Software technologies skills:

A graph-based study to capture their associations and dynamics

Konstantinos Georgiou
School of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
kageorgiou@csd.auth.gr

Maria Papoutsoglou
School of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
mpapouts@csd.auth.gr

Athena Vakali
School of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
avakali@csd.auth.gr

Lefteris Angelis
School of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
lef@csd.auth.gr

ABSTRACT

Software design and development technologies evolve very fast and in unpredicted rates, posing many challenges for programmers who strive to use them properly and to be up-to-date, especially since software development demands teamwork and collaboration. As a result, Question and Answer (Q&A) sites, like Stack Overflow, have seen large growth. The questions are characterized by tags, which support developers to easily trace their topic of interest. Very often, these tags refer to technologies that are connected or serve the same purpose. This work is motivated by the fact that despite the volume of questions and technologies change over time, tags inter-connections carry insightful information since they can be utilized to monitor the technology trends and their dynamics given technologies fast simultaneous evolution over time. This work recognizes the value of such connections, to reveal associations of technologies and to support the scientific community, which can be highly inspired by this study, using advanced data analysis tools. Benefits of the present study include early notification of the labor market needs, competency discovery and directions for educational planning. In the present study, graph theory principles and tools were employed to profile chronologically the evolution and the associations of technologies related to tags, and the experimentation carried out has involved the entire set of Stack Overflow questions posted during a decade, between 2008 and 2018. Very interesting conclusions are summarized based on the tags related to Computer Science Technologies (Hard Skills) analysis, revealing non-evident software technologies skills, associations and dynamics.

CCS CONCEPTS

• Social and professional topics • Software and its engineering • Collaborative and social computing

BCI'19, September 26–28, 2019, Sofia, Bulgaria
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7193-3/19/09…$15.00
https://doi.org/10.1145/3351556.3351565

KEYWORDS

Data Analysis, Hard Skills, Stack Overflow, Correlation, Networks, Graphs

ACM Reference format:

Konstantinos Georgiou, Maria Papoutsoglou, Athena Vakali and Lefteris Angelis. 2019. Technical software skills: A graph-based study of their evolution. In *Proceedings of 9th Balkan Conference on Informatics (BCI'19). September 26–28, 2019, Sofia, Bulgaria,* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3351556.3351565

1 Introduction

1.1 Motivation

Nowadays, information technologies have drastically changed the pace of life, including the working environments. Important events, everyday activities and even the interpersonal relationships flow at a much more rapid velocity, emphasizing on the immediate exchange of thoughts, reactions and feelings. Furthermore, it is not an exaggeration to claim that data, in every shape and form, is the driving force of human decisions and policies. Organizations, either public or private, as well as individuals produce continuously huge volumes of diverging data, contributing to the so-called "Big Data" challenge, as Computer Scientists commonly name the opportunities to extract knowledge from the data, aiming at prompt and efficient decisions. Additionally, software is everywhere nowadays, in the sense that it is very difficult to find an automated procedure or environment without the intervention of a software system.

Striving to extract knowledge from data and to develop flexible software for every type of automation, software companies continuously face the challenge for producing advanced technologies. Subsequently, the rapid evolution of the needs for new technologies creates new needs for technical competencies of professionals in the IT labor market. Moreover, these needs do not only concern single technical skills, but rather combinations of skills. All these issues related to trends in IT skills constitute the motivation of our work.

Therefore the main question of our research is the study of evolution of computer technologies in time, not individually but simultaneously, focusing on the associations among them and always under the perspective of how programmers are interested in them. It is our belief that such studies could facilitate the monitoring of trends related to the preferences of programmers and depict technologies that either endured in time or were rendered obsolete.

Since the study has clearly an empirical character, there is a need to find and utilize appropriate data, related to technical IT skills. An openly available and rich source of data and knowledge could be a "Questions and Answers" (Q&A) site, where programmers can pose questions related to computer technologies (frameworks, programming languages etc.) and provide answers to others. Questions can be made traceable by using tags related to the question's subject and to specific computer technologies.

