LOCAL IDENTIFICATION OF WEB GRAPH COMMUNITIES

RADBOUD UNIVERSITY NIJMEGEN
MASTER THESIS INFORMATION SCIENCE
JUNE 2007

AUTHOR:	MAX HINNE
THESIS NUMBER:	54 IK
SUPERVISOR:	PROF. DR. IR. TH.P. VAN DER WEIDE
SECOND CORRECTOR:	PROF. DR. H. P. BARENDREGT

LOCAL IDENTIFICATION OF WEB GRAPH COMMUNITIES

Max Hinne
Institute for Computing and Information Science, Radboud University, Toernooiveld 1, Nijmegen, Netherlands
mhinne@sci.ru.nl

Keywords: Web graph, local modularity, community identification algorithm

Abstract: In order to use knowledge of the Web graph in Information Retrieval, we provide a consistent overview, aiming firstly at global aspects of the graph such as degree distribution, and then proceed by examining local aspects of the graph: community identification. We discuss several community models and we implement a community identification algorithm that operates without a priori knowledge of the graph. To elaborate on the algorithm we introduce a notational framework for graph clusters. We run the algorithm on the Dutch domain (.NL) and from the results of this experiment we conclude that the Web consists of several clusters that are mutually connected through a core of hubs. In addition we evaluate the clustering quality of the algorithm, which provides a reputable basis for local community identification.

1 INTRODUCTION

In the past decade, the World Wide Web (WWW) has grown significantly. A recent study estimates the total number of websites at 11,5 billion (Gulli & Signorini, 2005) and this number is still increasing. Since the WWW has become such an important asset of our daily life, the Web has gained interest in the scientific community, which resulted in various studies concerning a wide variety of topics. One of these areas of research examines only structural properties of the WWW – the Web is seen as a *graph*, the contents of websites are mostly ignored. Using this approach one is able to analyse the evolution of structures and phenomena on the Web (Broder, et al., 2000). An interesting example of such a phenomenon is the scale-free degree distribution on the Web (Barabási, Albert & Jeong, 2000), which will be explained in detail in the following paragraphs. In this paper we continue the ongoing process of providing a model that accurately describes the Web. To do so we firstly provide a brief primer on basic graph theoretic concepts in paragraph 2. Thereafter the distinction between global and local graph characteristics is made. In paragraph 3 we discuss the current state of affairs concerning the Web graph globally. The attention will be directed to the scale-free degree distribution that has received so much attention lately and to connected components. After this global view we proceed with local aspects of the Web graph in paragraph 4 where the emerging of *graph communities* and *modularity* is discussed. Several community models will be reviewed. The question we consider here is: 'How is the Web graph organised on a local scale?'. As a means of answering this we implement a community identification algorithm based on *local modularity*, which can be seen as a measure of disconnectedness for clusters of the graph. To explain the algorithm we introduce a framework for describing local graph phenomena.

The results of the experiment with our community identification algorithm are provided in paragraph 5. Paragraph 6 concludes the paper and provides suggestions for further research.

2 PRIMER ON GRAPH THEORY

Before we proceed with modelling the Web graph, we cover some of the basics of graph theory.

We abstract from the content of websites and regard only their connectivity. An interesting side effect of this approach is that the Web can be compared to totally different networks – like the metabolic system. We define the Web graph as an ordered pair $G = (V, A)$. The set V contains the websites, which we will refer to as *nodes* or *vertices* $v \in V$ and the set A contains the *directed* hyperlinks, ordered pairs $(i, j) \in A \subseteq V^2$, which we will refer to as *arcs*.

A can be viewed as a binary relation over V. The notation $A(x, y)$ means that an arc from x to y exists. In a directed graph, this relation is asymmetric, so $A(x, y) \not\leftrightarrow A(y, x)$. In addition the predicate $A(x, Y)$ is used, indicating the vertices in the set $Y \subseteq V$ that x points to:

$$A(x, Y) \triangleq \{y \in Y | A(x, y)\}$$

The symbol $\triangleq$ is used as 'is defined as'. Secondly we introduce $A(X, y)$, the nodes in the set $X \subseteq V$ that point to y:

$$A(X, y) \triangleq \{x \in X | A(x, y)\}$$

Of special interested is the set of all nodes that connect to a specific vertex; its *neighbourhood*. In a directed graph two types of neighbourhoods exist: the set that points to a node and the set that are pointed to by a node:

$$A_{in}(x) \triangleq A(V, x) \qquad \text{and} \qquad A_{out}(x) \triangleq A(x, V)$$

The complete neighbourhood of x is then simply

$$A(x) = A_{in} \cup A_{out}$$

Later on we will also use sets of arcs instead of nodes. More specifically, we want to know all arcs from X to Y:

$$arcs(X, Y) \triangleq \{(x, y) \in A | x \in X \wedge y \in Y\}$$

It is sometimes desirable to view a directed graph as undirected, i.e. we make no distinction between a source and a destination vertex: $G = (V, E)$. The arcs in an undirected graph are *edges*. For an undirected graph the above predicates are defined analogously: The notation $E(x, y)$ indicates that x and y are connected. This relation is symmetric, i.e. $E(x, y) \leftrightarrow E(y, x)$. The predicate $E(x, Y)$ provides all the nodes in $Y \subseteq V$ that are connected to x:

$$E(x, Y) \triangleq \{y \in Y | E(x, y)\}$$

The neighbourhood of x in an undirected graph is given by

$$E(x) \triangleq E(x, V)$$

We also define a predicate for all edges between two sets:

$$edges(X, Y) \triangleq \{(x, y) \in E | x \in X \wedge y \in Y\}$$

In addition there is the notion of a *path* between two vertices. There exists a path in a graph between two vertices x and y if they are neighbours in one or more steps:

$$path(x, y) \triangleq A(x, y) \vee \exists_z [A(x, z) \wedge path(z, y)]$$

And in an undirected graph there can exist a chain of edges between two nodes x and y:

$$chain(x, y) \triangleq E(x, y) \vee \exists_z [E(x, z) \wedge chain(z, y)]$$

These predicates will play an important role in our community identification algorithm, to which we will return later.

3 GLOBAL STRUCTURE OF THE WEB GRAPH

When trying to find the connectivity structure of a large graph, in particular the WWW, we use a process called *crawling*. The crawler starts at a given seed vertex $v_0 \in V$ (or a seed set of vertices) and proceeds to add all neighbours $A_{out}(v_0)$ to its crawl frontier, or $E(v_0)$ in an undirected graph. This is then repeated in a breadth-first search process for each vertex in the frontier, adding all new vertices and arcs to the stored graph, until no new vertices to explore remain. Crawlers are often used by search engines, which in addition to storing the graph structure, index the documents based on their contents and structure.

By using such a crawl, Broder *et al.* (2000) have observed that if the Web is seen as undirected, about 10% of the vertices have no chain to any of the nodes in the other 90%, which form a connected component and as a consequence, not all vertices can be reached from the chosen seed of a crawl. It gets more interesting when directionality is taken into account. One can distinguish four different graph connectivity subsets: A strongly connected component (SCC), which is defined as a subset S of a directed graph G, such that any node in S has a path to all other nodes in S and S is not a subset of any larger such set:

$$SCC(S) \triangleq \left\{x \in V \,\middle|\, \forall_y [y \in S \leftrightarrow path(x, y)]\right\}$$

The SCC forms the central CORE of the web graph. The next two parts are referred to as IN and OUT, which respectively label the subset of nodes that have a path to a node in the central core, but cannot be reached from it, and the subset that has a path from a node in the central core, but cannot return to it:

$$IN(I,S) \triangleq \{x \in V - SCC(S) \mid \forall_y [y \in I \leftrightarrow path(y,x)]\}$$

And

$$OUT(O,S) \triangleq \{x \in V - SCC(S) \mid \forall_y [y \in O \leftrightarrow path(x,y)]\}$$

Finally there is the collection of sub graphs that cannot reach, and cannot be reached from, the SCC, but that are connected to either the IN or OUT component. These sets are called the TENDRILS of the World Wide Web. The CORE is the largest component with roughly 27% of the vertices, followed by the IN and OUT components that both consist of 21% of the graph. The TENDRILS make up for 22%, which means that 9% of the web graph is disconnected from the rest of the graph (which could also be considered as a fifth component). In Figure 1 the structure of the WWW, which Broder et al. refer to as the 'bow-tie', is visualized.

Donato, Leonardi, Millozzi & Tsaparas (2005) refined the bow-tie structure and introduced the so-called daisy model. In this model the IN and OUT components of the graph are jointly broken down into several *weakly connected components* (defined analogous to SCC, but for undirected graphs), that encircle the CORE like the petals of a daisy flower. These petals are each subsets of the IN and OUT components from the bow-tie model (see Figure 2).

