Evolution of Web Graph

Minhyung Kang
Stanford University
Stanford, CA 94305
dankang@stanford.edu

Abstract

In this paper, we study the evolution of structure of the internet. Rather than looking at a single snapshot of the web, which most of previous studies in the field have focused on, we look at the change in the structure across 12 years of data. We select a small subset of the data for an initial exploration. We develop a pipeline to extract the links from raw archive files and hash them to unique links. The links are then ingested to create webgraphs and perform analyses. The networks obtained are composed of few hundred millions of nodes and billions of edges, comparable to networks of many of the earlier work done in field. We note similarities between our observations and what has been reported by other authors, as well as a great discrepancy in component sizes, notably the biggest strongly connected component. We develop a few hypotheses to explain the issue, and perform an experiment to confirm that crawling process could be a cause to the problem. Lastly, we construct domain graphs from our links and study the changes in popular domains across the years.

1. Introduction

If one were to discuss the most influential network structures in the 21st century, many would pick the World Wide Web (WWW). Since its introduction in the 90s, WWW has connected people and information around the world. In its early days the structure of this intricate, rapidly changing network was only imagined, many claiming that it lacked structure whatsoever. After iconic study by Broder et al [5] in 1999 which claimed the network to be of 'bow-tie' structure, numerous studies on the network structure followed for next two decades. While there have been much work done on analyzing snapshots of web graphs, not enough work has focused on temporal changes in the network, possibly due to several reasons. First, there aren't many consistent crawl datasets across a long period of time that are publicly available and can be utilized by researchers to study the evolution of the web. Hence, they are forced to focus on rather limited temporal subset. Secondly, even equipped with such dataset, the size of the data is often on the numbers of terabytes of data per year, posing a computational challenge. Thirdly, changes in network structures are difficult to quantify and explain. A variety of different metrics need to be utilized, and it is unclear if they, once combined, can tell a coherent story regarding the changes in the web.

In this work, we look at evolution of the web from year 2003 to 2014. We utilize crawled web archives for each year, whose size ranges from 1.7 TB to 7.4 TB. Instead of parsing all data, here we focus on a smaller subset of data - notably the years 2003, 2004, 2007 and 2010. We extract link structures, filter valid links, and construct web graphs in yearly snapshots. We provide some general statistics about webgraph of each year, pointing out similarities as well as differences between our data and results reported by previous works in the field. Notably, we observe a huge discrepancy in the structure of web graph when we perform breadth first search to verify the bow-tie structure of the web. We attribute this to be due to crawling process and link extraction, and carry out an experiment to confirm that crawling process can indeed have a huge impact on our result. We then construct domain-level networks, and observe how popular domains have changed over the years.

2. Related Work

The structure of the web has been studied several times in past two decades. Broder et al. [5] investigated an AltaVista crawl of 200 million pages and 1.5 billion links and claimed that the web graph is of a bow-tie structure, composed of several well-defined components. These include: CORE, a strongly connected component; IN component, a set of nodes that can reach CORE but not be reached from it; OUT component, a set of nodes that can be reached by CORE but cannot reach CORE themselves; disconnected component (Tendrils, DISC, and Tube) which have no connection with CORE. They showed that each component is of roughly similar size, and claimed that two separate crawls revealed equivalent insight. Lastly, they observed that degree distributions seemed to follow power laws, all of which

Table 1: Data processing pipeline

Stage	Raw Data	Extract Links	Filter / Hash Links	Create Webgraph	Analyses
Platform	DFS	AUT, PySpark	PySpark	WebGraph (Java)	
Result Format	WARC	Parquet	TSV	.offset, .graph	Statistics
Result Size - 2003	1.7 TB	28 GB	47 GB	5.3 GB	Varied
Result Size - 2004	2.2 TB	21 GB	37 GB	4.4 GB	Varied
Result Size - 2007	3.0 TB	44 GB	71 GB	9.5 GB	Varied
Result Size - 2010	5.4 TB	86 GB	111 GB	14 GB	Varied

Table 2: Size of web archives in terabytes for each year

Year	**Size**	**Year**	**Size**
2003	1.7	**2009**	5.3
2004	2.2	**2010**	5.4
2005	6.1	**2011**	4.6
2006	6.6	**2012**	7.4
2007	3.0	**2013**	5.9
2008	5.0	**2014**	4.1
Total		57 TB	

have been accepted more or less by the academic community.

