Does the network of technologies used by developers represent a power-law structure?

Arjun Khurana
UNIVERSITY COLLEGE LONDON
6th January 2019

COMP0123: Complex Networks and Web
Coursework

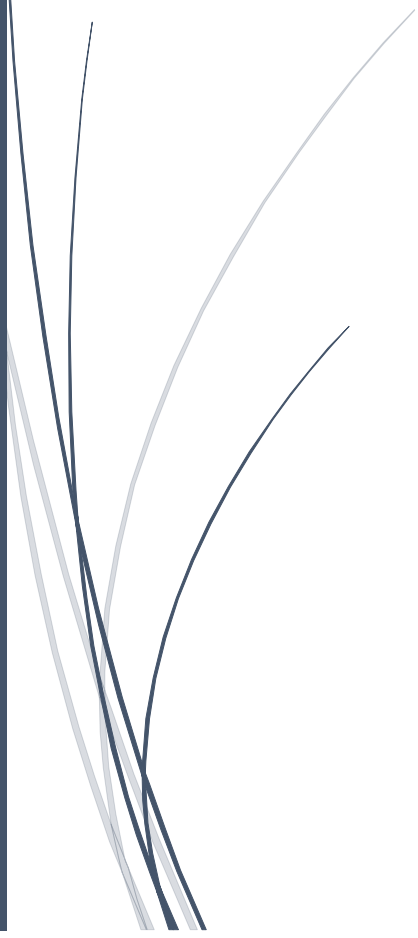

Table of Contents

1. Abstract

The tags used by a developer to describe themselves is a great indicator of their past experiences with technology. Data acquired from Stack Overflow, containing tags used to describe the technologies (i.e. frameworks and programming languages) used by developers is analysed to suggest that the network of technologies is a power law distribution network. Network was analysed via database queries and illustrations to confirm that this network follows a small-world network structure with a dissociative mixing pattern. Comparing the network structure to the AS (Autonomous System) network suggests that the structure of both networks is similar in that they are both scale-free networks.

2. Introduction

2.1 Problem and Significance

Very little research is available into the behaviour of developers. Even fewer relevant material relating to developers' decision-making regarding the technologies they use. Understanding the structure of past experiences with technologies can help forecast future trends, and more importantly, the level of success of emerging technologies.

There has not been much research into the decisions of developers regarding the frameworks and programming languages they use, hence this investigation is a novel perspective into the network of technologies used by developers. The results will also show the significance of certain nodes and edges in this network and how similar or different the structure of this network is to other common networks. The investigation might show that the seemingly random choice of technologies used actually follows a common network structure.

This paper aims to answer the following core questions: What is the inherent structure of the network of technologies used by developers? Does this network follow a power law structure such as the AS network? What do the network properties suggest regarding its structure?

2.2 Dataset

The original data is provided by Stack Overflow's Developer Stories' content. However, this investigation will be focusing on a subset of that data dump. This dataset is acquired by Kaggle (Stack Overflow, 2017) and collated by a data scientist working at Stack Overflow. The dataset represents the tags used by developers when describing their experiences and skills. Additionally, the dataset consists of links (or edges) between nodes (technology tags) to demonstrate the use of two individual tags within the same developer story.

Extracted the core data and carried out several queries in Microsoft Excel to analyse the dataset and obtain the core information required for the illustrations of the network. The visualisations were generated via Gephi.

2.3 Key Results

The network shows a structure similar to that of a scale-free network. Network connectivity and properties, as well as visualisations of the networks, were developed, targeting different attributes to test different perspectives. Comparing an ordered and a random set of data to evaluate the relationship between node size (tag popularity) and node position. Both graphs indicate a power-law-like behaviour such that most nodes have a relatively small node size, whereas a few nodes have a medium size and only a handful of nodes have a very large size. Later, this is examined in further.

The network of technologies used by developers does display a power law distribution, additionally, it has similar network properties to some other power-law networks such as the Internet network. This is explored further in the 'Discussion' section.

2.4 Evaluation Process

Compared the visualisations, graphs and network properties to those of well-established power-law network models, such as the AS (power-law) network. The comparison was used to measure the extent to which the network analysed in this study resembles existing complex networks. Additionally, network properties were used to identify the type of structures which are followed by the network.

3. Background

3.1 Complex Networks

Networks can be of varying types, including weighted, unweighted, directed, undirected, etc. (COMP0123: Complex Networks and Web). Additionally, networks can have many different structures, for example, 'Regular', 'Random' and 'Small-World Networks'. Each of these network structures mentioned has distinct properties differing from each other, however, even amongst each individual structure these properties can vary, resulting in an unimaginable number of potential networks.

Complex Networks consists of an even wider and more diverse spectrum of properties and distinctions from 'Random' and 'Regular' networks. However, the exploration of power-law networks has been an incredible source of new and interesting discoveries, especially for explaining the structure of common networks such as the network of academic collaborations (Newman, 2004).

There are many different properties a network may consist of, however, the key properties to be examined for a complex network are its degree distribution, clustering coefficient, average length, size of the biggest (giant) component (MIT, 2009) and network diameter.

3.2 Scale-Free and Power-Law Networks

Scale-Free networks are essentially networks which display a power-law degree distribution, such that there is a presence of a few highly connected nodes and many nodes which only have a few links.