Both the bow-tie model and the daisy model provide a general idea of how the Web is organised on a global scale. However, they provide no insight in how vertices tend to relate to each other. For this, we need another concept called the degree distribution of the graph.

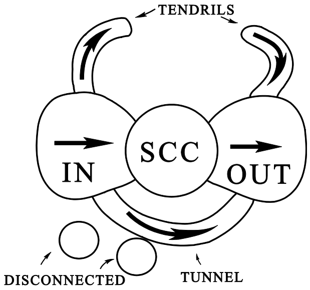

Figure 1: The bow-tie visualisation of the Web graph.

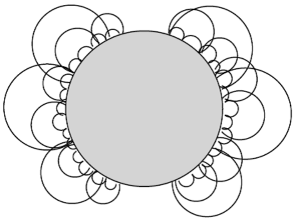

Figure 2: The daisy visualisation of the Web graph.

3.1 Degree Distributions and Scale-free Graphs

The degree distribution of the Web has received much attention in the scientific community, because it shows similarities to various other networks. To explain the concept some predicates require definition:

Let $indeg\ (x)$ be the in-degree of vertex x, defined as the number of neighbours that point to the vertex:

$$indeg\ (x) \triangleq |A_{in}(x)|$$

Similarly the out-degree of x is defined as the number of vertices x points to:

$$outdeg(x) \triangleq |A_{out}(x)|$$

The total degree of x, $deg\ (x)$, is defined as:

$$deg\ (x) \triangleq |A(x)| = indeg(x) + outdeg(x)$$

If a graph is seen as undirected, the total degree may also be written as $deg(x) \triangleq |E(x)|$, since $|A(x)| = |E(x)|$.

The degree distribution $P(k)$ of a graph gives the probability that a node x has exactly degree k:

$$P(k) \triangleq Prob(deg(x) = k|G)$$

This value is obtained by counting the number of nodes that have degree $k \in K$, where K is the set of all degrees that occur in the graph, and dividing by the total number of nodes in the graph, $N = |V|$:

$$P(k) = \frac{\#_{x \in V}\ deg(x) = k}{N}$$

The directed graph degree probabilities $P(k_{in})$ and $P(k_{out})$ are defined analogously.

Since the influential work by Paul Erdős and Alfréd Rényi (Erdős & Rényi, 1960) it has been the assumption that two nodes in a graph are connected with random probability p, which is independent of any other edge or node. If a node is connected to on average z other nodes and the total number of nodes in the graph is N, then it follows that $p = \frac{z}{N-1}$. For large N, p can be approximated by $\frac{z}{N}$. The degree distribution of such a graph is then:

$$P(k) = \binom{N}{k} p^k (1-p)^{N-k} \simeq \frac{z^k e^{-z}}{k!},$$

where the second equality becomes exact in the limit of large N (Newman, Strogatz & Watts, 2001). The distribution will be recognized as Poisson, which means that most of the vertices in the graph have a degree of (a value close to) z, while vertices with a much lower or higher degree are rare. However, as the reader might have guessed, it turns out that the Poisson degree distributed model (which we will refer to as the ER-graph, after Erdős and Rényi) does not do justice to various real-world graphs, such as power grids, metabolic systems, collaboration networks and food webs (see references in Newman et al., 2001). We will now proceed with another degree distribution that more accurately models the Web and other networks.

3.1.1 The Scale-free Model

Barabási, Albert & Jeong (2000), Huberman & Adamic (1999) and Faloutsos, Faloutsos & Faloutsos (1999) experimented on Web crawls and found that the degree distribution of the WWW follows a power-law; that is, the probability $P(k)$ is proportional to $k^{-\gamma}$ (c is a normalising constant):

$$P(k) \cong ck^{-\gamma}$$

Barabási et al. (2000), Broder et al. (2000), Kumar, Raghavan, Rajagopalan, Sivakumar, Tomkins & Upfal (1999) and Laura, Leonardi, Caldarelli & De Los Rios (2002) subsequently attempted to find the value for γ, which they estimated at $\gamma \approx 2.1$. In contrast to ER-graphs, this degree distribution is heavily right-skewed, which implies that many nodes with a low degree exist, but the probability that a node has an extreme degree (i.e. it is a hub) is still significant. Furthermore, only a small amount of vertices has degree z.

Because the degree distribution of these graphs can be said to follow a scale-free power-law, the type of graphs has been named 'scale-free' graphs by Barabási and Albert, but in the literature there has been some confusion as to what graphs are scale-free (or 'scale-invariant') and what are consequences of this property. The following claims are regularly associated with SF-graphs (Li, Alderson, Doyle & Willinger, 2005; Keller, 2005):

- The degrees of an SF-graph are distributed according to a power-law.
- An SF-graph can be generated by using a stochastic process, prominently *preferential attachment* (Barabási *et al.* 1999).
- SF-graphs have an extremely small diameter.
- SF-graphs are self-similar.
- SF-graphs have many hubs (nodes with a very high degree) that are supposed to 'hold the network together' and are said be to be the cause that SF-graphs are highly error-tolerant, but vulnerable to targeted attacks.

Each of these claims will be discussed subsequently in relation to the Web.

Typical examples for ER- and SF-graphs are given in Figure 3.

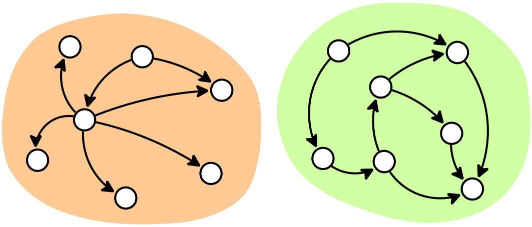

Figure 3: Two graphs with the same number of nodes, but a different degree distribution. The left graph exemplifies a scale-free graph where hubs occur relatively frequently, the graph to the right exemplifies a more traditional Erdős-Renyí-graph where most nodes have a close to average degree.

3.1.2 Scale-invariance

The terms power-law distribution and scale-free graph are used almost interchangeably since the original publications (Barabási *et al.* 2000, Albert, Jeong & Barabási, 1999), while they are actually different (but related) concepts and should be treated as such. The label 'scale-free' simply means that this power-law is independent of N, the number of nodes in the graph. Indeed, a power-law can trivially be shown to be scale-invariant (Keller 2005), but scale-invariance does not demand a power-law. In other words, graphs with empirically confirmed power-law distributions are scale-free, but scale-free graphs are not always distributed according to a power-law.

Li et al. (2005) have extended the scale-free graph theory by introducing a metric that defines if a graph is scale-rich or scale-free. Let G be a connected graph with a fixed degree sequence. The s-metric for G is then:

$$s(G) = \sum_{(x,y)\in E} deg(x) \cdot deg\,(y)$$

The higher the value for $s(G)$, the more scale-free the graph. The metric can be normalized if s_{max} is considered the maximum attainable value for s given the degree distribution of G (but any possible connectivity configuration): $S(G) = \frac{s(G)}{s_{max}}$ with $s_{max} = max_{\mathrm{E}\subseteq\mathrm{V}\times\mathrm{V}}(s(V,E))$.

Li et al. show that as a consequence of the Rearrangement Inequality the metric will be high for graphs where many hubs are interconnected (i.e. there is a ‘hub-like core’ in the graph, which is the case in the Barabási-Albert model), and low if many hubs are connected to low-degree nodes. By using this metric they redefine scale-free graphs as graphs with a scaling degree distribution *and* a high value for $S(G)$[1]. The advantage of this definition is that its formality makes it much less ambiguous and that the concepts of scale-invariance and power-laws are separated. The s-metric shows that it is possible to construct a graph that is scale-free, without being power-law distributed. Properties of empirically observed graphs are therefore not necessarily a consequence of the scale-free nature of these networks, but are caused by different mechanisms. In the following subparagraph we discuss some of the possibilities.

3.1.3 Generative Models for SF Graphs

A helpful procedure when trying to understand a graph is constructing a new one that has the same properties. In this regard various studies have attempted to simulate graphs that by some mechanisms result in a scale-free power-law degree distribution. The most widespread mechanisms were introduced by Barabási *et al.* (2000), and accompanied their finding of the power-law in the Web: *growth* and *preferential attachment* (PA).

