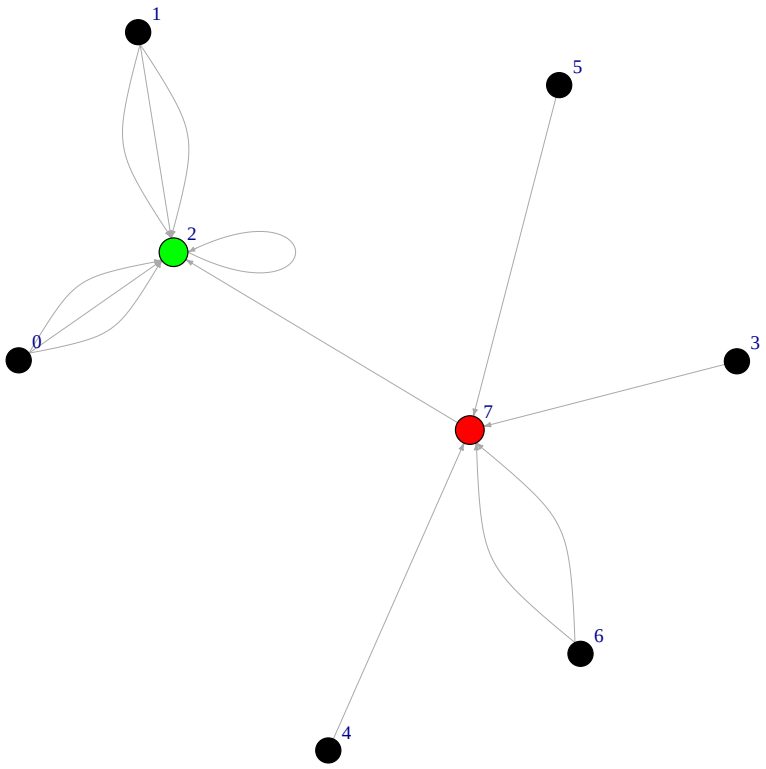

Figure 4.7: Sample of the blog graph before pruning.

4.7 Blog Cluster Analysis

The cluster analysis we are about to make will be applied to two distinct graphs: the pruned graph, that resulted from the data preparation process carried in Section 4.4 and aims at representing the blog network with a focus in link source diversity, and the original graph, crudely illustrating the blog network's reality, including link multiplicity and self-citations. Figures 4.7 and 4.8 illustrate the differences between both graphs, regarding the edge simplification process. As we can see in Figure 4.7, the graph structure includes edge multiplicity and loops. In this example, node 2 represents the most popular blog, with an in-degree of 8, against node 7's in-degree of 5. In Figure 4.8, we remove multiple edges and loops, which results in a different scenario. Now, node 7 appears as the most popular blog, with an in-degree of 4, against node 2's in-degree of 3. In reality, node 2 represents the most cited blog, linked by nodes 0 and 1, and, even though node 7 comes second, it is cited by a wider set of blogs, having in-links from nodes 3, 4, 5 and 6. So, there are two scenarios: a scenario of quantity and a scenario of diversity, which are both studied, based on the original graph and the pruned graph.

We partition the pruned blog graph — with 10,937 nodes and 48,399 edges — and the original blog graph — with 72,591 nodes and 459,737 edges — into slices of 1,000 blogs each, ordered by decreasing in-degree, and compute the mean and median values

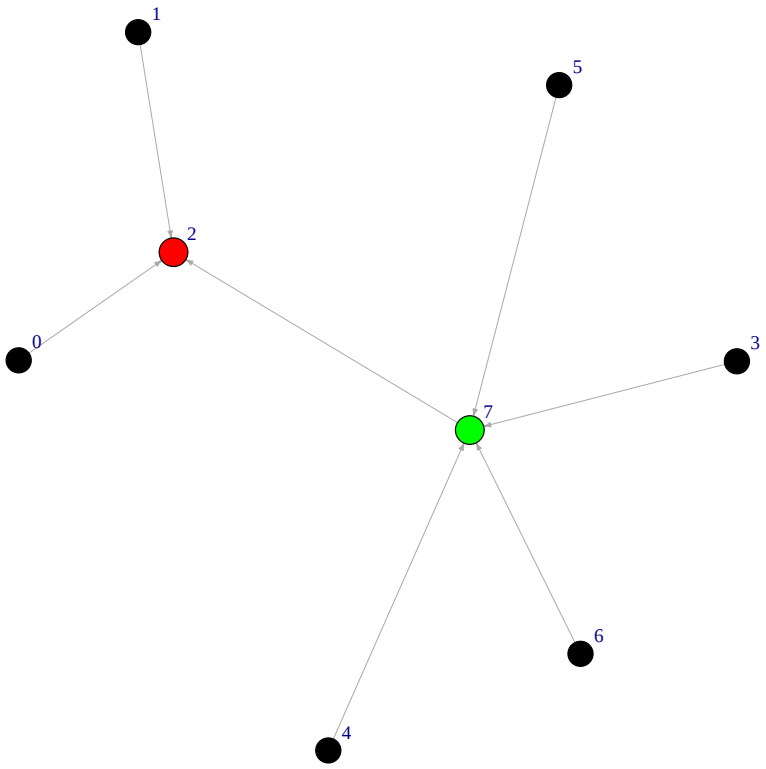

Figure 4.8: Sample of the blog graph after pruning.

