

Topic Recommendation from Tag Clouds

Ágnes Bogárdi-Mészöly*, András Rövid†, Hiroshi Ishikawa‡

*Department of Automation and Applied Informatics,
Budapest University of Technology and Economics, Hungary
Department of Computer Science, Shizuoka University, Japan
Email: agi@aut.bme.hu

†Institute of Applied Mathematics, Óbuda University, Hungary
Email: rovid.andras@nik.uni-obuda.hu

‡Department of Computer Science, Shizuoka University, Japan
Email: ishikawa@inf.shizuoka.ac.jp

Abstract—The spread of Web 2.0 has caused user-generated content explosion. Users can tag resources in order to describe and organize them. A tag cloud provides a rough impression of relative importance of each tag within the overall cloud in order to facilitate browsing among numerous tags and resources. The size of a tag cloud may be huge. Thus, the goal of our paper is to recommend topics based on the tag cloud and visualize the recommended topics like a tag cloud. Firstly, an algorithm has been proposed to construct a special tag graph from the tag cloud. Secondly, an algorithm has been provided in order to recommend topics using this tag graph by calculating the reference count of each node. Furthermore, a visualization has been introduced for recommended topics like a tag cloud using a special font distribution algorithm. The proposed graph and algorithms have been validated and verified on the tag cloud of a real-world thesis portal.

I. INTRODUCTION

With the appearance of Web 2.0 [1] and the spread of social media sites [2] users became from passive spectators to active content generators. Users can interact and collaborate with each other in virtual communities. Nowadays lots of social sites exist for various purposes: collaborative projects (Wikipedia), blogs (Twitter), content communities (YouTube), social networking sites (Facebook), virtual game worlds (World of Warcraft), virtual social worlds (Second Life), etc.

There are numerous books dealing with this topic denoted for introduction and marketing [3], for research [4], etc. Various significant companies have research groups for social computing: Microsoft [5], IBM [6], HP [7], etc.

In social networks power laws occur many times in many contexts [4]. A random variable is distributed according to a power law when its probability density function is given by $x^{-\gamma}$, where $x \geq x_{min}$ and $\gamma > 1$ [8]. γ is a constant parameter called exponent or scaling parameter, typically in the range of $2 < \gamma < 3$.

Usually on these sites users can assign tags to resources in order to describe and organize them. This tagging process [9] establishes associations between tags and resources, which can be applied to navigate to resources by tags, as well as, to tags based on related tags, etc. With tagging a folksonomy (folk (people) + taxis (classification) + nomos (management))

evolves, which is the vocabulary of tags emerged by the community [10]. The size of these vocabularies may be huge, moreover, they are incomplete and inconsistent. Thus, in connection with social tagging several challenges have emerged [4].

The paper is organized as follows. Section II covers the background. Section III introduces the experimental environment. Section IV proposes algorithms to construct a special tag graph from a tag cloud, in addition, to recommend topics using this tag graph by calculating the reference count of each node. Furthermore, it validates and verifies the novel algorithms on a real-world thesis portal. Section V introduces a visualization for recommended topics like a tag cloud using a special font distribution algorithm. Finally, Section VI reports the conclusion and future work.

II. BACKGROUND

In this section, definitions related to tagging are discussed. Tags are user-defined informal and personal strings, short descriptions related to resources, keywords associated with resources. They are helpful in browsing and searching. Resources are such identities which can be tagged, such as text, image, audio, video, document, etc. Tagging is the process of assigning existing and new tags to resources. Tag recommendation systems exist to help users in tagging based on own tags or tags of other community members.

There are lots of different kinds of tags: content-based, context-based, attribute, ownership, subjective, organizational, purpose, factual, personal, self-referential, tag bundles, etc. Furthermore, users have various motivations for tagging: future retrieval, contribution and sharing, attract attention, play and competition, self presentation (self referential tags), opinion expression, task organization, social signaling, money, technology ease, etc. [4].

