Annales Univ. Sci. Budapest., Sect. Comp. 42 (2014) 55–68

TREND ANALYSIS BASED ON SEMANTIC GRAPHS – A CASE STUDY*

Anna Bánsághi (Budapest, Hungary)
Attila Kovács (Budapest, Hungary)

Dedicated to the 70th birthday of Prof. András Benczúr

Communicated by János Demetrovics

(Received June 1, 2014; accepted July 1, 2014)

Abstract. Information published on the web is continuously growing and getting more and more complex. It is possible to accomplish quantitative trend analysis on collected data in various aspects. This paper describes a methodology in order to explore tendencies targeting the cloud and the quality characteristics (especially the testing domain) inside. The implemented prototype does frequency analysis on well prepared data sets of academic papers published on the web. The results are represented with a sequence of semantic graphs. This kind of time series data is an extension of the traditional tag cloud definitions. The paper describes the developed web crawling technology as well. The presented method can be applied as a reusable design pattern for any kind of trend analysis based on web data.

1. Introduction

We have arranged a civilization in which most crucial elements exceedingly depend on science and technology. Basically, the up to date technology can

*ACM Computing Classification: 1.2.4 Semantic Networks
Key words and phrases: cloud computing, semantic graphs, software testing
The project was supported by a special contract No. 18370-9/2013/TUDPOL with the Ministry of Human Recources.

be examined in three different aspects: we can focus on products and services which were produced during the technology, we can follow the processes and the activities which are parts of the production of these products and services, and we can also analyse the capacity of the external and internal environment of the corresponding technology arrangements. These three factors (products – activities – capacity) mutually depend on and affect each other. It is also clear that each of them can be layered. An example is the Gartner hype cycle model [5] which is an analytical tool for understanding the ontogeny of emerging technologies observing that technologies walk through a predictable pattern (technology trigger – peak of inflated expectations – trough of disillusionment – slope of enlightenment – plateau of productivity) before becoming accepted by customers and reaching a realistic level of productivity. The axes of the model are the time and the maturity which measure the level of ranking of the innovative services. This study introduces a similar experimental model in which the focus is, instead of the products, on the *activities corresponding to technologies*, specifically which are connected to academic researches.

The target of our analysis is connected to the research results achieved in cloud computing. To be more precise, the sources of the analysis are academic proceedings papers related to cloud computing and its quality characteristics, especially the testing domain inside. The diminution of the source has two advantages. First, the information is controlled and authentic due to the fact that the appropriate papers were collected from the digital library (www.computer.org) of the IEEE Computer Society. Second, the collection of data was performed in a semi-automating way. The aspect of the analysis (similar to the Gartner model) is the maturity, in our case the trends in cloud related quality and testing. The experimental evolutionary model represents quality and testing related topics in the cloud and emphasizes the shifts on these topics.

The problem of topic modeling started with the term frequency – inverse document frequency (*tf-idf*) scheme [10] which ranks the words based on how important they are to a document within a larger corpus. The algorithm is often used in quantitative analysis obtaining a list of the most important words for each article. A different probabilistic model, the Latent Dirichlet Allocation (LDA) model [3] uncovers the underlying semantic structure of a document collection based on a hierarchical Bayesian analysis of the texts. The hypothesis is that the observed complexity of usage patterns is produced by a smaller set of hidden variables such as list of words (topics).

Both models – linear algebra and probability modeling – can infer from networks. Dietz et al. [4] created a model that looks at citation networks and describes the flow of topics between papers, where documents are generated by topical innovation and topical inheritance via citations. Martens et al. [8] explored the topics of the Cloud Computing ecosystem which were aggregated

to a comprehensive model. Styvers et al. [11] proposed an extension of the LDA-based model, so called the Author – Topic Model, assuming that the topics are formed on the mixtures of authors writing a paper. Thiel et al. [12] introduced two methodologies to visualize topic shifts and change of vocabulary of scientific papers over a given period of time.