for blog features in each slice. So, for example, when we look at Figure 4.9, the slice of order 0 represents the group of blogs ranked from 1 to 1,000, and in the y-axis is the mean and median values for the number of words of the posts in this group of blogs (829 and 706, respectively); the slice of order 4 represents the group of blogs ranked from 4,001 to 5,000, having the values of 556 and 406 for the mean and median, and so forth.

Pruned Graph

At this point, we are working on the blog network with 10,937 nodes, that we partitioned into slices of 1,000 nodes — this represents 9.14% of the nodes in the pruned graph and 1.38% of the nodes in the original blog graph. For each slice of the pruned graph, we associate a membership value to the corresponding nodes on the original graph, so we can access the attribute data we lost after the pruning process.

Table 4.9 compares the characteristics for the whole blog network with the characteristics for the subgraph of the most cited slice (slice 0); the mean and median values for the slice's links per blog are directly extracted from the original version of the graph, using the membership attribute we set, therefore representing the real number of links per blog in slice 0. We verify that the subgraph that illustrates the most cited slice is a lot more

	Original Graph	Most Cited Slice Subgraph
Density	8.72e-5	5.58e-2
Reciprocity	0.31	0.24
Diameter	20	12
Edges	459,737	55,702
Vertices	72,591	1,000
Mean Links per Blog	6.33	77.64
Median Links per Blog	0	24

Table 4.9: Properties of the original graph and the most cited slice subgraph.

dense than the original graph, but less reciprocal, and has at least ten times more out-links per blog than the original.

One of the interesting results we found is related to the number of words per post (Figure 4.9). Blogs that are intensely cited gather the set of posts with the highest number of words — an average value of 829 and a median value of 706 —, decreasing consistently until slice 6 — with an average value of 532 and a median value of 360 — and increasing again until slice 9, the last positioned slice in the most cited blogs — with an average value of 762 and a median value of 574 words. This means that blogs that are more cited, by a wider set of blogs, also write longer posts and, the less cited blogs have the smaller number of words; however, there are also blogs that have almost no links pointing to them that write a lot — this is an unexpected behavior, however the edges of the pruned graph solely represent a connection between two blogs, without multiplicity, and therefore, in this case, popularity is synonym of a high variety of link sources, as opposed to a gross amount of link sources.

Looking at Figures 4.10 and 4.11, we find an interesting behavior regarding blog ages and newly created posts, for each slice, as the slices become less cited. Slices that are more linked gather a set of younger blogs and, as slices become less linked, the blog ages increase. The opposite behavior happens regarding posts. Blogs that are more cited also write the largest number of posts — reaching an average value of 5,691 monthly new posts for slice 1. As for the least cited blogs, we verify that the monthly number of new posts is constantly around 500. The conclusion we reach is that relevant bloggers, that get their posts cited throughout the blogosphere, represent a set of younger blogs and generate more new content than bloggers whose posts aren't widely cited, in spite of representing a set of older blogs.

Figure 4.12 depicts the evolution of the number of links per post, for progressively less linked slices. We verify that blogs with a high number of citations also link more to other blogs. The number of links per post is, however, smaller than 1 for every slice, meaning that, when we look individually at the post contents, we generally don't find a high number of links.