However, Serrano et al [12] later pointed out that a lot of statistics often used to describe a network, such as degree distribution, rather depend heavily on the crawl methods and may not be a good representative sample of the whole web. Indeed, some described the web graph as a daisy[7], while others described it as a teapot [13]. In a more recent study of the web, Meusel et al [10] showed such dependence of the findings on crawl process, pointing out that proportion of the four components were vastly different from what was proposed by Broder et al. Furthermore they observed that the degree distributions are not power laws, and newly provided distance-based feature measurements of the web.

Several works have also been done on studying the changes in structure of networks, such as development of email-based social networks [8, 9] or Internet routing network [6]. Many different measurements and corresponding algorithms were proposed to characterize changes in topology of networks, including time-varying communities [1], using units of different motifs to describe temporal networks [11], or frequency-based pattern characterization of evolution [3]. To our knowledge, no work has been performed studying temporal structure changes of the internet.

3. Dataset

The dataset that we use is crawled web archive data in WARC (Web Archive) format. The format stores a lot of information regarding the crawl such as payload content and control information that even with compression the file size is massive. The amount of data per year can be found in Table 2. One can quickly note that the amount of data is not consistent across the years. This is due to changes in crawling depth and duration and hence we cannot make estimations regarding growth or diminishing in size of the whole web graph. That is, we cannot make any statements about the whole graph unless we actually know how much of the whole webgraph we have crawled. Yet, it is possible to discuss changes in average statistics such as average in-degree or out-degree, or make statements about well-defined domains that are shared across the years. For example, it might be possible to state that within domain X the number of nodes in a strongly connect components have increased and that its diameter have decreased.

3.1. Data Processing Pipeline

While the raw data is massive in its size, the main piece of information we are interested in is the edges between webpages. Hence, once link extraction is completed, the size of the file we have to handle for our study is far reduced and much more manageable.

Refer to Table 1 for a summary of the processing pipeline and corresponding data sizes. We use open source *Archives Unleashed Toolkit* (AUT) [1], which provides a simple interface built around *Apache Spark* to analyze archives, to parse and extract link data from raw WARC files. Once the links are extracted, we use *PySpark* to filter to valid, unique links (e.g. disregarding links that start with 'mailto:'), and use *xxHash* algorithm [2] to hash the nodes to 64 bit numbers. We then use *WebGraph* package [4] to create webgraphs and perform analyses. While the package was developed more than a decade ago, it still is the most extensive framework to study web graphs with various functionalities and compression techniques that cannot be found with other available tools. Note that after this step, our data has been reduced by almost 4 orders of magnitude, making it much more manageable. The biggest compression mainly comes from two

[1] https://archivesunleashed.org/aut/
[2] https://github.com/Cyan4973/xxHash

steps: the first is link extraction, where we discard all unnecessary information of web pages. Note, that as we filter and hash links, we actually get a file greater in size as we output an uncompressed naive TSV file. Another major reduction in size arises from switch to *WebGraph* framework, which utilizes compression algorithms to provide compact representation.

The whole pipeline, as of the moment, is quite costly. The first link extraction step takes between a few days to a week depending on source size, the hashing step about a day. Once deduplicated links are extracted, webgraph generation and statistics computation can be carried out in hours or in some cases, minutes. Due to this costly extraction process, we initially only look at 4 years of data as noted in Table 1. We believe it to be beneficial to look at a sample of data before trying to make adjustments and optimizations for processing pipeline.