1.2 Research Questions

The initial question behind the current paper as described earlier, was related to the evolving interest of programmers on computer technologies. This interest is directly mapped to their hard skills in IT. While examining their preferences expressed as questions in Q & A sites, we noticed that some tags appear in pairs. In some cases, these pairs tend to appear more and more often as time goes by, while on the other hand the frequency of some other pairs is decreased. Moreover, many technologies develop connections and adapt their purposes by exploiting software expansions and upgrades. The influence and popularity of some technologies varies from time to time due to the emergence of competitive ones.

Considering the above remarks and taking into account some first analyses we posed the following research questions:

- How technologies are associated and how can we model the associations from the analysis of a Q&A site?
- Which technologies have special role/status among others as depicted from the Q&A site analysis?
- How can we study the evolution of the associations and status of technologies through time?

Thus, in this paper, we accumulated user questions data from Stack Overflow, a well-known Q & A site, and after analyzing them, we tried to explain the trends of use and the associations among technologies using notions from networks and graphs. Certain graph metrics (centrality etc.) were also used in order to better present the technologies' status.

2 Related Work

During the last decade, Q&A websites have started to become popular for their sharing of knowledge content. Quora[1] [17],[13] and Yahoo[2] [9] Q&A are two popular Q&A sites for general purposes use and variety of topics. In addition, Stack Exchange is a "family" from 174 Q&A sites and usually each one has a different specific purpose use. The most popular site is the Stack Overflow Q&A website with technological orientation. Through the collaborative learning process, users can ask and/or answer questions according to their technical expertise. Through knowledge sharing, users interact and share their current knowledge or learn from the experience of the other members. In recent scientific approaches, many authors try to detect experts using the content of Stack Overflow and activity of every user [20], [11]. The use of tags in Stack Overflow community seems to be a "communication bridge" between the members of community. Using a specific tag the asker lets the other members know that the tagged question is linked with a specific issue so that members with the same expertize can provide a possible solution. User generated tags are more than 50,000[3]; however as it has been pointed out [1] tags have many drawbacks as askers have the option to create even not meaningful ones and add them arbitrarily. For this reason, authors in [1] try to extract main topics from Stack Overflow questions and answers using a semi-automated approach. More specifically, they used tags as trends and they connected them with the topics they found from the discussions. They found that trends are correlated with the detected topics and the most popular topics were related to web and mobile development. In addition, the research approach of [18] used a target set of tags, which were related to mobile development and detected in Stack Overflow discussions, which concern mobile developers.

As the tags are specific technologies such as programming languages, IDEs, frameworks, libraries etc. that can be learned through experience or courses, taking into consideration the definition of hard skills, we can directly map certain tags to hard skills. The definition of a hard skill is "the ability to apply knowledge and use know-how to complete tasks and solve problems" [4]. It is worth mentioning that there are approaches, which used tags as hard skills and correlated them with soft aspects.

Authors in [16] use tags as a list of hard skills and try to connect them with soft skills extracted from Stack Overflow job advertisements using principal component analysis. Also, authors in [4] use technology tags and try to find correlation between the personality traits of top users and tags using a lexicon approach and correlation analysis. In [8] authors tried to implement a recommendation system enabling users to tag questions more easily. They rely on Topic Shifting, a term referring to the use of different tags for questions of the same subject. They conclude that this recommendation system is much more efficient than static tagging systems implemented in Q & A sites.

The approach of [10] focused on pinpointing the primary subjects in Stack Overflow posts, utilizing the LDA algorithm. This way, they managed to trace trends in technology usage along with metrics that define a topic's popularity and impact. Another use of tags in [19] studied the course of Stack Overflow tags with the aid of Affiliation Networks that connect a tag with corresponding ones based on the correlations.

[1] https://www.quora.com/

[2] https://answers.yahoo.com

[3] https://stackoverflow.com/tags

[4] https://europass.cedefop.europa.eu/education-and-training-glossary/s (see the term skill)

Afterwards, they develop a Machine Learning model, to predict future tag popularity. The model achieves 66% accuracy.