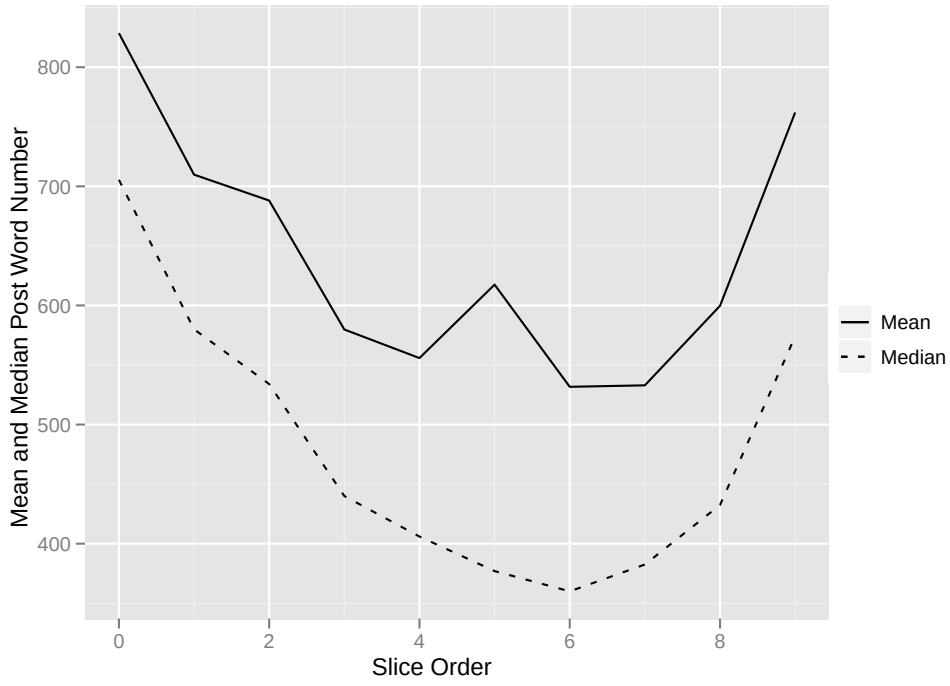

Figure 4.9: Number of words per post, for the pruned blog graph.

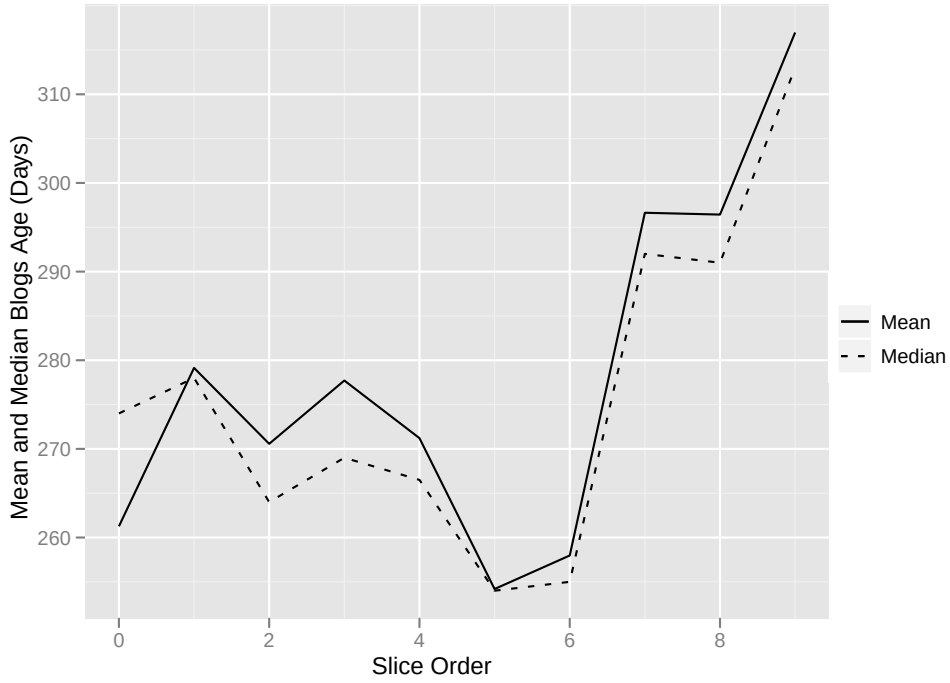

Figure 4.10: Blogs age, in days, for the pruned blog graph.

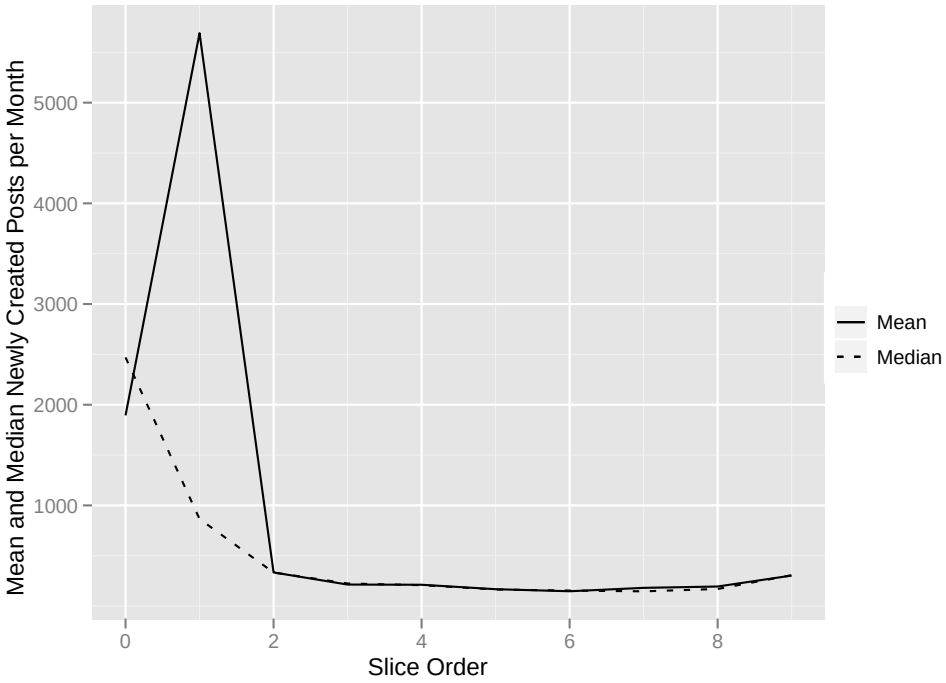

Figure 4.11: Newly created posts, per month, for the pruned blog graph.