Power law networks have two distinct properties, *"Very many nodes with only a few links"* and *"A few hubs with a large number of links"* (COMP0123: Complex Networks and Web). This is different to a Bell curve model which does not have any nodes with a significantly high number of links and most nodes have a similar degree (links).

3.3 Random, Regular and Small-World Networks

Random networks tend to have the following two properties, low clustering and a small average path length. In essence, this implies that random networks have a low edge locality and require fewer hops on average to travel between nodes. In contrast, regular networks tend to have high clustering and a large diameter.

Small-World networks, however, are a mix of both random and regular networks such that they tend to have a low diameter and high clustering (COMP0123: Complex Networks and Web). Small-world networks also have a very small average path lengths relative to their network size, roughly around 2 or 3 hops (Cancho & Sole, 2001).

4. Literature Survey

4.1 What are developers talking about? An analysis of topics and trends in Stack Overflow.

Trends relating to developers' behaviour is a great way to understand their past experiences with technology, as well as, for forecasting which technologies they are likely to use in the future. Barua, Thomas, & Hassan (2012) investigates this area in detail. The primary aim of this research is to find topics and trends in Stack Overflow using a *"semi-automatic methodology to analyse the textual content of Stack Overflow discussions"*. This methodology has been used and described in other research papers before. Latent Dirichlet Allocation (LDA) is the algorithm used for this study, in essence, LDA is a *"generative topic model extractor"* (Algorithmia, 2016). After extracting the necessary information trends were identified and analysed.

The results suggest that developers have a wide range of interests when interacting on Stack Overflow. This includes version control queries, job support, syntax queries for specific programming languages, etc. A key discovery was that Web Development, Mobile Development, MySQL and Git are the most popular technologies. Particularly the discussions regarding jQuery, Android has been increasing significantly and Java has been constantly occurring in discussions. In contrast, the popularity of the .NET framework has been decreasing.

The paper by Barua, et al. is relevant to the overarching investigation of this paper because it analyses Stack Overflow, which will be the source for the dataset used for this study. Additionally, the target of both investigations is related to trends regarding developers' behaviour. Furthermore, both studies involve analysing relationships between developer activity.

The research paper briefly explains the current state of the art as well, relating to user activity and social interactions in online software community-based websites. The paper explains that the *"unstructured nature of the posts, which are written in natural language, prohibits most conventional data mining techniques from being effective"* (Hassan, 2008). Additionally, the findings from this paper can help improve existing tools for analysing large repositories, as well as, supporting the development of tools better suited for future trends.

The optimal parameters of the LDA algorithm have not been identified, therefore, there is a risk that the findings could be non-optimal and even invalid for the larger community. Secondly, the greatest potential issue is temporary or even false popularity of topics. Since the research only reviews discussions from a set time period some topics might just be trending temporarily during the time of the investigation. Therefore, the study might fail to represent overall and accurate topic trends.

4.2 A Topological Analysis of The Open Source Software Development Community.

The Open Source Software (OSS) community is a great indicator of how developers behave and the type of technologies they use. Xu, Gao, Christley, & Madey (Jan, 2005) *"perform a quantitative analysis of Open Source Software developers by studying the entire development community at SourceForge."* The paper looks at collaborations amongst members, as well as, the effects of different types of developers on the OSS community.

The results suggest that the development network displays scale-free and small world properties. Therefore, the paper might present an alternative view of developer activity to the findings of this paper. In particular, the literature explains the SourceForge developer network has a small diameter and a high clustering coefficient. Another key result is that co-developers and active users with a weak link to the community might actually be a significant aspect for OSS development.

The research is relevant because it analyses the different types of core network properties defining a network. Additionally, the paper focuses on developers and their experiences/projects which is the core target for this investigation as well. Finally, both papers' dataset would be from similar developer-based communities.

The OSS community has made a significant impact on widely used technologies such as Linux, Apache, etc. *"The phenomenon of OSS success is not yet fully understood"*, therefore, investigating this field and explaining the reasons for this success and efficiency can help explain the underlying social structure topology of the wider developer community.

Some of the limitations of this paper are that it does not provide any visualisations of the network itself aside from the power law graphs, therefore lacking a more comprehensive presentation and explanation of the entire topology. Additionally, the paper only looks at projects on one service (SourceForge), ignoring GitHub which is one of the largest sources of open-source software projects. This omits a significant portion of OSS projects.

4.3 Popularity, Interoperability, and Impact of Programming Languages in 100,000 Open Source Projects.

Investigating the popularity of programming languages is going to be a key aspect of this investigation. Additionally, understanding the relationships between these languages will help generate the network of technologies. Bissyande, Thung, Lo, Jiang, & Reveillere (July, 2013) aim to investigate a large number of open source projects available on GitHub to assess *"the "popularity", "interoperability" and "impact" of various languages measured in different ways"*.

The results of the study suggest that C and JavaScript are amongst the most commonly used technologies in open-source projects on GitHub. Additionally, the paper explains why many languages such as C++, Java, Objective-C and C# interoperate incredibly well with C, due to the fact that these languages are derived from C itself. Finally, the investigation finds that C and C++ are the most compatible programming languages in terms of interoperability.

The investigation is incredibly relevant to this study because both papers look at the popularity of programming languages and the connections (interoperability) between them. Additionally, the findings from the study can help support or contradicts the results of this study.

The amount of research attempting to explain the *"popularity"*, *"interoperability"*, and *"impact"* of technologies, in general, is lacking, especially for programming languages. Thus, the paper introduces some novel discoveries. Additionally, the result can help identify future trends within the OSS community.