In our approach, we selected to study the tags from the perspective of simultaneous evolution across time. To the best of our knowledge there, is no relative methodology for evolving technological trends. Only the authors in [14] used the time evolution to detect changes over time in technology interest using topic modeling in the text body of questions and answers. Aiming to analyze the time evolution of technological tags in specific time windows, we selected the network/graph approach.

Graph approach becomes a commonly popular method for analysis in Stack Overflow data during the previous years. A profusion of approaches ([15],[5],[12]) used Gephi software [3] for the visualization process of their network. Generally, from social network content it is possible to identify specific collaboration pattern from co-occurence tagging [7]. More specifically authors defined a social network using a quantitative method to analyze the development of the network through graph theory for a co-occurence network. In a co-occurence network users and tags are essential properties of the network [6]. These properties are related to network properties of folksonomies. In our approach, we chose to use the igraph package [5], which is available as an R library. This package provides different graph algorithms and network metrics. Furthermore, processed data from previous steps of our proposed methodology (see Section 3) can be directly inserted and analyzed. The main idea of our approach started from the simple visualization of Stack Overflow tags using igraph from the open dataset available in Kaggle but this approach did not take into consideration the aspect of time change[6]. However, the aforementioned graph analysis was used for exploratory purposes comparing to our approach where we use advanced metrics of graph theory.

3. Methodology

The methodology of our approach consists of a three-step process: (i) data collection and preparation, (ii) graph modeling and descriptive statistics and (iii) visualization and interpretation of preliminary results.

3.1 Data Collection and Preparation

Starting to implement our approach, we selected Stack Overflow as the source of information. Stack Overflow consists of many different parts of information such as questions, answers, users etc. In addition, every feature, like the aforementioned, contains many sub-details. For example, the structure of a question contains the title of the question, the main body, tags related to question, user who posted the question etc. The helpful part of the question for our current approach is the set of tags, which are related to the question. As a result, we collected the tags of all questions from the origins of Stack Overflow (August 2008) until December 2018. Querying Stack Exchange Data Explorer[7], we collected more than 17.000.000 questions using SQL queries. However, the first problem we faced was the large number (millions) of unique tags.

As this number of information is impossible to be handled and provide meaningful results, we selected to apply a filter and select the most popular technical skills. More specifically, we selected to analyze the 50 most frequent technologies. We filtered and processed the tags, based on the Developer Survey of 2018, conducted by Stack Overflow. We counted the single appearance frequencies of every tag we kept, in all questions and then we counted the frequency of pairs of tags. A pair of tags is the same if we change their order, i.e. (tag1, tag2) and (tag2, tag1) are counted once.

Our next step involved counting the correlation between each tag with all the rest. Since the appearances of tags are not represented by numerical variables so as to use classic correlation coefficients, we used the following formula appropriate for binary variables:

$$\boldsymbol{C(tag1, tag2)} = \frac{\boldsymbol{P(tag1, tag2)}}{\boldsymbol{P(tag1) * P(tag2)}} = \frac{\boldsymbol{\#(tag1, tag2)}}{\boldsymbol{\#tag1 * \#tag2}} * \#(\boldsymbol{Qs\ per\ year})$$

where by P(tag) we denote probability of occurrence of tag, the # symbol means "frequency of" and "Qs" stands for Questions. The rationale of the formula is that a number close to 1 would show independence of tags appearances, according to the classic definition of independence of events in probability theory. On the other hand, values lower than 1 show that there is some type of dependence in the negative sense, i.e. a specific pair of tags occurs less frequently than expected under independence assumption. Furthermore, values higher than 1 show dependence in the positive sense, i.e. the tags occur together more than expected under the independence assumption. In this paper we are only interested in positive dependence, so in the graphs that will be described in the next section, edges are drawn only if $C(tag1, tag2) > 1$.