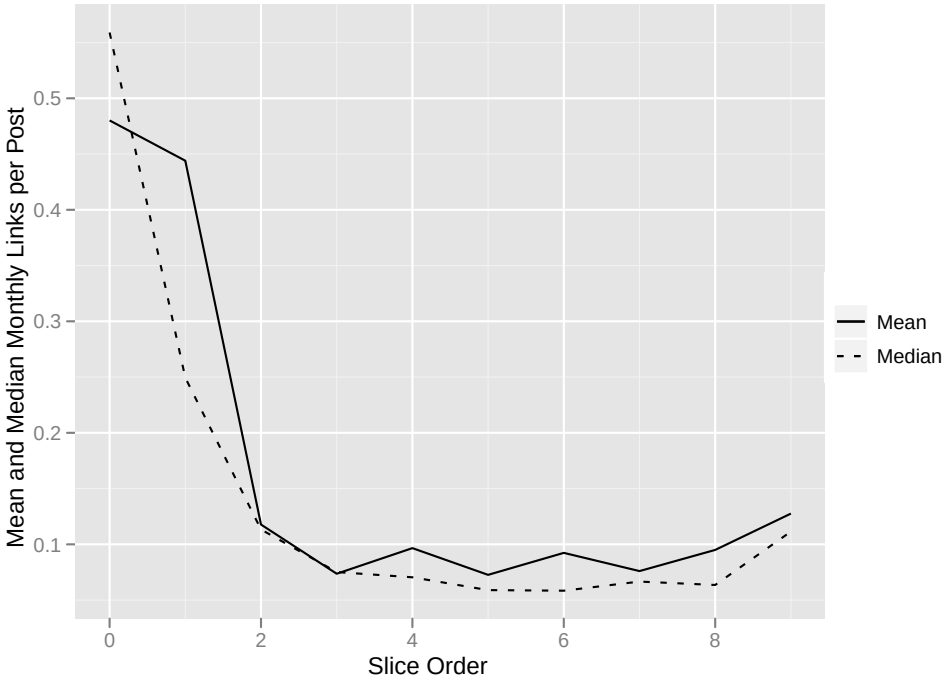

Figure 4.12: Monthly number of links per post, for the pruned blog graph.

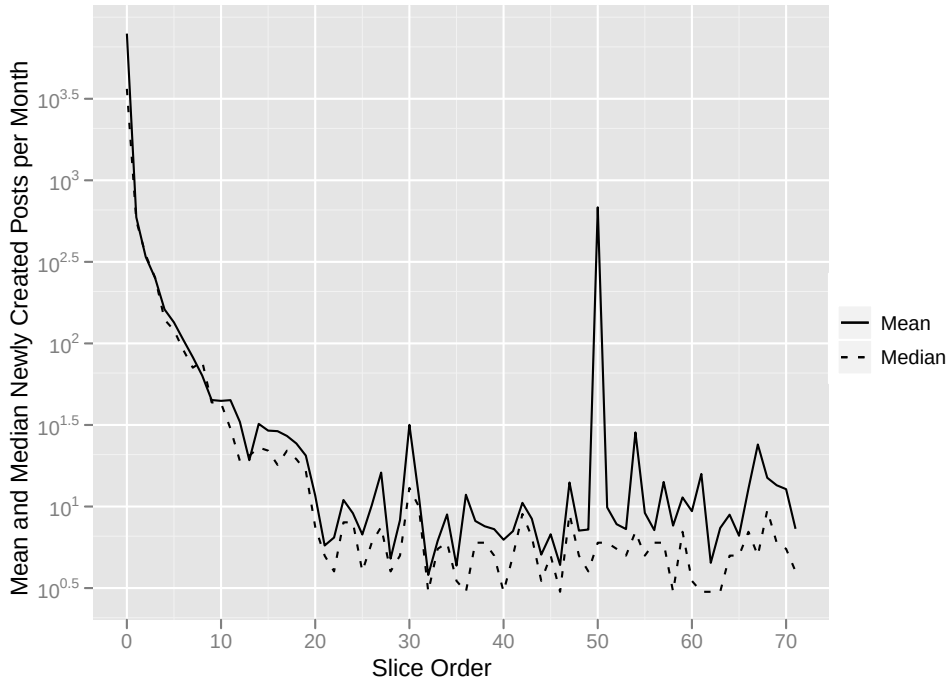

Figure 4.13: Newly created posts, per month, for the original blog graph.