The paper is lacking in certain areas, primarily this is the limited dataset examined. Only 30 technologies are tested, with technologies related to web development missing from the dataset. Therefore, the study is missing a significant portion of relevant information. Finally, the dataset does not necessarily cover a diverse range of developer backgrounds.

5. Methodology

5.1 Data Preparation

Firstly, the data was acquired from Kaggle (Stack Overflow, 2017). The publisher presented the data within two tables. One dataset represented the individual tags, i.e. nodes, and their respected popularity, i.e. node size. The second dataset represented the links between each node, such that if two tags were used together there would be a link between them as this represented the connectivity of the entire network. The second dataset also includes a field to represent the value of an edge between two nodes, such that if two tags are commonly used together the edge between them would have a higher value than an edge between two nodes which are not as frequently occurring together.

After acquiring the dataset from the online data repository, the next step was to collate the necessary information from both tables into a master table. This table would contain the key information relating to the network, including network properties, such as degree

distribution. Slight sanitisation of the dataset was required as each link was included twice even though the edges were undirected. One of the key aspects was the degree of each node. To calculate this from the original dataset, a simple database query "$COUNTIF(links! A: A, A11)$" was run which finds the number of occurrences of each tag within the links table, counting the number of unique instances of each node within the links' table. Each occurrence of a node within this table would represent an edge with another node, thus, the greater the number of occurrences the greater the links connected to a particular node, and so, the higher its degree (k). Additionally, counting frequency of each degree (N_k) was carried out by another database query "$COUNTIF(collated\_dataset! C: C, A5)$". This was done to help with calculating the probability of each degree occurring (P(k)). $P(k) = Nk/N$, where N = number of nodes in the network. Key note to keep in mind, the queries would change the cell references as appropriate for different nodes.

Another table containing randomised dataset was also created for the node size-node position graph by assigning a random number to each node and then sorting the table in ascending order by the random value column. This reshuffled the entire table, thus, listing the table entries in a random order.

5.2 Methods

Used excel and database query to calculate and measure the necessary information and then create the graphs. For the power law graph (figure 1) and the node size-position graphs (figures 2 and 3), Excel provided built-in mechanisms which made this process incredibly quick and effective.

5.3 Algorithm

Simple counting algorithm was used to obtain the number of times a unique tag occurs within table two (for the links between nodes). For this only one of the two columns were used because the edges were essentially undirected, however, they were duplicated in the table. For example, a link from *".net"* to *".asp.net*" exists, whereas, a link from *".asp.net*" to *".net"* also exists in the dataset. The edge value for both these links was the same because the edge is supposed to be undirected, however, the dataset fails to distinguish this. Thus, when necessary this was dealt with appropriately.

5.4 Tools

Microsoft Excel was used to carry out the dataset queries, including collating the entire dataset together and sifting through it to acquire any necessary information. Additionally, this was used to create the degree distribution graph, depicting the k-P(k) representation of the network and the node size structure graphs.

For the visualisations of the dataset, Gephi was used. Multiple variations of the dataset were created, each focussing on a unique node property. Three illustrations were created to alter visible node size and colour based on three different attributes. The first graph (figure 4) involved calibrating the node display size according to its degree. Secondly, the node size was proportional to the node's betweenness centrality (figure 5). The betweenness

centrality is essentially a measure of how often a node lies in the shortest path of other nodes. Finally (figure 6), the node's visual properties were based on its node size value, acquired from the original dataset.

Another visualisation was created to represent the network with the nodes' assigned group (figure 7). This information was included in the original dataset, such that, tags which have a similar focus are grouped together.

Visualisations provided a different perspective on how the network behaves when the target attribute changes. By focussing on multiple different node attributes, the visualisations as a whole provide an in-depth view of the complete network structure, including its degree distribution, clustering and component connectivity.

6. Results

Some core network properties (measured by the network visualisation analysis) of this network are as follows:

- Average Degree = 4.261
- Network Diameter = 10
- Average Clustering Coefficient = 0.615
- Average Path Length = 4.5

6.1 Degree Distribution

To create the power law graph, after collating the necessary data, Microsoft Excel was used to create a scatter graph to represent the degree-probability (k-P(k)) relation. Following this, a trend line depicted that the network seemed to follow a power-law relation. Such that, with an increasing degree, the probability of a node with the corresponding degree level decreased. Most importantly, the trend line seemed to become more horizontal near the bottom end of the graph, this characteristic is a trademark of a power-law network. The gradient of the trend line is 0.9131.