The solution proposed in this paper – similar to the tag cloud [13] – is the building of semantic graph sequences based on the *tf-idf* scheme. The traditional tag cloud represents the frequent keywords in a collection of documents. Usually, the tag cloud is in an alphabetical order and the size of the keywords reflects to its frequency. In our case some changes have been done regarding the original definition. On the one hand each separated tag cloud was dedicated to each considered time period. Consequently, a sequence of tags (containing thirteen tags) was built from the articles collected over the past thirteen years. On the other hand, the traditional tag cloud is a set of nodes, however, in our model the *relationships* among the keywords were taken into account. There is an edge between two keywords whether they appear in the same article. Furthermore, the keywords have been classified, hereby the graphs could be split further into sub-graphs. Different sub-graphs will be marked with different colors. Thus, all the keywords belonging to a certain category are represented in a colored graph and have different size of the keywords.

Figure 1 and Figure 2 present the difference of a tag cloud and the semantic graph.

Figure 1. Tag cloud of the software quality, source: Technische Universität München

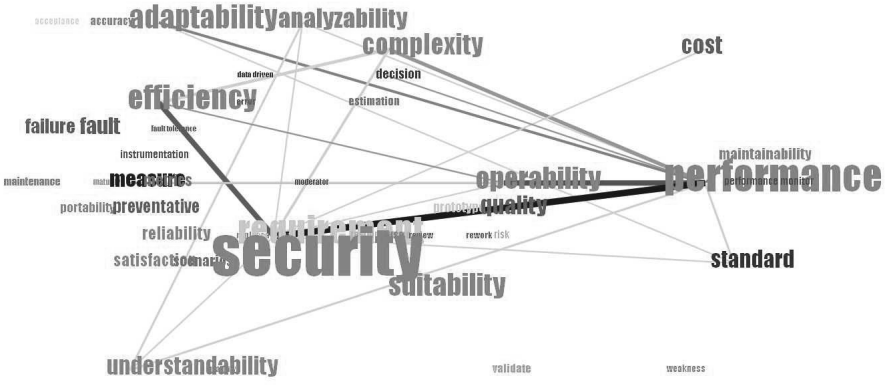

Figure 2. The semantic graph of the year 2005.

2. Building the model

The basic steps of the quantitative analysis are similar to [14]. The first steps are the extraction of valuable information such as collection, organization, analysis and representation of the source data.

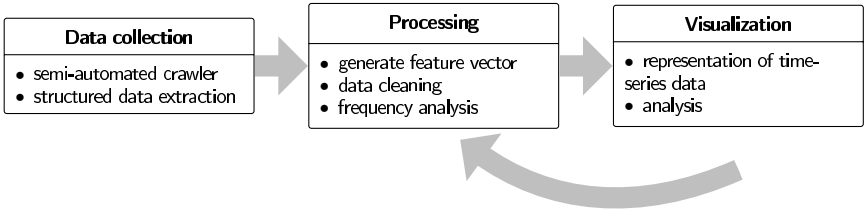

Figure 3. Components of the graph-based quantitative analysis and their process flow.

2.1. Data collection

The articles, which are from the cloud computing domain and were published between 2001 and 2013, have been selected from the collection of articles of the digital library mentioned above. At the initial stage of the collecting process both structured and unstructured textual data (several conference pro-

ceedings) were identified focusing on cloud computing. Although all of the potential candidates provide a premium quality in the field of cloud computing (such as the ACM Symposium on Cloud Computing or the International Conference on Cloud Computing Technology and Science Proceedings) the final choice was fixed on the IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid) and on the International Conference on Cloud Computing (CLOUD). The reason is that CCGrid symposium series serves as the major international forum ranging from clusters and grids to clouds and data centers, and was debuted at 2001. In similar reasons, the IEEE Society launched a series of events to promote the growing field of cloud computing. The CLOUD conference series was established in 2009.