The model assumes that the generation of a graph starts with a collection of nodes: N_0. At each time step t we add a new node x to this collection (growth). This new node forms m links to the old nodes ($m \leq N_0$). The nodes that x links to are not determined uniformly, instead the model states that this probability is dependent on the degree of the nodes already present:

$$Prob\big((x,y) \in E_{t+1} \big| (V_t + x)\big) =$$

$$\frac{deg\,(y)}{\sum_{z\in V} deg\,(z)} = \frac{deg\,(y)}{2\cdot|E|}$$

At time t there are $N_0 + t$ nodes and mt edges. The degree distribution that results from this model can now be derived. Let $k_x(t)$ be the degree of node x at time t, which we will approximate as a continuous random variable. Then $\kappa_x(t) = \mathcal{E}(k_x(t))$ is the expectation value of the degree of a node. The growth rate of $\kappa_x(t)$ can be determined:

$$\frac{d\kappa_x(t)}{dt} = \frac{\kappa_x(t)}{2t}$$

$$k(t) = D \cdot \sqrt{t}$$

The value of the constant D can be determined by looking at the initial condition $k(t_0) = m = D \cdot \sqrt{t}$, so $D = \frac{m}{\sqrt{t_0}}$. Thus it follows that

$$k(t) = m \cdot \sqrt{\frac{t}{t_0}}$$

From this point we can obtain the degree distribution $P(k)$ as the derivative of the cumulative probability $P(k_x(t) < k)$:

$$P(k_x(t) < k) = P(t_0 > \tfrac{m^2t}{k^2})$$

$$= 1 - \ P(t_0 \leq \tfrac{m^2t}{k^2})$$

$$= 1 - \frac{m^2t}{(N_0 + t)\cdot k^{-2}}$$

$$P(k) = \frac{d\left(1 - \frac{m^2t}{(N_0+t)\cdot k^{-2}}\right)}{dk} = \frac{2m^2t}{(N_0+t)} \cdot \frac{1}{k^3}$$

So for large values of t we have $P(k) = \frac{2m^2}{k^3}$, which predicts a value of 3 for γ and in addition provides an estimate of the constant c (see 3.1.1). Barabási et

[1] A graph with a power-law degree distribution provides an example of a graph with high $S(G)$.

al. suggest that the difference between this analytic value and the one found on the Web can be explained by additional mechanisms, such as the rewiring of already existent edges. Nonetheless, a power-law degree distribution is indeed obtained by this model.

Laura, Leonardi, Millozzi, Meyer & Sibeyn (2003) have implemented two models to generate web-like graphs as well. The first one is called the Evolving Network model and is essentially a combination of growth and PA (i.e. based on the mechanisms as given by Barabási *et al.*), although in their paper Laura et al. limit the PA to the in-degree of the node (as opposed to total degree). The second model is the Copying model, based on the theory developed by Kumar *et al.* (1999), where new nodes have probability α that they copy an edge of a prototype node p, and probability $1-\alpha$ that they connect to a randomly selected other node from the total graph. Laura *et al.* conclude that both of these generative models result in graphs with statistics similar to the Web, in particular they show a power-law degree distribution.

Similar to the Evolving Network model, Pennock, Flake, Lawrence, Glover & Giles (2002) suggest the Network Growth model, in which they combine preferential attachment with a uniform probability distribution for the adding of new nodes. By using this model, they are able to explain the structure of specific subregions of the web (i.e. university webpages or newspaper webpages) more precisely than with PA alone.

Although the ‘growth and PA’-model is strikingly intuitive, the Copying model and the Network Growth model show that Li *et al.* were right in their criticism: indeed there are multiple explanations for real-world graphs with scale-free power-law degree distributions. Even more models and/or refinements have been proposed (see for example Cooper & Frieze, 2003; Dorogovtsev, Mendes & Samukhin, 2000; Pandurangan, Raghavan, & Upfal, 2002), which means that only more experiments can unveil what the true underlying mechanisms for the Web graph are.

Besides the fact that multiple models can explain the scale-invariant power-law distributions as found on the Web, these models seem to be incomplete. Newman (2002) shows this by looking at graph assortativeness. In general, an assortative graph is a network with nodes that connect to each other because they have some similarity, while in a disassortative graph nodes connect to each other because they are different. In practice, assortativeness is usually associated with node degree. In an assortative graph, nodes with a high degree connect to other nodes with a high degree, and vice versa for nodes with low degree. Newman defines the assortativity coefficient r $(-1 \leq r \leq 1)$ that captures the assortativeness of an entire graph and emprically determines that the Web crawl by Barabási et al. (2000) is disassortative ($r = -0.065$), while the growth and preferential attachment model suggests an assortativity coeffcient of exactly $r = 0$. The question remains open what refinements of the models are required to capture the Web.

3.1.4 Small-world Properties

A further characteristic of the Web graph is its diameter, in social networks also referred to as the ‘degree of separation’. This concept became widespread after a famous experiment by Milgram in 1967, who proposed the ‘small-world’ hypothesis: everyone on the earth is connected to everyone else through no more than six steps – the ‘six degrees of separation’.

The diameter of G can be defined as the average shortest path between all pairs of vertices (Albert et al. 1999), or, in case not all the nodes are connected, the average *connected* shortest path (Broder et al. 2000). We adhere to the latter definition. Let $d(x,y)$ be the length in vertices of the shortest path from x to y. The average shortest path of G is then given as:

$$diameter(G) = \frac{1}{|A^+|} \sum_{(x,y)\in A^+} d(x,y)$$

A graph is considered a small-world graph if its expected diameter is a function of the logarithm of N: $\mathcal{E}(diameter(G)||V| = N) = \log(N)$. Several studies show that indeed the Web graph is a small-world graph (Albert et al. 1999; Broder et al. 2000; Bollobás & Riordan, 2002; Chung & Lu, 2002; Cohen & Havlin, 2003). The models these studies propose suggest diameters as small as 3.14 (Cohen & Havlin, 2003), while the actual observed diameters range from 16 to 21. Although the models cannot be considered very accurate, the Web has an extremely small diameter nonetheless.

3.1.5 Self-similarity

Self-similarity in graphs refers to the concept that subsets display the same properties as the entire graph; for example in the sense of degree distribution or diameter length. In scale-free graphs, the combination of the slow (logarithmic) increase of the graph diameter and the power-law degree distribution provide an indication that such a graph cannot be self-similar. If it would be the actual case, then a scale-invariant power-law relationship between N and $diameter(G)$ would be expected (Song, Havlin & Makse, 2005). Interestingly, Song et al. were able to reconcile the degree distribution and diameter and by using a box-covering technique[2] they found that their case study scale-free graphs (one of which was the same web crawl Albert et al. used in 1999) exhibited self-similarity.

Earlier, Dill, Kumar, McCurley, Rajagopalan, Sivakumar & Tomkins (2002) empirically tested the Web for self-similarity and obtained some interesting results. In their experimental setup, they generated seven disjoint random subsets out of a web crawl consisting of 60 M pages. Interestingly, these subsets where distributed according to (significantly close to) the same power-law degree exponents $\gamma_{in} = 2.1$ and $\gamma_{out} = 2.23$. In addition, the ratios of the different components (recall the CORE, etc.) in the random subsets were consistent with those found by Broder et al. (see Fig. 3). Dill et al. concluded that the web is self-similar and that this self-similarity is pervasive, i.e. it holds for several parameters (degree distributions, component sizes). In combination with the findings by Song et al. these results provide a strong indication for self-similarity in the Web. In paragraph 5 we return to the subject of self-similarity when we compare communities to the Web graph.

3.1.6 Resilience of SF Graphs

A network can suffer from two kinds of failures: errors and attacks. The former refers to the malfunctioning of random nodes, while the latter refers to the removing of specific targeted nodes. The resilience of a graph can be tested by measuring the change in diameter after such a failure has occurred. If the diameter increases significantly, the nodes that have been removed were crucial in several paths through the network. If the diameter stays (almost) the same, then the removed nodes apparently played only a minor role.

In real-world situations, many networks are highly resistant against errors. For example, downtime of a website rarely affects the accessibility of another website, because there are other paths available. This quality is often ascribed to redundant graph edges, i.e. edges that serve only as backup in case of errors, but Albert, Jeong & Barabási (2000) show that error-resistance occurs only in scale-free graphs and is not a consequence of redundant wiring, but of the power-law degree distribution. In an ER-graph, the removal of any node causes the same damage to the network as would any other node. In SF-graphs however, many nodes can be removed without any harm (the nodes with a low degree, through which only a few paths run and therefore hardly affect the diameter). On the other hand, if a hub vertex is removed the resulting network may break apart into several disconnected components.

Crucitti, Latora, Marchiori & Rapisarda (2003) show that when 2% of the nodes of a scale-free network are removed at random, the graph is still hardly affected. If these 2% are targetted at high-degree vertices however, the network quickly falls apart. Since the Web is a power-law distributed, the same rules for its resilience apply. A well-placed attack on a couple of large news-sites for example could severely damage the connectivity of the graph. How such network catastrophes can be avoided remains a hot topic in graph theory.