3.2. Framework

As mentioned, we utilize various packages in our pipeline for extraction and analysis, including *AUT, PySpark, JAVA*, and *WebGraph*. We performed our experiments using a cluster provided by InfoLab of Stanford University, which consists of Intel(R) Xeon(R) CPU E7- 4870 server with 80 CPUs.

4. Results

4.1. Degree Distribution

It has been observed in the literature that the in-degree and out-degree distribution of web graph networks tend to follow so called 'power-law' [2]. That is, probability $P(k)$ that a vertex interacts with k other nodes (or in other words, the probability of having k in or out degree) decays as power law, as $P(k) \sim k^{-\gamma}$. However, it was noted in [10] that this is rather a naive observation from a log-log plot, and that in fact many other graphs can portray linear tendency in log-log plot. Hence, we just make two observations here, and make no further statements about the model behind it, though further analysis could try to fit different functions. First, there is resemblance of linear tendency of degree distribution in a log-log plot. The shape of the distribution resembles that of other studies, notably the triangularly linear shape of indegree distribution and slightly concave form of outdegree distribution. Secondly, overall shape of the distribution is rather consistent across the years, though there is slight difference in outdegree distribution. We see higher proportion of nodes with medium sized outdegree (around 10^2) and lower proportion of nodes with low outdegree (less than 10^1). We suspect this might mean the network has become more connected across the years. Indeed, that is what we observe in average degree, which we discuss in the next section.

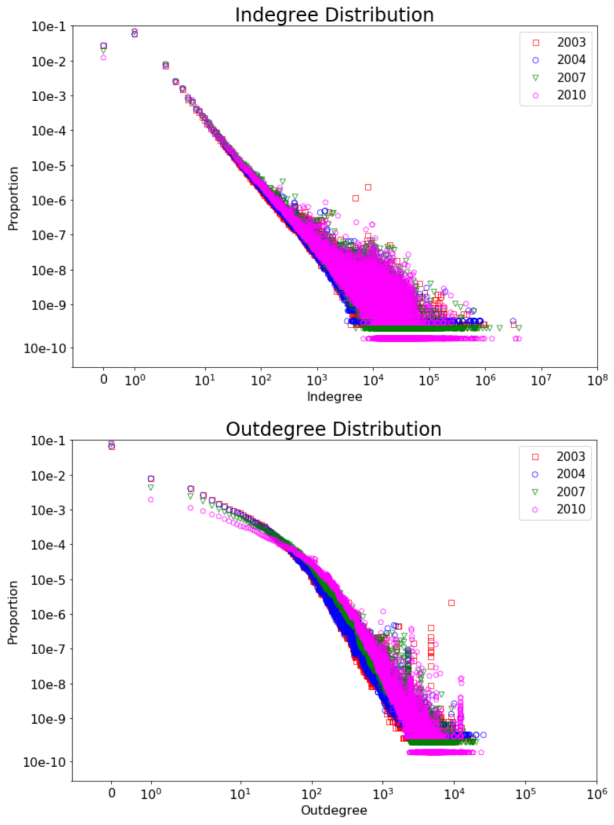

Figure 1: In-degree and Out-degree distribution of three years. Note that they are both log-log plots, where x-axis is the size of in/out degree, and y-axis is the proportion of the whole network.

4.2. Summary of Statistics

Summary of statistics of 4 years of data can be observed in Table 3. Along with our collected data, we include data from [5, 7, 10]. Note that some numbers are not reported, such as proportion of dangling nodes, and hence we leave them as blank.Some other numbers were indirectly reported, such as stating that weakly connected component was greater than 90% of whole graph by [7], or with different precisions. We can make several observations regarding our data and compare it with previous results.