3.2 Graph Modeling and Descriptive Statistics

Correlation matrices for every year were constructed thus concluding in eleven matrices representing years 2008-2018. In each matrix, rows and columns corresponded to all the tags while the matrix elements contained the value of the correlation between them. All the diagonal elements were set to zero, as the correlation formula is meaningful only for different tags.

[5] https://igraph.org/r/

[6] https://www.kaggle.com/devisangeetha/network-visualizations-with-igraph

[7] https://data.stackexchange.com/

The tables are symmetric, as each pair is counted once regardless of the order of tags in it.

In the final step, we constructed graphs (networks), where each vertex (node) represents a tag and the edges refer to the correlation between a pair of tags. If there is a correlation greater than 1 between tag1 and tag2 in the correlation tables, then they will appear connected in the graph. We constructed eleven graphs, corresponding to the correlation tables. We then analyzed some basic technologies, based on their appearances throughout the years and extracted some basic graph metrics (connectivity, centrality, community detection) to compare the structure of graphs along the time interval of the study. Graph metrics that were used for analyzing the graphs are the following [1]:

- Degree Centrality: Degree of each vertex, indicating technology interaction with other technologies.
- Closeness Centrality: The reverse sum of all shortest paths from vertex to vertex, indicating technologies that essentially are reference points to many others.
- Betweenness Centrality: Metric calculating the number of shortest paths in a graph passing through a specific vertex for all pairs of vertices indicating technologies that are required for passing between technologies.
- Eigenvector Centrality: Calculates the influence of a vertex to other vertices in the sense that a high value shows a vertex that has many connections with other highly connected vertices, indicating important technologies that depend and influence other important ones.
- Connectivity: This is not a metric, it is rather a characterization of the entire graph and has to do with existence of isolated parts. To investigate graph connectivity, we examined visually the structure of the graphs focusing on isolated vertices and we also calculated cliques consisting of more than one vertex. The presence of more cliques in the graphs is indicative of strongly connected technologies.

Finally, we created tables showing the pairs of technologies with highest correlations between technologies and the evolution of cliques in time. It should be noted, that even if a pair has high value of correlation, this does not necessarily mean that it has strong presence in the whole network, since some pairs may correspond to technologies fulfilling specific requirements. Additionally, competing technologies appear to have high correlation, as developers opt to use one of them.

4. Results

In this section, we present selected plots due to space limitation. All results can be found in the link: *https://tinyurl.com/y5h3665v* .Specifically, Figures 1(a) and 1(b) show the correlation graphs of technologies for 2008 and 2018 respectively. We can observe that the graph has evolved through the years, so the most recent one contains much more technologies and connections. Indeed, there is a larger number of vertices in the 2018 graph, containing technologies that do not exist in the 2008 graph.

The different centrality metrics that we applied to the graphs, displayed some very interesting results. Vertices with higher degree centrality in general are MacOS, Redis, Memcached, Cloud and Heroku, showing higher interaction with other technologies. Regarding closeness centralities, we found no notable differences in high numbers. Here it is interesting to depict the smallest vertices showing relative isolation, and these are Matlab, Neo4j, Excel/VBA and HTML/CSS/JQuery. In other metrics, vertices like Ajax, Windows/Linux/MacOs, Cloud, Django and Azure show the highest betweenness centrality and therefore the necessity of the corresponding technologies for connecting others. Finally, larger influence, as can be measured by the eigenvalue centrality, can be traced in vertices like Windows, MacOs, Linux/Bash/Shell, Hadoop and Redis/Cloud/Memcached.

Regarding the evolution of centralities in time, the vertices with high degree centralities (e.g. Memcached, Redis, Windows, MacOS, Cloud) more or less remain the same after 2012. An interesting finding is that Ruby appears to have the largest degree centrality in 2008 but has small values in later years. As mentioned earlier the closeness centralities are useful in determining isolated technologies, not connected to others. Such technologies are VBA/Excel, SharePoint, C#, Matlab and R, which remain stable throughout the study period.