In our tag clouds, tags are content-based, tagging is motivated by contribution and sharing. With content-based tags the actual content of the resources can be identified. By the contribution and sharing as motivation, tags describe resources, and add them to conceptual clusters or refined categories for known and unknown audience.

Tag clouds are visually depicted tags in order to facilitate browsing among numerous tags and resources. It gives a rough impression of relative importance of each tag within the overall clouds.

In some situations to answer various questions browsing in tag clouds are more useful than searching [11]. Search interface is preferred if the needed information is specific. Tag clouds are preferred if the seeked information is more general.

For this reason, the visualization of tag clouds is one of the most important and complicated consideration [12]. Tag clouds have two dimensional representations. Tags can be ordered alphabetically, based on semantic similarity or any kind of clusters [13], [14]. Relevant tags can be visually emphasized using such visual properties as shape, color, position, etc.

Each tag cloud is visualized in its own unique way. The basis of the used methods is similar, but there are no two tag clouds whose visualization is the same. Numerous font distribution algorithms exist [15] [16].

In this paper such tag clouds are investigated. In our tag clouds all tags are represented simply alphabetically ordered and visually weighted by letter size. The further improvement of visualization is a subject of future work.

III. EXPERIMENTAL ENVIRONMENT

The Faculty of Electrical Engineering and Informatics of the Budapest University of Technology and Economics has a web portal to manage all theses of the faculty for the whole workflow starting from description to review [17]. (Now the language of its user interface is only in Hungarian, the English version is proceeded.)

This portal has been implemented as a three-tier ASP.NET web site. The presentation layer is in HTML and jQuery. In the business logic layer there are C# classes. In the data access layer LINQ and stored procedures are used mixed. The database is in Microsoft SQL Server. The provided algorithms have been implemented in SQL stored procedures, C# classes using LINQ, and MATLAB functions [18].

On this thesis portal tags are assigned to theses to describe and organize them in order to be helpful in browsing and searching. The portal has tag clouds in Hungarian and English languages.

In our previous work, novel algorithms have been provided to improve tag clouds with vocabulary refinement and enhanced reference counts [19]. Moreover, an improved font distribution algorithm has been provided based on the power law distribution and arbitrary percentages [20]. In this paper, topics are recommended using a special tag graph constructed from the tag cloud and by calculating the reference count of each node.

IV. SPECIAL TAG GRAPH

In this section a special tag graph is proposed, which can be constructed from a tag cloud, and can be used for topic recommendation. Table I summarizes the notations of algorithms. Furthermore, the novel algorithms are validated and verified on a real-world thesis portal.

TABLE I
NOTATIONS OF ALGORITHM

Notation	Description
τ	set of tags
τ_u	set of uninvestigated tags
t	a tag
$t.dn$	display name property of tag t
$t.rc$	reference count property of tag t
$t.wc$	word count property of tag t
$dn.w_i$	i^{th} word of a display name
$dn.w_{i \rightarrow j}$	concatenated word starting from i^{th} word and ending at j^{th} word of a display name dn
ν	set of nodes
ε	set of edges
v	a node
v_s	source node of a directed edge
v_t	target node of a directed edge
$e_{i,j}$	an edge from node v_i to node v_j
$v.id$	identifier property of node v
$v.dn$	display name property of node v
$v.wt$	weight property of node v
$v.rc$	reference count property of node v

A. Algorithm to Construct a Tag Graph

A directed, weighted graph $G = (\nu, \varepsilon)$ is defined as a set of nodes ν and edges ε . The ij^{th} entry of the adjacency matrix A is one if there is a directed edge from node i to node j , and zero if such edge does not exist. Only the nodes have nonnegative weights, the edges are not weighted.

This graph can be constructed from a tag cloud using Algorithm 1. The notations of this recursive algorithm can be seen in Table I.

This is a recursive algorithm. The tags are investigated ordered ascending by their word count. If the word count is one, then create a new node for it, and return with their identifier in Steps 9–11.