The size of the graphs are comparable to some of the earlier studies done on networks. We see that number of nodes range from 300 million to 500 million, and the number of directed arcs between 1.5 billion and 3.3 billion. This is comparable to 200 million and 360 million nodes, respectively, and 1.5 billion edges in [5] and [7]. This number, however, is much smaller than 3.563 billion nodes and 129 billion edges used in [10]. It is possible that the bigger graphs in our dataset have more than a billlion nodes, but we suspect it unlikely that any single year we have will have

Table 3: Summary of statistics of our results and three previous papers. All numbers are in millions except for average degree, and the number in parentheses are portion of all nodes, in percentage.

	2003	**2004**	**2007**	**2010**	**Broder00**	**Donato05**	**Meusel14**
# Nodes	376	305	491	677	203	185	3,563
# Arcs	2,071	1,602	3,274	5,198	1,466	1,500	128,736
Avg degree	5.50	5.25	6.67	7.68	7.5	-	36.8
Max SCC	9.2 (2.4)	8.5 (2.8)	17.2 (3.5)	22.0 (3.3)	56.4 (27.7)	44.7 (32.9)	1,827.5 (51.3)
Max WCC	357.5 (94.9)	291.1 (95.4)	478.3 (97.4)	661.7 (97.8)	186 (91.6)	166.5 (90+)	3,349.2 (94)
IN	47.5 (12.6)	37.8 (12.4)	48.5 (9.9)	47.5 (7.0)	43.3 (21.3)	14.4 (10.6)	1,138.9 (32.0)
OUT	81.6 (21.6)	69.2 (22.7)	175.3 (35.7)	295.5 (43.7)	43.2 (21.2)	53.3 (39.3)	215.4 (6.1)
Dangling	249.1 (66.1)	202.6 (66.4)	366.1 (74.6)	557.0 (82.3)	-	-	-

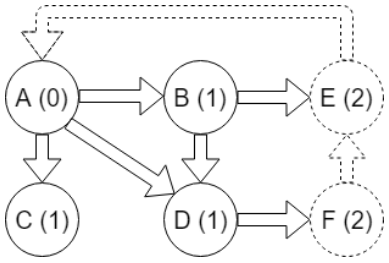

Figure 2: Toy crawling process with seed node A. Alphabets are node ids and number in parentheses are BFS depth. Dotted line indicates components (nodes and edges) that exist but never explored by our crawler, while regular lines are components that were explored by our crawler.

as big of a graph as in [10].

The average degree of directed graph is between 5.2 and 7.7 and increases across the years. This is aligned with 7.5 as reported by [5], but much smaller than 36.8 observed by [10]. Again, it will be interesting to see how the measurements in networks from later years change, but we suspect this difference to be partly due to link construction which we discuss in the next section.

4.3. Components of Webgraph

As noted in Section 2, [5] devised an experiment to analyze the structure of a web graph, which has since been repeated many times across the years. After identifying the biggest strongly connected component (SCC), one can perform a breadth first search (BFS) from nodes within them. By subtracting size of SCC from number of reachable nodes, we have an estimate of the OUT component. We can perform a BFS on a reversed graph in a similar manner, and get estimate of the IN component. We perform the experiments on webgraph we constructed, the result of which is in Table 3.

First observation, which is somewhat concerning, is how disjoint the graph seems. The biggest strongly connected component only takes up about 2.3 to 3.5 % of all nodes, which is far smaller than numbers reported by other papers, which range from 25% to 50%. On the other hand, the size of weakly connected component takes up around 95% of the whole graph, which aligns with numbers reported from previous works. We also observer a very large number of dangling nodes, with 65 to 82% of all nodes being terminal nodes. Unfortunately, we could not find comparable statistics in previous works.

In terms of other components of the graph, we already see vastly different tendencies between previous works. [5] claimed that all components were of similar size, [7] said IN component was smaller than SCC and OUT, while [10] said the biggest SCC took up half of the graph IN component about 30%, and OUT component to be only around 6%. In comparison, we see that our OUT component to be much larger than IN component, the proportion changing from 1:2 to 1:6 across the years. So where could these differences arise from? We believe there are two possible explanations.