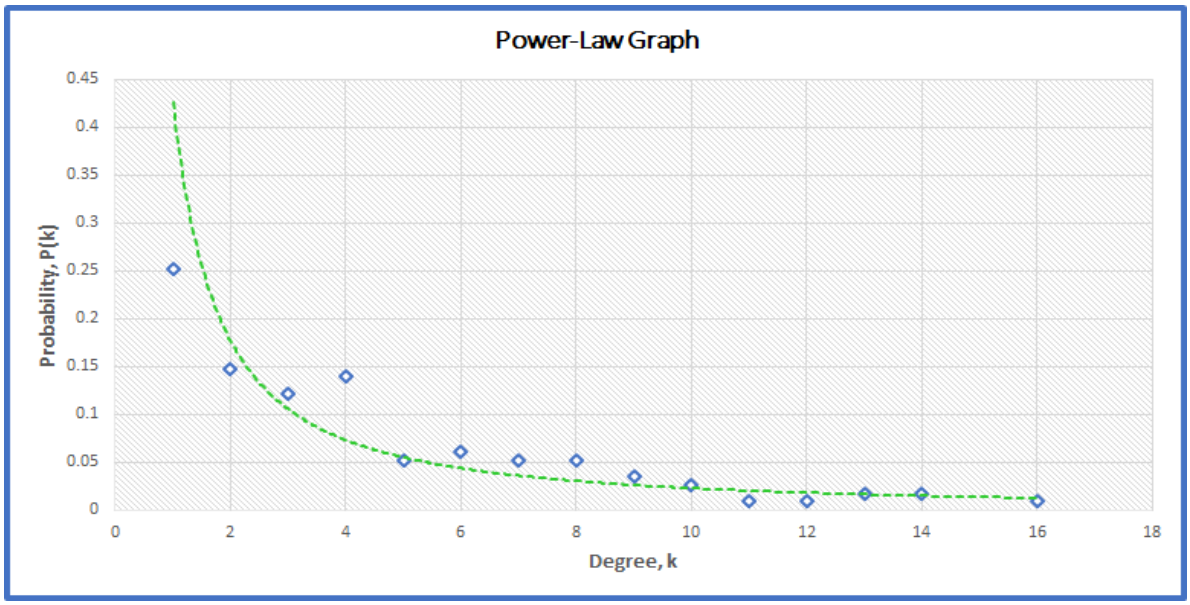

Figure 1: Graph representing the degree distribution of the network, showing a power-law relation between degree, k, and the probability of degree, P(k).

Since the number of tags available to analyse was only 115, the dataset is relatively small. Additionally, since most developers tend to focus on specific areas of computer science it is understandable that the maximum degree would be smaller than in other power-law networks such as the network of Autonomous Systems. Therefore, the raw data points were used instead of a typical log-log graph as the graph would be slightly incomprehensible for analysis otherwise.

6.2 Node Size and Tag Popularity Analysis

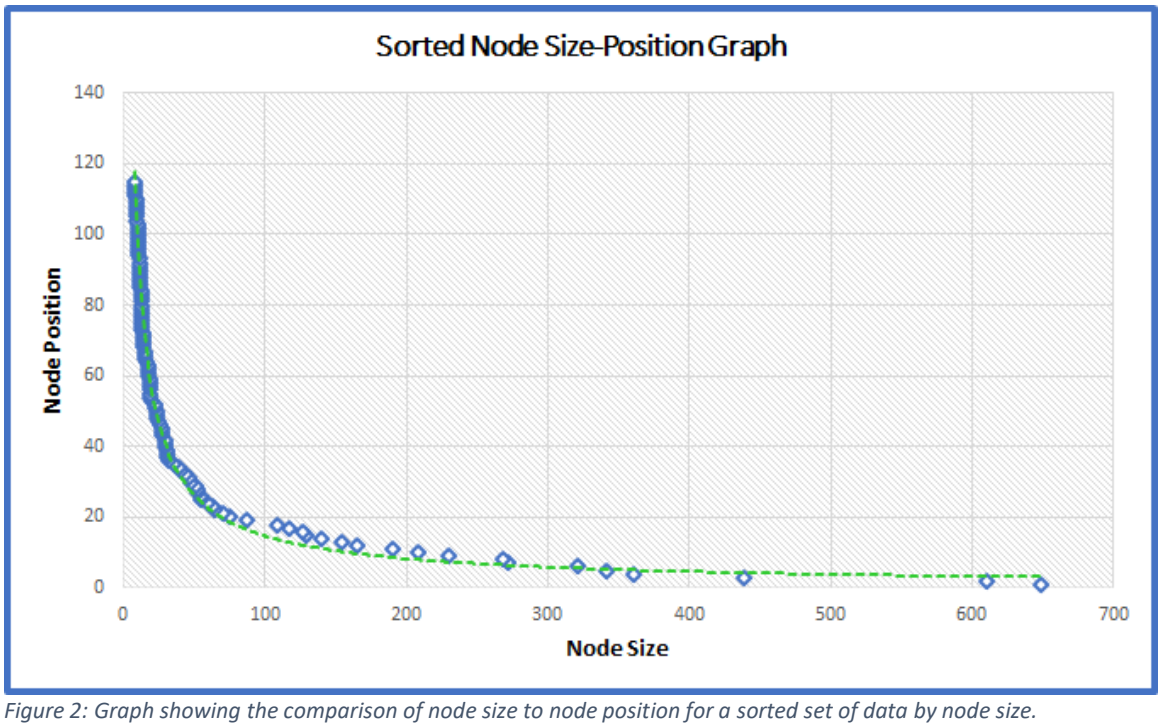

Figure 2: Graph showing the comparison of node size to node position for a sorted set of data by node size.