4 LOCAL STRUCTURE OF THE WEB GRAPH

Now that the global structure of the Web graph has been discussed, we turn to local phenomena: graph communities. A community is a collection of nodes in a graph that are somehow related. Some examples of communities in other types of graphs than the Web are protein-clusters that together have specific functions in the metabolic system, or power grids that together provide electricity for an area. Such community structures, or clusters as they are sometimes called, are also meaningful in the Web. The most obvious implementation of a web-community would be collections of pages that share a topic. When trying to allocate vertices to clusters by topic, we are looking at the *contents* of the vertex. However, as explained before, in this paper we ignore vertex content and focus on the

[2] The algorithm by Song et al. uses a box-covering technique. They would create 'boxes' of a certain size l_B (this size corresponded to the distance nodes in these boxes were away from each other) and cover the entire network with N_B of these boxes. Their result shows a power-law relationship between the size of the boxes and the number of boxes that were needed to cover the entire graph, indicating self-similarity. For an in-depth explanation of the algorithm we suggest Song et al. (2005) and Song, Havlin & Makse (2006).

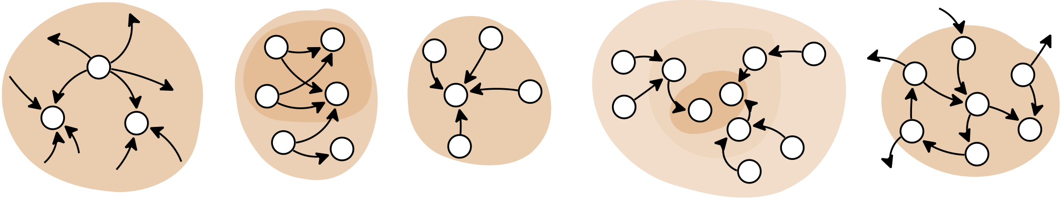

Figure 4: From left to right: the community models 1-5 (see text).

connectivity of vertices – on *structural* communities. In the following paragraph we will discuss several models for structural communities as found in the available literature.

4.1 Community Models

There is a wide range of approaches to graph communities. We list the basic idea behind some of the more striking models (for a more complete overview we suggest Danon, Diaz-Guilera, Duch & Arenas (2005)) and subsequently we will proceed to examine the community model we will adopt in this paper in greater detail. The various models are exemplified visually in Figure 4.

1. Gibson, Kleinberg & Raghavan (1998) define a community as the combination of a set of 'authorative' vertices ($indeg(x) \gg outdeg(x)$) and a set of hub vertices ($outdeg(x) \gg indeg(x)$). These hubs and authorities are connected in their model. By this definition, the communities consist mostly of nodes with a high degree, leaving out affiliated but less definitive nodes.
2. Kumar, Raghavan, Rajagopalan & Tomkins (1999) describe a community as a densely connected bipartite subgraph (a bipartite graph is a graph where the set of vertices can be divided into two disjoint sets V and U such that no edge has both end points in V or U) containing at least one complete bipartite subgraph. While the idea of a bipartite subgraph would result in a very precise notion of a community, it is quite likely that some vertices that are related to the community, would need to be in both partitions since they have neighbours in both sets. According to this model they would not be added to the community.
3. Popescul, Flake, Lawrence, Ungar & Giles (2000) adopt another take and view communities as popular nodes (highest in-degree) and all the nodes pointing to it. Essentially the model takes a local authority and adds its neighbourhood to the community. Here, the problem is that the model does not allow for multiple authorative vertices.
4. Zhou, Wen, Ma & Zang (2002) take yet another turn and portray a community as a collection of concentric circles of nodes. The smallest circle contains the core of the community, the proceeding circles each contain affiliated pages on a ranked scale. Affiliation is defined as linking to nodes in the smaller concentric circle. The model by Zhou et al. is a refinement of the previous model, but it still excludes multiple authorative sources.
5. Flake, Lawrence & Giles (2000) and Flake, Lawrence, Giles & Coetzee (2002) define a community as the collection of nodes that have more links between them than to nodes outside the community. It is a natural definition and captures a strong concept. The model aims to have authorities and hubs inside the community and in addition the nodes that are connected to these vertices. One could rephrase the definition as: a community is a collection of nodes that is separated (but not disconnected) from the rest of the graph. The combination of simplicity and intuitiveness makes this model the model of choice for this thesis.

The simplicity of the definition by Flake et al. leaves room for two interpretations, which are labelled the 'strong community' and the 'weak community'. They are defined as follows (Radicchi, Castellano, Cecconi, Loreto, & Parisi, 2004): Let C be a subset of the graph G. C is a strong community if

$$strong(C) \triangleq \forall_{v \in C}[|A(v,C)| > |A(v,V-C)|]$$

And C is a weak community if

$$weak(C) \triangleq |arcs(C,C)| > |arcs(C,V-C)|$$

It follows that a strong community is also a weak community, while the reverse in general does not hold.

From paragraph 3.1.1 it follows that the Web graph contains a significant large amount of hub-vertices, i.e. vertices with a high out-degree. According to the strong community definition, if such a hub is a member of the community, over half of its neighbours must be within the community. Such a condition is too restrictive for a useful community model, since this would make it near impossible to include hubs in communities. For example, websites from wikis or major news agencies tend to be hubs and could therefore not be included in a community, unless more than half of the websites they are connected with are in the community as well. The weak definition makes it possible that these nodes are added into the community. The community definition that will be used in the remainder of this paper is therefore adjusted into the second alternative, that of a *weak community*. The following paragraph elaborates on how such communities can be found in a graph.

4.2 Graph Modularity

As there are many different community models, it makes sense that there exist multiple implementations of community identification algorithms accompanying these models. This is also the case for identification algorithms that specifically follow the definition by Flake et al. The implementations differ on terms of result, complexity and on whether they operate on the global graph or automate locally. Newman & Girvan (2004) have proposed a mechanism that can evaluate identification results that has become widely accepted. Given a community identification result consisting of n disjoint communities, we can define an $n \times n$ matrix e where each element e_{ij} corresponds to the fraction of all links pointing from community C_i to community C_j:

$$e_{ij} = \frac{1}{|A|}\left|arcs(C_i, C_j)\right|$$

If the network does not show signs of community structure, or if the division of communities was chosen at random instead of by using an adequate algorithm, the expected value of the number of intercommunity links can be approximated, since this is the probability that a link begins in C_i: $\frac{1}{n}$, multiplied by the probability that a link ends in C_j (also $\frac{1}{n}$): $\frac{1}{n^2}$. Since we know the real value of e_{ii} (all links within the community) we can calculate the summed difference between the current community partitioning and uniform partitioning, the *modularity* measure

$$Q(\{C_1, C_2 \dots, C_n\}|G) = \sum_i \left(e_{ii} - \tfrac{1}{n^2}\right).$$

Note that the modularity is a characteristic for the entire graph.

In the extreme case that n communities within a network have been identified, with no links between them, Q will have the value $1 - 1/n$, which tends to 1 for large values of n (Danon 2005), indicating a clear non-random community structure. If this value tends to 0, the community decomposition was unsuccessful. According to Newman and Girvan, the value of Q typically ranges between 0.3 and 0.7 for networks with strong communities, with higher values being rare.

Since finding a high modularity implies that many of the communities accord to at least the definition of a weak community, the modularity itself could be the basis for a community identification algorithm. The algorithm would have to find the maximum $Q(\{C_1, C_2 \dots, C_n\}|G)$ for all possible divisions of the network, which would result in optimal communities. Unfortunately, this process would be very costly in terms of complexity and require an exponential amount of time. For networks with more than say twenty nodes, this is already beyond any practical application, let alone for Web applications such as search engines, so this option can quickly be put aside. Newman (2004) suggests to iteratively calculate the difference in modularity when two communities are joined together. That is, starting with a matrix where each element contains a single node, for each possible combination of two communities we calculate how the modularity of the clustering changes:

$$\Delta Q \triangleq Q(\{C_1, C_2, C_3, \dots, C_n\}) - Q(\{C_1 + C_2, C_3, \dots, C_{n-1}\})$$

The contribution in modularity by C_i and C_j initially was:

$$Q(\{C_i, C_j\}) = \left(e_{ii} - \tfrac{1}{n^2}\right) + \left(e_{jj} - \tfrac{1}{n^2}\right)$$

But after these communities are combined the contribution is:

$$Q(\{C_i + C_j\}) = Q(\{C_i, C_j\}) + \left(e_{ij} - \tfrac{1}{n^2}\right) + \left(e_{ji} - \tfrac{1}{n^2}\right)$$

Therefore the difference in modularity when two communities are joined together is:

$$\Delta Q(\{C_i + C_j\}) = \left(e_{ij} - \tfrac{1}{n^2}\right) - \left(e_{ji} - \tfrac{1}{n^2}\right)$$

This calculation can be done in constant time, resulting in a total complexity for the algorithm of $O(n^2)$ for sparse graphs. The algorithm provides clear community structures and would be useful for our experiment, were it not for the fact that it presupposes that the total graph is known and stored. In practice, the total (size of the) Web is not known and efficient calculation on a graph of this magnitude is infeasible. Therefore a community identification algorithm is needed that can operate locally, i.e. without a priori knowledge of the Web graph. An algorithm that functions on a local scale is proposed by Clauset (2005). It keeps a complexity of $O(k^2)$ where k is a user-given upper bound for the number of vertices to be processed. The algorithm is inspired on modularity as used by Newman, and introduces a new measure R, that of *local modularity*. The algorithm and its underlying model will be explained in the next sections.