The first explanation is the crawling process. It has been noted in [10] that the webgraph structure greatly depends on the crawling process. While we do not know of the crawling process that was used to generate our data, we could hypothesize the following. Just as any other crawler, we can assume that ours were kicked off from various seeds in a BFS manner. Now, assume that BFS is set to terminate after 10 steps. Then there are webpages that are visited in the 10th level. If we extract the links from these webpages, we have links to other websites which, if we do not already have them in our visited pages, we never visit in that BFS run. In such case, all nodes that would have been visited in 11th level end up being dangling pages. For a simple example, refer to Figure 2, and assume we perform BFS of 1 step from node A. Then we would visit B, C, D, and our current link construction scheme would include links to E and F as well, even though we never visit those nodes. Moreover, E and F will be falsely identified as dangling nodes. Hence, This might explain our great number of danging nodes, as well as our small proportion of greatest SCC. We test this hypothesis in Section 4.4.

Table 4: Changes in extraction due to filtering process

Original	Nodes	A, B, C, D, *E, F*
	Links	AB, AC, AD, BD, *BE, DF*
Filtered	Nodes	A, B, C, D
	Links	AB, AC, AD, BD

Table 5: Statistics of year 2003 before and after filtering out the destination pages. All number in million except for average degree and BFS. Number in parentheses is proportion of all nodes in percentage.

	2003	**2003 (filtered)**
# Nodes	376	99
# Arcs	2,071	997
Avg degree	5.50	10.02
Max SCC	9.2 (2.43)	9.2 (9.24)
Max WCC	357.5 (94.86)	96 (96.13)
IN	47.5 (12.61)	47.5 (47.76)
OUT	81.6 (21.64)	4.97 (5.00)
Dangling	249.1 (66.09)	8.9 (8.98)
BFS depth	$756 \sim 775$	$763 \sim 772$
BFS depth (reversed)	$130 \sim 139$	$130 \sim 141$

Table 6: Statistics of created domain graph

	2003	**2004**	**2007**	**2010**
# Nodes	7.69 M	7.24 M	12.16 M	9.25 M
# Arcs	41.62 M	34.77 M	34.54 M	25.33 M
Avg degree	5.41	4.79	2.80	2.74
Max SCC	72.59 K (0.94)	49.78 K (0.69)	40.21 K (0.33)	22.78 K (0.25)
Max WCC	7.68 M (99.99)	7.24 M (99.99)	12.16 (99.99)	9.24 (99.99)
BFS depth	7	$5 \sim 6$	$5 \sim 6$	$5 \sim 6$
BFS depth (reversed)	8	$6 \sim 8$	$6 \sim 7$	$6 \sim 7$

The second possibility is the link construction step. While [5, 7] do not explicitly mention the link construction process, [10] explains that they constructed links by using both a and $link$ HTML elements, as well as redirects contained in HTTP header. In comparison, the extraction package we use, *AUT*, extracts only the a elements. Hence, our process theoretically covers fewer links per page. This explains the great average degree in [10], and can help explain the large portion of biggest SCC.

4.4. Filtering Destinations

To test the hypothesis that the peculiar structure of our webgraph might be due to crawling process, we carry out the following experiment. From the archive file of 2003, we extract all the urls of pages we have visited and downloaded. Then from the links we have from previous extraction process, we only filter the links whose destination pages are in visited URLs. This is effectively reducing our crawling level by 1, and would reduce large number of edges. However, in this way, we can ensure that we only look at links between pages that we have visited. The difference in our link extraction, using our example from Figure 2 again, is demonstrated in Table 4.