In addition, it was an important aspect that the IEEE Computer Society web site provides the possibility of crawling its Digital Library and of extracting *structured data* from it. By selecting an appropriate subpage of the `www.computer.org` domain as a start URL, the needed information is scraped by our semi-automated web crawler component which is written in Python and used the Scrapy application framework. The data type of the desired information was defined as a class of properties, such as the unique URL, the title and the abstract of the article, and the list of keywords. The crawling rules for following links and for extracting data from matched pages were formulated as XPath queries. The following HTML portion of the code represents an article sample, which can be found in the digital library (`www.computer.org`):

```
<div id="dlcontent">
    <div id="abs-articleinfo"></div>
    <div id="abs-info-left">
        <div id="abs-proceedingTitle">
            2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid
                Computing
        </div>
        <div id="abs-articleTitle">
            Automated, Elastic Resource Provisioning for NoSQL Clusters Using TIRAMOLA
        </div>
        <div id="abs-issue-left-p"></div>
    </div>
    <div id="asciiText"></div>
    <div id="BibTex"></div>
    <div id="RefWorks"></div>
    <div id="abs-authors"></div>
    <div id="abs-doi-left"></div>
    <div id="abs-abscontent">
        <div class="abs-tagline"></div>
        <div class="abs-articlesummary">
            This work presents TIRAMOLA, a cloud-enabled, open-source framework ...
        </div>
    </div>
    <div id="abs-additionalinfo">
        <div class="abs-tagline"></div>
        <div class="abs-index-terms">
            <span class="terms">Index Terms:</span>
            <div>
                Open-source,Cloud Resource Provisioning,Elasticity,NoSQL,Markov Decision ...
            </div>
```

```
        <div class="abs-citation"></div>
    </div>
</div>
```

Observe that the title of the article can be found in the `abs-proceedingTitle` div tag. Likewise, the abstract of the article can be found in `abs-articlesummary`, as well as the keyword of the article in the `abs-index-terms` div tags. These tags and styles were selected by the following XPath expressions:

```
/div[@id="abs-articleTitle"]/text()
/div[@class="abs-index-terms"]/div/text()
/div[@id="abs-abscontent"]/div[@class="abs-articlesummary"]/text()
```

Table 1 shows the number of processed papers annually and cumulatively.

Year	CCGrid	CLOUD	Sum
2001	73		73
2002	43		43
2003	81		81
2004	80		80
2005	126		126
2006	137		137
2007	94		94
2008	97		97
2009	71	26	97
2010	111	69	180
2011	51	92	143
2012	119	119	238
2013	86	106	192

Table 1. The processed research papers.

2.2. Data processing

The data processing consists of three subtasks. In the first step software quality and testing related expressions of the feature vector were built. In the second step the data were cleaned. In the last step the word frequency matrix was computed. Let us discuss the concept behind the feature vector construction. The selection and classification [6, 9] of the software testing related keywords were done manually. The 363 identified expressions were classified into 21 categories which can be grouped further into four main classes (see Table 2).

CLASSES	NUMBER OF KEYWORDS
Organization level test processes	
Test policy	6
Test strategy	43
Test approach	36
Overall risk management	11
Test and quality standards	20
Project level test processes	
Test plan	29
Test management	15
Test processes	24
Product quality assurance	31
Test measurements	4
Testing tools	29
Test types and techniques	
Static testing	18
Non-functional testing	7
Regression-related testing	7
Experience based testing	9
Specification based testing	25
Structural based testing	24
Test levels	
Unit test	2
Integration test	7
System test	2
Acceptance test	14

Table 2. The four main classes of the keywords and their further categorization.

The test policy and strategy, as well as the testing approach, risk management and standards belong to the *organization level test processes*. This level contains the policy and strategy concepts, the test approaches, the identification, analysis and treatment of the risks, and the organizational test standards.

The *project level test processes* – such as test management, test processes, test plan, selection and usage of test tools necessary for operative tasks, composition of measurements of quality assurance for the products of services – constitute another category.

The third category consists of *test types and techniques*, like functional, non-functional, static and regression based techniques, and test design techniques

(specification based, structure based, experience based tests).