Original Graph

Similarly to what we did in the previous section, we partition the original graph into 72 slices, ordered by decreasing in-degree, consisting of 1,000 blogs each. We repeat the same studies for this uncleaned version of the graph, observing analogous behaviors regarding the newly created posts per month (Figure 4.13) and the monthly number of links per post (Figure 4.14). However, the evolution of number of words per post (Figure 4.15) and the evolution of blog ages (Figure 4.16), for progressively less cited blogs, show a different behavior from the pruned graph.

On the pruned version of the graph, intensely cited blogs have posts with a high number of words, decreasing as the slices are less cited and then increasing again for the least cited blogs. On the original graph, the number of words decrease progressively, as blogs become less cited, showing a clear relation between post length and popularity — the larger the content produced, the more citations a blog has.

Concerning blog ages, on the pruned version of the graph, highly cited blogs are younger than blogs with few citations and, on the original graph, intensely cited blogs are older than blogs with less citations. This means that younger blogs are cited more widely around the portuguese blogosphere, but older blogs gather the highest number of citations, even if not from such a variety of blogs.

There is an evident contrast between both versions, since the original graph represents the reality of the blogosphere's sample and the pruned graph represents a selected reality,

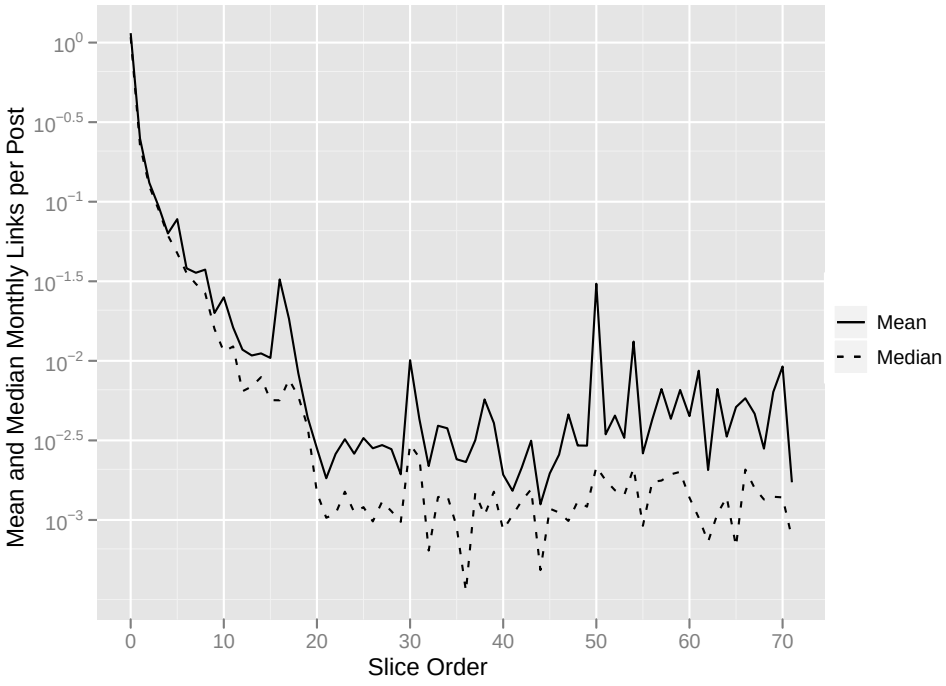

Figure 4.14: Monthly number of links per post, for the original blog graph.

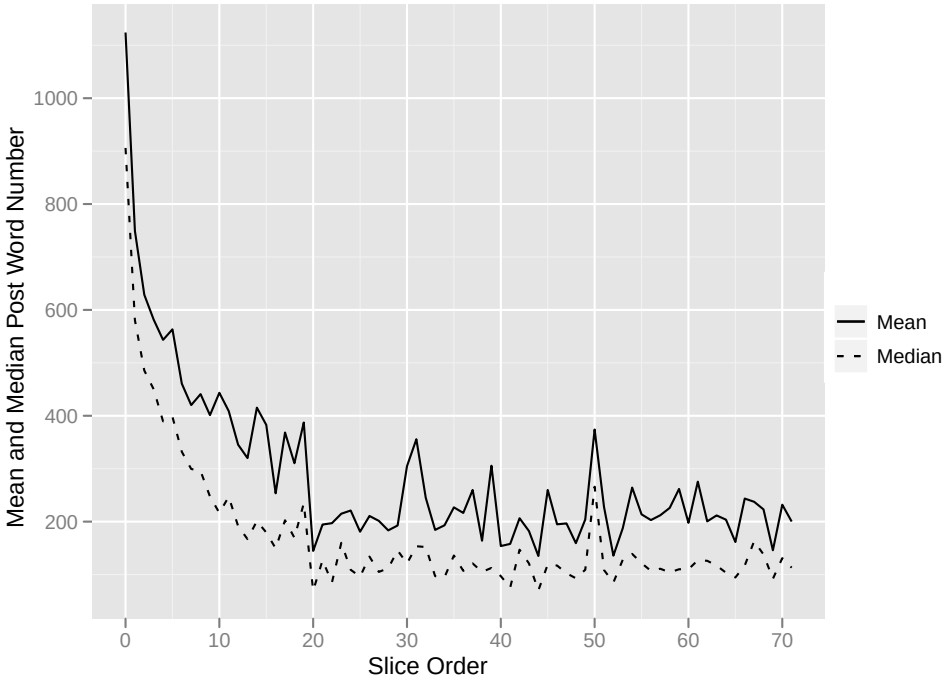

Figure 4.15: Number of words per post, for the original blog graph.