If the display name of the tag contains more words, then search for a matching node investigating the display name from its begin (Steps 16–17) and its end (Steps 23–24) starting from length of word count minus one until one. If a matching node exists, then investigate whether the other part of the display name exists as a node (Steps 18–19, 25–26). If there is no matching for the other part, then create appropriate nodes for it by recursive calls (Steps 21–22, 28–29).

If there is no matching node investigating the display name from its begin and its end starting from length of word count minus one until one (Step 30), then create a new node for the first word (Steps 31–32), and create appropriate nodes for the other part of the display name by recursive calls (Steps 33–34).

After the shorter parts of the display name exist, then create a new node for the whole display name (Step 35). Moreover, create two directed edges from two nodes with shorter parts to the node of the whole display name (Step 36).

Algorithm 1 Pseudo code of algorithm to construct tag graph from tag cloud

```

1:  $\tau_u = \tau$ 
2:  $\nu = \emptyset, \varepsilon = \emptyset$ 
3:  $k = 1$ 
4: while  $\tau_u \neq \emptyset$  do
5:   for  $t \in \tau_u$ , where  $t.wc = \min(t_u.wc), \forall t_u \in \tau_u$ 
6:      $k = \text{AddNode}(k, t.dn, t.wc, t.rc) + 1$ 
7:
8:   function  $\text{AddNode}(k, dn, wc, rc) : i$ 
9:   if  $wc = 1$  then
10:     $v.id = k, v.dn = dn, v.wt = rc, \nu = \nu \cup \{v\}$ 
11:    return  $v.id$ 
12:   else
13:     $i_1 = 0$ 
14:    for  $i = 1 \rightarrow wc - 1$  do
15:      if  $i_1 = 0$  then
16:        if  $\exists v_1 \in \nu$ , where  $v_1.dn = dn.w_{1 \rightarrow wc-i}$  then
17:           $i_1 = v_1.id$ , where  $v_1.dn = dn.w_{1 \rightarrow wc-i}$ 
18:          if  $\exists v_2 \in \nu$ , where  $v_2.dn = dn.w_{wc-i+1 \rightarrow wc}$  then
19:             $i_2 = v_2.id$ , where  $v_2.dn = dn.w_{wc-i+1 \rightarrow wc}$ 
20:          else
21:             $i_2 = \text{AddNode}(k, dn.w_{wc-i+1 \rightarrow wc}, i, 0)$ 
22:             $k = i_2 + 1$ 
23:          else if  $\exists v_1 \in \nu$ , where  $v_1.dn = dn.w_{i+1 \rightarrow wc}$  then
24:             $i_1 = v_1.id$ , where  $v_1.dn = dn.w_{i+1 \rightarrow wc}$ 
25:            if  $\exists v_2 \in \nu$ , where  $v_2.dn = dn.w_{1 \rightarrow i}$  then
26:               $i_2 = v_2.id$ , where  $v_2.dn = dn.w_{1 \rightarrow i}$ 
27:            else
28:               $i_2 = \text{AddNode}(k, dn.w_{1 \rightarrow i}, i, 0)$ 
29:               $k = i_2 + 1$ 
30:            if  $i_1 = 0$  then
31:               $v_1.id = k, v_1.dn = dn.w_1, v_1.wt = 0, \nu = \nu \cup \{v_1\}$ 
32:               $i_1 = v_1.id$ 
33:               $i_2 = \text{AddNode}(k + 1, dn.w_{2 \rightarrow wc}, wc - 1, 0)$ 
34:               $k = i_2 + 1$ 
35:             $v.id = k, v.dn = dn, v.wt = rc, \nu = \nu \cup \{v\}$ 
36:             $\varepsilon = \varepsilon \cup \{e_{i_1, v.id}, e_{i_2, v.id}\}$ 
37:            return  $v.id$ 

```

B. Experimental Results for Construction

An example part of the constructed tag graph can be seen in Fig. 1. The tags are the following: 'data', 'text mining', 'data mining', 'data mining competition'. The reference counts of such nodes which do not correspond to real tags are zero (Steps 21, 28, 33 of Algorithm 1). The in-degree of nodes can be zero or two: zero if the display name of a given node contains only one word, in addition, two if it is a compound word.