The last category consist of *test level related concepts.*

The keywords in the feature vectors and the abstracts were preprocessed with the Python procedures of the Natural Language Toolkit [2]. In both cases the same stemming algorithm run over the lower case converted text. During the feature vector construction it was considered that the text should only contain lower case characters of the English alphabet and space. During the abstracts parse the special characters such as hyphen, comma, slash were replaced by space.

All the articles were represented by a 4-tuple (URL – title – keywords list – abstract). Then, this data set was purged (tuples with empty abstracts, duplicate titles, setc., were deleted). We note that beside the previously mentioned character replacement there were other processes performed during the cleaning of the abstracts, such as the unification of misspelled or different spelling words. Depending on the results these steps were performed iteratively. As a result of this step, the previously described article was transformed into the 4-tuple represented in the Table 3, while Table 4 shows its term – frequency pairs.

URL	http://www.computer.org/csdl/proceedings/ccgrid/2013/4996/00/4996zcvr-abs.html
keywords	open source, cloud resource provisioning, elasticity, nosql, markov decision process, policy based optimization, distributed datastores
abstract	thi work present tiramola, a cloud enabled, open sourc framework ...
title	automated, elast resourc provis for nosql cluster use tiramola

Table 3. Cleaned and stemmed representation of an article.

TERM	FREQUENCY
adapt	1
decis	3
perform	2
standard	1

Table 4. Word frequency of the same article

After the data cleaning and stemming the collection of abstracts were appropriate for analyzing the frequency of words. Usually, the word frequency matrix of textual documents looks as follows: the columns of the matrix are annotated with words, the rows of the matrix are annotated with titles of the

documents. Cells represents the presence of the words in a document collection. In our case different measurements were used: the word frequency was computed by the *tf-idf* measure by

$$w^i_{year} = \frac{f^i}{F^i} \log\left(\frac{N_{year}}{n^i_{year}}\right). \tag{2.1}$$

The weight of the i-th word can be computed by multiplying two sub-formulas regarding to a given abstract. The first sub-formula is responsible for the word frequency, where f^i means the occurence of the i-th word in the abstract, and F^i denotes the number of words in the same abstract. The second sub-formula is the logarithmic function of the variables of n^i and N, where n^i is the number of the abstracts in which the i-th word occure, and N is the number of all abstracts. According to the thirteen years observation period, thirteen individual sets of abstracts were analyzed on the stipulation that the columns of the word frequency matrix were restricted to the words and expressions of the feature vector. Thus, the weight of the i-th word from the feature vector in a given year was calculated by the Equation 2.1.

Table 5 shows three keyterms and their associated information.

ID	TERM	*tf-idf*	f^i	n^i
107	fault	0.28099599005279186	20	8
108	fault multipl	–	–	–
109	fault toler	0.2586709274833676	16	5

Table 5. A piece of the union graph of the year 2013

2.3. Data visualization

The visualization of the computed data was made via graphs. Graph visualization algorithms accepts the description of the graph as an input and returns the image of the graph as an output. There are three main drawing approaches of graph visualizations, namely the hierarchical, the topology-shape-metrics and the force-directed approaches [1, 7]. The hierarchical approach is suitable for visualization of acyclic, directed graphs. In the second approach the nodes are represented with circles and rectangles, connected with dashed lines.

In our case, for the representation of simple, undirected graphs, the third approach was used. The force-directed solution is typical. Another advantage is that it provides an animation feature. The idea behind the force-directed method is that the graph is modeled as a mechanical system where the vertices are replaced with steel rings, and the edges between them are replaced with

springs. The drawing algorithm starts the system from a random unstable state, which moves until it gets into a minimal energy state. In this study the prototype uses the force directed algorithm of the D3 JavaScript library in which the animation technique is switched off. Consequently, the vertices of the thirteen graphs appear in alphabetic order on the same position of the screen.

In the following, the implementation of the previously described structures will be presented, where the physical structure is based on graphs. In this step, the Python NetworkX package was used.