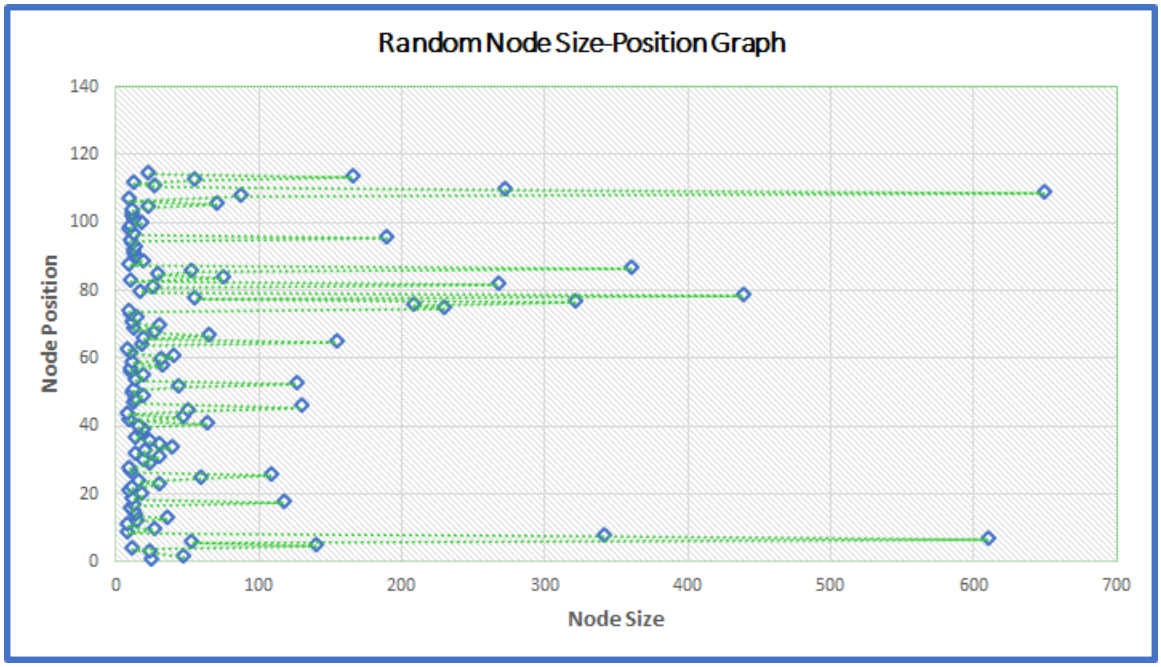

Figure 3: Graph showing the comparison of node size to node position for a randomised set of data.

The graphs above figures 2 and 3 represent the same dataset, however, the first is when the dataset was first ordered in descending order by node size and the second dataset is after randomising the data entries. This was to check whether the ordering of the nodes in the table would affect the findings. Evidently, both graphs illustrate the same conclusion regarding node size/popularity.

The node size – node position graph evidently shows that most nodes have a node size below 100. In fact, around two-thirds of the nodes have a node size of roughly 40. In contrast, less than 5% of the nodes have a size of greater than 300. Furthermore, only two of the 115 nodes have a node size greater than 600. This clearly shows that even the size of nodes within this network follow a scale-free structure. Since node size represents the popularity of a tag, i.e. how often a tag is used within developers' stories, this result indicates that some tags significantly outweigh all the other technologies.

JavaScript has the highest node size, hence, the greatest popularity amongst developers, followed closely by Java. Alternatively, tags with the lowest node size include Drupal, LINQ and Vue.js, and so, these technologies are comparatively used quite rarely. The finding regarding JavaScript being the most popular technology used by developers is supported by the findings in the paper by Bissyande, Thung, et al. (July, 2013). Additionally, since LINQ and Drupal are comparatively not very common technologies it makes sense that their usage is not as wide as some of the other technologies. Hence this result can be confirmed to be valid and a good representative of the wider developer community.

6.3 Network Connectivity

To ascertain the network connectivity and properties, multiple graphs were made. Each focusing on different attributes. In the following visualisations, the size of an edge (link) is proportional to its link *'value'* as provided in the original dataset. This link value is a

representation of the link strength, i.e. tags which are used together more frequently have a greater link value.

6.3.1 Node Degree Analysis

The figure below (figure 4) shows the network when focussing on node degree. It is evident that technologies related to web development and Microsoft Corporation technologies have the greatest degree on average. This can be understood by the incredible interoperability of web technology with other no-web technologies and the fact that Microsoft produces a very diverse set of commonly used technologies. Additionally, jQuery, JavaScript, C# and AngularJS are generally combined with other technologies for a project, thus, explaining their high degree value. Hence both these groups would have many links with other groups.