For the betweenness centralities, in 2008 Ruby had the highest value, but this seems to decrease in later years. In contradiction, MacOS again seems to have a reemerging value of betweenness centrality, along with Windows, Linux, Cloud and Azure. Even though MacOS did not have high influence (eigenvector centrality) in some years, the use of this specific Operating System established its high betweenness centrality value. Finally, an interesting observation is that, in 2012, Cordova and C# have high values, a result aligned with the increased need for smart phones.

Regarding the eigenvector centralities, a general finding is that some of the highest values belong to Operating Systems (e.g. Windows. MacOS, Linux). Interestingly enough, MacOS appears in 2008 and 2009, gaining great influence from 2012 and beyond, as smart phones began to emerge in the global market.

Apart from Operating Systems, a large part of high eigenvector centrality values belongs to Database Managements Systems like MongoDb, Redis, Cassandra etc.

To draw results for the graph connectivity, we calculated the number of cliques that are formed in every year's graph. In Table 1, we can see that after a rapid increase in connectivity and a high peak in 2013 (833 cliques), the number becomes stable with the 2018 graph having 621 cliques.The study of graphs can reveal several interesting technologies that appear to change their tendencies through time or gain a notable popularity despite being relatively recent. Technologies such as Xamarin, Kotlin and AWS Services appear frequently and