The statistics for graph constructed after filtering step can be found in Table 5. There are many easily noticeable differences. First, the number of nodes were reduced to almost quarter of its original size, and number of links were halved. Average degree has doubled, and we see that the proportion of max SCC size is now of around 9%, which is still smaller than what was reported in other works, but is more reasonable. Note, the size of max SCC does not change, which is reasonable; as all edges in SCC, by definition, have both source and destination nodes explored by our crawler, they should not be removed. We see that it was mostly the falsely danging nodes that were vastly reduced, as number of dangling nodes decreased by 240 million. Such change also greatly reduced the size of OUT component. The IN component did not change in size, and hence we now see a disproportionately big IN component. If we were to have deeper levels of BFS, or extract more links from *links* HTML elements and redirection, it is possible that we have more nodes in IN component become part of the SCC.

4.5. Domain Graph

Next, we construct a new graph by extracting the links from domain level. For each site-level link, we extract the domain, and group them by links between domains. Note that here, we do not weigh the domain link by number of links between websites of each domain. Then we construct a webgraph using these domain links. The result is summarized in Table 6.

We observe, as expected, that the size of the graph has greatly reduced in size, by almost 2 orders of magnitude. The number of nodes are now in orders of millions, and the number of arcs in order of tens of millions. The average degree actually decrease over the years, even though we saw it increase in page-level graphs. We see the strongly connected component to be even smaller proportion of all nodes, while weakly connected component basically takes up the whole graph. It is also worth noting that the depth of BFS is between 6 and 8, much smaller compared to that of page level graphs.

We can also look at the most popular source and destination domains by grouping our links by their domains. The source domains with most out-links across years is

Table 7: Top 10 source domains across years

	2003	2004	2007	2010
1	da.ru	tripod.lycos.com	tripod.lycos.com	tripod.lycos.com
2	directory.google.com	webbound.com	ljudmila.org	gy.com
3	anywho.com	cyber.law.harvard.edu	google.com	mybloglog.com
4	dominion-web.com	anywho.com	paginegialle.it	directory.google.com
5	tollfree.att.net	tollfree.att.net	choicehotels.com	google.com
6	att.net	salon.com	gy.com	at-la.com
7	suchmaschine.com	att.net	directory.google.com	mister-wong.de
8	dmoz.org	directory.google.com	educationplanet.com	hotsheet.com
9	newhoo.com	gy.com	hotsheet.com	rhymeswithright.mu.nu
10	asia.dir.yahoo.com	excite.co.uk	stumbleupon.com	choicehotels.com

Table 8: Top 10 destination domains across years

	2003	2004	2007	2010
1	adobe.com	adobe.com	adobe.com	facebook.com
2	microsoft.com	microsoft.com	microsoft.com	youtube.com
3	geocities.com	geocities.com	google.com	twitter.com
4	amazon.com	amazon.com	geocities.com	adobe.com
5	members.aol.com	google.com	amazon.com	google.com
6	google.com	members.aol.com	en.wikipedia.org	en.wikipedia.org
7	yahoo.com	yahoo.com	apple.com	microsoft.com
8	cnn.com	cnn.com	nytimes.com	amazon.com
9	apple.com	nytimes.com	cnn.com	nytimes.com
10	nytimes.com	apple.com	members.aol.com	maps.google.com

listed in Table 7, and similar information regarding destination domains is listed in Table 8. We observe that many of top source domains are web hosting websites, such as `tripod.lycos.com`, `gy.com`, or `da.ru`. As it is additional cost to purchase a custom domain, it is understandable that many websites would be under the tripod domain. We also observe portal and directory websites with many links to other businesses, including `directory.google.com`, `paginegialle.it`, `choicehotels.com`, and `hotsheet.com`. There are also some potential spam/hacker websites, such as `ljudmila.org`.

When we look at domains with most in-links in Table 8, we can spot many familiar names and make various observations. First, we see a curiously large number of links to Adobe. After looking at the links themselves, we observe that they are related to Acrobat Reader or Flash, both very widely used products of Adobe and very often embedded in websites. Similar observation can be made about Microsoft, though no specific product seems to be in play. We also see decline of Yahoo over the years. For example `geocities.com`, which was a popular web hosting service offered by Yahoo, was discontinued in 2009; hence, we see it being in top 10 list until 2007 but disappear in 2010. Of course, we observe rise of great social media and user content websites in 2010, such as `facebook.com`, `youtube.com`, and `twitter.com`/ It is also worth noting that there are consistently many links to news websites, such as `nytimes.com` or `cnn.com`.