The histogram of out-degree of nodes is depicted in Figs. 2 and 3. There are numerous nodes whose out-degree is zero, namely, they are not building items of nodes with longer display name. However, there are some nodes which are frequently used building items. The weight and out-degree of

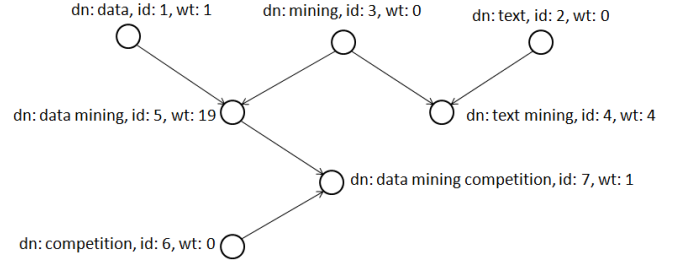


Fig. 1. An example part of the tag graph

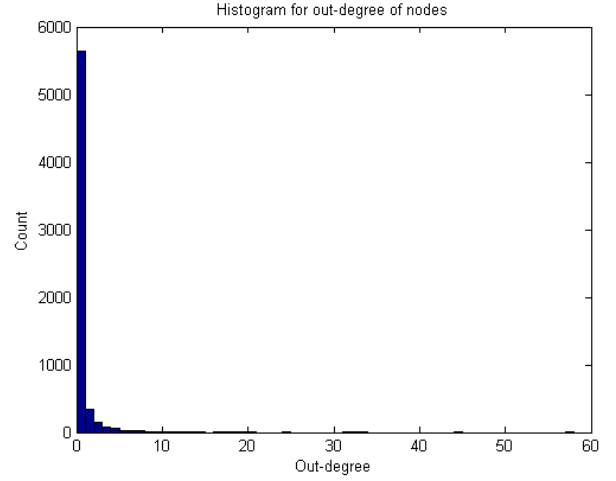


Fig. 2. Histogram for out-degree of nodes

nodes influences the reference counts of nodes, namely, the recommended topics.

C. Algorithm to Recommend Topics from Tag Graph

The reference counts of nodes can be calculated by Algorithm 2. The notations of this recursive algorithm can be seen in Table I.

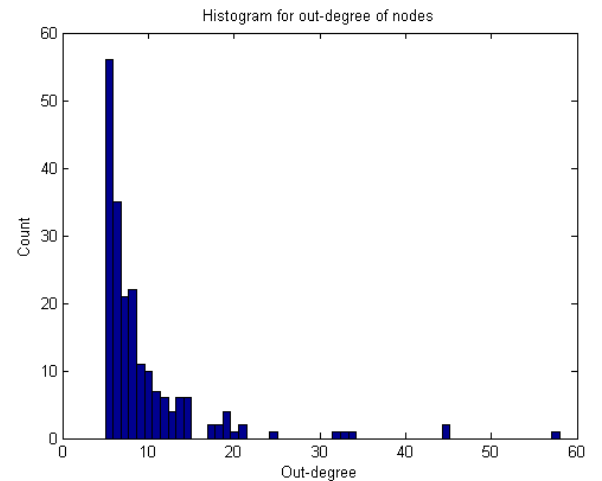


Fig. 3. Histogram for out-degree of nodes only from out-degree 5

Algorithm 2 Pseudo code of algorithm to calculate reference counts of nodes

```

1: for all  $v \in \nu$  do
2:    $v.rc = v.wt + Count(v, 0)$ 
3:
4: function  $Count(v_s, rc) : rc$ 
5:    $\nu_t = \{v_t, \text{ where } \exists e_{s,t}\}$ 
6:   if  $|\nu_t| \geq 1$  then
7:      $rc = rc + \sum_{\forall v_t \in \nu_t} v_t.wt$ 
8:   for all  $v_t \in \nu_t$  do
9:      $rc = Count(v_t, rc)$ 
10: return  $rc$ 

```

TABLE II
CONSTRUCTION OF TAG GRAPH

	Count
Tags	4152
Nodes	6408
Edges	5404
Nodes whose in-degree = 0	3706
Nodes whose in-degree = 2	2702
$rc \geq 10$	274
$wt \neq rc$ ($rc \geq 10$)	236
$wt = 0$ ($rc \geq 10$)	70

The reference count of a node is calculated as the sum of weights of such nodes which can be reached from the given node via directed edges.