In the first step the graphs of abstracts were created. The four-tuple related to a given article was represented by a complete undirected graph, in which the expressions from the abstract provide the vertices of the graph. The graph is complete, due to the fact that each of the expressions appears in the same abstract, the weight of the edges are all equal to one. The labels of the vertices are complex data storing information which is needed for frequency analysis. In this step the data comprise of components such as the numeric identifier of the vertex, the keywords, the number of the different tokens in the abstract and the number of the occurrences of the keywords. Formally, in a given *year* the notations are as follows:

V_{year}	vertices labeled by keywords
$V^i_{year} \subset V_{year}$	vertices assigned to the keywords of the article i
$E^i_{year} \subseteq V^i_{year} \times V^i_{year}$	set of edges between each pair of vertices of the article i
$k_{year} : V^i_{year} \to \text{ID} \times \text{Keyterm} \times \text{Frequency}$	vertex labeling function
$w_{year} : E^i_{year} \to \text{Frequency}$	weight function of the edges

In the second step the previously created graphs were joined (union) according to the word frequency analysis. In order to define the static location of the vertices in the visualization a template was used as a static graph of the keywords of the feature vectors. Thus, thirteen union graphs were generated. If in a given year a given keyword did not occur the related node is invisible during the visualization but it preserves its location. The information stored in a node of the union graph is the following: the numeric identifier of the vertex, the keyword, the computed *tf-idf* value and the category of the given keyword. The set of edges was produced by the union of the set of edges of the graphs of the abstracts. If there were multiple edges between two nodes, these edges were unified into a single edge and its weight was accumulated. Finally, the time ordered sequence of the semantic graphs were produced.

V_{year}	the set of vertices of the union graph
$E_{year} \subseteq \cup_{i=1}^{n} E_{year}^{i}$	the set of edges of the union graph
$k : V_{year} \to \mathbb{ID} \times \text{Keyterm} \times \text{Freq-}tf\text{-}idf$	vertex labeling function
$w_e : E_{year} \to \mathbb{N}, \quad w_e = \sum_{i=1}^{n} w_e^i$	weight function of the edges

Before we were able to use the D3 JavaScript library for visualization, in the third step, the union graphs were transformed once more. Technically, the main part of this transformation was the merging of the thirteen graphs into one single JSON object. The set of nodes of the merged graph contains the keywords but one node stores all of the information from all of the thirteen years. The node stores the identifier, the keyword, and the category of the given keyword as well. Each keyword is classified into one category, thus, the label of the classification clearly defines the color of the node during the visualization. The thirteen *tf-idf* values regarded to the thirteen years were stored in list. Similar to these lists, the weights of the edges were stored in lists as well. Figure 4 shows the analysis results of the year 2011. The full semantic graph sequence can be found at `http://phd.mamikon.net`.

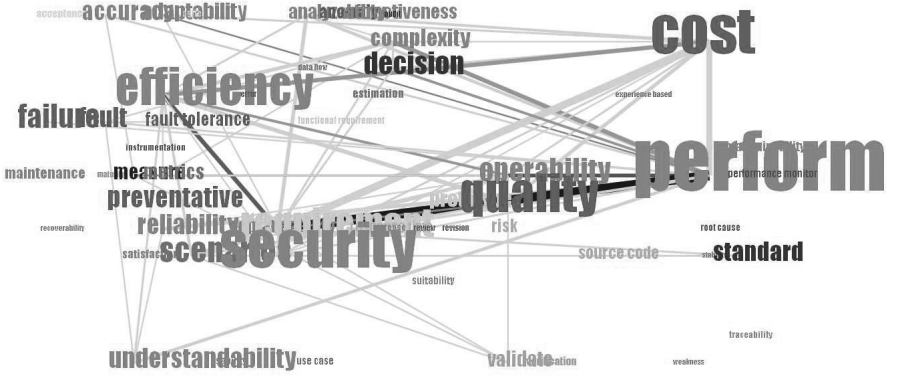

Figure 4. The semantic graph of the year 2011.