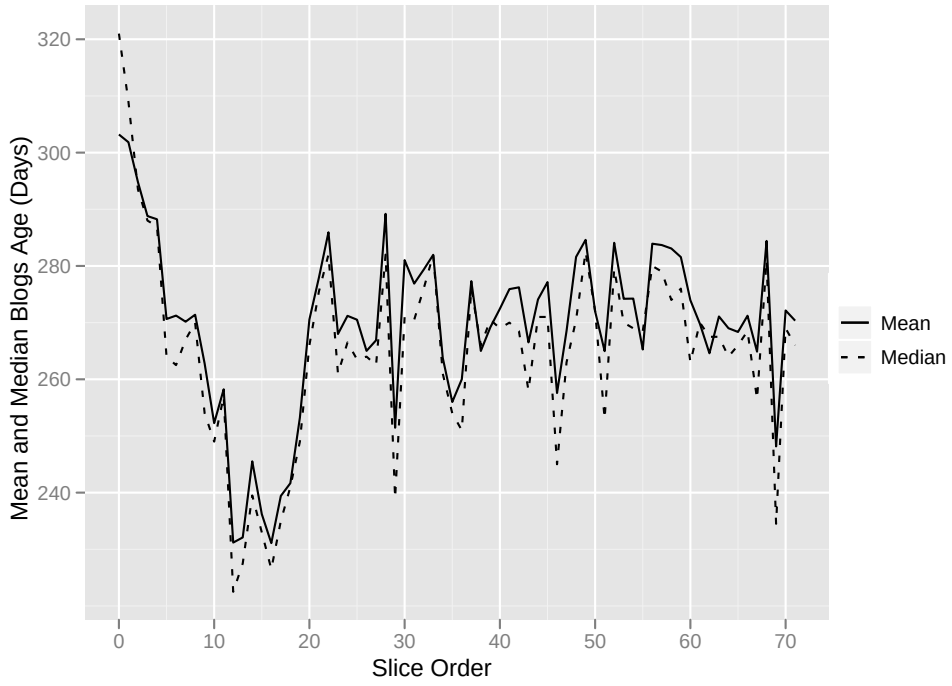

Figure 4.16: Blogs age, in days, for the original blog graph.

where the focus is the variety of links, not the quantity, allowing us to attribute a degree of importance to each blog that isn't merely dependent on the number of in-links, but also on the diversity of the sources of those links.

4.8 Summary

In this chapter, we described the blog graph structure and explained how it was generated by extracting explicit and embedded links from post content. We have studied a large sample of the portuguese blogosphere, by partitioning it, using the number of citations as criteria, and have examined several behaviors for those parts, in order to understand whether the most popular blogs have different characteristics from the remaining blogs. By analyzing the mean and median values for the monthly newly created posts and the number of links and words per post, we identified a pattern where change consistently happens as we move from the highly cited blogs to the less cited. The characteristics of the slices in what concerns posting activity, linking pattern and number of words varies strongly. There are evident differences between the highly cited slices and the remaining ones, illustrating the contrast between popular and less popular blogs.

Chapter 5

Conclusions

We surveyed the methodologies for the analysis of the blogosphere, covering several blog characterization approaches, link analysis research and community detection heuristics. We used a recent snapshot of SAPO's blog collection, previously studied by Pinto [Pin08], regarding trend detection, by Branco [Bra08], regarding blog ranking, and by Couto [Cou09], regarding blogging activity. By applying some of the reviewed methodologies, we briefly characterized the portuguese blogosphere. Based on the results of the characterization, we identified an increase in blogging activity after the second half of 2007 and validated the data set, removing the month of September 2009 for having almost 50% bogus blogs.

In order to analyze the differences between the highly cited and the less cited blogs, we represented the link structure of the blogosphere by means of a blog graph, studying the evolution of several features depending on the number of in-links. This allowed for a better understanding of the behaviors of popular blogs when compared to less popular blogs, making it clear that popular blogs create a higher number of posts per month, with more links per post and a higher number of words.

5.1 Main Contributions

The conducted study was the first centered on the link structure of the portuguese blogosphere and studied the differences between popular and less popular blogs. When analyzing the link ecosystem, we looked at the features in the context of the blog network. Deriving from the concept of blog A-List — the set of most influential blogs — and assuming a popularity criterium based on the number of citations, we partitioned a large sample of the portuguese blogosphere into several ordered slices — groups of 1,000 blogs, ranked by in-degree — and analyzed the mean and median values of the

characteristics of each slice, observing the evolution for progressively less cited slices. The model we used can be tuned by defining a different size for the slices, changing the granularity, or by using a different heuristic, for instance PageRank or HITS, or even the h-index application proposed by Branco [Bra08], to determine blog relevance. Using this method, we found evident differences between the highly cited slices and the remaining ones, illustrating the contrast between popular and less popular blogs.