The proposed special tag graph with the calculated reference counts can be used for topic recommendation. The recommended topics are the nodes with the most reference counts. The limit in reference counts or the maximum number of topics can be chosen arbitrarily.

D. Experimental Results for Recommendation

The construction of the tag graph is described in Table II. The nodes whose reference count is greater or equal to 10 are identified as topics. The reference counts and the weights of topics are different numbers in more than 85 percentages, thus, the proposed reference counts of nodes are an important improvement of the reference count of tags. More than 25 percentages of nodes are such topics, which are not existing tags, hence, the provided tag graph is a significant enhancement of the original tag cloud.

V. VISUALIZATION OF RECOMMENDED TOPICS

In tag clouds the tags are classified according to their reference counts. The number of classes is an arbitrary parameter. In our topic cloud, tags are classified into four classes. In our previous work, several font distribution algorithms have been tested. The linear distribution algorithm simply divides linearly the whole range (from minimum count to maximum) count by the number of classes. The logarithmic algorithm divides logarithmically equal intervals. Since the reference

TABLE III
DISTRIBUTION OF NODES AMONG CLASSES

	Percentage	Count of nodes
Class 1	95	255
Class 2	3	8
Class 3	1.5	4
Class 4	0.5	1

.net (64) • 3d (44) • 62305 (12) • acquisition (12) • active (13) • ad (11) • ajax (16) • algorithm (24) • analysis (25) • and (12) • **android** (1124) • android application (13) • application (73) • application development (14) • architecture (11) • artificial (13) • asp.net (38) • asp.net mvc (16) • automation (19) • automotive (11) • autonomous (15) • avr (16) • azure (10) • based (34) • bluetooth (24) • board (10) • bus (16) • business (16) • c# (40) • c++ (12) • camera (13) • can (41) • car (13) • card (18) • cell (22) • circuit (15) • cisco (10) • client (11) • cloud (28) • cloud computing (24) • code (13) • communication (20) • computer (20) • computer graphics (16) • computing (28) • content (10) • control (34) • controller (12) • crm (11) • current (14) • css (10) • data (66) • data acquisition (11) • data mining (20) • database (26) • design (37) • detection (26) • development (34) • digital (27) • distributed (10) • dynamic (11) • e (16) • eclipse (10) • ee (32) • electric (16) • electrical (11) • embedded (28) • embedded system (13) • energy (30) • engine (24) • enterprise (12) • entity (13) • entity framework (13) • environment (12) • estimation (11) • ethernet (21) • exchange (15) • facebook (12) • fi (15) • filter (16) • filtering (10) • finite (10) • flash (12) • flow (18) • formula (15) • fpga (26) • framework (36) • frequency (17) • fuzzy (10) • game (47) • game development (11) • game engine (10) • generation (11) • google (28) • gpu (17) • gps (38) • gpu (11) • graph (10) • graphics (21) • grid (15) • gsm (14) • gui (12) • gwt (11) • hardware (14) • health (11) • high (13) • home (15) • html5 (13) • ieee (10) • image (36) • image processing (31) • in (16) • indoor (11) • integration (13) • intelligence (20) • intelligent (14) • interface (30) • internet (21) • inverter (11) • iphone (10) • ipv6 (12) • **Java** (95) • java ee (32) • javascript (23) • jquery (15) • labview (17) • laser (14) • lcd (10) • learning (29) • lightning (12) • line (16) • linear (16) • linux (19) • live (13) • load (16) • localization (10) • location (12) • location based (10) • low (10) • lt (17) • machine (22) • management (70) • map (13) • market (15) • matlab (31) • measurement (36) • medical (14) • memory (10) • metering (11) • method (17) • micro (10) • microcontroller (48) • microsoft (48) • microsoft sharepoint (12) • microsoft sql server (11) • mining (27) • mobil (27) • **mobile** (38) • model (44) • modeling (15) • modelling (11) • module (12) • monitoring (18) • motion (13) • motor (18) • mpeg (11) • msz (17) • multi (12) • mvc (21) • mysql (19) • **network** (119) • networks (26) • neural (13) • non (10) • of (46) • on (14) • ontology (10) • openflow (10) • openpi (11) • optical (16) • optimization (25) • p2p (10) • phone (20) • php (34) • planning (24) • platform (13) • plc (23) • positioning (11) • **POWER** (64) • process (25) • **processing** (61) • processor (10) • programming (10) • project (10) • protection (20) • quality (12) • radio (26) • real (17) • real time (12) • remote (10) • rfid (12) • ria (10) • robot (20) • routing (24) • sap (24) • search (14) • security (36) • sensor (34) • sensor network (13) • series (11) • server (30) • service (26) • shader (10) • sharepoint (12) • signal (22) • signal processing (14) • silverlight (28) • simulation (42) • simulator (13) • sip (10) • smart (21) • smartphone (10) • soa (14) • soap (11) • social (20) • social network (11) • software (11) • software development (14) • solar (23) • solar cell (11) • soldering (19) • space (10) • speech (22) • sql (21) • sql server (11) • stock (14) • storage (12) • studio (13) • supply (12) • switch (12) • synchronous (12) • **system** (100) • systems (37) • technology (13) • test (37) • testing (27) • text (17) • time (32) • time series (11) • to (13) • tracking (19) • traffic (16) • transformer (17) • usb (27) • user (30) • user interface (16) • video (12) • virtual (15) • visualization (18) • voltage (21) • wcf (31) • **web** (90) • web application (12) • webportal (11) • webservice (16) • wi (15) • wi fi (15) • wind (24) • windows (38) • windows phone (17) • wireless (20) • workflow (15) • wpf (19) • xml (15)

Fig. 4. Resulted topic cloud

counts obey a power law, a power law and percentage based approach can led to correct visual impression. The resulted distribution of nodes among classes is summarized in Table III.

The identified topics are visualized as a tag cloud alphabetically ordered and visually weighted by letter size. The resulted topic cloud are depicted in Figs. 4 and 5. The calculated reference counts are in brackets after the display name of topics. The tags with their reference counts, which fill parts of the given node, are shown on tooltips. In Fig. 5 see for example the tooltip for node 'processing', who is not in the original tag cloud, who is not an itself existing tag, but identified as an important topic.