4.3 Local Modularity

Instead of dividing a graph into several communities, as the global approach in the previous paragraph suggests, it is in practical applications more useful to find the community that surrounds a given vertex. This way the algorithm does not have to process the entire graph, but only a subset (for example in ranking retrieved websites based on their community membership). The local modularity measure as introduced by Clauset (2005) works according to this concept. We will explain the algorithm and the framework it is built on subsequently.

A *cluster* is a group of nodes from the entire population: $C \subseteq V$, $C \neq \emptyset$ of which we know all link structure (only outbound links on the Web). The cluster is usually not isolated; there are some connections between outsiders and the cluster nodes. These outsiders are referred to as the *universe* U of the community:

$$U(C) \triangleq \{u \in V - C \mid A(C,u) \neq \emptyset\}$$

Not all of the nodes in C have to be connected to U. In fact, a tight community would actually have only a few members that exchange links with outsiders, while most nodes connect only to other community members. The vertices that do connect to U are said to be in the *boundary* B of C (see also Figure 5):

$$B(C) \triangleq \{b \in C \mid A(b,U) \neq \emptyset\}$$

Analogously to global modularity as given by Newman & Girvan, we are interested to what degree the cluster is isolated from these outsiders. This can be expressed by looking at the sharpness of the boundary in relation to the universe, i.e. the number of links from the boundary to the cluster versus the number of links to the entire network. By examining the number of links of the boundary instead of the total cluster, clusters of different sizes can be compared better. The achieved fraction is the *local modularity* $R(C)$ of a graph subset C, defined as 0 when $B = \emptyset$ and when $B \neq \emptyset$:

$$R(C) = \frac{|arcs(B(C), C)|}{|arcs(B(C), V)|} \qquad \textbf{(1)}$$

The local modularity measure is a characteristic of a subgraph that shows how much a cluster is separated from the rest of the graph. If for example $R = 0.9$ and $|C| \ll |G|$, we have a subgraph that is only thinly connected to the rest of G. Such a cluster is a *community*, if its local modularity measure exceeds a given threshold d:

$$community(C) \triangleq R(C) \geq d$$

In this paper it is assumed that $d = \frac{1}{2}$, since this is the threshold at which the weak community definition as given in section 4.1 is true.

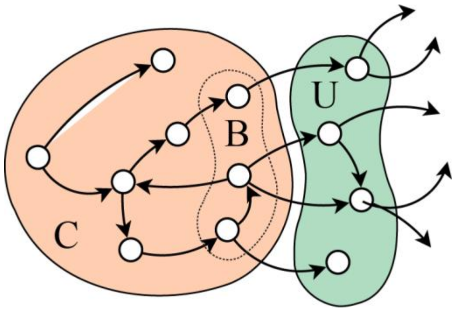

Figure 5: The local modularity model. The blob to the left is the cluster, the sub-blob is its boundary, connected to the blob to the right: its universe. In this situation, $R(C) = 2/6$.

4.3.1 Community Identification Algorithm

Because local modularity does not require knowledge of the entire graph, we can find communities with maximum local modularity from a single seed vertex v_0. The basic idea is simple: for each neighbour of the cluster (i.e. its *universe*) we evaluate which vertex would increase the modularity the most if it was added to the cluster.

Table 1: The local modularity algorithm.

```
C := ∅;
v := $v_0$;
**repeat**
   C := C+v;
   v := argmax{R(C+u)|u∈U(C)}
**until** |C| = k **or** R = d
```

We can continue this greedy algorithm indefinitely if we allow the maximum increase to be negative: i.e. if no vertex would increase the modularity, we add the node that provides the least decrease. The algorithm is described using pseudocode in Table 1.

The stop condition of the algorithm is arbitrary. Either the algorithm processes k nodes, or the process continues until a given local modularity threshold $0 < d \leq 1$ is reached. We return to this criterion in section 4.5 where we examine actual clusters. Before that we continue with the analysis of the algorithm.

To calculate which vertex $u \in U$ is the best candidate (i.e. it has the highest $\Delta R(C)$ of all candidates in U) we could simulate the adding of each vertex and calculate $R(C)$ by using (1). In most situations this is inefficient. A better solution is to derive the difference in modularity for each vertex: $\Delta R(C,u)$:

$$\Delta R(C,u) = R(C+u) - R(C) \qquad \textbf{(2)}$$

In order to use this equation, we need to know how $arcs(B(C),C)$ and $arcs(B(C),V)$ change when u is added to the cluster, which depends on how $B(C)$ changes. To analyse this we start with nodes from the boundary that had u as their exclusive neighbour in the universe. These will not be in $B(C+u)$:

$$D(C,u) = \{b \in B(C) \big| A(b, V-C) = \{u\}\}$$

For these nodes the following property holds:

Lemma 1:

$$x \in D(C,u) \rightarrow A(x,V) = A(x,C) + u$$

There can now be two distinct situations:

1. u will not become a boundary member of $C+u$, i.e. $A(u, V-C) = \emptyset$ or
2. u will become a boundary member of $C+u$, i.e. $A(u, V-C) \neq \emptyset$.

Situation 1: $u \notin B(C+u)$:

$$B(C+u) = B(C) - D(C,u)$$

From this we derive that

$$arcs(B(C+u), C+u) = arcs(B(C),C) + arcs(B(C),u) - arcs(D(C,u),V)$$

And

$$arcs(B(C+u),V) = arcs(B(C),V) - arcs(D(C,u),V)$$

Situation 2: $u \in B(C+u)$:

$$B(C+u) = B(C) - D(C,u) + u$$

We derive that

$$arcs(B(C+u), C+u) = arcs(B(C),C) + arcs(B(C),u) - arcs(D(C,u),V) + arcs(u,C)$$

And

$$arcs(B(C+u),V) = arcs(B(C),V) - arcs(D(C,u),V) + arcs(u,V)$$

Both situations are illustrated with hypothetical examples, also visualised in Figure 6.

Example situation 1:

$$V = \{1,2,3,4,5\}$$
$$A = \{(2,1),(2,4),(3,4),(3,5)\}$$
$$C = \{1,2,3\}$$

The universe and boundary of this cluster are therefore

$$B(C) = \{2,3\}$$
$$U(C) = \{4,5\}$$

If we consider the adding of node 4 to the cluster, then

$$D(C+4) = \{2\}$$
$$B(C+4) = B(C) - D(C,4) = \{3\}$$

By using the above formulae, we derive that $arcs(B(C+4), C+4) = \{(2,1)\} + \{(3,4)\} - \{(2,1)\} = \{(3,4)\}$

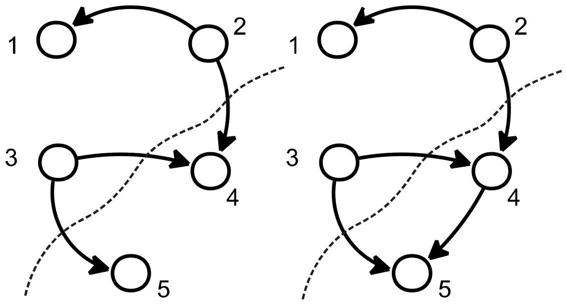

Figure 6: In the diagram on the left, node 4 has no arcs to $V - C$, therefore it is not in $B(C+4)$. In the diagram on the right the node does have an arc to a node in $V - C$, therefore node 4 is in $B(C+4)$. See text for a detailed analysis.