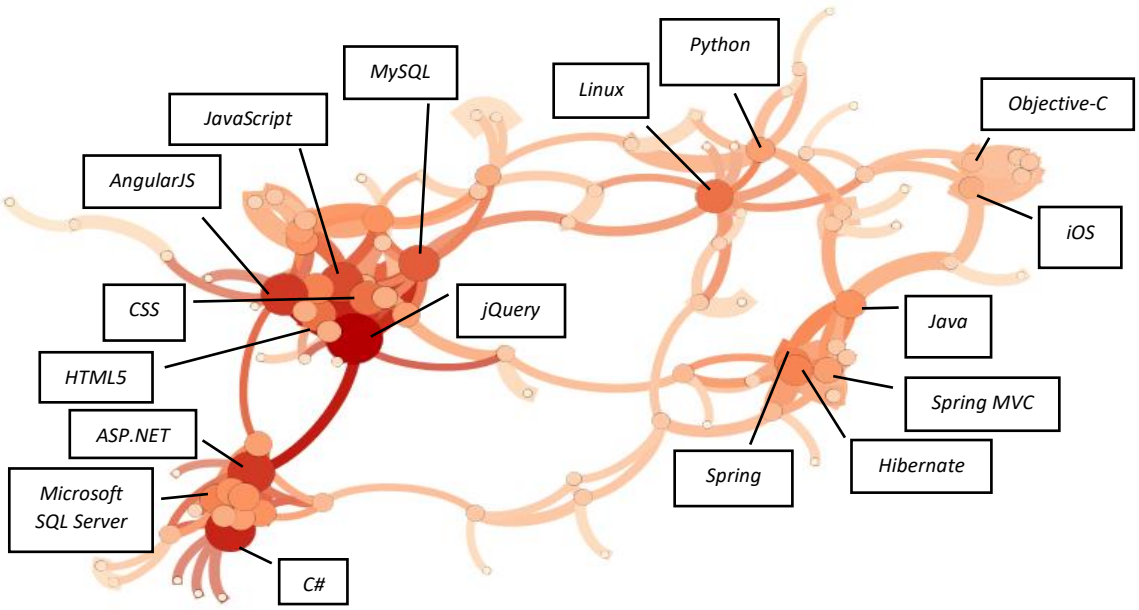

Figure 4: Dataset visualisation, focussing on node degree. Node size is proportional to its degree.

6.3.2 Node Betweenness Centrality Analysis

The figure below (figure 5) shows the network when focussing on node betweenness centrality value. Web technologies such as jQuery and ASP.NET, and database and server related technology such as MySQL and Apache are very significant when it comes to their presence in the shortest paths. This is likely due to the core responsibility these technologies are associated with within a project. Additionally, it is understandable that Linux and Python are also significant in this graph because an operating system and programming language would play vital roles in developing software.

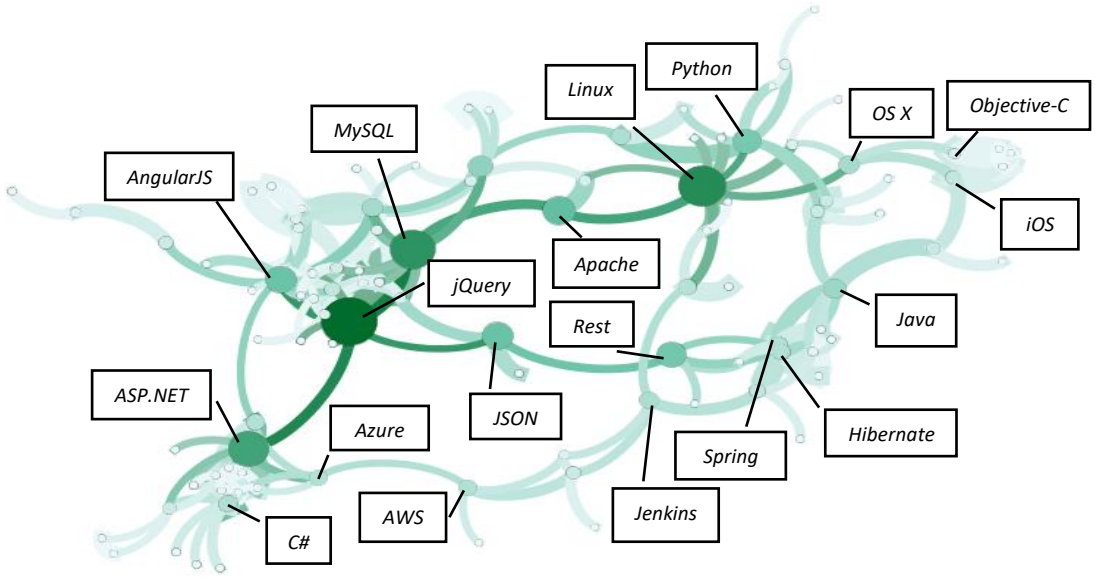

Figure 5: Dataset visualisation, focussing on betweenness centrality. Node size is proportional to its betweenness centrality value.

6.3.3 Node Size Analysis

The figure below (figure 6) shows the network when focussing on node size (popularity). The most significant technologies are JavaScript, Java, Python, HTML and CSS. This is understandable since these technologies are generally the most popular amongst developers. Further explanation of this variant of the network can be found in section 6.2 above.