5. Conclusion

In this work, we looked at 4 years of data from a dataset of 12 years of crawled web archives. We have developed a pipeline that utilizes multiple frameworks to extract links, hash and deduplicate them, generate webgraphs, and perform analyses. We obtained networks of few hundred million nodes and billions of edges, and perform experiments as carried out in other works in the field. We noted big discrepancies in the structure of the web and proposed a few hypotheses to explain them. We carried out an experiment and confirmed that one of them contributes to the differences. We created domain graphs and observed how popular domains have changed across the years.

There are obviously many future work that can be done. One would be looking into bridging the gap between our results and other reported results. A possible first step would be changing our extraction procedure to extract redirections and *link* HTML components as links. Another interesting

question would be that of optimizing the pipeline to speed up the processing. We could work on rewriting the extraction code, which has been the biggest throttle in our pipeline, to maximize parallelism. We could also look at how tasks are carried out in Spark, and optimize the calls and parameters. Then we can utilize our whole dataset and actually start looking at the overall trend across 12 years of data.

References

[1] M. Araujo, S. Papadimitriou, S. Günnemann, C. Faloutsos, P. Basu, A. Swami, E. E. Papalexakis, e. V. S. Koutra, Danai", T. B. Ho, Z.-H. Zhou, A. L. P. Chen, and H.-Y. Kao. Com2: Fast automatic discovery of temporal ('comet') communities. pages 271–283, 2014. 2

[2] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. 3

[3] M. Berlingerio, F. Bonchi, B. Bringmann, e. W. Gionis, Aristides", M. Grobelnik, D. Mladenić, and J. Shawe-Taylor. Mining graph evolution rules. pages 115–130, 2009. 2

[4] P. Boldi and S. Vigna. The webgraph framework i: Compression techniques. *Proceedings of the Thirteenth World-Wide Web Conferences*, pages 595–601, 2004. 2

[5] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Proceedings of the 9th World-Wide Web Conferences on Computer Networks: the International Journal of Computer and Telecommunications Networking Archive*, pages 309–320, 2000. 1, 3, 4, 5

[6] J. Chan, J. Bailey, and C. Leckie. Discovering correlated spatio-temporal changes in evolving graphs. *Knowledge and Information Systems*, pages 53–96, 2008. 2

[7] D. Donato, S. Leonardi, S. Millozzi, and P. Tsaparas. Mining the inner structure of the web graph. *WebDB*, 2005. 2, 3, 4, 5

[8] K. Juszczyszyn, K. Musial, P. Kazienko, and B. Gabrys. Temporal changes in local topology of an email-based social network. *Computing and Informatics*, 28:763–779, 2009. 2

[9] K. Juszczyszyn, K. Musial, A. Musial, and P. Brdka. Molecular dynamics modelling of the temporal changes in complex networks. *Evolutionary Computation*, 2009. 2

[10] R. Meusel, C. B. S. Vigna, and O. Lehmberg. Graph structure in the web — revisited: A trick of the heavy tail. *Proceedings of the 23rd International Conference on World Wide Web*, pages 427–432, 2014. 2, 3, 4, 5

[11] A. Paranjape, A. R. Benson, and J. Leskovec. Motifs in temporal networks. *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 601–610, 2017. 2

[12] M. Serrano, A. Maguitman, M. Bogu, S. Fortunato, and A. Vespignani. Decoding the structure of the www: A comparative analysis of web crawls. *ACM Transactions on the Web (TWEB)*, 2007. 2

[13] J. Zhu, T. Meng, Z. Xie, G. Li, and X. Li. A teapot graph and its hierarchical structure of the chinese web. *Proceedings of the 17th International Conference on World Wide Web*, pages 309–320, 2008. 2