And $arcs(B(C+4), V) = \{(2,1), (2,4), (3,4), (3,5)\} - \{(2,1), (2,4)\} = \{(3,4), (3,5)\}$

The change in modularity that the addition of node 4 causes, is:

$$\Delta R(C,4) = \frac{|\{(3,4)\}|}{|\{(3,4),(3,5)\}|} - \frac{1}{4} = \frac{1}{2} - \frac{1}{4} = \frac{1}{4}$$

Example situation 2:

$$V = \{1,2,3,4,5\}$$
$$A = \{(2,1), (2,4), (3,4), (3,5), (4,5)\}$$
$$C = \{1,2,3\}$$

The universe and boundary of this cluster are therefore

$$B(C) = \{2,3\}$$
$$U(C) = \{4,5\}$$

If we consider the adding of node 4 to the cluster, then

$$D(C+4) = \{2\}$$
$$B(C+4) = B(C) - D(C,4) = \{3,4\}$$

By using the above formulae again, we derive that $arcs(B(C+4), C+4) = \{(2,1)\} + \{(2,4),(3,4)\} - \{(2,1),(2,4)\} + \{\,\} = \{(3,4)\}$

And $arcs(B(C+4), V) = \{(2,1), (2,4), (3,4), (3,5)\} - \{(2,1), (2,4)\} + \{(4,5)\} = \{(3,4), (3,5), (4,5)\}$

The change in modularity that the addition of node 4 causes, is:

$$\Delta R(C,4) = \frac{|\{(3,4)\}|}{|\{(3,4),(3,5),(4,5)\}|} - \frac{1}{4} = \frac{1}{3} - \frac{1}{4} = \frac{1}{12}$$

These trivial examples show the working of the algorithm, but obviously it is intended to be applied to real-world graphs. The next section discusses the local modularity model and algorithm in relation to the Web.

5 EXPERIMENTAL RESULTS

By running the community identification algorithm as presented in the previous section, we acquire knowledge about the structure of the Web. Most prominently, if a large percentage of the identified clusters have a high local modularity value, then this indicates that the Web is modular in structure, i.e. several components are connected through hubs, but have little to no mutual connections. If on the other hand most of the clusters have only a low value of $R(C)$, then the web is much more interconnected. In addition, if many communities are identified, then these clusters are likely to be valuable for information retrieval purposes. The underlying assumption is that these clusters exhibit their high local modularity value because their nodes somehow relate to each other. We return to the cluster hypothesis in section 5.3; for now we proceed by elaborating on our experimental setup.

Our community identification algorithm was implemented in Microsoft C#. The main routine is globally given in Table 1. Instead of running the algorithm on the Web in real time, we decide to use a Web crawl, because slow responding Web servers would delay the batch process greatly. It should be noted that no meta-information about the crawl is used in the implementation of the algorithm and the experiment is entirely repeatable real-time. We restricted the crawl dataset to unique second-level domains on the Dutch top-level domain[3] .NL (± 1.5 million nodes), obtained in April 2007. We defined an arc to exist between two nodes A and B if these domains had at least one hyperlink from domain A

[3] In the URL ‘www.foo.bar’, ‘bar’ is the top-level domain and ‘foo’ the second level domain.

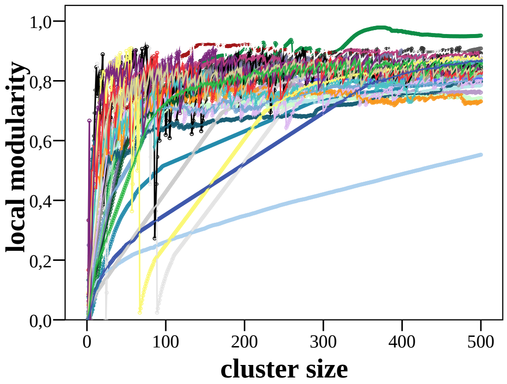

Figure 7: The modularity progression for 30 random seed vertices.

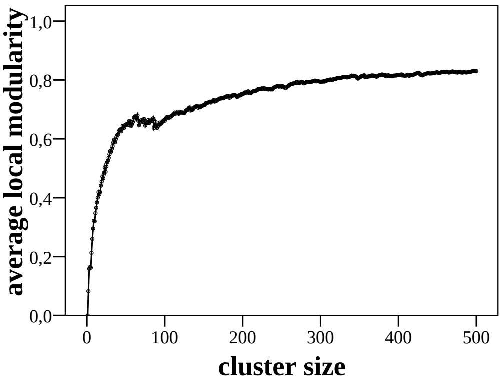

Figure 8: The average modularity progression for 30 random seed vertices.

to domain B; the actual number of links is not taken into account[4].

Our experiment is divided into two parts. In the first part clusters were initialised with a random node v_0. Seed vertices with an out-degree of 0 were excluded. The second part uses the same crawl, but instead of random v_0 we pick specific second-level domains, so we can evaluate the relations between the seed node and the other nodes in the achieved cluster manually. We return to these results in 5.3.

The results from the first set are shown in Figure 7. The chart shows that for 25 out of 30 clusters, the local modularity exceeds the 0.5 barrier before $|C| = 50$. Furthermore, 25 out of 30 clusters consistently reach $R(C) = 0.8$ before $|C| = 100$.

From this part of our experiment, a few of the clusters show behaviour that differs remarkably from others. The blue line in Figure 7 for example (the line corresponding to the lowest modularity value after $|C| = 150$) corresponds to an extremely high average degree of 14.5, as opposed to $\pm$ 4.3 in the entire graph. Other anomalies are clusters that are strongly isolated from the rest of the graph. At $|C| \pm 75$ their only neighbour in U is a hub with a high degree. As soon as this hub is added to the cluster, it greatly decreases the modularity score. When the surroundings of this hub are added to the clusters one by one, the modularity slowly increases again. This shows that the community that was identified before this hub was added, was a component of the community that is identified when $|C|$ approaches 500.

Besides a few explainable anomalies, the clusters all show a remarkable similar pattern. The modularity score skyrockets when the algorithm starts, varying in the range $0.6 < R(C) < 0.9$ before the clusters contain more than 100 nodes. At this point, the $R(C)$ is quite stable in the range $0.7 < R(C) < 0.9$. These findings are made more clear in the chart that plots the average modularity progression, as given in Figure 8. It is noteworthy that in this experiment the average local modularity can be accurately approximated by a logarithmic function: $R(C) \cong c\ln(|C|) + d$, where c and d are constants with respective values 0.116 and 0.127 in this fit. The coefficient of determination R^2 between this formula and the average empirical data is 0.953, indicating a relationship between the size of the cluster and its modularity value. This relationship can be explained by looking at the equations for $\Delta R(C + u)$ (the derivative of $R(C)$) in the previous paragraph. In situation 1, some rearranging of the terms in (2) provides:

$$\Delta R(C+u) = \frac{arcs(B(C),u) + (R-1)\cdot arcs(D(C,u),V)}{arcs(B(C),V) - arcs(D(C,u),V)}$$

$B(C)$ is a function of C, and $1 \leq |B(C)| \leq |C|$. When $|C|$ and $|B(C)|$ are large enough, the above equation is determined by $|arcs(B(C),V)|$, which is independent of u:

$$|arcs(B(C),V)| = \sum_{c \in B(C)} outdeg(c)$$

So in situation 1, $\Delta R(C + u)$ can indeed be approximated by $\frac{d}{|C|}$ where d is a constant determined by the relation between $B(C)$ and C and the average degree in the cluster. The same line of reasoning, mutatis mutandis, is applied to situation 2:

[4] This way we prevent certain recurring hubs to be added in every cluster, like www.google.com, as many websites have a google search option on each of their pages.

$$\Delta R(C+u) = \frac{arcs(B(C),u) + (R-1) * arcs(D(C,u),V)}{arcs(B(C),V) - arcs(D(C,u),V)}$$

Again, this equation is determined by $|arcs(B(C),V)|$ when the cluster and its boundary are large enough, so $\Delta R(C+u)$ can be approximated by $\frac{d}{|C|}$.

5.1 Pruning

The community identification algorithm is greedy and therefore some nodes will be added to the clusters that should in hindsight better be left out. We introduce a pruning module to run after the identification process is completed (i.e. the community size has reached its limit or a modularity threshold is reached) to remove nodes from $B(C)$ that negatively influence the modularity of the cluster. This procedure is described in Table 2.

Table 2: Pruning.

```
repeat
   v := argmax{R(C-u)|u∈B(C)-$v_0$}
   C := C-v;
until R(C-v) ≤ R(C) or |B(C)|=1
```

The details of the working of the procedure are analogous to those of the original algorithm and are omitted.

By pruning clusters the modularity value increases (or in the worst case scenario, stays the same). We reran the community identification algorithm on the same 30 seed vertices as before, but after the cluster size reached 100 nodes we applied the pruning algorithm. This time our community size threshold was 100, so we could see how pruning affects the anomalies as discussed in the previous section.

The average modularity value at $|C| = 100$ was $R(C) = 0.661$, much lower than after pruning was applied, which resulted in an average local modularity of 0.756. The difference between these values shows that it is possible to construct clusters on the Web graph with even higher modularity. The clusters that benefited the most from pruning were those that showed the anomalies in Figure 7, since the hub that greatly decreased the modularity was removed by pruning.