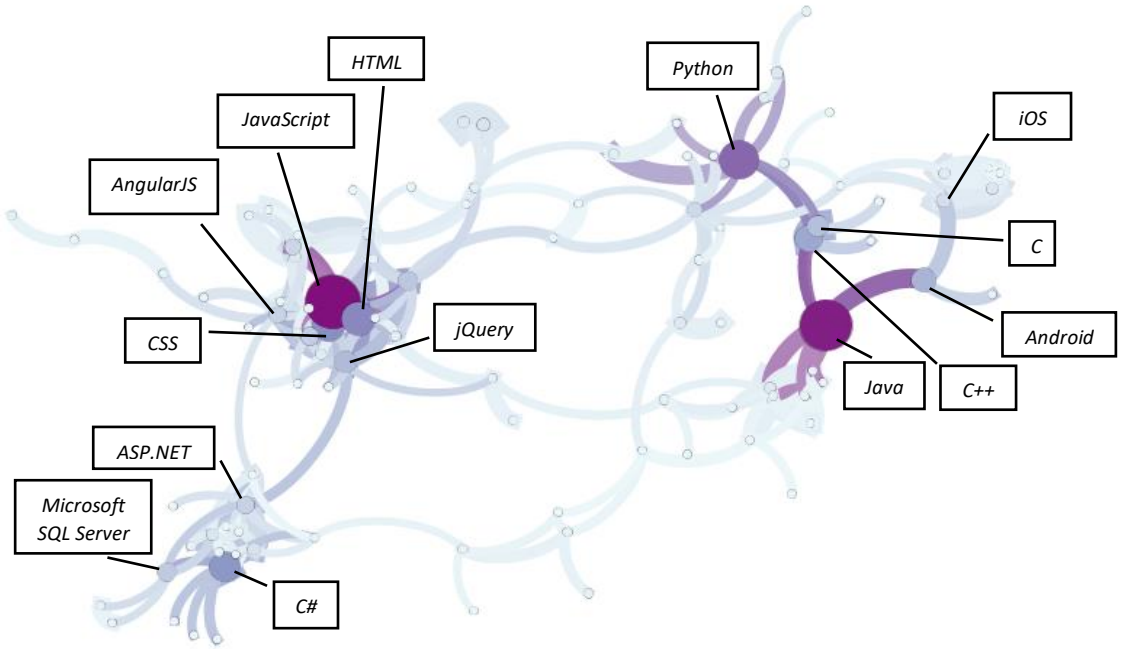

Figure 6: Dataset visualisation, focussing on node size. Node physical size is proportional to its node size value from the original dataset, proportional to its popularity.

6.3.4 Node Group Analysis

The figure below (figure 7) shows the network when focussing on nodes' groups, by colour coordinating the nodes based on which group they belong to. There are two to four major groups of nodes. The two major groups are web development programming languages (including JavaScript, jQuery, CSS) and technologies developed or related to the Microsoft Corporation (such as VB.NET, C#, ASP.NET). The two slightly less significant, but still important groups are Java related programming languages and frameworks (including Java, Spring, Hibernate) and database related technologies (such as MongoDB and PostgreSQL).

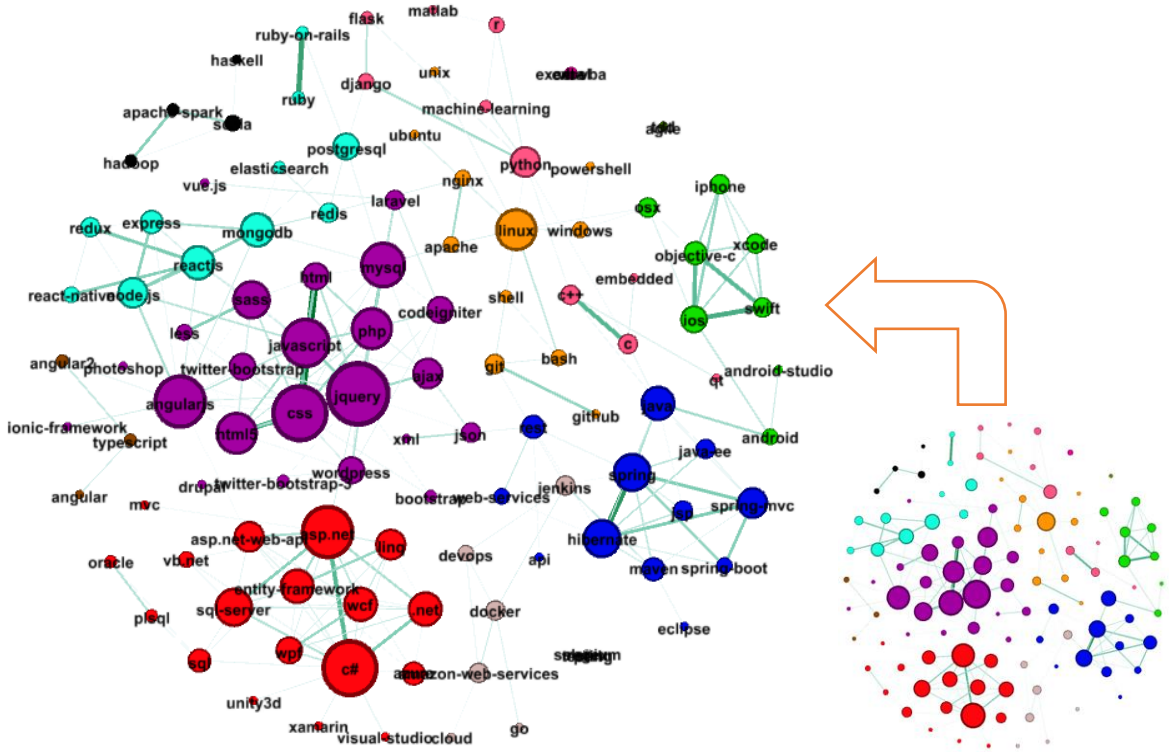

Figure 7: Dataset visualisation, focussing on node group. Nodes are coloured according to their respective groups, as assigned in the original dataset. Tags similar to each other are assigned to the same group. Node size is proportional to its degree.

7. Discussion

7.1 Observations and Insightful Thoughts

The network structure varies dramatically when focusing on node degree compared to betweenness centrality and node size (popularity). However, 'jQuery' has been the largest node for each variant. The network also has a large giant component, encompassing almost all the nodes. This seems reasonable as technologies tend to be quite intertwined in terms of interoperability.