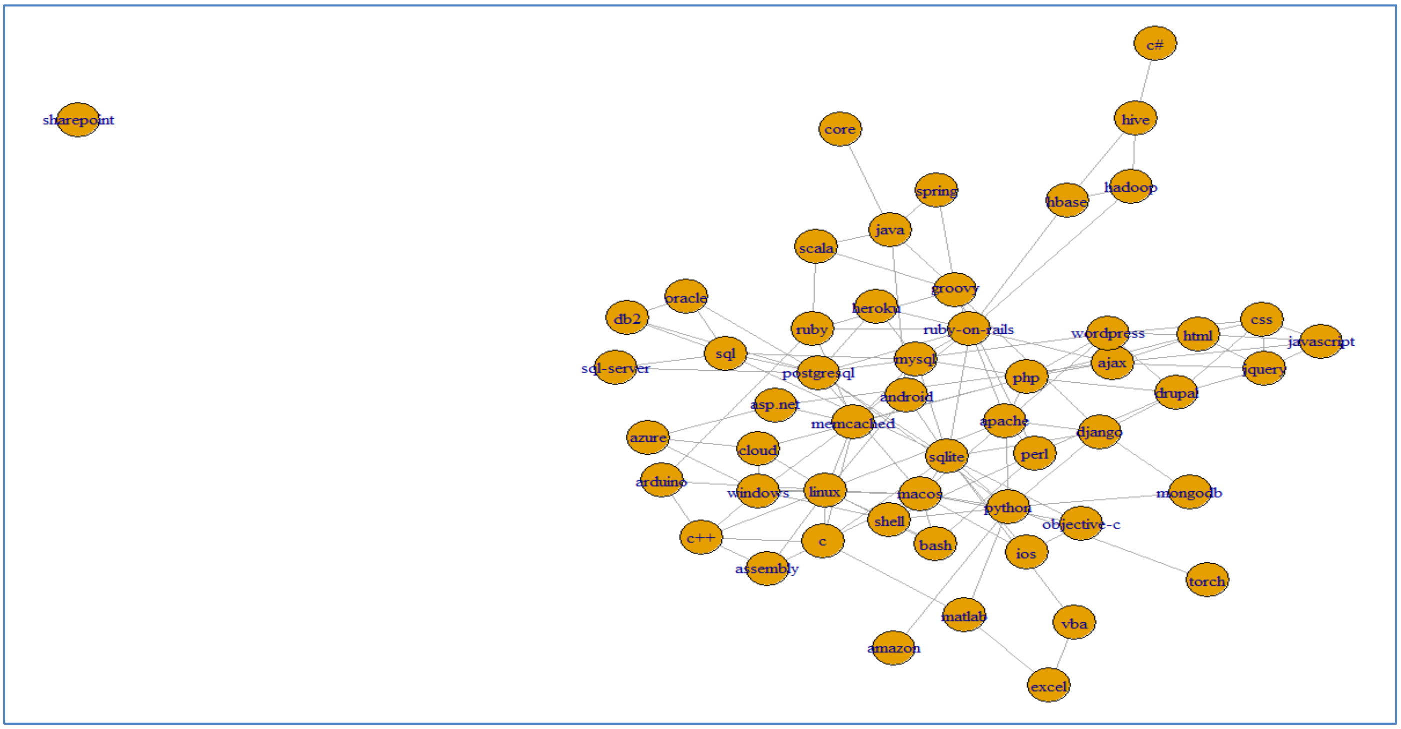

Figure 1(a) Correlation Graph of 2008

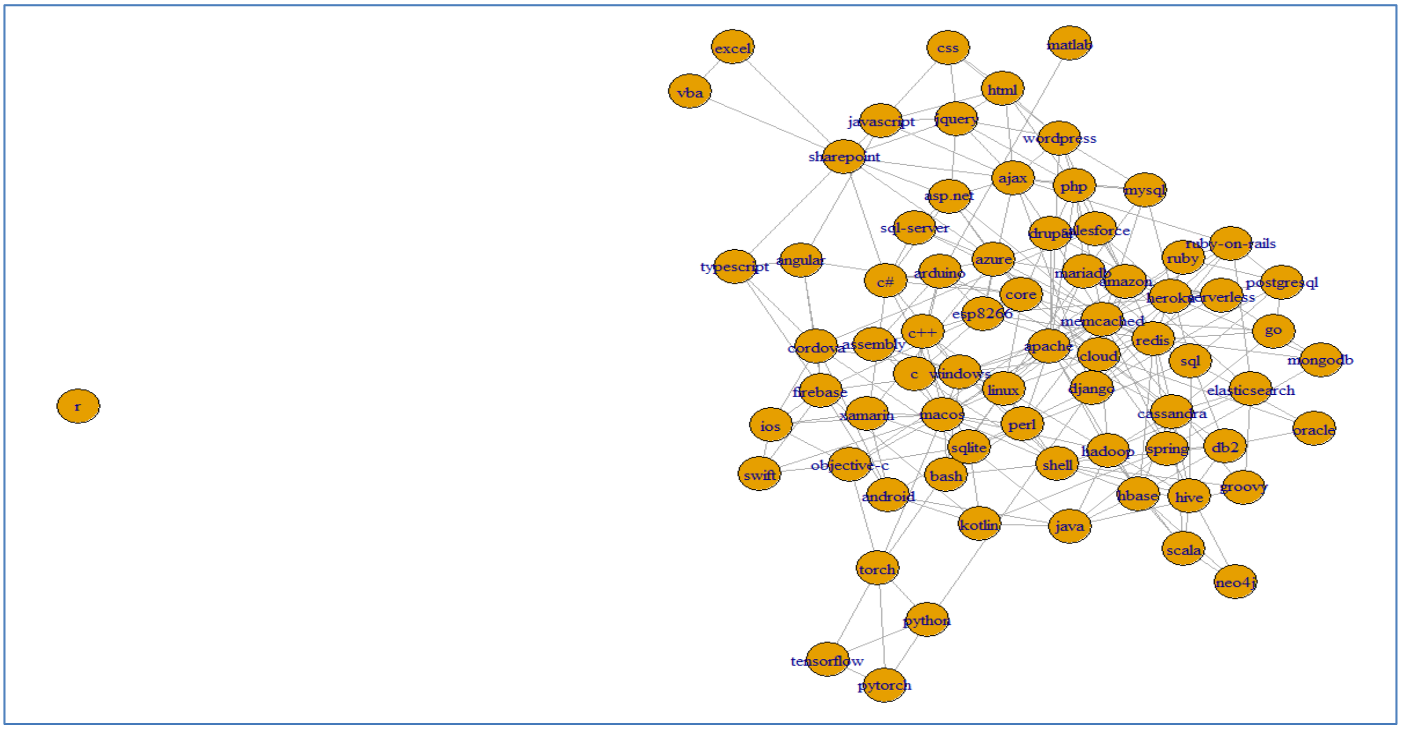

Figure 1(b) Correlation Graph of 2018

Table 1: Number of cliques through years of study

Year	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2017
No of cliques	218	273	491	704	642	833	636	637	635	632	621