.net (64) • 3d (44) • 62305 (12) • acquisition (12) • active (13) • ad (11) • ajax (16) • algorithm (24) • analysis (25) • and (12) • **android** (1124) • android application (13) • application (73) • application development (14) • architecture (11) • artificial (13) • asp.net (38) • asp.net mvc (16) • automation (19) • automotive (11) • autonomous (15) • avr (16) • azure (10) • based (34) • bluetooth (24) • board (10) • bus (16) • business (16) • c# (40) • c++ (12) • camera (13) • can (41) • car (13) • card (18) • cell (22) • circuit (15) • cisco (10) • client (11) • cloud (28) • cloud computing (24) • code (13) • communication (20) • computer (20) • computer graphics (16) • computing (28) • content (10) • control (34) • controller (12) • crm (11) • current (14) • css (10) • data (66) • data acquisition (11) • data mining (20) • database (26) • design (37) • detection (26) • development (34) • digital (27) • distributed (10) • dynamic (11) • e (16) • eclipse (10) • ee (32) • electric (16) • electrical (11) • embedded (28) • embedded system (13) • energy (30) • engine (24) • enterprise (12) • entity (13) • entity framework (13) • environment (12) • estimation (11) • ethernet (21) • exchange (15) • facebook (12) • fi (15) • filter (16) • filtering (10) • finite (10) • flash (12) • flow (18) • formula (15) • fpga (26) • framework (36) • frequency (17) • fuzzy (10) • game (47) • game development (11) • game engine (10) • generation (11) • google (28) • gpu (17) • gps (38) • gpu (11) • graph (10) • graphics (21) • grid (15) • gsm (14) • gui (12) • gwt (11) • hardware (14) • health (11) • high (13) • home (15) • html5 (13) • ieee (10) • image (36) • image processing (31) • in (16) • indoor (11) • integration (13) • intelligence (20) • intelligent (14) • interface (30) • internet (21) • inverter (11) • iphone (10) • ipv6 (12) • **Java** (95) • java ee (32) • javascript (23) • jquery (15) • labview (17) • laser (14) • lcd (10) • learning (29) • lightning (12) • line (16) • linear (16) • linux (19) • live (13) • load (16) • localization (10) • location (12) • location based (10) • low (10) • lt (17) • machine (22) • management (70) • map (13) • market (15) • matlab (31) • measurement (36) • medical (14) • memory (10) • metering (11) • method (17) • micro (10) • microcontroller (48) • microsoft (48) • microsoft sharepoint (12) • microsoft sql server (11) • mining (27) • mobil (27) • **mobile** (38) • model (44) • modeling (15) • modelling (11) • module (12) • monitoring (18) • motion (13) • motor (18) • mpeg (11) • msz (17) • multi (12) • mvc (21) • mysql (19) • **network** (119) • networks (26) • neural (13) • non (10) • of (46) • on (14) • ontology (10) • openflow (10) • openpi (11) • optical (16) • optimization (25) • p2p (10) • phone (20) • php (34) • planning (24) • platform (13) • plc (23) • positioning (11) • **power** (64) • process (25) • **processing** (61) • processor (10) • programming (10) • project (10) • protection (20) • quality (12) • radio (26) • real (17) • real time (12) • remote (10) • rfid (12) • ria (10) • robot (20) • routing (24) • sap (24) • search (14) • security (36) • sensor (34) • sensor network (13) • series (11) • server (30) • service (26) • shader (10) • sharepoint (12) • signal (22) • signal processing (14) • silverlight (28) • simulation (42) • simulator (13) • sip (10) • smart (21) • smartphone (10) • soa (14) • soap (11) • social (20) • social network (11) • software (11) • software development (14) • solar (23) • solar cell (11) • soldering (19) • space (10) • speech (22) • sql (21) • sql server (11) • stock (14) • storage (12) • studio (13) • supply (12) • switch (12) • synchronous (12) • **system** (100) • systems (37) • technology (13) • test (37) • testing (27) • text (17) • time (32) • time series (11) • to (13) • tracking (19) • traffic (16) • transformer (17) • usb (27) • user (30) • user interface (16) • video (12) • virtual (15) • visualization (18) • voltage (21) • wcf (31) • **web** (90) • web application (12) • webportal (11) • webservice (16) • wi (15) • wi fi (15) • wind (24) • windows (38) • windows phone (17) • wireless (20) • workflow (15) • wpf (19) • xml (15)

Fig. 5. Resulted topic cloud with tooltip about tags belonging to node of "processing"

VI. CONCLUSION AND FUTURE WORK

The visualization of huge tag clouds is one of the most important and complicated consideration. In our thesis portal tag clouds have a very important role to facilitate browsing and searching among numerous tags and theses. In our previous work, tag clouds have been improved with vocabulary refinement and enhanced reference counts. Moreover, fonts have been distributed based on power law distribution and arbitrary percentages.

In this paper, novel algorithms have been proposed to recommend topics based on the tag cloud and visualize the recommended topics like a tag cloud. The further improvement of visualization is a subject of future work. Novel algorithms have been proposed to construct a special tag graph from the tag cloud, and to recommend topics using this graph by calculating the reference count of each node. Furthermore, a visualization has been introduced for recommended topics like a tag cloud using a special font distribution algorithm.