Since pruning removes nodes from $B(C)$, the boundary of the clusters shrinks as more and more nodes are removed. On average the boundary of a cluster of 100 nodes consists of 21.65 nodes. After pruning, this value is more than halved to 10.69, with an average cluster size of 89.04. So in conclusion, the original algorithm with the pruning module combined result in clusters which are connected to the rest of the Web by only 12% of their nodes and by following an arc from one of these nodes, the average chance to 'exit' the cluster (i.e. follow an arc to a node outside C) is $1 - R(C) = 0.244$. The following sub paragraph discusses the relation between these findings and the Web graph model.

5.2 Relation Between Local and Global Web Graph Phenomena

None of the clusters reached a local modularity value of 1 except in trivial situations where $|C| = 2$ (excluded in Figure 7). With the bow-tie model in mind we expected a significant part of the clusters to reach a modularity of 1 before 500 nodes were accumulated, which would correspond to the OUT-component or an outward TENDRIL. The lack of this event seems to indicate that the bow-tie model is too crude. In the daisy model the IN- and OUT-components are split up into several smaller components, that allow for some overlap. This is more in line with our experience, where clusters of nodes are connected to other nodes through their boundary. The smaller the boundary and the higher the modularity of the cluster, the more isolated the petal of the daisy flower model. Boundary nodes that occur in various clusters can be said to be in the CORE of the model. These nodes are usually mega-hubs that provide services as Web statistics or link portals; websites with little actual content of their own. Pruning removes these nodes from the clusters, leaving only the petals.

Earlier we examined the degree distribution of the Web in detail and discussed the scale-free power-law that is often associated with the Web and other real-world graphs. The fact that the community identification algorithm will only add hub nodes when it either has no other choice or the hub is indeed relevant to the cluster, can be seen when we compare the degree distribution plots from the clusters combined and the total graph. The average degree on the .NL-crawl is 4.25, while the clusters together provide an average degree of 3.29, significantly lower. Although the average degree in the clusters is lower, both the clusters as well as the total .NL-crawl show the by now familiar power-law degree distribution. The respective parameters for these distributions are $\gamma_{total} = 1.34$ and

$\gamma_{clusters} = 1.89$, a further confirmation that the clustering algorithm excludes mega-hubs.

5.3 The Cluster Hypothesis

In line with Van Rijsbergen's cluster hypothesis (Van Rijsbergen, 1979), we assume that closely associated nodes in a graph are mutually related. For the local modularity algorithm this implies that clusters with a high modularity score contain websites that have some relation to the other nodes and v_0. To test this we ran a second batch of community identification processes, but instead of randomly selecting a seed node v_0 we chose specific websites that enabled us to evaluate the obtained clusters. The results of three of such clusters are shown in Table 3. The modularity score of 0.5 (the rightmost column) is significant, as can be seen when it is compared to the worst-case local modularity:

$$R_{worst\ case}(C) = \frac{|C|}{\sum_{c \in C} outdeg(c)}$$

This value is shown below the cluster's content. The difference between this value and the actual value of the cluster (0.5) shows that the websites are indeed associated in a graph-theoretical way and inspection of the contents of the clusters shows that the clusters indeed contain related websites. In fact, of the 51 listed vertices (54 are listed in Table 3, but we exclude the seed vertices since they are

Table 3: Nodes in clusters with $R(C) \geq 0.5$. Blank descriptions belong to websites that were inaccessible.

$\boldsymbol{v_x}$	**Website**	$\boldsymbol{outdeg(x)}$	**Description**	$\boldsymbol{R(C+v_x)}$
0	overheid.nl	18	Dutch Government Services	0.0
1	minocw.nl	4	Ministry of Education, Culture and Science	0.091
2	postbus51.nl	0	Government Information Desk	0.182
3	regering.nl	0	Dutch Government	0.273
4	bedrijvenloket.nl	2	Government Services for Businesses	0.333
5	e-overheid.nl	0	Electronic Government Services	0.375
6	info-wmo.nl	0	Information on the Act of Social Support	0.417
7	minvws.nl	6	Ministry of Traffic and Water management	0.467
8	kiesbeter.nl	0	Health Care Counseling	**0.500**
Worst case modularity 0.267				
$\boldsymbol{v_x}$	**Website**	$\boldsymbol{outdeg(x)}$	**Description**	$\boldsymbol{R(C+v_x)}$
0	ru.nl	39	Radboud University Nijmegen	0.0
1	stw.nl	7	Technology and Science Foundation	0.043
2	sentinels.nl	4	Dutch Security Research Program	0.080
3	nwo.nl	0	Dutch Organisation for Scientific Research	0.120
4	ictregie.nl	0	Developments in ICT-research	0.140
5	wisweb.nl	0	Mathematical Applications for High School	0.160
6	*snnonline.nl*	0		0.180
7	beevee.nl	1	Biology Students Union	0.200
8	nanoned.nl	0	Nanotechnology Network	0.220
9	fom.nl	0	Fundamental Matter Research	0.240
10	gx.nl	0	Website Content Management	0.260
11	wisfaq.nl	0	Mathematical Q&A for High School	0.280
12	wetland-ecology.nl	1	Master Class on Climate Change	0.300
13	embedded-systems.nl	0	Program of STW	0.320
14	minez.nl	0	Ministry of Economy	0.340
15	kizz.nl	0	Administrational Student Services for RU	0.360
16	betabedrijvenbeurs.nl	0	Science and Business Fair	0.380
17	marie-curie.nl	0	Astrophysics Student Union	0.400
18	*nedstat.nl*	0	Web Statistics	0.420
19	*azn.nl*	0		0.440
20	astron.nl	0	Dutch Astronomy Foundation	0.460
21	bioinformatics.nl	0	Bioinformatics Web Portal	0.480
22	jacquard.nl	0	Software Engineering Research Program	**0.500**
Worst case modularity 0.423				

Table 3: Continued.

v_x	Website	$outdeg(x)$	Description	$R(C + v_x)$
0	bnn.nl	33	TV-Network	0.0
1	omroep.nl	4	National channel	0.054
2	vara.nl	6	TV-Network	0.093
3	vpro.nl	2	TV-Network	0.133
4	nederland3.nl	6	TV-Channel	0.157
5	uitzendinggemist.nl	22	Missed transmissions	0.178
6	nederland2.nl	0	TV-Channel	0.205
7	novatv.nl	18	Programme on news	0.231
8	nederlandkiest.nl	5	Mirror of nos.nl	0.260
9	nos.nl	6	National channel union	0.284
10	nederland4.nl	0	Online channel	0.314
11	nederland1.nl	0	TV-Channel	0.333
12	publiekeomroep.nl	16	Mirror of omroep.nl	0.356
13	zapp.nl	0	omroep.nl youth division	0.381
14	cinema.nl	0	Movie-related news	0.407
15	funx.nl	2	Radio- and TV-channel	0.424
16	radio2.nl	2	Radio channel	0.433
17	nps.nl	4	Dutch TV-Programme foundation	0.443
18	llink.nl	2	TV-Network	0.452
19	*ingeborgdouwecentrum.nl*	0		0.460
20	*korrelatie.nl*	0	Mental and social support foundation	0.468
21	*esthervanderheiden.nl*	0	Music teacher / conductor	0.476
22	*vakantietaal.nl*	0	Language courses	0.484
23	*dogsincluded.nl*	0	Information about travelling with dogs	0.492
24	avro.nl	0	TV-Network	**0.500**
Worst case modularity 0.195				

obviously related to themselves), only 8 show no relation to the other vertices of the cluster. These have been listed in italics and they include websites that were unavailable for inspection.

From the three examples it stands out that each of them was seeded with a hub vertex. This was done because these websites might be meaningful to the reader, so the contents of the rest of the cluster can be judged. However, starting with a hub vertex is not a necessity. For example, what happens when instead of seeding with ru.nl, we initiate the clustering with stw.nl, sentinels.nl or beevee.nl is the following. In the first two situations, the seed vertices have a decent degree and will not be inclined to include a hub vertex. They will grow a small community and eventually add ru.nl to their cluster, after which the cluster grows similar to the second cluster in Table 3. The difference is that the first few nodes are tightly related to the seed vertex, while the nodes after ru.nl has been added are less related. In the situation of beevee.nl, the only neighbour of this node is ru.nl, so this hub will be added right away.

6 CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

In this paper we discussed the current state of affairs of the Web graph. Ample attention has been given to aspects of the Web that concern the graph as a whole. In particular we examined the scale-free power-law degree distribution that is typical for the Web and we provided a precise and consequent notation for its associated equations. The degree distribution and the models of connected components give an indication as to how the Web is organised on a global scale, but provide little information about small components of the graph. To investigate these local aspects, we discussed several community models and the concepts *modularity* and its cousin *local modularity*. Local modularity is seen as a measure for the disconnectedness of a cluster and the rest of the graph. We implemented a clustering algorithm based on local modularity, which benefits from being able to be employed on a large and dynamic graph such as the WWW, without having a priori knowledge of this network.