Table 2: Highest Correlation Values in 2018

Pair	Pytorch- Torch	Arduino - Esp2866	Hadoop – Hive	Hbase - Hadoop	Bash - Shell	Ruby - RubyOnRails	Excel - VBA	Memcached - Redis	Linux - Shell	IOS - Objective-C
Correlation Value	460	344	115	107	50	43	39	37	22	20

have strong presence in the graphs, both in terms of graph metrics and correlation. Meanwhile, frameworks like Angular, appear in later years but grow exponentially in their use and popularity.

We also traced technologies like Pytorch or Tensorflow that can be detected much more recently but appear to be rise in popularity from the beginning of their use. Finally, there were pairs such as Arduino and ESP2866 that were strongly correlated with each other but were isolated from the other technologies. Their status in the network proves that there are pair of technologies that cannot be used for many purposes but are extremely important in some cases. Competitive technologies like Memcached or Redis providing solutions to the same problem can also develop strong associations, as users in their questions ask which one to use.

The general results of our research, in combination with our questions in Section 1 are the following:

- Association between technologies can be inferred by the tags extracted from a popular Q&A site and can be modeled using graphs with nodes and edges based on a measure of correlation.
- The role/status of technologies can be studied by specific centrality measures.
- The graph-based modeling year by year is useful for inference regarding the evolution of the entire body of technologies through time.

In Table 2 we can see the 10 pairs of tags showing highest correlations in 2018. They can be spotted in pairs like Hadoop – Hive, Hbase - Hadoop, Arduino – ESP2866 and Pytorch – Torch. As mentioned above, competing technologies like Memcached – Redis are correlated due to conflicts of use while some other pairs have to do with specific use (Arduino – ESP2866, Ruby – Ruby-On-Rails) or Operating Systems (Linux – Shell).

5. Conclusions

The goal of this paper was to analyze tags from Stack Overflow Questions. The studied tags are related to technologies and therefore to technology skills that software developers are interested in and tend to acquire throughout the years.

The basic idea was to study an entire "system" of tags/technologies not each one alone but all together, along with the connections between them. Our approach was based on graph/network theory with tags represented by vertices connected with edges showing their correlation. The notion of correlation between tags was defined as the systematic co-occurrence of a pair of tags. The graph/network approach has many benefits as it permits the visualization of all technologies together, facilitates the computation of many metrics that can show tendencies and the status of special technologies in the entire system and furthermore can be used to depict the evolution of the entire system throughout time by presenting graphs separately for each year.

A basic conclusion is that newer technologies constantly expand and create new connections way easily than older ones. This can be attributed to software connectivity and continuous upgrades they receive. However, this does not prevent some stable technologies, like Hadoop, Hive or Arduino to be strongly connected with others. Correlations vary, depending on the use of some technologies, but some pairs are inseparable, due to their specific goal.

On the other hand, analyzing the metrics of our graphs, we understood the influence and special role of some constantly important and long lasting software and computer technologies, like Java or Javascript, that influence a large number of similar technologies, although not necessarily having high correlations with them.

Some interesting future research suggestions include the further implementation of clustering techniques to trace communities, in combination with studying cliques present in the graphs. Another suggestion is the study of Soft Skills related tags, along with Hard Skills to get a clearer picture of user preferences and trends. Finally, an ambitious goal would be the accumulation of all the question and answer data of Stack Overflow and various others Q & A sites in order to repeat the process and get much more representative results.

ACKNOWLEDGMENTS

This research is co-financed by Greece and the European Union (European Social Fund- ESF) through the Operational

Programme «Human Resources Development, Education and Lifelong Learning» in the context of the project "Strengthening Human Resources Research Potential via Doctorate Research" (MIS-5000432), implemented by the State Scholarships Foundation (IKY).