Introducing a characterization based on the link structure allowed for a cluster-centered study. It brought two types of analysis together, applying blog characterization to specific groups of blogs, extracted through a link analysis process. This approach not only allowed for the clustering and characterization of blogs, but also offered the chance to compare the behavior of different groups of blogs. We verified that highly cited or popular blogs do indeed behave differently when compared to the remaining blogs.

Popular blogs have shown a higher activity regarding post creation, with the most cited slice creating an average of 7,934 posts per month. As a general rule, other highly cited slices have an average post creation value superior to 100, reaching 594 for the second most cited slice. Groups of blogs that are less cited usually create between 5 and 30 posts a month. Note that, since these values represent groups of 1,000 blogs, they might be a bit dim regarding the characterization of posting activity.

Highly cited blogs also have the greatest number of out-links per post. However, posts usually don't gather a great number of links. The most cited slice has an average number of out-links per post of 1.15. This value is less than 0.25 for the remaining slices, reaching values as low as 1 link in 1,000 posts.

Regarding the length of the post content, popular blogs gather the longest texts, with an average of 1,124 words for the most cited slice. For the remaining slices, the average number of words per post ranges from 135 to 749, showing a constant but not very accentuated decrease, as we go from popular to less popular blogs.

Even though not the most important feature, blog age is also a factor that influences popularity. Older blogs, with an average age of 303 days, are also the most cited. However, in general, slices of order superior to 20 tend to gather a set of blogs with ages around 270 days, differing only a month from the most cited slice.

5.2 Future Work

The analysis of the portuguese blogosphere's link ecosystem leaves an open door for future studies. Based on the methods used by Kumar et al. [KNRT05] for the analysis of the bursty evolution of the blogosphere, we could study the evolution of blog popularity, in order to understand what influences a blog to become a reference in the blogosphere. Several prefix graphs, for different time frames, could be extracted from the blog graph

and partitioned using a ranking heuristic. Having different blog members for the corresponding slices of each prefix graph, an analysis of the rank evolution of the most popular blogs could be made, accompanied by a study of their features evolution.

Another lead for future research is the detection and characterization of portuguese blog communities, eventually applying the algorithms implemented in igraph to detected densely connected subgraphs. We could also study the link polarity for the various communities, identifying whether a community is densely connected because it loves a certain subject or because it negatively criticizes another, and perhaps identify a group of central blogs as the target of discussion.

Referências

[BEL05] U. Brandes, M. Eiglsperger, and J. Lerner. GraphML Primer. In *Graph Drawing*, 2005.

[BKM+00] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the Web. *Computer Networks*, 33(1-6):309 – 320, 2000.

[Bra08] José Mário Branco. Aplicação do h-index em blogues. Master's thesis, Faculdade de Engenharia da Universidade do Porto, July 2008.

[CK06] E. Cohen and B. Krishnamurthy. A short walk in the Blogistan. *Computer Networks*, 50(5):615–630, 2006.

[CN06] Gabor Csárdi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[CNM04] A. Clauset, MEJ Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):66111, 2004.

[Cou09] Orlando Telmo Couto. Characterizing the Portuguese Blogosphere. Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2009.

[CPH09] Meeyoung Cha, Juan Antonio Navarro Pérez, and Hamed Haddadi. Flash Floods and Ripples: The Spread of Media Content through the Blogosphere. In *International AAAI Conference on Weblogs and Social Media (ICWSM'09)*. AAAI, 2009.

[DB00] A. Descartes and T. Bunce. *Programming the Perl DBI*. O'Reilly & Associates, Inc. Sebastopol, CA, USA, 2000.

[FR91] T.M.J. Fruchterman and E.M. Reingold. Graph drawing by force-directed placement. *Software- Practice and Experience*, 21(11):1129–1164, 1991.

[Fre79] L.C. Freeman. Centrality in Social Networks Conceptual Clarification. *Social networks*, 1(3):215–239, 1979.

[HHB+03] Mark S. Handcock, David R. Hunter, Carter T. Butts, Steven M. Goodreau, and Martina Morris. *statnet: Software tools for the Statistical Modeling of Network Data*. Seattle, WA, 2003. Version 2.0.

[HSKW07] S.C. Herring, L.A. Scheidt, I. Kouper, and E. Wright. A Longitudinal Content Analysis of Weblogs: 2003-2004. *Blogging, Citizenship and the Future of Media. London: Routledge*, 2007.

[Kle02] Jon Kleinberg. Bursty and Hierarchical Structure in Streams. *Data Mining and Knowledge Discovery*, 7(4):373 – 397, 2002.