Similar to the Internet network as measured via AS dataset. The Internet Topology also follows a power law network structure, with a few nodes highly connected with others and most nodes only connected with a small number of other nodes (Zhou & Mondragón, 2004). Additionally, both networks show a rich-club tendency where the highly-connected nodes

are well interconnected with each other. Figure 4 shows this clearly for the technologies network, as jQuery, AngularJS, ASP.NET, C# and JavaScript are nodes with the highest degree (links) and between these nodes the average hop distance is very small (one to two hops).

7.2 Conclusions

The network of technologies used by developers is inherently a scale-free network. It certainly follows a power-law structure, similar to that of the AS network. This can be seen throughout the many different variants of the network visualisations as well as through graphs created during the network analysis. Additionally, it is evident that the network has a disassortative mixing pattern since the highly connected networks are mostly connected to lowly connected nodes and vice versa. Additionally, since the network has a somewhat high clustering and to some extent small diameter, along with a very small average path length, it is fair to confirm that the technologies network displays a small-world network structure. To conclude, the network of technologies used by developers has shown to be similar to the Internet (AS) network and that it represents power-law network properties and structure.

7.3 Limitations

Due to a limited time frame and resources available for analysis, the dataset used for this investigation was just a subset of the complete set of data dumped by Stack Overflow. Therefore, the findings of this study act as a prediction and forecast of the potential structure of the complete network of technologies used by developers. A conclusive judgement of the entire network's structure can only be made after analysing the complete set of data relating to tags used by developers.

The study only reviewed developers using Stack Overflow, thus some proportion of developers might be omitted. Additionally, there is a possibility that developers will omit details of past technologies they used if these technologies are not popular or used anymore. Another limitation related to developers is that they might only include technologies which they are proficient at, thus omitting technologies which they have technically used but not frequently. This means the network between tags would lack some links, or at least affect the link value.

7.4 Future work and advice

In the future, repeating this study but with an immensely larger set of data will prove highly beneficial for analysing the complete network of technologies. This will provide a much deeper understanding of how the entire network behaves, as well as its structure.

Another aspect which might benefit from a deeper investigation is the algorithm and methodology used for carrying out the network analysis. From research of related studies, it is clear that there many different methods and algorithms available to analyse a network such as this. Therefore, carrying out similar research on the same type of dataset but with an alternative algorithm and/or methodology could help support the findings of this study and further explain the structure of the network of technologies used by developers.

8. Acknowledgements

All data and user contributions have originated from Stack Overflow and licensed under CC-BY-SA 3.0 with attribution required.

9. References

Algorithmia. (2016). *LDA*. Retrieved January 2019, from www.algorithmia.com: https://algorithmia.com/algorithms/nlp/LDA/

Barua, A., Thomas, S. W., & Hassan, A. E. (2012, November 01). What are developers talking about? An analysis of topics and trends in Stack Overflow. (J. Whitehead, Ed.) *Springer Science+Business Media*, 36. Retrieved January 2019

Bissyande, T. F., Thung, F., Lo, D., Jiang, L., & Reveillere, L. (July, 2013). Popularity, Interoperability, and Impact of Programming Languages in 100,000 Open Source Projects. *2013 IEEE 37th Annual Computer Software and Applications Conference* (p. 10). Kyoto, Japan : IEEE. Retrieved January 2019, from https://ieeexplore.ieee.org/document/6649842

Cancho, R. F., & Sole, V. R. (2001, November 07). The Small World of Human Language. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 5. doi:https://doi.org/10.1098/rspb.2001.1800

COMP0123: Complex Networks and Web. (n.d.). Week 1, Slide Set 2: Basic Network Properties. *Lecture Slides*.

COMP0123: Complex Networks and Web. (n.d.). Week 2, Slide Set 4: The Small World. *Lecture Slides*.

COMP0123: Complex Networks and Web. (n.d.). Week 2, Slide Set 5: Power-law networks. *Lecture Slides*.

Hassan, A. E. (2008). The Road Ahead for Mining Software Repositories. *2008 Frontiers of Software Maintenance* (p. 11). Beijing, China: IEEE. Retrieved January 2019, from https://ieeexplore.ieee.org/document/4659248

MIT. (2009). *Studying Complex Networks*. (R. Hernandez-Lopez, Editor) Retrieved Jan 2019, from web.mit.edu: http://web.mit.edu/8.334/www/grades/projects/projects10/Hernandez-Lopez-Rogelio/structure_2.html

Newman, M. E. (2004, April 06). Coauthorship networks and patterns of scientific collaboration. *PNAS*, 6. doi:https://doi.org/10.1073/pnas.0307545100

Stack Overflow. (2017, September 28). Stack Overflow Tag Network. *Network (links and nodes) of Stack Overflow tags based on Developer Stories*. (J. S. Overflow), Ed.) Kaggle. Retrieved January 06, 2019, from https://www.kaggle.com/stackoverflow/stack-overflow-tag-network

Xu, J., Gao, Y., Christley, S., & Madey, G. (Jan, 2005). A Topological Analysis of the Open Souce Software Development Community. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences* (p. 10). Hawaii, USA: IEEE. Retrieved January 2019, from https://ieeexplore.ieee.org/document/1385642

Zhou, S., & Mondragón, R. J. (2004, December 03). Accurately modeling the Internet topology. *Physical Review E*, 20. doi:https://doi.org/10.1103/PhysRevE.70.066108