REFERENCES

[1] Barabási, A. L. (2016). Network science. Cambridge university press.

[2] Barua, A., Thomas, S. W., & Hassan, A. E. (2014). What are developers talking about? an analysis of topics and trends in stack overflow. Empirical Software Engineering, 19(3), 619-654.

[3] Bastian, M., Heymann, S., & Jacomy, M. (2009, March). Gephi: an open source software for exploring and manipulating networks. In Third international AAAI conference on weblogs and social media.

[4] Bazelli, B., Hindle, A., & Stroulia, E. (2013, September). On the personality traits of stackoverflow users. In 2013 IEEE international conference on software maintenance (pp. 460-463). IEEE.

[5] Bosu, A., Corley, C. S., Heaton, D., Chatterji, D., Carver, J. C., & Kraft, N. A. (2013, May). Building reputation in stackoverflow: an empirical investigation. In 2013 10th Working Conference on Mining Software Repositories (MSR) (pp. 89-92). IEEE.

[6] Cattuto, C., Schmitz, C., Baldassarri, A., Servedio, V. D., Loreto, V., Hotho, A., ... & Stumme, G. (2007). Network properties of folksonomies. Ai Communications, 20(4), 245-262.

[7] Feicheng, M., & Yating, L. (2014). Utilising social network analysis to study the characteristics and functions of the co-occurrence network of online tags. *Online information review*, *38*(2), 232-247.

[8] Gruetze, T., Krestel, R., & Naumann, F. (2016, June). Topic shifts in stackoverflow: Ask it like socrates. In International Conference on Applications of Natural Language to Information Systems (pp. 213-221). Springer, Cham.

[9] Hertzum, M., & Borlund, P. (2017). Music questions in social Q&A: an analysis of Yahoo! Answers. Journal of Documentation, 73(5), 992-1009.

[10] Johri, V., & Bansal, S. (2018, January). Identifying trends in technologies and programming languages using Topic Modeling. In 2018 IEEE 12th International Conference on Semantic Computing (ICSC) (pp. 391-396). IEEE.

[11] Kapitsaki, G. M., & Foutros, P. (2017, August). Dear developers, your expertise in one place. In 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 371-374). IEEE.

[12] MacLeod, L. (2014, May). Reputation on Stack Exchange: Tag, You're It!. In 2014 28th international conference on advanced information networking and applications workshops (pp. 670-674). IEEE.

[13] Maity, S. K., Kharb, A., & Mukherjee, A. (2018). Analyzing the Linguistic Structure of Question Texts to Characterize Answerability in Quora. IEEE Transactions on Computational Social Systems, (99), 1-13.

[14] Neshati, M., Fallahnejad, Z., & Beigy, H. (2017). On dynamicity of expert finding in community question answering. Inform

[15] Odiete, O., Jain, T., Adaji, I., Vassileva, J., & Deters, R. (2017, July). Recommending programming languages by identifying skill gaps using analysis of experts. a study of stack overflow. In Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization (pp. 159-164). ACM.

[16] Papoutsoglou, M., Mittas, N., & Angelis, L. (2017, August). Mining People Analytics from StackOverflow Job Advertisements. In 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 108-115). IEEE.

[17] Patil, S., & Lee, K. (2016). Detecting experts on Quora: by their activity, quality of answers, linguistic characteristics and temporal behaviors. Social network analysis and mining, 6(1), 5.

[18] Rosen, C., & Shihab, E. (2016). What are mobile developers asking about? a large scale study using stack overflow. Empirical Software Engineering, 21(3), 1192-1223.

[19] Westwood, S., Johnson, M., & Bunge, B. Predicting Programming Community Popularity on Stack Overflow from Initial Affiliation Networks

[20] Yan, J., Sun, H., Wang, X., Liu, X., & Song, X. (2018, September). Profiling Developer Expertise across Software Communities with Heterogeneous Information Network Analysis. In Proceedings of the Tenth Asia-Pacific Symposium on Internetware (p. 2). ACM.