The resulted topic cloud is a significant enhancement of the original tag cloud, since lots of such topics are in the resulted topic cloud, which are not existing tags in the original tag cloud, in addition, the popularity of topics is calculated more properly, furthermore, popular topics can be identified easily.

The proposed algorithms have been implemented in SQL stored procedures, C# classes using LINQ, and MATLAB functions. They have been validated and verified on tag clouds of a real-world thesis portal.

ACKNOWLEDGMENT

First of all we wish to thank SUZUKI Foundation for sponsoring this research. This work is connected to the scientific program of the "Development of quality-oriented and harmonized R+D+I strategy and functional model at BME" project. This project is supported by the New Hungary Development Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

REFERENCES

- [1] T. O'Reilly, *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*, International Journal of Digital Economics, No. 65, pp. 17–37, March 2007.
- [2] A. M. Kaplan, M. Haenlein, *Users of the world, unite! The challenges and opportunities of Social Media*, Business Horizons, Vol. 53(1), pp. 59–68, 2010.
- [3] J. Zimmerman, D. Sahlin, *Social Media Marketing All-in-One For Dummies*, Wiley Publishing, ISBN 978-0-470-58468-2, 2010.
- [4] C. C. Aggarwal, *Social Network Data Analytics*, 1st Edition, Springer, ISBN 978-1-4419-8461-6, 2011.
- [5] Microsoft Research - Social Computing Group, <http://research.microsoft.com/scg/>.
- [6] IBM Research - Social Computing Group, http://www.research.ibm.com/haifa/dept/imt/ct_st.shtml.
- [7] HP Labs - Social Computing Lab, <http://www.hpl.hp.com/research/sc/>.
- [8] A. Clauset, C.R. Shalizi, and M.E.J. Newman, *Power-law distributions in empirical data*, SIAM Review, Vol. 51(4), pp. 661–703, 2009.
- [9] S. A. Golder, B. A. Huberman, *The Structure of Collaborative Tagging Systems*, 2005.
- [10] D. H. Pink, *Folksonomy*, New York Times, December 11, 2005.
- [11] J. Sinclair, M. Cardew-Hall, *The folksonomy tag cloud: when is it useful?*, Journal of Information Science, Vol. 34(1), pp. 15–29, 2008.
- [12] S. Lohmann, J. Ziegler, L. Tetzlaff, *Comparison of Tag Cloud Layouts: Task-Related Performance and Visual Exploration*, INTERACT, Part I, LNCS 5726, pp. 392–404, 2009.
- [13] Y. Hassan-Montero, V. Herrero-Solana, *Improving Tag-Clouds as Visual Information Retrieval Interfaces*, International Conference on Multidisciplinary Information Sciences and Technologies, Merida, Spain, 2006.
- [14] J. Salonen, *Self-Organizing Map Based Tag Clouds*, OPAALS Conference, 2007.
- [15] kentbye's blog, *Tag Cloud Font Distribution Algorithm*, <http://www.echochamberproject.com/node/247>, June 24, 2005.
- [16] K. Hoffman, *In Search Of... The Perfect Tag Cloud*, <http://files.blog-city.com/files/J05/88284/b/insearchofperfecttagcloud.pdf>.
- [17] Thesis Portal, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, <http://diplomaterivik.bme.hu>.
- [18] MATLAB, The Language of Technical Computing, <http://www.mathworks.com/products/matlab/>.
- [19] Á. Bogárdi-Mészöly, A. Rövid, H. Ishikawa, *Algorithms to Improve Tag Clouds*, 5th International Conference on Emerging Trends in Engineering & Technology, Himeji, Japan, 2012.
- [20] Á. Bogárdi-Mészöly, A. Rövid, H. Ishikawa, *An Improved Font Distribution Algorithm for Tag Clouds*, 6th International Conference on Soft Computing and Intelligent Systems, Kobe, Japan, 2012.