[KNRT05] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the Bursty Evolution of Blogspace. *World Wide Web*, 8(2):159–178, 2005.

[KRRT99] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the Web for emerging cyber-communities. *Computer Networks*, 31(11-16):1481 – 1493, 1999.

[New06] MEJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):36104, 2006.

[OBS99] M.A. Olson, K. Bostic, and M. Seltzer. Berkeley DB. In *Proceedings of the FREENIX Track: 1999 USENIX Annual Technical Conference*, pages 183–192, 1999.

[OFS+05] J. O'Madadhain, D. Fisher, P. Smyth, S. White, and Y.B. Boey. Analysis and Visualization of Network Data using JUNG. *Journal of Statistical Software*, 10:1–35, 2005.

[Pin08] José Pedro Pinto. Detection Methods for Blog Trends. Master's thesis, Faculdade de Engenharia da Universidade do Porto, July 2008.

[PL05] P. Pons and M. Latapy. Computing communities in large networks using random walks. *Lecture notes in computer science*, 3733:284, 2005.

[QRSA07] Vahed Qazvinian, Abtin Rassolian, Mohammad Shafiei, and Jafar Adibi. A Large-Scale Study on Persian Weblogs. *Proceedings of 12th International Joint Conference on Artificial Intelligence, Workshop of TextLink2007*, 2007.

[R D09] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009. ISBN 3-900051-07-0.

[RB06] J. Reichardt and S. Bornholdt. Statistical Mechanics of Community Detection. *Physical Review E*, 74(1):16110, 2006.

[SAP09] SAPO. Blogs do SAPO. http://blogs.sapo.pt, Accessed in October 2009.

[Wic09] Hadley Wickham. *ggplot2: elegant graphics for data analysis*. Springer New York, 2009. ISBN 978-0-387-98140-6.

Appendix A

R Functions for Cluster Analysis

A.1 Graph Slicing

- **graph.degree.cut**(*graph*, *from*=1, *to*=1000, *mode*="in") — Returns the subgraph for the nodes ranked between `from` and `to`, in the in-degree top. The argument `mode` can be adjusted to use the out-degree top ("out") or the sum of both ("total").
- **graph.slice**(*graph*, *slice.size*=1000) — Returns a list of graphs, resulting from partitioning the `graph` into slices of `slice.size` nodes, considering the in-degree rank.

A.2 Cluster Analysis

All of the following methods take a `graph` argument, that represents a cluster subgraph, and a `cluster` argument, either representing a set of indices or a set of boolean values that determine whether a node is to be considered or not, when accessing the original graph for attributes.

- **cluster.general.properties**(*graph*) — Returns a `data.frame` with the density, the reciprocity, the diameter, the number of edges and the number of vertices of the graph.
- **cluster.blogs.over.time**(*graph*) — Returns a `data.frame` with a column `Date`, in the format YYYY-MM-DD, for intervals of a day, a column `Blogs` with the number of blogs created for the given date, and a column `Age` with the pre-calculated number of days since the blog was created.
- **cluster.blogs.over.time.months**(*graph*) — Returns a `data.frame` with a column `Month`, in the format YYYY-MM, and a column `Blogs` with the number of blogs created for the given month.
- **cluster.posts.over.time**(*graph*, *cluster*) — Returns a `data.frame` with a column `Date`, in the format YYYY-MM-DD, for intervals of a day, and a column `Posts` with the number of posts created for the given date.
- **cluster.posts.over.time.months**(*graph*, *cluster*) — Returns a `data.frame` with a column `Month`, in the format YYYY-MM, and a column `Posts` with the number of posts created for the given month.

- **cluster.posts.over.time.hours**(*graph*, *cluster*) — Returns a `data.frame` with a column `Hours`, in the format H, and a column `Posts` with the number of posts created for the given hour.
- **cluster.links.per.blog.over.time.months**(*graph*, *cluster*) — Returns a `data.frame` with a column `Months`, in the format YYYY-MM, and a column `LinksPerBlog` with the number of links per blog for the given month.
- **cluster.links.per.post.over.time.months**(*graph*, *cluster*) — Returns a `data.frame` with a column `Months`, in the format YYYY-MM, and a column `LinksPerPost` with the number of links per post for the given month.
- **cluster.post.size.distribution**(*graph*, *cluster*) — Returns a `data.frame` with a column `Size` and a column `Posts`, representing the number of posts with the given size (number of characters).
- **cluster.word.count.distribution**(*graph*, *cluster*) — Returns a `data.frame` with a column `Words` and a column `Posts`, representing the number of posts with the given number of words.
- **cluster.external.link.distribution**(*graph*, *cluster*) — Returns a `data.frame` with a column `Links` and a column `Blogs`, representing the number of blogs with the given number of links, pointing to the outside of the cluster.
- **cluster.internal.link.distribution**(*graph*, *cluster*) — Returns a `data.frame` with a column `Links` and a column `Blogs`, representing the number of blogs with the given number of links, pointing to the inside of the cluster.