3. Conclusion

In this study a model of processing cloud computing related academic articles was described from the aspects of quality and testing. The aim of this

research was visualizing the sequence of semantic graphs (trends) applying quantitative analysis. Our graph representation provides an intuitive frame for the organization and analysis of information. The traditional definition of tag cloud was extended, not only the frequency analysis of the terms was used but also their categories and relationships were considered. Another aim was an implementation of a well modularized, easily parametrized prototype and achieving a model which can easily be validated.

Examining the graph representation the following conclusions can be established:

- Among the investigated 21 categories the commonly occuring items were the *Product quality assurance*, the *Test strategy* and the keywords of these categories such as *quality, cost, efficiency, performance* for all examined years with increasing frequency.
- Although the expressions related to quality are common, the emphasis on testing itself is small, for example there are just a few keywords from the category *Test types and techniques*.
- There is a significant strong connection between the keywords *performance* and *requirements*, and among the triplet *cost – performance – efficiency*.
- It is interesting that in real projects the importance of the combination of words *Cost of Quality* is high, the examined academic papers do not reflect on this phenomenon, they occur together rarely.
- The frequency (importance) of the keywords *security* and *risk* were growing during the examined years.

References

[1] **Battista, G.Di, P. Eades, R. Tamassia and I.G. Tollis,** *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.

[2] **Bird, S., E. Klein and E. Loper,** *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*, O'Reilly, 2009.

[3] **Blei, D.M., A.Y. Ng and M.I. Jordan,** Latent Dirichlet allocation, *The Journal of Machine Learning Research*, **3** (2003), 993–1022.

[4] **Dietz, L., S. Bickel and T. Scheffer,** Unsupervised prediction of citation influences, *Proc. 24th Int. Conf. on Machine Learning,* ACM, New York, NY, USA, 2007, 233–240.

[5] **Fenn, J. and M. Raskino,** *Mastering the Hype Cycle: How to Choose the Right Innovation at the Right Time (Gartner)*, Harvard Business School Press, Boston, 2008.

[6] **Graham, D., E. Van Veenendaal, I. Evans and R. Black,** *Foundations of Software Testing: ISTQB Certification*, Intl Thomson Business Pr., 2008.

[7] **Kafumann, M. and D. Wagner (eds.),** *Drawing Graphs Methods and Models*, Lecture Notes in Computer Science **2025**, Springer, Berlin, 2001.

[8] **Martens, B. and F. Teuteberg,** Understanding the cloud computing ecosystem: Results from a quantitative content analysis, *Proc. 10. Internationale Tagung Wirtschaftsinformatik, Zürich, 2011,* 16–18.

[9] **Morgan, P., A. Samaroo and B. Hambling,** *Software Testing: An ISTQB-ISEB Foundation Guide*, British Computer Society, 2010.

[10] **Salton, G. and M.J. McGill,** *Introduction to modern information retrieval*, McGraw-Hill, 1983.

[11] **Rosen-Zvi, M., T. Griffiths, M. Steyvers and P. Smyth,** The author-topic model for authors and documents, *Proc. 20th Conf. on Uncertainty in Artificial Intelligence,* AUAI Press, Arlington, Virginia, USA, 2004, 487–494.

[12] **Thiel, K., F. Dill, T. Kotter and M.R. Berthold,** Towards visual exploration of topic shifts, *IEEE Int. Conf. on Systems, Man and Cybernetics, 2007,* 522–527.

[13] **Venetis, P., G. Koutrika and H. Garcia-Molina,** On the selection of tags for tag clouds, *Proc. Fourth ACM Intern. Conf. on Web Search and Data Mining* **WSDM '11**, ACM, New York, NY, USA, 2011, 835–844.

[14] **Weiss, S.M., I. Nitin and Z. Tong,** *Fundamentals of Predictive Text Mining*, Springer, 2010.

Anna Bánsághi
Eötvös Loránd University
Budapest, Hungary
anna.bansaghi@mamikon.net

Attila Kovács
Eötvös Loránd University
Budapest, Hungary
attila.kovacs@compalg.inf.elte.hu