To explain the community identification algorithm we introduced a notational framework that can easily be used in other algorithms or graph theoretical concepts. Application of the local community identification algorithm on the Dutch .NL-domain shows that the Web graph consists of various components that have only a few connections to the rest of the graph, by one or more hub vertices. We found that on average a cluster obtained by our algorithm has a modularity of ± 0.65 when the cluster contains 50-100 vertices, only to increase when more nodes are added, staying consistently at ± 0.8 after the cluster has accumulated 300 vertices. Furthermore, pruning increases the modularity value of these clusters, showing that by enhancing the algorithm clusters can be identified that are even more separated from the rest of the graph. These results indicate that vertices that are not extremely large hubs, are likely to be part of a cluster of the Web graph that is fairly disconnected from the rest, which corresponds globally to the daisy model. Once again, the importance of hub vertices in the Web graph is stressed. Interesting further research should examine these in greater detail, as they are valuable for IR-purposes as well as in understanding graph resilience.

Besides increased knowledge about the structure of the Web graph, we have also evaluated the contents of communities. We preliminarily tested three identified clusters to the Van Rijsbergen cluster hypothesis and we confirmed that by using this algorithm, clusters with a modularity ≥ 0.5 (these have more links from their boundary to the cluster than from their boundary to external nodes) contain semantically related websites. However, in order to fully validate the clustering qualities of our algorithm, further work is required. We propose to investigate if cluster membership can be used in page ranking in search engines and also to evaluate the relation between vertices in clusters using more conventional IR-methods such as document similarity.

Finally, we are looking forward to alternative concepts that can be construed using the framework we used in this paper. A first suggestion might be to look at the status of clusters, as opposed to their disconnectedness, by a new measure:

$$R'(C) = \frac{arcs(C, B(C))}{arcs(V, B(C))}$$

ACKNOWLEDGEMENTS

The author is grateful to Theo van der Weide for his valuable supervision and discussions and to Slatman Internet Services for providing the dataset of the .NL-domain.

REFERENCES

Albert, R., Jeong, H., & Barabási, A.-L. (1999). Diameter of the World Wide Web. *Nature , 401*, 130.

Albert, R., Jeong, H., & Barabási, A.-L. (2000). Error and attack tolerance of complex networks. *Letters to nature , 406*, 378-381.

Barabási, A.-L., Albert, R., & Jeong, H. (2000). Scale-free characteristics of random networks: The topology of the World-Wide Web. *Physica A , 281*, 69-77.

Bollobás, B., & Riordan, O. (2002). The diameter of a scale-free random graph. *Combinatorica , 24* (1), 5-34.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., et al. (2000). Graph structure in the Web. *Computer Networks , 33*, 309-320.

Chung, F., & Lu, L. (2002). The average distances in random graphs with given expected degrees. *PNAS , 99* (25), 15879-15882.

Clauset, A. (2005). Finding local community structure in networks. *Physics Review E , 72*, 026132.

Cohen, R., & Havlin, S. (2003). Scale-free networks are ultrasmall. *Physical Review Letters , 90* (5).

Cooper, C., & Frieze, A. (2003). A general model of Web graphs. *Random Structures & Algorithms , 22* (3), 311-355.

Crucitti, P., Latora, V., Marchiori, M., & Rapisarda, A. (2003). Efficiency of scale-free networks: error and attack tolerance. *Physica A , 320*, 622-642.

Danon, J., Diaz-Guilera, A., Duch, J., & Arenas, A. (2005). Comparing community structure identification. *Statistical Mechanics P09008* .

Dill, S., Kumar, R., McCurley, K. S., Rajagopalan, S., Sivakumar, D., & Tomkins, A. (2002). Self-similarity in the web. *ACM Transactions on Internet Technology , 2* (2), 205-223.

Donato, D., Leonardi, S., Millozzi, S., & Tsaparas, P. (2005). Mining the inner structure of the Web. *8th Workshop on the Web and Databases.* Baltimore, Maryland, USA.

Dorogovtsev, S. N., Mendes, J. F., & Samukhin, A. N. (2000). Structure of Growing Networks with Preferential Linking. *Physics Review , 85* (21).

Erdős, P., & Rényi, A. (1960). On the Evolution of Random Graphs. *Publ. Math. Inst. Hung. Acad. Sci. , 5* (17).

Faloutsos, C., Faloutsos, M., & Faloutsos, P. (1999). On power-law relationships of the internet topology. *ACM SIGCOMM 99, 29*, pp. 251-262.

Flake, G. W., Lawrence, S., & Giles, C. L. (2000). Efficient identification of Web Communities. *6th Int. Conf. on Knowledge Discovery and Data Mining*, (pp. 150-160).

Flake, G. W., Lawrence, S., Giles, C. L., & Coetzee, F. M. (2002). Self-organisation and identification of web communities. *IEEE Computer , 35* (3), 66-71.

Gibson, D., Kleinberg, J. M., & Raghavan, P. (1998). Inferring Web Communities from Link Topology. *Proc. of the ACM Symposium on Hypertext and Hypermedia*, (pp. 225-234). Pittsburg, PA, USA.

Gulli, A., & Signorini, A. (2005). The Indexable Web is more than 11.5 billion pages. *Int. WWW. Conf. Special interest tracks and posters of the 14th international conference on World Wide Web*, (pp. 902-903).

Huberman, B. A., & Adamic, L. A. (1999). Growth dynamics of the WWW. *Nature , 401*, 131.

Keller, E. F. (2005). Revisiting "scale-free" networks. *BioEssays , 27* (10), 1060-1068.

Kim, B. J., Yoon, C. N., Han, K., & Jeong, H. (2002). Path finding strategies in scale-free networks. *Physical Review E , 65* (2), 027103.1-027103.4.

Kumar, R., Raghavan, P., Rajagopalan, S., & Tomkins, A. (1999). Trawling the Web for emerging cyber-communities. *Computer networks , 31*, 1481-1493.

Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A., & Upfal, E. (1999). Stochastic models for the web graph. *41st FOCS*, (pp. 57-65).

Laura, L., Leonardi, S., Caldarelli, G., & De Los Rios, P. (2002). A Multi-Layer Model for the Web Graph. *2nd Int. Workshop on Web Dynamics.* Honolulu.

Laura, L., Leonardi, S., Millozzi, S., Meyer, U., & Sibeyn, J. F. (2003). Algorithms and Experiments for the Web Graph. *Proc. of the European Symposium on Algorithms.*

Li, L., Alderson, D., Doyle, J. C., & Willinger, W. (2005). Towards a theory of scale-free graphs: Definition, properties and implications. *Internet Mathematics , 2* (4).

Newman, M. E. (2002). Assortative mixing in networks. *Physical Review Letters , 89* (20), 208701.1-208701.4.

Newman, M. E. (2004). Fast algorithm for detecting community structure in networks. *Physics Review E , 69, 066133*

Newman, M. E., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physics Review E , 69*, 026113.

Newman, M. E., Strogatz, S. H., & Watts, D. J. (2001). Random graphs with arbitrary degree distributions and their applications. *arXiv:cond-mat/0007235 , 2* (7).

Pandurangan, G., Raghavan, P., & Upfal, E. (2002). Using PageRank to characterize web structure. *8th Ann. Int. Computing and Combinatorics Conf.*, (pp. 330-339).

Pennock, D. M., Flake, G. W., Lawrence, S., Glover, E. J., & Giles, C. L. (2002). Winners don't take all: Characterizing the competition for links on the web. *PNAS , 99* (8), 5207-5211.

Popescul, A., Flake, G. W., Lawrence, S., Ungar, L. H., & Giles, C. L. (2000). Clustering and Identifying Temporal Trends in Document Databases. *IEEE Advances in Digital Libraries*, (pp. 173-182). Washington, DC.

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., & Parisi, D. (2004). Defining and identifying communities in networks. *PNAS* , 2658-2663.

Van Rijsbergen, C. J. (1979). *Information Retrieval, 2nd edition.* Glasgow: Dept. of Computer Science, University of Glasgow.

Song, C., Havlin, S., & Makse, H. A. (2006). Origins of fractality in the growth of complex networks. *Nature Physics , 2*, 275-283.

Song, C., Havlin, S., & Makse, H. A. (2005). Self-similarity of complex networks. *Nature , 433*, 392-395.

Zhou, W.-J., Wen, J.-R., Ma, W.-Y., & Zang, H.-J. (2002). *A Concentric-Circle Model for Community Mining in Graph Structures.* Redmond: Microsoft Research.